Music Genre Classification

Suryank Tiwari
MT19019

Rose Verma
MT19052

***Abstract*—This project tackles the various machine learning challenges associated with classification of musical genres. Members of a music genre share common characteristics - instruments, tempo, harmonic and rhythmic content but the problem remains challenging given the subjective nature of the field. Comparative and exploratory analysis of the pre-processing routines applied, feature extraction techniques and the models used for identification are enclosed. The work that follows can serve as a basis for music tagging and recommendations for audio streaming or vending services.**

***Index Terms*—music, genre, classification, GTZAN**

I. Problem statement and motivation

Genre of a song is an umbrella term that clusters members with similarity in the musical instrumentation used, the harmonic and rhythmic content, structure, pitch, and other related properties. It is not a well defined concept and is highly subjective in nature. These definition have arisen over time through complex and tangled historical and cultural factors.
Automatically extracting music information aids in organization of the abundance of of music files available digitally on the Web. Genre hierarchies, typically created manually by human experts, are currently one of the ways used to structure music content on the Web. [1]

The motivation of the project is to create a machine learning framework that explores and then extracts crucial features from musical content. Musical preprocessing techniques that perform better results and a comparative analysis of machine learning models upon the same data to solve the problem of music genre classification. The work can serve as a basis for several other auditory applications.

II. Literature review

G. Tzanetakis and P. Cook worked on the GTZAN dataset to extract features representing timbral texture, rhythmic content and pitch content in 2002. They achieved an accuracy of 0.61. [1] Dong and Mingwen achieved an average accuracy of 0.70 with CNNs that parallel the human accuracy for this work (70 percent as well). [2] Changsheng Xu [3] and team applied SVM over the musical genre classification problem to achieve over 0.90 accuracy for a 4 genre problem. D. Huang, A. Serafini and E. Pugh [4] from Stanford achieved testing accuracy of 0.82 in their work on GTZAN genre classification using CNNs.

Oramas, S., Barbieri, F., Nieto, O. and Serra, X. in 2018 [5] combined auditory, textual and visual representations of music to a single model and they achieved an AUC score of 0.936 for 10 genre classification on Million Song Dataset with two CNNs for audio and visual processing.

III. Dataset details

GTZAN: This is a publically available dataset which and is quite popular for this problem statement. It comprises of 1000 audio tracks each of which is 30 seconds long. There are 10 genres in total and each genre has a 100 tracks in it. The dataset is balanced and homogeneous. The tracks are all 22050Hz Mono 16-bit audio files in .wav format. Total size of the data is 1.23 GBs.

Dataset Analysis

This congrous nature of the dataset makes it a suitable choice for proceeding with the problem, but the small size of the dataset restricts proper learning. This is further discussed in the Dataset Augmentation section.
The distribution of number of tracks per genre can be seen in figure 1.

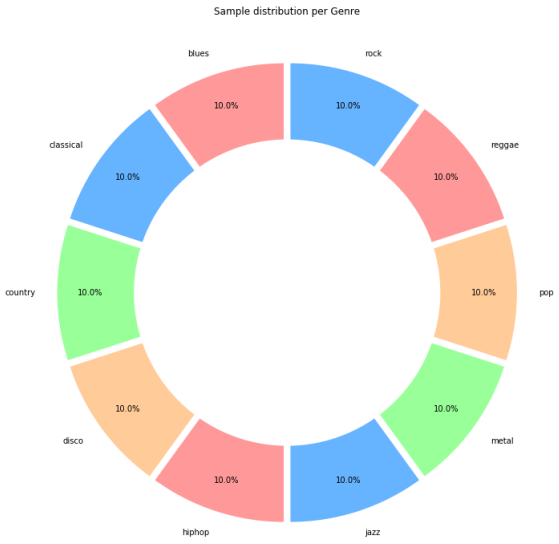

Fig. 1. Track sample distribution pie plot

After loading all the tracks from GTZAN dataset to a feature matrix, and reducing the dimensions to 2 using PCA, the hex scatter distribution is noted in Fig 2. The data distribution is evidently dense and most of the samples overlap, and the task of genre segregation is not an easy one. Clearly, feeding song data to machine learning models will give terrible results given

the nature of the distribution. Hence, proper feature extraction is a must and is discussed in further sections in detail.

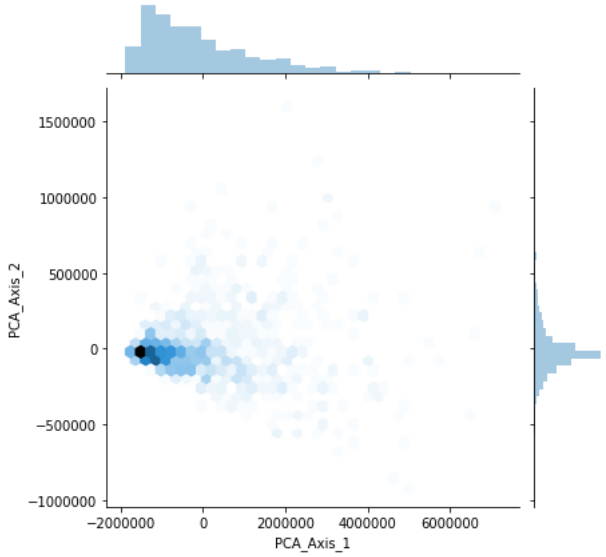

Fig. 2. GTZAN data PCA 2D hex plot

IV. Work Done

A. Preprocessing and Dataset Augmentations

There are 100 samples per genre and each sample has a length of 30 seconds. There are two issues with the dataset in consideration that are observed initially:

1) The small nature of the dataset doesn't allow enough samples per genre for training which in turn affects performance badly. There needs to be more samples per genre for proper learning.

2) While extracting features, properties like mean, standard deviation and variance of the feature data will be considered. A sample length of 30 seconds will have several silences, noise altered portions along with different types of musical themes as are usually present in a song. That is, a scalar unit represents the vector quantities. Scalars representing lengthy vectors will be more erroneous and less representative of the actual data. A length of 30 seconds is too lengthy for proper scalar representation.

These issues are resolved by creating non overlapping samples of shorter length from the original music tracks in the datasset. Thence increasing the number of samples and decreasing the sample length for better representation with scalars.
The original dataset had a 1000 tracks of 30 seconds each. Each of these samples was subdivided into sub-samples of a constant length and number of samples in a genre increased from 100 to

$$\frac{30}{n} * 100 \text{ samples per genre} \quad (1)$$

The tested values of n are - 1, 3, 5 and 10 resulting in 3000, 1000, 600 and 300 samples for each genre. This is further shown in fig 3.

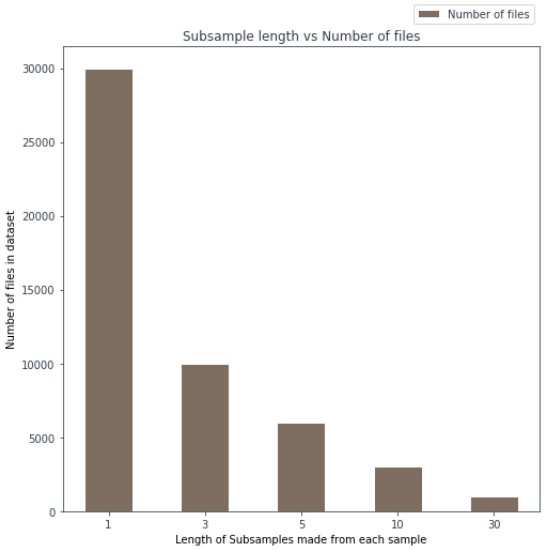

Fig. 3. Number of files in the entire dataset for each subsample length

This procedure of dataset augmentation significantly improved the classification performance. 30 was the original length of the samples and it is the worst performer out of the bunch of samples we have taken. Best results have been obtained with subsample length 5, then 3, followed by 10 and 1, and lastly 30. This is further discussed in the results section.

B. Feature Extraction

Librosa library has been utilized for feature extraction purposes from the dataset. All of the features extracted below have been demonstrated on a specific 5 second long audio sample, however different lengths of subsamples were used and compared in the classification process for comparison of results. The features that have been extracted and then considered for selection are as follows:

Mel-Frequency Cepstral Coefficients
MFCCs are signal bound features which describe the shape of the spectral envelope, i.e. frames of energy signals.

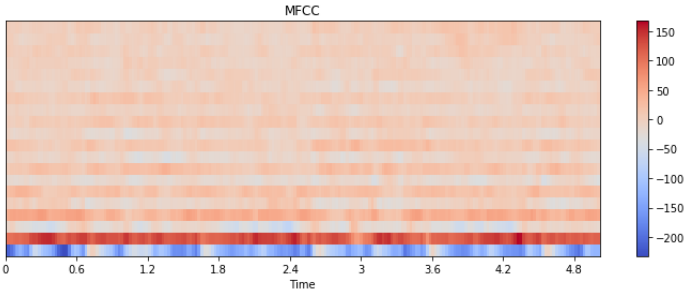

Fig. 4. MFCC feature plot

MFCCs are an important feature for sound classification and have been used in most classical work.

Power Spectrogram Chromagram
Short term Fourier transform (STFT) of the chromagram is taken as the feature.

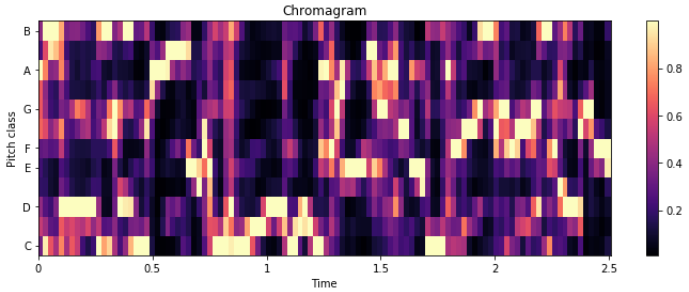

Fig. 5. Chroma_stft feature plot

Chroma Energy Normalized (CENS) Chromagram
CENS features take statistics over large frames of tempo, articulation, and musical ornaments. CENS is a good feature for audio matching and similarity.

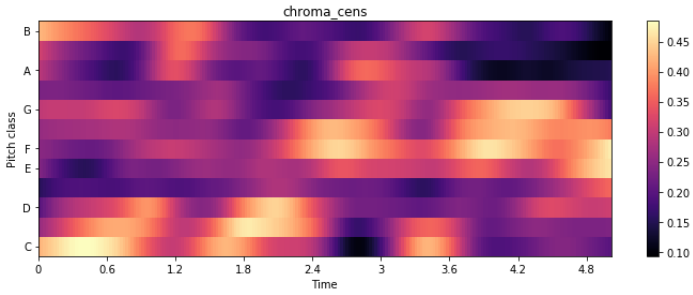

Fig. 6. Chroma_cens feature plot

Constant-Q Chromagram
The constant-Q transform uses a logarithmically spaced frequency axis which is smilar to the Mel scale but unlike the fourier transform.

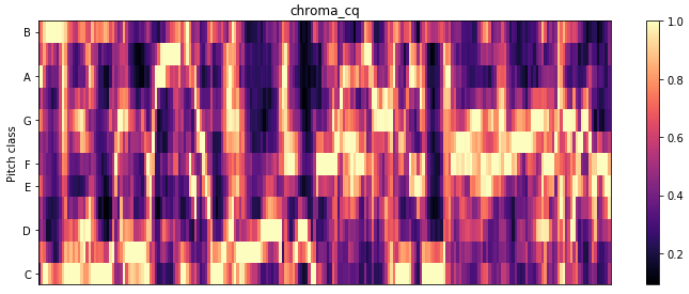

Fig. 7. Chroma_cqt feature plot

Mel-scaled spectrogram
This feature coverts the spectrogram frequencies to the mel scale and gives them as output.

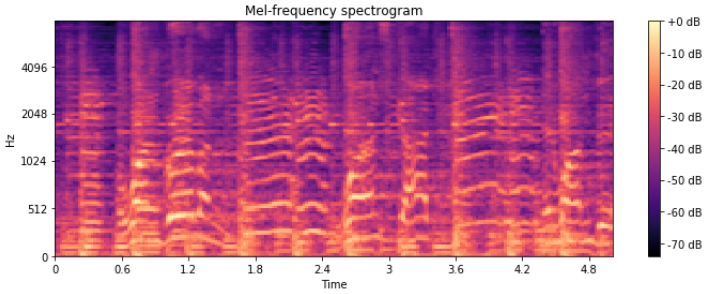

Fig. 8. Melspectrogram feature plot

Flatness
Flatness is a measure of quantification of noisy nature of an audio clip. A flatness of 1 indicates similarity to white noise.

Zero crossing Rate
The rate at which an audio sample crosses the zero threshold, i.e. the rate of the sign change along a signal.

Root Mean Square The RMS or root mean square value for each frame from the audio samples y is computed and taken as a feature.

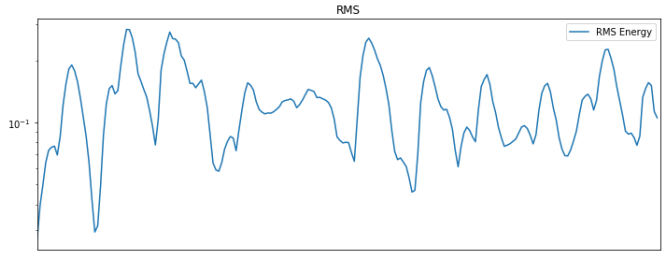

Fig. 9. RMS feature plot

Tempogram
Tempo is the beats per minute and hence speed of a musical piece. It is given by the reciprocal of the beat period. Tempogram indicates the prevalence of certain tempi at each moment in time.

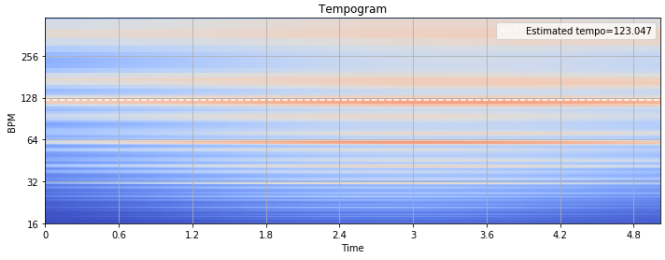

Fig. 10. Tempogram BPMS feature plot

Polynomial features
This feature is the collection of coefficients of a **nth order polynomial** that has been fit upon the columns of a spectro-

gram. For this problem polynomials of order 0, 1 and 2 were chosen.

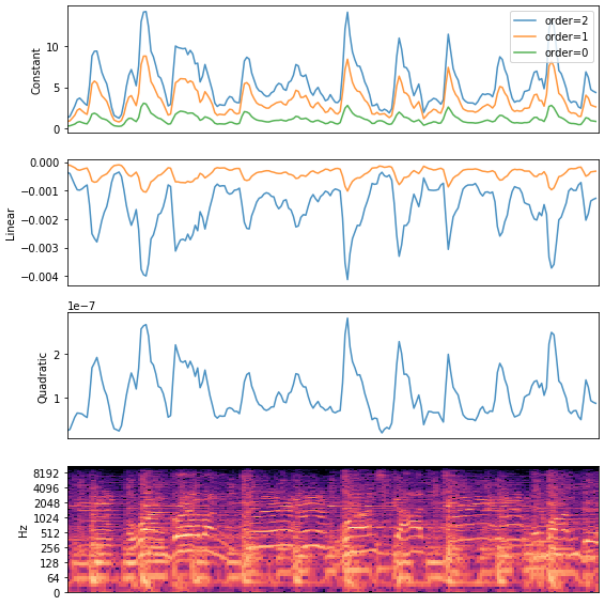

Fig. 11. Polynomial feature plot

Spectral Centroid

It is the weighted mean of the frequencies present in the sound. It is similar to treating musical frequencies as weights and then finding the center of mass for the group.

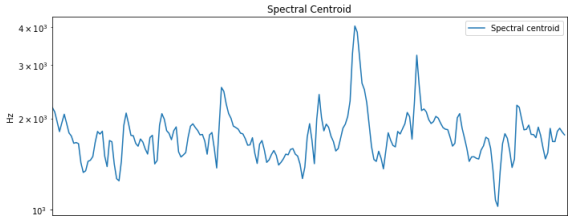

Fig. 12. Spectral Centroid feature plot

Spectral Contrast

For each sub-band in spectrogram frames, contrast is the comparison of peak energy to valley energy. A high spectral contrast denotes clear signal, while a low spectral contrast denotes broad band noise.

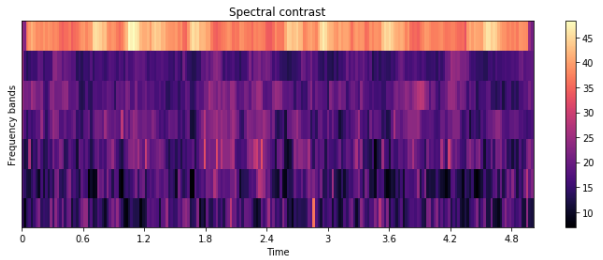

Fig. 13. Spectral Contrast feature plot

Spectral Bandwidth

For each sub-band in spectrogram frames, contrast is the comparison of peak energy to valley energy. A high spectral contrast denotes clear signal, while a low spectral contrast denotes broad band noise.

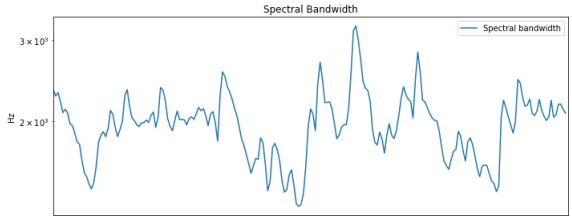

Fig. 14. Spectral Bandwidth feature plot

Spectral Rolloff

Spectral rolloff is the frequency below which a specified percentage of the total spectral energy

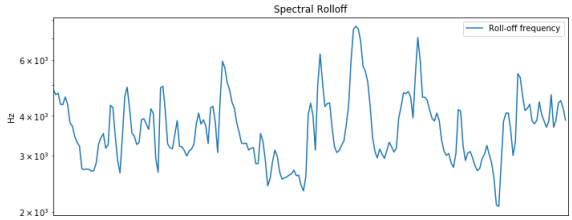

Fig. 15. Spectral Rolloff feature plot

Tonal centroid features

A Tonnetz can be derived by connecting successive perfect fifths, major thirds, and successive minor thirds with lines. This creates a 2 dimensional lattice which can be used as a feature for representation of a track.

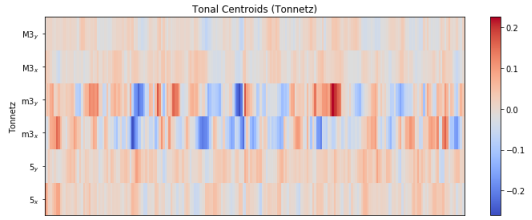

Fig. 16. Tonnetz feature plot

For each feature except for MFCC, mean, variance and standard deviation of features is taken. Poly features contains three polynomials, 0th order, 1st order and 2nd order. A total of 16 features excluding the MFCC are obtained. While computing MFCC, 20 vectors are obtained, and for each vector, the mean, variance and standard deviation is taken. This leads to total features of:

$$\text{Total Features} = 16 * 3 + 20 * 3 = 108 \quad (2)$$

Two features are also added in the dataframe for filename, and genre label but these are popped out before training. Including these, the total feature vector length becomes 110.

C. Feature Selection

The dataset obtained after feature extraction and dataset augmentation consisted of 110 features. Number of features is too much and there is need for dimensionality reduction or feature selection. PCA was applied as a dimensionality reduction technique but always gave a drop in the net performance. Therefore, feature selection was performed in the following ways.

Variance Thresholding

Features with small variance are not as important and can be removed safely. Features having a variance of less than 0.1 are removed from consideration. After Variance Thresholding, 84 features remain from the set of 110 features.

Removal of Correlated Features

Correlated Features increase redundancy and do not contribute to the model differently other than giving the same data bias multiple times. Therefore, correlated features are removed, and the feature set size is reduced to 56.

Selecting Features using F-ANOVA Tests

ANOVA-F value between each feature and the target value indicates important features. Select KBest algorithm variant with fclassif score function is used to compute the same. The final feature set size is 40 based on trial and error of performances achieved.

Fig. 17. Histogram plot of selected features

Fig 17 plots the histogram of the 40 features that have been selected after feature extraction and selection. Fig 18 plots the correlation matrix of these 40 features.

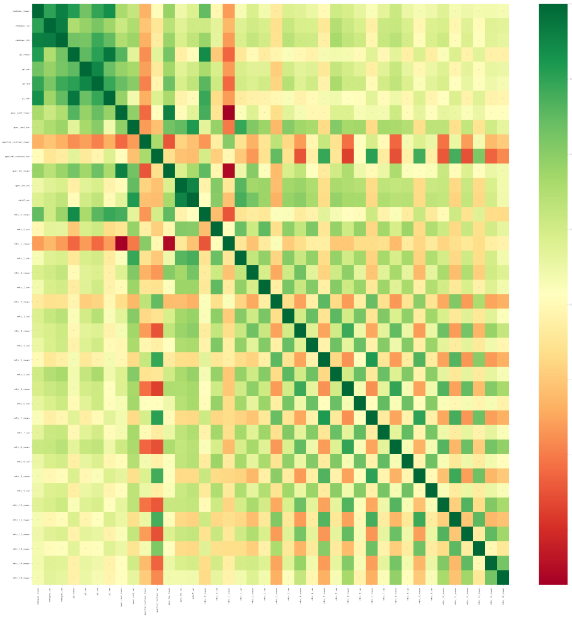

Fig. 18. Correlation matrix for selected features

D. Model Selection

A set of models consisting of 12 classifiers was applied to the final oversampled and feature selected dataset. The performance was compared on testing ratio which was 30% of the total samples and on 3 fold cross validation. For models that showed some promise, hyper parameter tuning was performed with GridSearchCV algorithm to improve performance further. The models that give the best performance are:

1) HistGradient Boosting Classifier
2) LGBM Classifier
3) Random Forest Classifier
4) Gradient Boosting Classifier

E. Proposed Architecture

Failed Techniques, Challenges and Progression:

1) Initially only MFCCs (20) features were extracted and with Random Forest (RF) Classifier a cross validation accuracy of around 0.625 was achieved. This result is comparable to some of the results in the literature reviews.
2) To improve the performance, the following features were extracted now:
   - chroma stft
   - rmse
   - spectral centroid
   - spectral bandwidth
   - rolloff
   - zero crossing rate
   - MFCC (20) features

   But a drop in cross validation accuracy was noted to 0.594.

3) We further increased the features to the full set explained in the features section but a significant improvement wasn't observed.
4) Poor performance of the model led to suspicion of the small size of the dataset behind the improper learning. Therefore we augmented the dataset with nonoverlapping subsamples. This lead to a major boost in performance. An accuracy of 0.830, which was the fartheset leap we have noticed in performance in this project. Subsampling was done on 3 second length subsamples.
5) After a drastic performance improvement, the reason we inferred is that scalars could represent the data better when the samples were shorter. To find the ideal sample length we created multiple augmented datasets and compared the performances. Best performance was obatained at subsamples of length 5. Performance noted with a hyperparameter tuned RF was 0.842.
6) PCA for dimensionality reduction but the cross validation accuracy always dropped therefore the final model does not include dimensionality reduction.
7) A total of top 40 features were selected with hit and trial from 110 in total. Another performance gain was observed with cross validation accuracy obtained from RF shooting up to 0.866
8) Finally for model selection we used 12 different models on the same dataset to obtain the noted results.

Final Proposed Architecture:

1) Create multiple non overlapping sub samples of shorter length out of the existing samples of length 30 from GTZAN to create an augmented dataset. The optimal sub-sample length is 5 seconds. So each file is split into 6 files of 5 seconds each. The total files in the dataset become 6000.
2) Extract 110 specified features from the data for each sample in the new augmented dataset.
3) Select top 40 features using Variance thresholding, removing correlated features and then by using F-ANOVA tests.
4) The obtained feature dataset of (6000, 64) shape is supplied to a list of models for testing and k-fold cross validation accuracy. The best performing

V. Results

For dataset sub-sampling augmentation, following are the results of different length of sub samples on Random Forest Classifier (non hyperparameter tuned):

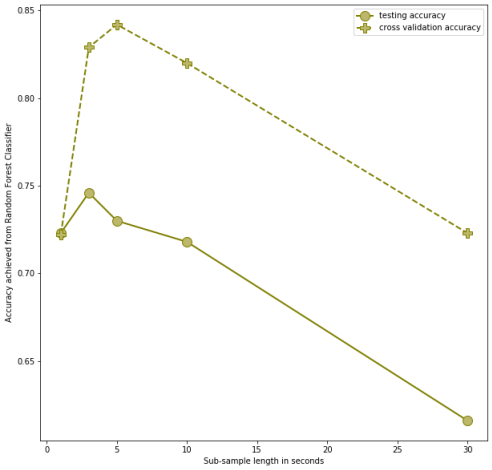

Fig. 19. Subsample length in augmented dataset vs accuracy achieved

Fig 19. clearly shows that when the dataset has samples of length 30, performance is the worst. The best results are observed at samples of length 5 seconds each.

Following performance observations were noted with listed models when performing classification on 5 second subsample-oversampled dataset, after feature selection had been applied. Hence, the shape of the dataframe input to these models was (6000, 40) and their testing accuracy on 30% unseen data samples and cross validation accuracy with 3 folds is reported below.

Model	**Testing Accuracy**	**Cross Val. Accuracy**
Linear SVC	0.152	0.182
SVC	0.303	0.293
Bernoulli NB	0.307	0.299
Ada Boost Classifier	0.240	0.326
Logistic Regression	0.352	0.346
Gaussian NB	0.442	0.435
Bagging Classifier	0.767	0.767
XGB Classifier	0.777	0.769
Gradient Boosting Classifier	0.846	0.853
LGBM Classifier	**0.853**	**0.859**
Random Forest Classifier	**0.858**	**0.866**
Hist Gradient Boosting Classifier	**0.886**	**0.884**

Hist Gradient Boosting Classifier and Random Forest Classifier are the two best performers and with the proposed architecture maximum accuracy of 88.4% is achieved.

Confusion matrix obtained with the best performing model on the proposed architecture is portrayed by fig 20.

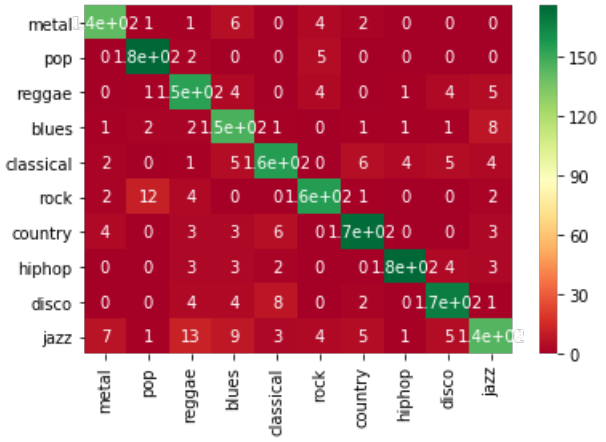

Fig. 20. Confusion Matrix using Hist Gradient Boosting Classifier

VI. Analysis of results

Subsamples of length 5 gave the best performance out of all. This leads to the belief that samples that are too small do not contain any data, and samples that are too large, lead to loss of data when represented by scalars. Sample length should be balanced. It is seen that the removal of correlated features and redundant features helped in reducing the training time, improved the overall performance of the model and helped reduce the data sparsity and harmful bias from the data. Tree based models generally performed well given their independence in nature and Decision Tree based ensembles decorrelate the features well. Random Forests are fast, correct and robust to noise and outliers. They also have low bias and moderate variance which is highly desriable in a machine learning model.

VII. Inferences and Conclusions

Largest improvement in results was observed after creating a new dataset with subsamples from the original. The problems with GTZAN dataset are that it has too little and too long samples in it. Absence of important features is a setback and needs to be resolved but just as important is the balancing of the dataset and its samples. The shortcomings of the data and the challenge we face were overcome by Dataset Manipulation, Feature Set Extraction and Feature Selection.

The accuracy achieved by the proposed architecture is comparable to that of Deep Learning models and there is further scope of improvement with exploration of more features and hence feature selection techniques.

VIII. Contributions

After project inception, we read up on research papers and articles to design a road map to achieving satisfactory results for this problem. The writing of project source code, project proposals, presentations and this final report has been a joint effort. Our group has collaborated closely on the majority of this project. We discussed the code and the problems we face with each other so the intricacies of the projects are known thoroughly to both of us. The stumps we faced while getting poor results to the ideas we had which improved the performances, has been a highly cohesive collaboration.

References

[1] Tzanetakis, George Cook, Perry. (2002). Musical Genre Classification of Audio Signals. IEEE Transactions on Speech and Audio Processing. 10. 293 - 302. 10.1109/TSA.2002.800560.

[2] Dong and Mingwen, “Convolutional Neural Network Achieves Human-level Accuracy in Music Genre Classification,” arXiv.org, 27-Feb-2018. [Online]. Available: https://arxiv.org/abs/1802.09697

[3] Changsheng Xu, N. C. Maddage, Xi Shao, Fang Cao and Qi Tian, ”Musical genre classification using support vector machines,” 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03)., Hong Kong, 2003, pp. V-429, doi: 10.1109/ICASSP.2003.1199998.’

[4] http://cs229.stanford.edu/proj2018/report/21.pdf

[5] Oramas, S., Barbieri, F., Nieto, O. and Serra, X., 2018. Multimodal Deep Learning for Music Genre Classification. Transactions of the International Society for Music Information Retrieval, 1(1), pp.4–21. DOI: http://doi.org/10.5334/tismir.10