Project report for IITB DS203 Programming and Data Science 2021: Music Genre Classification

Vineet Bhat
Roll No: 180260042

Souvik Pal
Roll No: 18D110011

Anish Chaurasiya
Roll No: 180260007

Aryan
Roll No: 180260009

***Abstract*—Rapid advances in machine learning algorithms have paved the way for high quality prediction and classification results in a wide variety of tasks. Music Genre Classification (MGC) is an important problems statement in the subject of Music Information Retrieval (MIR). There are more than 40 different types of genres existing and this classification problem becomes a key to a large number of downstream tasks. In this work, we explore this problem statement in detail, provide extensive data analysis on a few popular datasets and compare various machine learning and deep learning frameworks on the classification task. Our experiments demonstrate that neural networks achieve the best classification accuracy rate of 99%. We also provide a novel method to create a new dataset for music genre classification to aid future research.**

***Index Terms*—music, classification, machine learning, augmentation**

I. Introduction

For years, people have downloaded and purchased music from both offline and online stores. Majority of us have our favourite genre and we prefer to buy and listen music specific to that genre. However, not all music available is conveniently classified into its genre. Music genre classification is an important but a widely unexplored branch. Majority of the music datasets do not contain any type of genre classification. According to the Music Genre List[], there exists 41 primary genres of music and within those primary categories, 337 sub categories of music. However, majority of the music datasets do not follow this convention. With such large amounts of music available across many languages, manual genre classification is clearly unfeasible. Machine learning helps in solving this problem.

Music genre classification has mainly 2 steps - 1) Generating features and 2) Using algorithms to classify the data. Audio data is usually used in the raw forms as well converted into Mel-spectrogram features before passing it through classification algorithms. We experiment with both these methods of feature engineering and experiment with various Machine Learning and Deep Learning models and compare results. We categorize our experiments into 2 classes - 1) Experiments on traditional machine learning algorithms and 2) Experiments on Deep Learning frameworks. In traditional ML algorithms, we obtain the best classification accuracy of 74% using a Random Forest classifier and using deep learning, we obtain the best classification accuracy of 96% using InceptionV3.

II. Background

As mentioned before, this problem statement received its boost very recently at the start of the 20th Century. Scaringella et al [1] provides an extensive survey into this task and mentions majority of the ML models used for this classification problem. Tzanetakis and Cook [2] were one of the first to use ML algorithms for music genre classification. In this paper, they created the popular GTZAN dataset, which has become an industry and academia standard for MGC. Changsheng Xu et al [3] used Support Vector Machines acting on raw audio data for classification. Riedmiller et al [4] was the first one to use Unsupervised Learning in creating feature vectors for music data specific for MGC. Scheirer [5] provided a novel approach for this problem - utilizing a beat tracking algorithm which is used as a feature vector for classification. Aaron et al [6] was the first one to use Mel spectrograms as input features for classification models. Gwardys et al [7] used transfer learning by using a pretrained DL model on image classification and used this to finetune on a spectrogram dataset for MGC.

III. Methodology

A. Datasets

There are 2 important datasets for MGC - 1) FMA Dataset and 2) GTZAN Dataset.

1) FMA Dataset: Free Music Archive (FMA)[8] is a largescale dataset used for a wide variety of problems in Music Information Retrieval. It contains more than 8000 hours of music data across 16k+ artists and 14k+ music albums. The data has been classified into 161 genres. The authors of the dataset also provide features for these music files which can be directly used for downstream tasks.

Track listens is one of the most important metrics in analysing music data. Across different genres, we can identify the ones which are the most popular using the corresponding track listens. Figure 1 provides a visual representation of track listens across some of the popular genres.

Music data often consists of various languages. During feature extraction, sometimes its important to use the language information to model features differently for specific languages. For eg, the English language has clearer pronunciations compared to Spanish and Italian, and as a result, its features are more prominent and give better results in various MGI tasks. Fig 2 gives the breakdown of the FMA dataset into various languages. Fig 3 provides the frequency histogram with respect to the duration of the audio files.

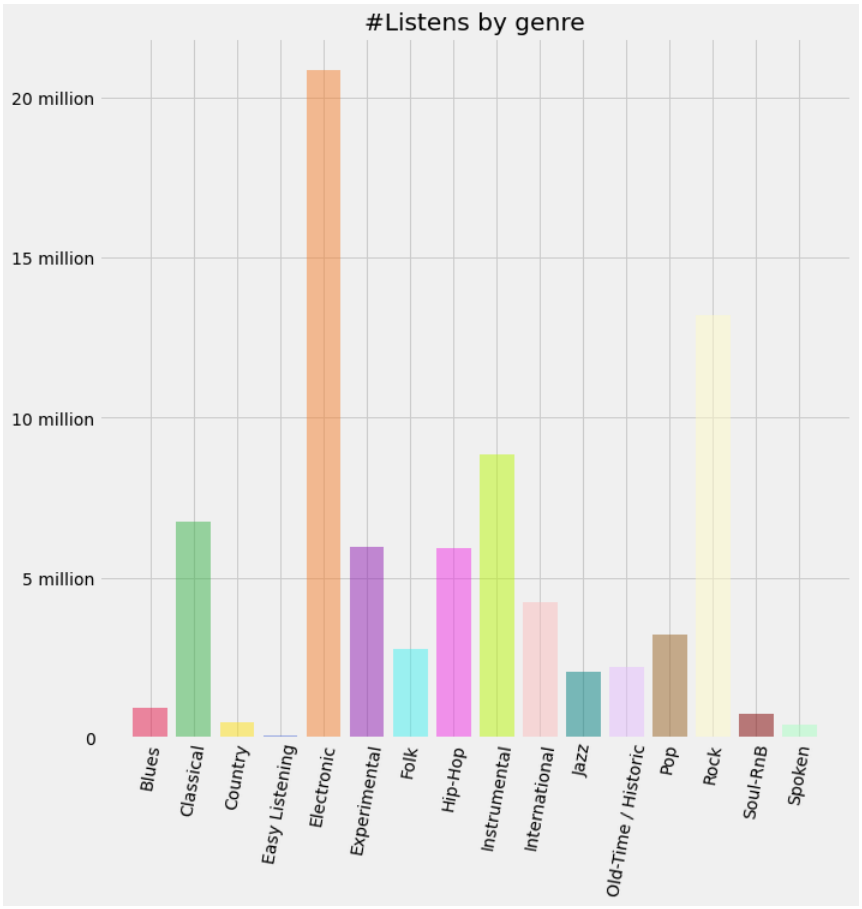

Fig. 1. Track listens across various genres

2) GTZAN Dataset: The GTZAN dataset [9] has become an industry standard for MGC. This dataset was so popular that it is often called the 'MNIST of sounds'. The corpus contains 10 genres of 100 audio files each. Each audio file is 30 seconds long. The corpus also contains the mel spectrogram images of all these audio files which makes it easier for researchers to directly work on images. Additionally the dataset also contains audio features for each file stored in 2 csv files. One file has for each song (30 seconds long) a mean and variance computed over multiple features that can be extracted from an audio file. The other file has the same structure, but the songs were split before into 3 seconds audio files (this way increasing 10 times the amount of data we fuel into our classification models). With data, more is always better while training ML models.

Extensive explorative data analysis has been performed in our submitted code.

B. Traditional Machine Learning algorithms

We performed experiments on the GTZAN dataset as well as a custom dataset created by us. The GTZAN dataset also contained feature vectors corresponding for every audio file. Two files with features of audio cropped to 3 sec and audio cropped to 30 sec were provided. We directly applied various ML algorithms to these features and report the accuracy. Thereafter, we also performed our own feature engineering on 3sec and 30sec clips through calculating the chroma and mfcc features of the datasets creating a 75 x 1 feature vector for every audio file. These feature vectors were passed through 7 machine learning algorithms - Naive Bayes, Stochastic Gradient Descent classifier, K-Nearest Neighbour, Decision Tree, Random Forest, Support Vector Machine and Logistic Regression.

We also provide a novel approach in creating our own custom data set for music genre classification. We utilize the Google audio dataset [10] and write our own scripts to extract and categorize the audio files. This dataset contains time stamps of YouTube videos and the corresponding genres. We use our scripts to download these youtube videos, convert them to the lossless wave files and categorize them into the provided genres. This dataset contained audio files classified into 7 genres. We applied feature engineering to calculate a 97-dimensional spectral feature vector for every audio file. Thereafter, we applied three algorithms - Logistic Regression, Random Forest and Support Vector Machine on these features.

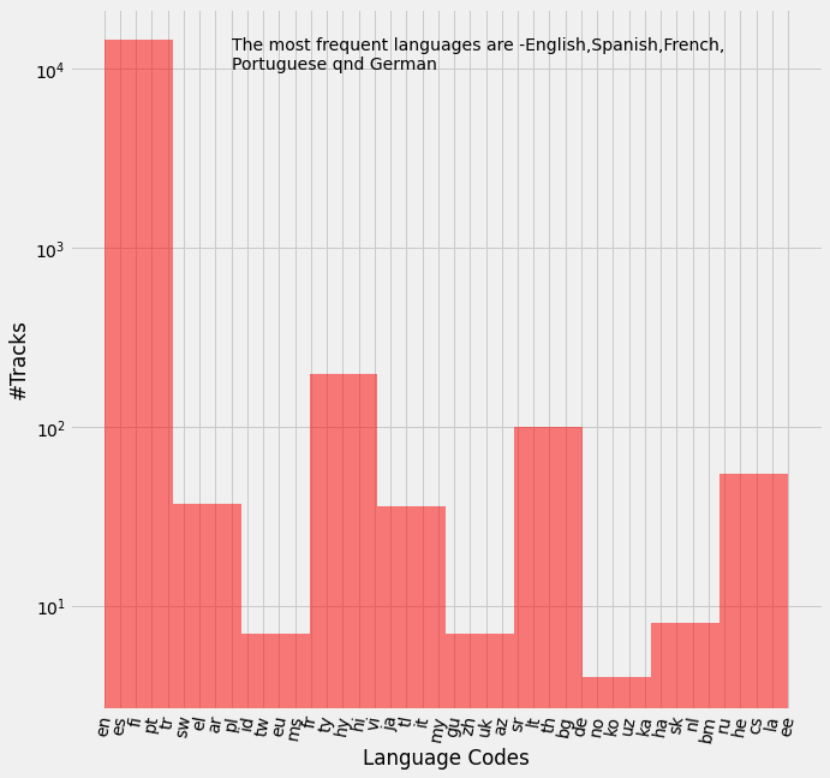

Fig. 2. Track listens across various genres

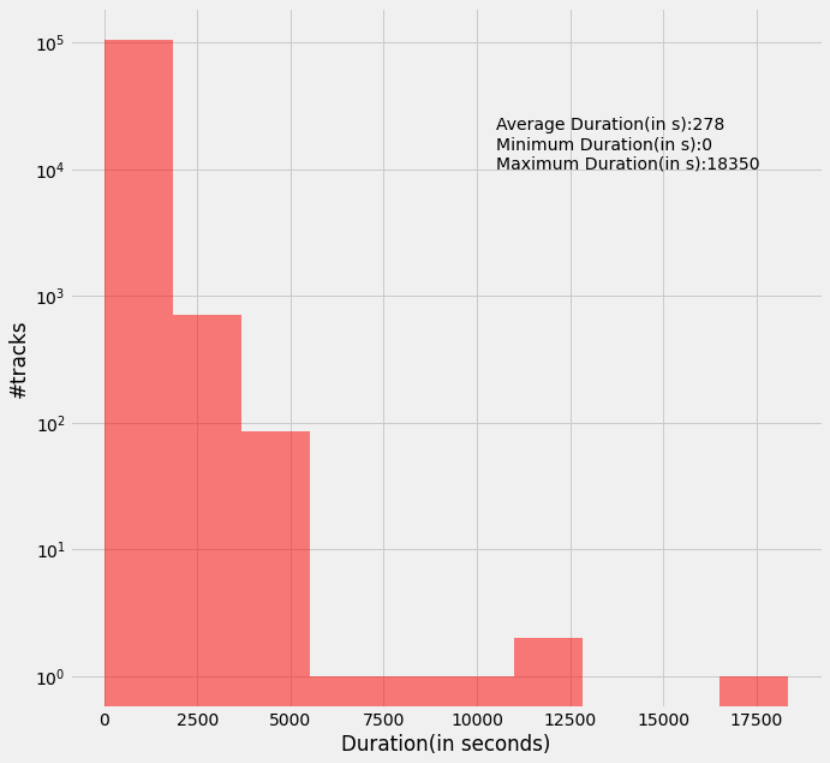

Fig. 3. Frequency histogram of duration of sounds

Dataset	Naive Bayes	Stochastic Gradient Descent	KNN	Decision Tree	Random Forest	SVM	Logistic Regression
GTZAN - 3 sec	51.52%	66.8 %	80.41%	65.93%	**80.81%**	74.78%	68.4%
GTZAN - 30 sec	56.33%	67 %	61.67%	59.33%	**78%**	71.33%	67%
GTZAN - 3 sec (FE)	59.33%	66.67 %	61%	44.33%	65.67%	**71.33%**	69%
GTZAN - 30 sec (FE)	46.35%	62.79 %	70.59%	58.64%	73.71%	**74.30%**	65.91%
Custom Dataset	47.16%	48.76%	43.64%	45.41%	**52%**	50%	52%

TABLE I
RESULTS OF TRADITIONAL ML MODELS; FE STANDS FOR DATASETS WHERE WE PERFORMED OUR CUSTOM FEATURE ENGINEERING

C. Deep Learning models

Deep learning models are data hungry. Since MGC is a low resource problem statement, utilizing neural networks is a tricky task. Since the GTZAN dataset does not have extensive data, we used spectral augmentation techniques to achieve good results. We experimented with both the available spectrogram images as well as generated our own mel spectrograms to augment the data and train the models.

IV. RESULTS

A. Traditional ML frameworks

Using the 3 sec cropped features provided with the GTZAN dataset, we achieve the best accuracy of 80.82% and an accuracy of 78% using the 30 sec features of the GTZAN dataset. Both these results were obtained using the Random Forest Classification algorithm. Through our feature engineering, we obtain the best classification accuracy of 71.33 % using SVM on 3sec clips. However, the best results on 30sec clips through our feature engineering were not promising as we could obtain only a best classification accuracy of 50 % using SVM. Table 1 provides the summary of our various experiments.

B. Deep Learning Models

We initially experimented with the feature vectors provided by the GTZAN corpus. For this purpose, a simple dense NN with multiple layers was used (Sequential Model A). We achieved a test set accuracy of 92 % using this approach. However, when we experimented on the spectrogram images using simple 1D-CNN and 2D-CNN models, we did not achieve good results. The train set accuracy was high and around 99% however the test set accuracy saturated at around 60%. This gave clear indications of overfitting, a phenomenon where the model is so powerful that it starts remembering the data rather than learning it. Overfitting can be solved by increasing the dataset, but this was a difficult and time consuming task. Hence, we decided to perform time masking and loudness masking augmentations on the original dataset to generate new data. This was combined with the original data and trained again on 1D-CNN and 2D-CNN models. Using augmentations, the problem of overfitting was solved, as the model achieved an excellent test set accuracy of about 97%. Table 2 shows our results across various experiments.

V. CONCLUSION

Through our experiments, we demonstrate the performance of wide variety of machine learning algorithms on the Music Genre Classification task. As a part of the project, we perform extensive data exploration on the FMA dataset and the GTZAN dataset and also provide a method to generate a new dataset utilizing the google audio dataset. Through spectral augmentations and a simple 2D-CNN model, we achieve a very high classification accuracy of 98%. Therefore, in our project, we chose an important problem statement and utilized various ML algorithms taught in class to our best use in tackling the problem and achieving a high accuracy.

VI. CONTRIBUTION BY EACH TEAM MEMBER

1) Aryan: Exploratory Data Analysis on FMA and GTZAN datasets

2) Souvik: Training Traditional ML algorithms on GTZAN dataset and novel method to create a new Music Genre dataset

3) Anish: Training various DL algorithms and experiments

4) Vineet: Training DL algorithms with Anish, documentation of report and creating the demo notebook

REFERENCES

[1] N. Scaringella, G. Zoia, and D. Mlynek. "Automatic genre classification of music content: a survey". In: *IEEE Signal Processing Magazine* 23.2 (Mar. 2006), pp. 133–141. ISSN: 1053-5888. DOI: 10.1109/MSP.2006.1598089. URL: http://ieeexplore.ieee.org/document/1598089/ (visited on 11/25/2021).

Dataset	Model Name	Epochs trained	Train Set accuracy	Test set accuracy
GTZAN - available features	Sequential Model A	600	99.69%	92%
GTZAN - available spectrograms	1D-CNN	20	99.87%	56.78%
GTZAN - available spectrograms	2D-CNN	20	99.75%	59.30%
GTZAN - available spectrograms + augmentation	1D-CNN	20	99.81%	97.62%
GTZAN - available spectrograms + augmentation	2D-CNN	20	99.86 %	98.05 %

TABLE II
RESULTS OF DEEP LEARNING MODELS

[2] G. Tzanetakis and P. Cook. “Musical genre classification of audio signals”. In: *IEEE Transactions on Speech and Audio Processing* 10.5 (July 2002), pp. 293–302. ISSN: 1063-6676, 1558-2353. DOI: 10.1109/TSA.2002.800560. URL: https://ieeexplore.ieee.org/document/1021072/ (visited on 11/25/2021).

[3] Changsheng Xu et al. “Musical genre classification using support vector machines”. In: *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).* Vol. 5. Hong Kong, China: IEEE, 2003, pp. V–429–32. ISBN: 9780780376632. DOI: 10.1109/ICASSP.2003.1199998. URL: http://ieeexplore.ieee.org/document/1199998/ (visited on 11/25/2021).

[4] Jan Wülfing and Martin Riedmiller. “Unsupervised learning of local features for music classification”. In: (Jan. 2012).

[5] Eric D. Scheirer. “Tempo and beat analysis of acoustic musical signals”. en. In: *The Journal of the Acoustical Society of America* 103.1 (Jan. 1998), pp. 588–601. ISSN: 0001-4966. DOI: 10.1121/1.421129. URL: http://asa.scitation.org/doi/10.1121/1.421129 (visited on 11/25/2021).

[6] Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. “Deep content-based music recommendation”. In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'13. Lake Tahoe, Nevada: Curran Associates Inc., Dec. 2013, pp. 2643–2651. (Visited on 11/25/2021).

[7] Grzegorz Gwardys and Daniel Grzywczak. “Deep Image Features in Music Information Retrieval”. In: *International Journal of Electronics and Telecommunications* 60.4 (Dec. 2014), pp. 321–326. ISSN: 2300-1933. DOI: 10.2478/eletel-2014-0042. URL: http://journals.pan.pl/dlibra/publication/101675/edition/87690/content (visited on 11/25/2021).

[8] admin. *Music Genre List - A complete list of music styles, types and genres*. en-US. Dec. 2012. URL: https://www.musicgenreslist.com/ (visited on 11/25/2021).

[9] URL: http://marsyas.info/downloads/datasets.html (visited on 11/25/2021).

[10] *AudioSet*. URL: https://research.google.com/audioset/ (visited on 11/25/2021).