Machine Learning Project Proposal

Music Genre Classification

Problem Statement:

Many companies nowadays use music genre classification for the purposes such as recommendation, or just to recognize the music like shazam. One of the best ways to categorize or classify music is by its genre. For this purpose, supervised machine learning approaches are considered. Music Genre Classification reads the audio files as input dataset and classifies them based on its genre. This can be done using Convolution Neural Networks (CNN). For this purpose, we plan to use Keras which is an open-source neural network library as it supports convolution neural networks and recurrent networks and the combination of two. It runs on top of the TensorFlow, CNTK, or Theano.

In the Music Genre Classification system, the aim of the proposed system is to compare the classic approach of extracting the features using a classifier and also using a Convolution Neural Networks on the representations of the audio. This report explores the application of machine learning algorithms to identify and classify the genre of the given audio file. The features which will be extracted by sampling the audio into four random parts, thus augmenting the data into 8000 clips as a raw input file [1]. This raw data is also converted into mel-spectrograms. The features from both these input data are then fed to the Convolution Neural Network which are further trained to classify the given audio file.

Our approach consists of collecting the data, preprocessing it so that we can extract features such as time domain and frequency domain, building the model, training the classifier by feeding it with the data. For which, CNN based deep learning model will be implemented and compared to check which model works the best by using evaluation metrics on every model.

Dataset:

The data chosen is from the GTZAN genre collection dataset. This dataset provides us with 1000 30-second audio clips, all labeled as one out of 10 possible genres and presented as .au files.

Link: http://marsyas.info/downloads/datasets.html

From each clip, sampling is done at four random locations, thus augmenting the data into 4000 clips of two seconds each. This leaves us with 44100 features for the raw audio input. Also, this raw audio file is converted into mel-spectrogram for performance increase. Mel-spectograms are a commonly used method of featuring audio as they closely represent how humans perceive audio (i.e. in log frequency). To convert files into mel-spectrogram, Fourier transform must be applied on the sampled raw input files.

The dataset consists of many folders containing audio files (having .au extensions) categorized into following music genres:

- Blues
- Classical
- Country

- Disco
- Hip-hop
- Jazz
- Metal
- Pop
- Reggae
- Rock

Steps and Approaches:

The major aim of the Music genre classification is as the name suggests, to be able to successfully read the data so that it can be trained well, and the system should be able to correctly classify the unobserved data into its respective genre category.

1. **Data pre-processing**: Firstly, the data needs to be in an understandable format, for which we will transform the raw data into an acceptable format for further processing in the models. We will drop the unwanted or irregular behaviors of the data for better results.

2. **Comparison:** For this project, we plan on using different machine learning models (e.g.: SVM, kNN, CNN etc) and compare the accuracies amongst them.

   Support Vector Machines (SVM) – This is a supervised learning algorithm, which can be used for both regression and classification. It finds a hyperplane in given features and distinctly classifies the data.
   K- Nearest Neighbors – K Nearest Neighbor algorithm centers the classes of data by choosing the nearest data samples most relevant to the class. Using kNN we will segregate the music genre based on frequency and other aspects of music files.

   Convolution Neural Network (CNN) – This is a deep learning algorithm which, based on features of input data, generates weights and biases, to learn from it. It uses multilayer perceptrons and relatively minimum data pre-processing is needed for it.

Related works:

[1] This is a follow up on the research paper on Music Genre Classification where the author(s) has implemented a variety of classification algorithms with different types of input. Here, the author has experimented with the RBF kernel support vector machine, k-nearest neighbors, a basic feed-forward network, and an advanced convolutional neural network. The dataset is obtained from GTZAN (same as ours) for all the musical data. The input to the algorithms were raw amplitude data as well as transformed mel-spectrograms of that raw amplitude data. Mel-spectrogram represents an acoustic time-frequency representation of a sound. Then displayed an output of a predicted genre out of 10 common music genres. By converting the raw audio into mel-spectrograms, it produced better results on all the implemented models, with convolutional neural network showing better accuracy than the rest. In the results however, the models couldn’t categorize pop music whilst retaining strong performance on classical music. As the pop music graphs are more scattered comparatively, this caused discrepancies in the results as there’s lack of distinct style in case of pop music whereas classical music has a much clearer definition.

Related to our approach:
The data used here is same as what we are using. The preprocessing of the data and the approach to improving the performance by converting the raw data to mel-spectrogram is like the approach we are doing.

In [2], the author has widened the approach for automatic music genre detection and tagging using convolutional neural networks. The author is using the same dataset as us, which is GTZAN. Along with it, he is even using MagnaTagATune using the audio clips mel-spectograms as input. For GTZAN, the dataset is splitted into 33% for validation and the rest for training. Training is done using Adam optimizer with learning rate of 0.001 and decay of 0.01. For MagnaTagATune, the dataset is splitted into 20% for test and 80% for training. 20% of of the training dataset is used for validation. Training is done using Adam optimizer with an initial learning rate of 0.0001 and exponential learning rate decay. He calculated accuracy for the same with a decent accuracy as a result. Maybe the author didn't train for enough epochs, as he couldn't get accuracies more than 85% for either of the datasets.

Related to our approach:
The data used here is the same as what we have used. Rest other approaches are impressive and could be considered in the future; especially the Adam Optimizer methodology. But other than using CNN, K- Nearest Neighbors, nothing else is related to the approach we are using.

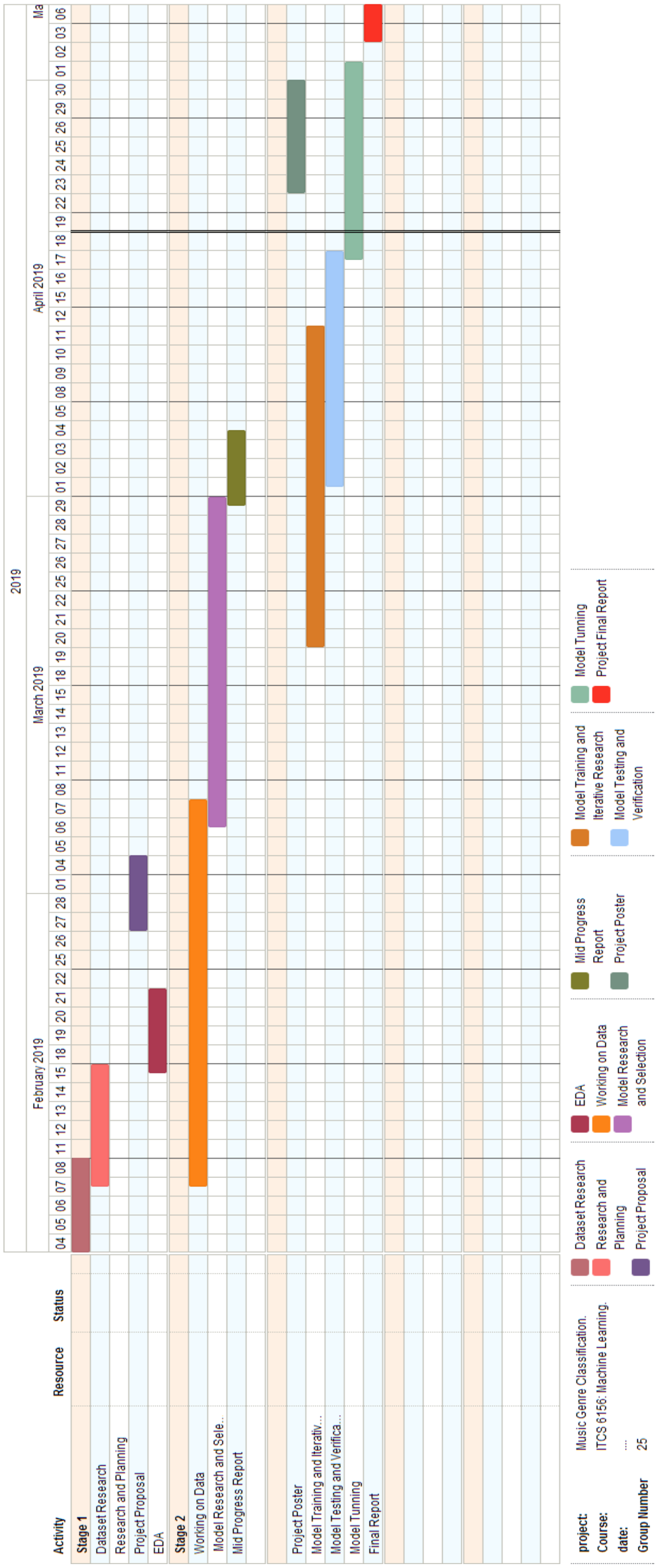
Approach overview:

- We have to clean the data and pre-process it
- We want to combine both the methods we found in the related work and build a solution using both k-means and CNN.
- We can first use SVM to reduce the dimensions
- Then the k-means to reduce the dimensionality even further
- and then use CNN with Adam optimizer and ReLU activation function

Tech Stack:

- OpenCV for SVM (Support vector machine)
- Python
- Pandas
- and Scikit-learn.

The timeline for work plan:



Team members:

Mayuri Kingre: 801073866

Sayali More: 801076191

Nishita Kalyanpur: 801076677

Grishma Varma: 801076310

Team Members	Contributions
Mayuri Kingre	1. Data Collection 2. Pre-processing the dataset a. Data Reading b. Data Cleaning. 3. Implementing Support Vector Machine. 4. Using evaluation metrics
Sayali More	1. Pre-processing the dataset a. Using Optimizers 2. Implementing K- Nearest Neighbor. 3. Comparing the different model's accuracy.

Nishita Kalyanpur	1) Preprocessing the dataset: a. Feature Extraction/Feature Selection. b. Converting the raw input data to mel-spectrograms 2) Data Visualization and Analysis. 3) Calculating accuracy. 4) Analysing model performances using different performances metrics.
Grishma Varma	1) Selecting dataset. 2) Data Partitioning. 3) Implementing Convolution Neural Network. 4) Implementing Cross validation. 5) Using Encoders

Questions that you want to answer during the projects :

This project will answer the following questions:

1. How machine learning models can be useful for music and media related applications?

2. Which model exhibits the best compatibility with our data set?

3. How does one genre vary from another and based on which aspects of the songs?

Things that we expect to learn:

In this project,

1. We will learn how to train a model, and handle data set pre-processing

2. Learn implementation of various machine learning models and their comparisons

Is your idea a novel?

There are multiple applications of music-related machine learning applications. We would try to achieve optimal performance and better results than existing applications. The approach used in this project will also help us in applying it to other similar domains.

References:

[1] Huang, Derek A., Serafini Arianna A., Pugh Eli J., et al. “Music Genre Classification”. 2018. http://cs229.stanford.edu/proj2018/report/21.pdf

[2] Flores, Miguel, et al. “Deep Music Genre”. 2017 http://cs231n.stanford.edu/reports/2017/pdfs/22.pdf

[3] G. Tzanetakis, P. Cook, et al. “Musical genre classification of audio signals”, IEEE, 2002 https://ieeexplore.ieee.org/document/1021072

[4] Michael Haggblade, Yang Hong, Kenny Kao, et al. “Music Genre Classification”. 2011. http://cs229.stanford.edu/proj2011/HaggbladeHongKao-MusicGenreClassification.pdf