Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

CSU44061-202223 MACHINE LEARNING
Group Project Report

Group 35_2, 21344349 21344426 22312293

December 3, 2022

Contents

1 Introduction

The topic we chose for this project was Music Genre Classification, we have all used music apps, and have come across the suggested by Genre feature, but how accurate is it? Is it accurate enough to be used with a Voice assistant to Classify the genre of the music a user listens to, so that a matching list of songs in the same genre can be provided, there are several use cases of a more accurate genre classification model. This was the main reason for us to choose this project, to solve the problem of accuracy by using low-level features based on music abstractions, which will be further explained in the report.

1.1 Input and Output

The input to our algorithm is an wav audio file which is then processed into a 2d array , we then use CNN model to train and predict the labels of the audio.

1.2 Results

As we are using the low level features to abstract the data of a music file to gets its floating point time series, and use the time series to find the 3 main features of an audio file i.e Spectogram: which is a visual way of representing the signal loudness of a spectrum of a signal as it varies with time. Chroma feature: which is a tool to analyze the features whose pitches can be meaningfully categorized and the approximate value for tuning is to the equal tempered scale. and Mfcc which are Mel frequency spectral coefficient's (MFCCs) of a signal are a small set of features (20 in our case usually between 10 to 20) which concisely describe the overall shape of a spectral envelope.

2 Dataset and Features

2.1 Description

To gather the data we went to YouTube as YouTube can provide us with reliable data in bulk, and as other APIs were providing us with maximum 50 to 100 songs, therefore, a choice to download a playlist from YouTube was made, also as downloading multiple playlists would have resulted in a lot of repeated songs but having a single playlist from a source which has enough songs without repetition.We imported 200 files per genre, After importing these files we first trimmed them to 5 files with duration of 20 secs, thus creating 5 different audio files,now resulting in a data set of 1000 files per genre.

2.2 Pre-processing of Data

As we needed the lower level audio data to train our model. We are using the librosa package (a package in python to to analyse music and audio files) to create the following features:-
Zero crossing rate: Zero crossing rate is the rate at which a sound signal goes from positive to zero to negative and vice versa. It is used extensively for differentiating sounds from percusion instruments in training models such as ours.

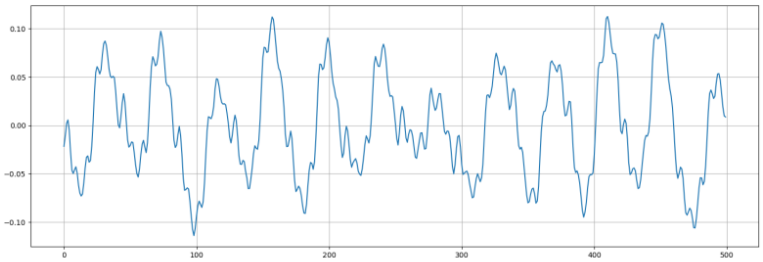


Figure 1: Zero crossing rate of our sample audio

Spectral centroid: In a spectrogram, the spectral centroid is used to measure and locate the centre of mass of the spectrum. Figure 2 left shows the spectral centroid for a blues music sample whereas Figure 2 right shows the spectral centroid for metal.

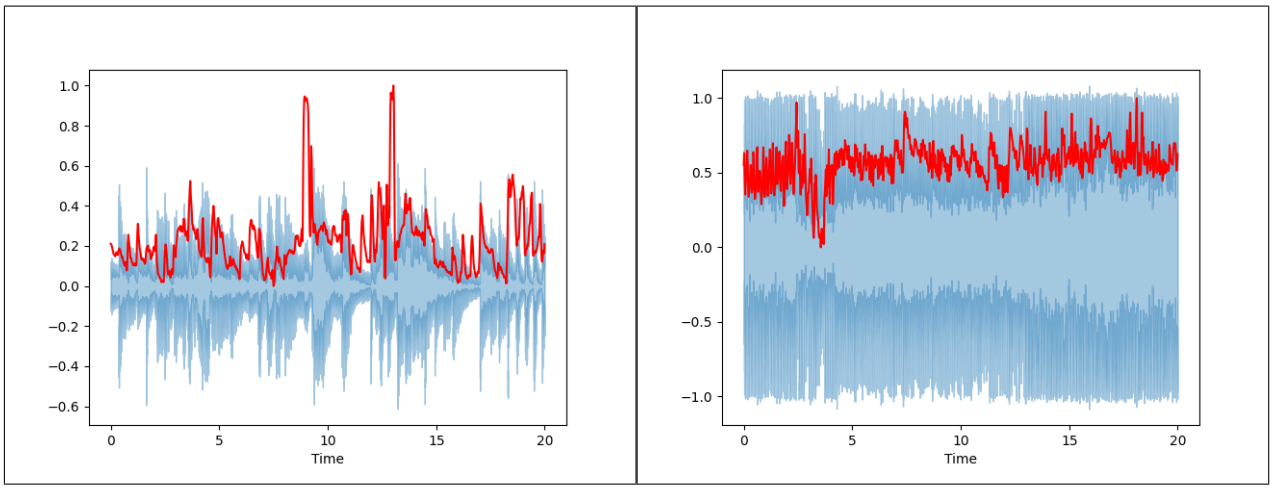


Figure 2: left=Spectral centroid for Blues audio file , right=Spectral centroid for metal audio file

As seen in the figures, the blues spectral centroid in the fig 2 has its peak in Fig 2. On the other hand, the spectral centroid for metal peaks towards the end and its centre of mass also fluctuating rapidly but generally remaining at a higher amplitude than blues music.

Spectral Rolloff: It is the frequency blow which a specified percentage of the total spectral energy lies. The default of this speceified percentage in librosa for spectral rolloff is 85 %. 3 shows the spectral rolloff for our blues audio file

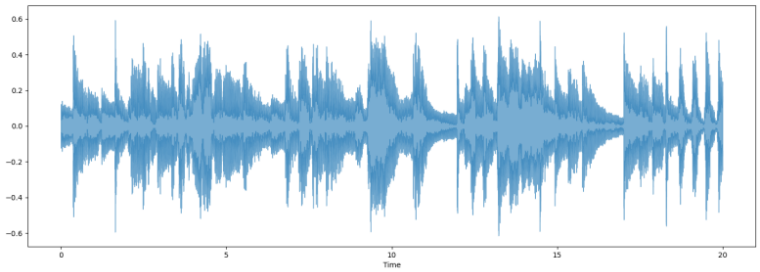


Figure 3: Spectral roll off for the blues audio file

Spectral Bandwidth: Also known as spectral spread is derived from spectral centroid.The spectral range of interest around the centroid,basically the sum of maximum deviation of the signal form both sides is considered as the bandwith of the signal at the frame
Mel-Frequency Cepstral Coefficients: Mel-Frequency Cepstral Coefficients is used to extract features from audio that correctly distinguish the several characteristics of the human voice by giving the overall shape of the vocal tract and can help us classify the music more accurately. Fig 6 shows the Mel-Frequency Cepstral Coefficients for the sample audio file with a feature set of 20. **harmony:** The characteristics of an audio files which are not distinguishable from human ears and are low level.
Perceptual: They are the shock waves representing sound rhythm and emotions inside an audio file
Tempo: This is basically beats per minute. it is the measurement of how fast or slow a piece of music is

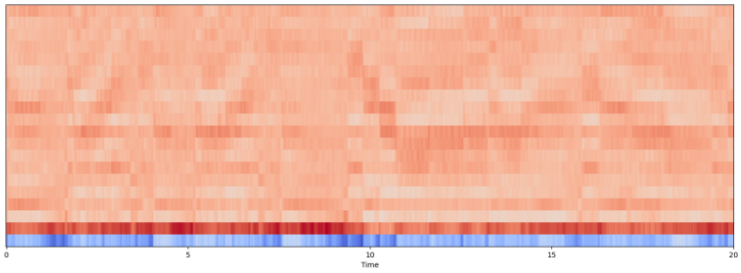


Figure 4: MFCC with 20 feature set

performed

RMS: An acronym for root mean square, which is a metering tool that measures the average of loudness of an audio track within a window of roughly 300 millisecond.

As all the features except tempo gives us a time series in a 2d array format,therefore,to populate our data.csv file using theses features, we calculate the mean and variance of the 2d array. As we are doing the data and feature extraction individually for all 10 genres, we label all the audios by making label as another column in our data.csv while also adding name and length of the audio file, making the total columns 60. Now we have a data.csv file with a shape of (10000,60). Time taken to prepare this file was 236 minutes 20 seconds and 3 millisecond through a python script.
As explained further in the report, this part goes as an alternate to Conv2d.

3 Methods

Models : -
The models we are using to train our data are KNN and CNN. While CNN is the more sophisticated model that we are using for training our model, we are also keeping KNN to compare as our baseline.

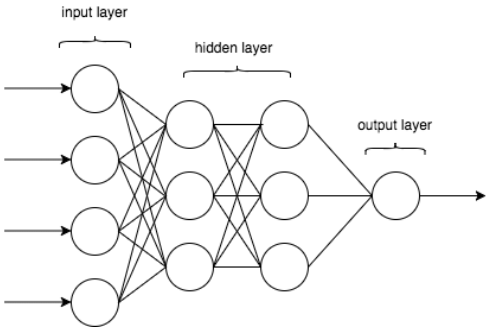


Figure 5: Image Source

CNN - Finally, we are using CNN to train our model. As we are not using images as our training data, we have features extracted from audio files to train our model because a convolution network cannot be used on audio files directly. Although, using spectrograms and Mel-Frequency Cepstral Coefficients images can let us use convolution layers in our neural net, it doesn't yield as much accuracy as feature extracted from audio files does. Our dataset, as described in the preceding sections, is a 2D matrix which is created in a way that it has 58 features extracted from each audio file and can act as a replacement for the output from convolution layers had we been using images. We are, therefore, taking the feature-extracted dataset as our

input layer to feed it into our dense layers. A dense layer can be defined as a layer that is deeply connected with its preceding layer, i.e. the neurons of the layers are connected to every neuron of its preceding layer as shown in 5. The hidden layer in this image can represent multiple dense layers as used in our model. The output shape from our first dense layer is (None,512), our second dense layer is (None, 256), and our third dense layer is (None, 128). These dense layers use relu activation function to perform filtration to extract fine features based on the output of the previous layer. Our final layer is a dense softmax layer that does the final classification.
6 shows these parameters in details.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 512)               30720

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 256)               131328

 dropout_1 (Dropout)         (None, 256)               0

 dense_2 (Dense)             (None, 128)               32896

 dropout_2 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 128)               16512

 dropout_3 (Dropout)         (None, 128)               0

 dense_4 (Dense)             (None, 10)                1290

=================================================================
Total params: 212,746
Trainable params: 212,746
Non-trainable params: 0
```

Figure 6

4 Experiments

We did a 70:30 split on our data. As the total number of data points in our dataset was 10000, our training dataset had 7000 data points and our test dataset had 3000 data points. To run our CNN model, We took the epoch value as 60 by looking at the loss model graph as shown in 7. This is because as the number of iterations for epoch increases, the validation loss decreases till it reaches a minimum value. On further increasing the number of iterations for epoch, the the validation loss increases due to overfitting. This validation loss is shown in the graph in 7

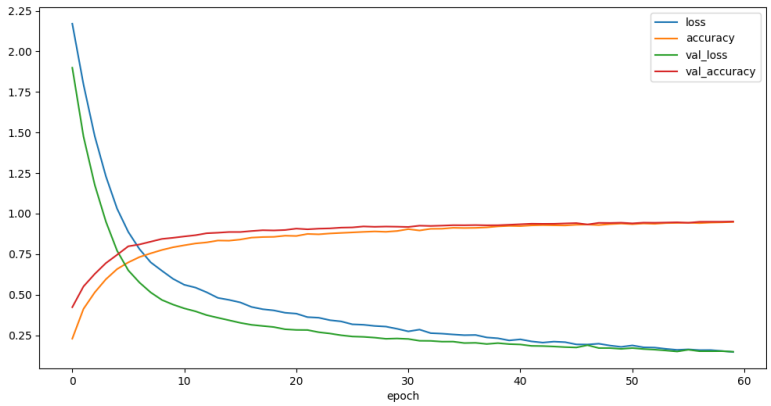
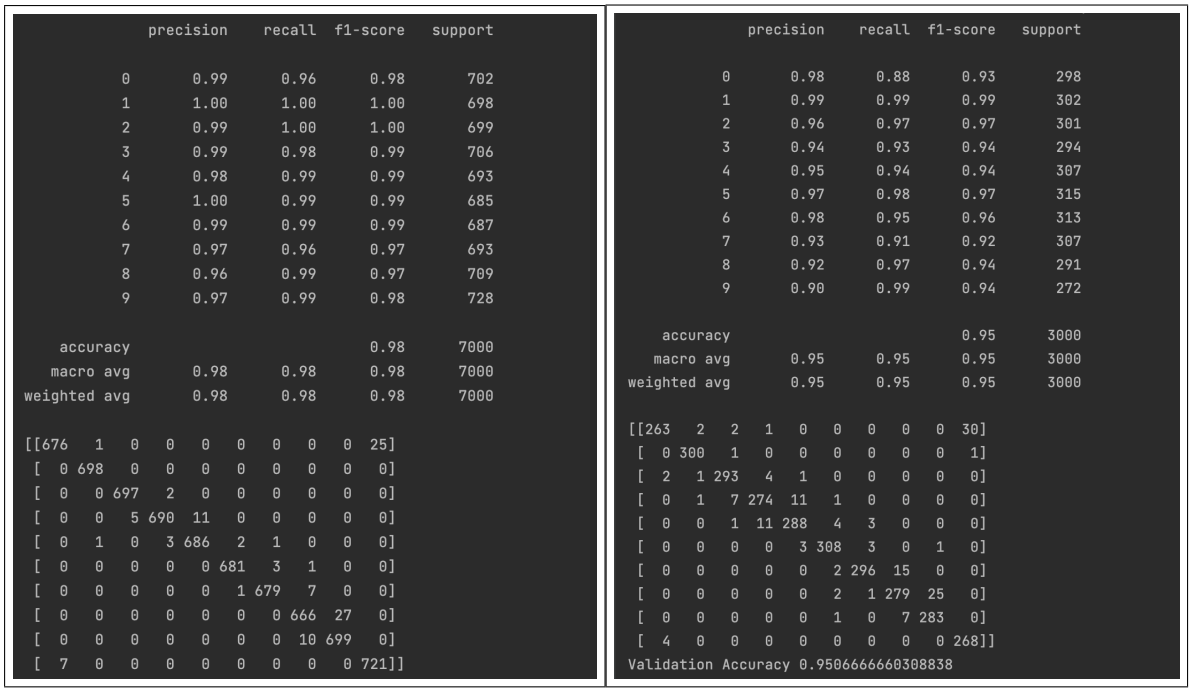


Figure 7

As our Baseline Classifier, we are using a KNN model. It gets us an accuracy of 79.29% with 5 fold cross

validation. We are using KNN because it is one of the more sophisticated machine learning algorithm for multi-class classification.It is a supervised machine learning algorithm which is used to perform multilabel classification. It calculates the euclidean distance from the neighbors and select the 'K' neighbor which has the least distance. We are using multiple 'K' values and doing 5 cross validation to find the accuracy.However, our accuracy with CNN was 97% whereas the one with our baseline is 59%.

5 Results



	precision	recall	f1-score	support
0	0.99	0.96	0.98	702
1	1.00	1.00	1.00	698
2	0.99	1.00	1.00	699
3	0.99	0.98	0.99	706
4	0.98	0.99	0.99	693
5	1.00	0.99	0.99	685
6	0.99	0.99	0.99	687
7	0.97	0.96	0.97	693
8	0.96	0.99	0.97	709
9	0.97	0.99	0.98	728
accuracy			0.98	7000
macro avg	0.98	0.98	0.98	7000
weighted avg	0.98	0.98	0.98	7000

```
[[676   1   0   0   0   0   0   0   0  25]
 [  0 698   0   0   0   0   0   0   0   0]
 [  0   0 697   2   0   0   0   0   0   0]
 [  0   0   5 690  11   0   0   0   0   0]
 [  0   1   0   3 686   2   1   0   0   0]
 [  0   0   0   0   0 681   3   1   0   0]
 [  0   0   0   0   0   1 679   7   0   0]
 [  0   0   0   0   0   0   0 666  27   0]
 [  0   0   0   0   0   0   0  10 699   0]
 [  7   0   0   0   0   0   0   0   0 721]]
```

	precision	recall	f1-score	support
0	0.98	0.88	0.93	298
1	0.99	0.99	0.99	302
2	0.96	0.97	0.97	301
3	0.94	0.93	0.94	294
4	0.95	0.94	0.94	307
5	0.97	0.98	0.97	315
6	0.98	0.95	0.96	313
7	0.93	0.91	0.92	307
8	0.92	0.97	0.94	291
9	0.90	0.99	0.94	272
accuracy			0.95	3000
macro avg	0.95	0.95	0.95	3000
weighted avg	0.95	0.95	0.95	3000

```
[[263   2   2   1   0   0   0   0   0  30]
 [  0 300   1   0   0   0   0   0   0   1]
 [  2   1 293   4   1   0   0   0   0   0]
 [  0   1   7 274  11   1   0   0   0   0]
 [  0   0   1  11 288   4   3   0   0   0]
 [  0   0   0   0   3 308   3   0   1   0]
 [  0   0   0   0   0   2 296  15   0   0]
 [  0   0   0   0   0   2   1 279  25   0]
 [  0   0   0   0   0   1   0   7 283   0]
 [  4   0   0   0   0   0   0   0   0 268]]
Validation Accuracy 0.9506666660308838
```

Figure 8: left=Training accuracy and its respective confusion matrix , right=Test accuracy and its respective confusion matrix

In the figure above, we are seeing a training accuracy of 98 % and test accuracy of 95 %. Our validation accuracy is also 95%. We have also provided the confusion matrices of our results.

6 Summary

The main purpose of doing this project was to use feature extracted audio files in order to classify them in genres. This is because doing CNN on image data from each audio file to classify them gives lower accuracy results. We found out through our project that feature extraction on audio files before running CNN on it can yield much higher accuracy results than any other ML model or the conventional CNN method with images extracted from audio.

7 Contributions

We all contributed to different sections which were important for the project to come together, but we also worked on the main model training together as it was an integral part of the project and had a lot of learning experience in it, therefore we chose to work on the model together on top of working on our individual tasks

Data Collection: - Saumya Bahuguna,Chirag Saxena
Feature Extraction: - Kinjal Bhattacharyya, Saumya Bahuguna
CNN: - Saumya Bahuguna, Kinjal Bhattacharyya and Chirag Saxena
Experiments: - Chirag Saxena, Kinjal Bhattacharyya

Figure 9: Signatures