Music Genre Classification

Masab Bakhtiar | Syed M. Mahad | Rafay Akram

National University of Computer and Emerging Sciences, Karachi campus

19k-1259

19k-1283

19k-1490

Abstract

Audio processing is one of the most complex tasks in data science as compared to image processing and other classification techniques. One such application is music genre classification which aims to classify the audio files in certain categories of sound to which they belong. The application is very important and requires automation to reduce the manual error and time because if we have to classify the music manually then one has to listen out each file for the complete duration. So To automate the process we use Machine learning and deep learning algorithms and this is what we will implement in this article.

Introduction

We can define our project problem statement as like given multiple audio files, and the task is to categorize each audio file in a certain category like audio belongs to Disco, hip-hop, etc. The music genre classification can be built using different approaches in which the top 4 approaches that are mostly used are listed below.

- K-Nearest Neighbors
- Multi-Layer Perceptron Classifier
- Kernel SVM
- Convolutional neural network

Dataset

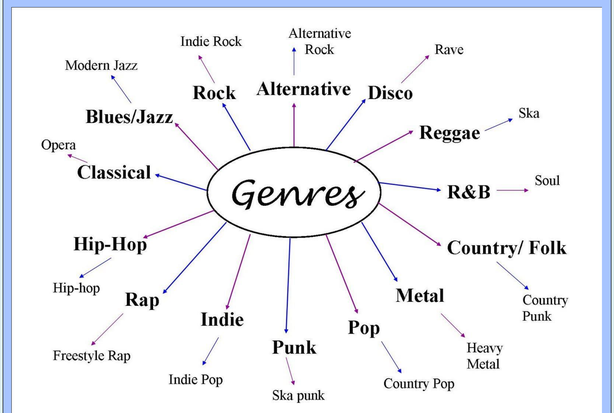
The gtzan8 audio dataset contains 1000 tracks of 30 second length. There are 10 genres, each containing 100 tracks which are all 22050Hz Mono 16-bit audio files in .wav format.
The genres are:

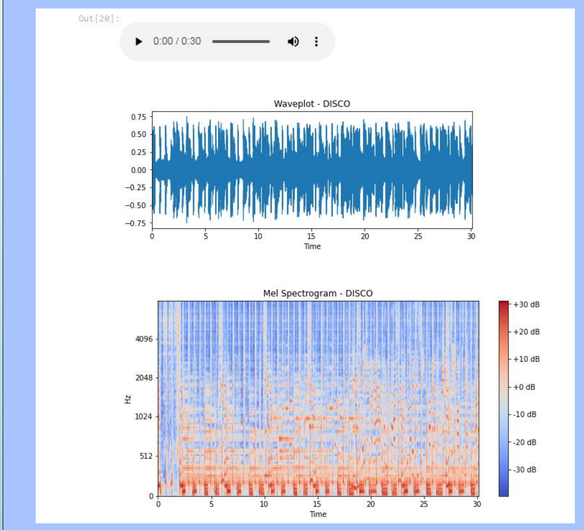
- blues
- classical
- country
- disco
- hip-hop
- jazz
- metal
- pop
- reggae
- rock

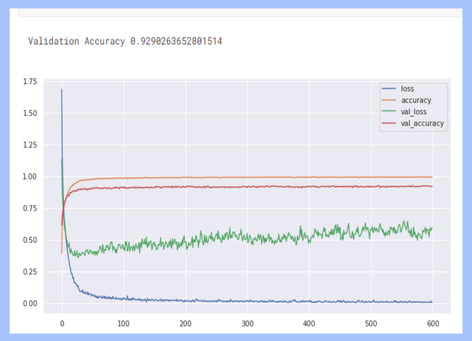
Methodology

- The dataset contains 10 genre, each containing 100 audio flles of 30sec in total 1000 audio files of 30sec 1GB data. Then we use exploratory data analysis (EDA) for the analysis of data. After data preprocessing and feature engineering, we split the data into train and test data and train the data using four technology K-Nearest Neighbors, Multi-Layer Perceptron Classifier, Kernel SVM, Convolutional neural network and then compare the accuracy of the four methods. After that, we test the model using test data and check the outputs

Figure







Results

```
[ ] sample = X_test[101]
    sample = sample[np.newaxis, ...]
    prediction = model.predict(sample)
    predicted_index = np.argmax(prediction, axis = 1)
    print("Expected Index: {}, Predicted Index: {}".format(results[y_test[101] + 1]
    , results[predicted_index[0] + 1]))
```

```
Expected Index: rock, Predicted Index: rock
```

```
[ ] sample = X_test[980]
    sample = sample[np.newaxis, ...]
    prediction = model.predict(sample)
    predicted_index = np.argmax(prediction, axis = 1)
    print("Expected Index: {}, Predicted Index: {}".format(results[y_test[980] + 1]
    , results[predicted_index[0] + 1]))
```

```
Expected Index: country, Predicted Index: country
```

Conclusions

The main thing to identify and divide the audio into different features is amplitude and frequency that changes within a short span of time.
We can visualize the audio frequency wave of amplitude and frequency with respect to time in form of a wave plot that can be easily plotted. Our model is successfully classifying the songs with approx 90% accuracy

References

Google
GitHub
Kaggle
YouTube