# Music Genre Classifier using Machine Learning

## Project Overview

This project aims to classify music into different genres using machine learning techniques. It involves loading audio files, extracting features, and training a model to predict genres.

## Working of the Program

### 1. Importing Required Libraries

The program uses several Python libraries for data handling, visualization, and audio processing:  
- `numpy`, `pandas` for numerical and tabular data handling  
- `matplotlib`, `seaborn` for visualization  
- `librosa` for analyzing and processing audio files  
- `IPython.display.Audio` for playing audio within Jupyter Notebook

### 2. Loading the Dataset

The dataset containing music genre labels and file paths is loaded using `pandas.read\_csv()`. This dataset is essential for training the classification model.  
- The dataset structure typically includes columns for file paths and corresponding genre labels.  
- The distribution of genres is checked using `value\_counts()`.

### 3. Audio File Processing and Visualization

To analyze the music files, the program performs the following steps:  
- `librosa.load(path)`: Loads an audio file and returns an array representing the waveform along with the sample rate.  
- `librosa.display.waveshow(x, sr=sr)`: Displays the waveform of an audio file.  
- `Audio(path)`: Enables playing an audio file within the Jupyter Notebook for listening.

### 4. Feature Extraction

Feature extraction is a crucial step in music genre classification. The program extracts key audio features such as:  
- \*\*MFCCs (Mel-Frequency Cepstral Coefficients)\*\*: Represents the power spectrum of the audio signal.  
- \*\*Chroma Features\*\*: Captures harmonic and melodic characteristics of the music.  
- \*\*Spectral Contrast\*\*: Measures the difference between peaks and valleys in the spectrum.

### 5. Machine Learning Model Training

The extracted features are used to train a machine learning model for genre classification. Possible models include:  
- \*\*Support Vector Machines (SVM)\*\*: Effective for high-dimensional data.  
- \*\*Random Forest Classifier\*\*: Uses multiple decision trees for classification.  
- \*\*Neural Networks\*\*: Deep learning approaches for improved accuracy.

## Main Functions and Their Working

Here are the key functions used in the program:

1. \*\*`librosa.load(path)`\*\* - Loads an audio file and returns the waveform and sample rate.

2. \*\*`librosa.display.waveshow(x, sr=sr)`\*\* - Displays the waveform of an audio file.

3. \*\*`Audio(path)`\*\* - Plays an audio file within Jupyter Notebook.

4. \*\*`pandas.read\_csv('file.csv')`\*\* - Reads the dataset containing music genre labels and file paths.

5. \*\*`value\_counts()`\*\* - Counts occurrences of each genre in the dataset.

## Conclusion

This project provides an approach to classifying music genres based on audio features. It involves data preprocessing, feature extraction, and machine learning techniques. Further enhancements may include deep learning models for improved accuracy.