Department of Electronics and Communication Engineering, MNNIT Allahabad, Session 2021-2022

Music Genre Classification

Under Guidance of

Prof. V K SRIVASTAVA

Project Presented by Group of

Divas Gupta (20185116)

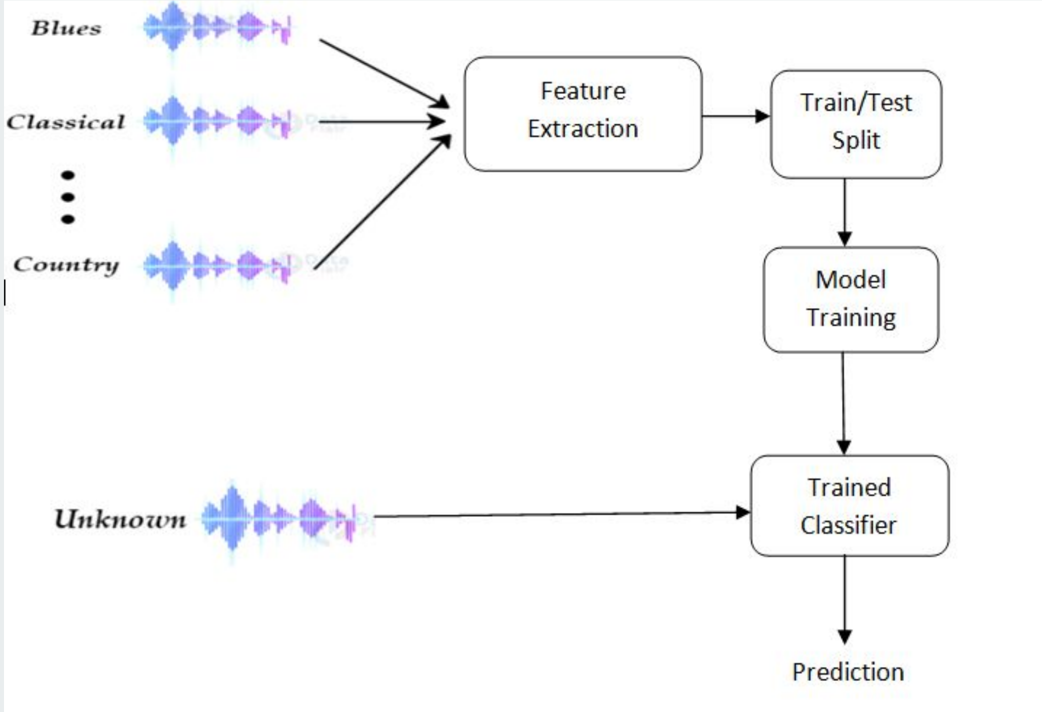
Priyanka Soni (20185053)

Dipesh Sharma Poudel (20185105)

Contents

INTRODUCTION

- There are 1264 micro - genres of popular music.
- Genre recognition is a way to recognise/predict the genre with the help of the audio track and features of dataset.
- It is a machine learning model to automatically classify different musical genres from audio files.



Why Genre classification?

- Genre classification can be of great utility to musical information retrieval systems.
- Genre is intrinsically built on the similarities between pieces of the same genre and differences between pieces of different genres.
- An automated genre recognition system would make it possible to classify and search large electronic music libraries.
- In order to generate better recommendation, user generated rating & reviews, genre or books metadata is used.

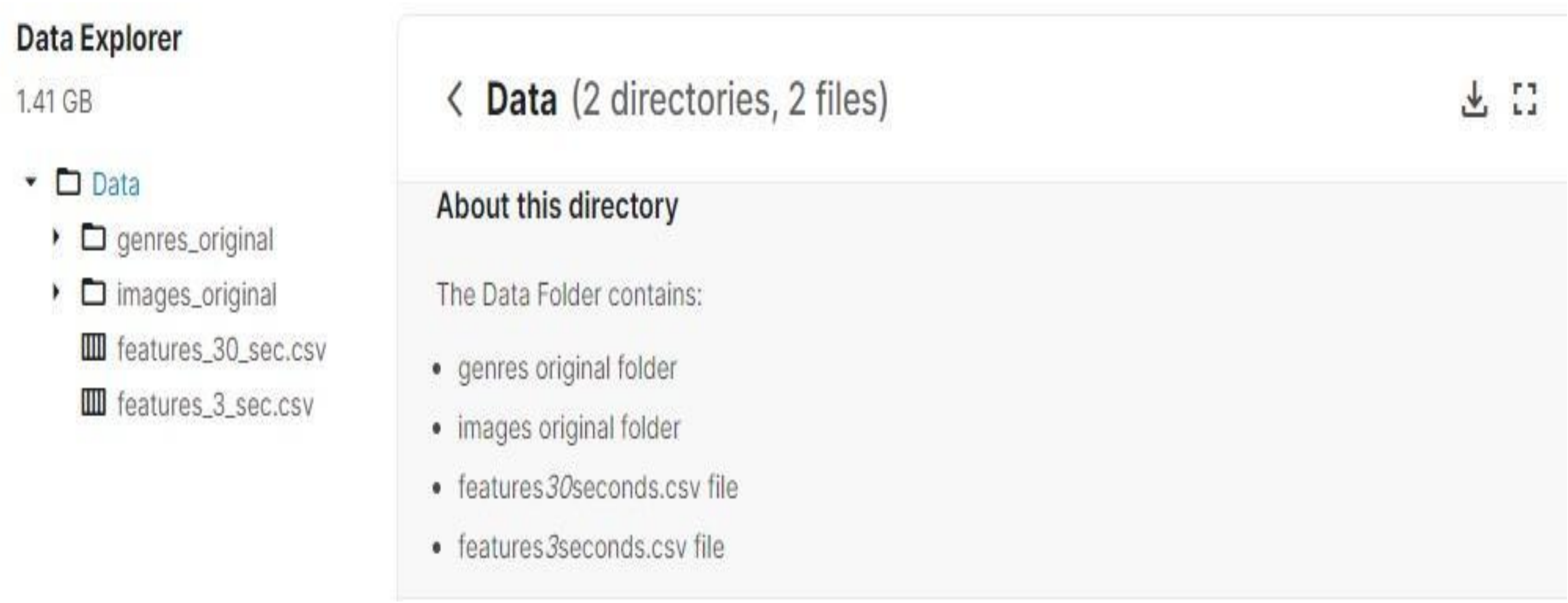
What is Genre?

- By definition , genre is a category of literary composition, determined by literary techniques, tone , content or even length.
- Genre is characterized by common features of pieces belonging to it such as Instrumentation, texture, dynamics, rhythmic characteristics, melodic gestures, harmonic contents.

Dataset (GTZAN genre collection)

- Contains 1000 music files. Dataset has ten types of genres with uniform distribution. Dataset has the following genres: blues, classical, country, disco, hiphop, jazz, reggae, rock, metal, and pop. Each music file is 30 seconds long.
- **Genres original** - A collection of 10 genres with 100 audio files each, all having a length of 30 seconds.
- **Images original** - A visual representation for each audio file. One way to classify data is through neural networks. Because NNs (like CNN, what we will be using today) usually take in some sort of image representation, the audio files were converted to Mel Spectrograms to make this possible.
- The files were collected in 2000-2001 from a variety of sources including personal CDs, radio, microphone recordings, in order to represent a variety of recording conditions.

Dataset (GTZAN genre collection)



Technology Used

- Python
- ML Algorithms
- Sklearn
- Matplotlib
- Numpy & pandas
- Scipy
- Librosa
- python_speech_features

Workflow

1. Preprocess the data using FFT (to convert original data into frequency domain).
2. Feature Extraction using MFCC, Delta, Delta- Delta and rhythmical features.
3. Feature Reduction using Principal Component Analysis.
4. Optimisation of hyper-parameters using Grid Search with Cross-Validation.
5. Training the classifier using various classification algorithms.
6. Testing the data and predicting the genre of our data files.
7. Compare performances of different classifiers using different benchmarks.

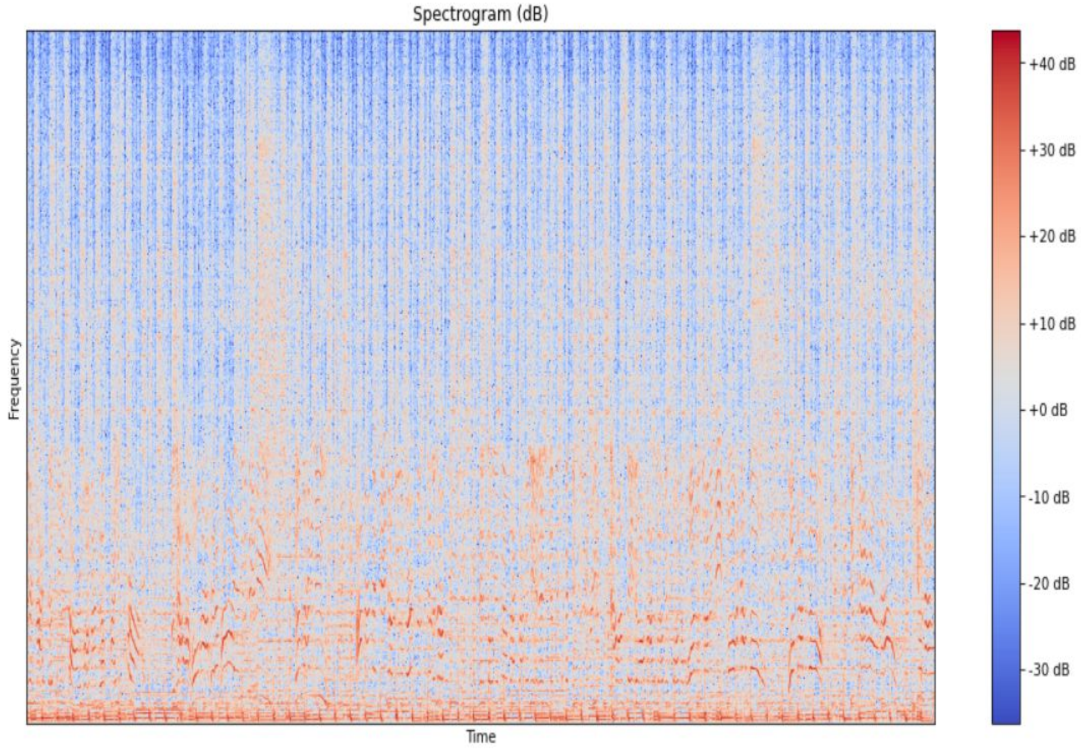
Preprocessing using FFT

- A FFT is a mathematical method to obtain DFT(Discrete Fourier Transform) for a sequence or the inverse of a sequence. A Fourier analysis is performed to obtain a frequency domain representation of the original domain. Rapid computation of this transform by the factorization of Discrete Fourier Transform Matrix into a sparse factors' product is job done by an FFT.

Feature Extraction Techniques for audio files

- **Time domain:** These are extracted from waveforms of the raw audio. Zero crossing rate, amplitude envelope, and RMS energy are examples.
- **Frequency domain:** These focus on the frequency components of the audio signal. Signals are generally converted from the time domain to the frequency domain using the *Fourier Transform*. Band energy ratio, spectral centroid, and spectral flux are examples.
- **Time-frequency representation:** These features combine both the time and frequency components of the audio signal. The time-frequency representation is obtained by applying the Short-Time Fourier Transform (STFT) on the time domain waveform. Spectrogram, mel-spectrogram, and constant-Q transform are examples.

Spectrograms



A spectrogram is a visual depiction of the spectrum of frequencies of an audio signal as it varies with time. Hence it includes both time and frequency aspects of the signal. It is obtained by applying the Short-Time Fourier Transform (STFT) on the signal. In the simplest of terms, the STFT of a signal is calculated by applying the Fast Fourier Transform (FFT) locally on small time segments of the signal.

Mel Spectrograms

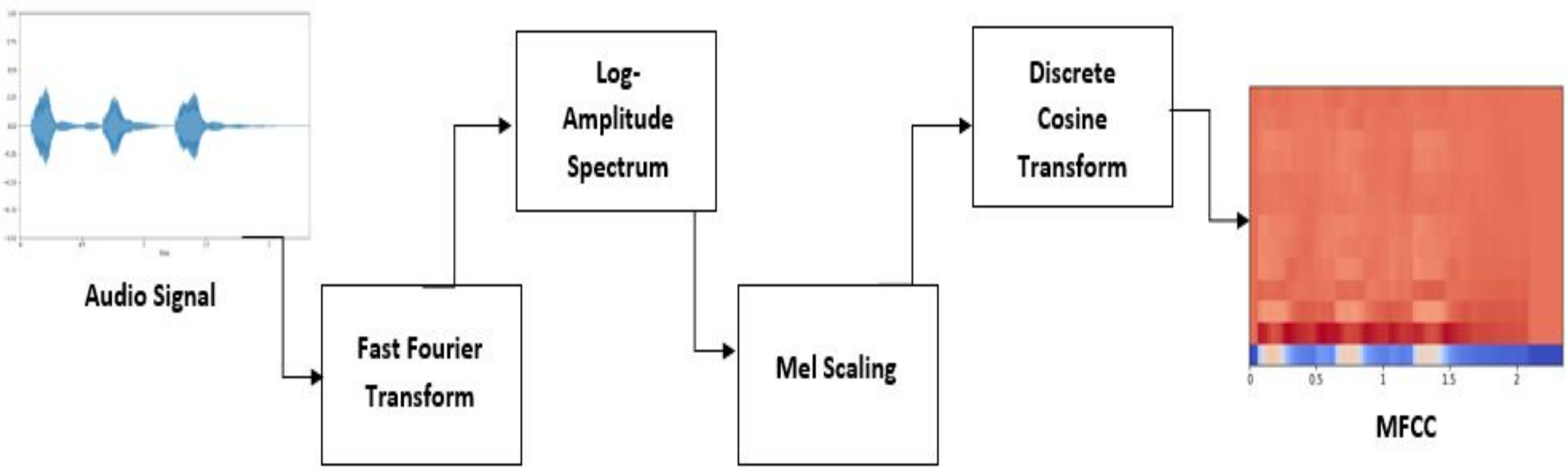
Apparently, we humans perceive sound logarithmically. We are better at detecting differences in lower frequencies than higher frequencies. For example, we can easily tell the difference between 500 and 1000 Hz, but we will hardly be able to tell a difference between 10,000 and 10,500 Hz, even though the distance between the two pairs is the same. Hence, the **mel scale** was introduced.

Conversion from frequency (f) to mel scale (m) is given by:-

$$m=2595.\log(1+f/500).$$

What are MFCC's(Mel Frequency Cepstral Coefficient)

Mel Frequency Cepstral coefficients emphasize on obtaining the exact structure of the audio signal to extract linguistic features and discard the background noise.



FEATURE REDUCTION - Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster.

We have used PCA in our work so as to carry out the dimension reductibility. We have reduced from 40 features (13 mfcc , 13 delta ,13 delta-delta , 1 tempo) to 7 important features by applying explained variance to our data set.

HOW DO YOU DO A PCA?

1. Standardize the range of continuous initial variables
2. Compute the covariance matrix to identify correlations
3. Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components
4. Create a feature vector to decide which principal components to keep
5. Recast the data along the principal components axes

Hyperparameter and Grid Search

- Parameters are the variables that are used by the Machine Learning algorithm for predicting the results based on the input historic data but Hyperparameters are the variables that the user specify usually while building the Machine Learning model. For example, max_depth in Random Forest Algorithms, k in KNN Classifier.
- Grid Search uses a different combination of all the specified hyperparameters and their values and calculates the performance for each combination and selects the best value for the hyperparameters.
- In GridSearchCV, along with Grid Search, cross-validation is also performed. Cross-Validation is used while training the model. As we know that before training the model with data, we divide the data into two parts – train data and test data. In cross-validation, the process divides the train data further into two parts – the train data and the validation data.

Training using K-Nearest Neighbor (KNN)

1. Load the training data.
2. Prepare data by scaling, missing value treatment, and dimensionality reduction as required.
3. Find the optimal value for K:
4. Predict a class value for new data:
    1. Calculate distance(X, Xi) from i=1,2,3,….,n.
       where X= new data point, Xi= training data, distance as per your chosen distance metric.
    2. Sort these distances in increasing order with corresponding train data.
    3. From this sorted list, select the top ‘K’ rows.
    4. Find the most frequent class from these chosen ‘K’ rows. This will be your predicted class.

Training using Random Forest Classifier

- Random forest is a *Supervised Machine Learning Algorithm* which is an ensemble of decision trees, that is *used widely in Classification*,important feature of the Algorithm is that it can handle the data set containing *continuous variables.*

- There are more trees (dataset is divided into smaller partitions) in a forest which makes this algorithm more robust. It uses divide-and-conquer approach to improve the performance. The decision trees are weak learners whereas random forest is a strong learner. When a new input is entered, it is run down in all of the trees. The result may be average, or weighted average of the terminal nodes which are reached. Random forest is able to deal with unbalanced and missing data as well.

Training using Random Forest Classifier

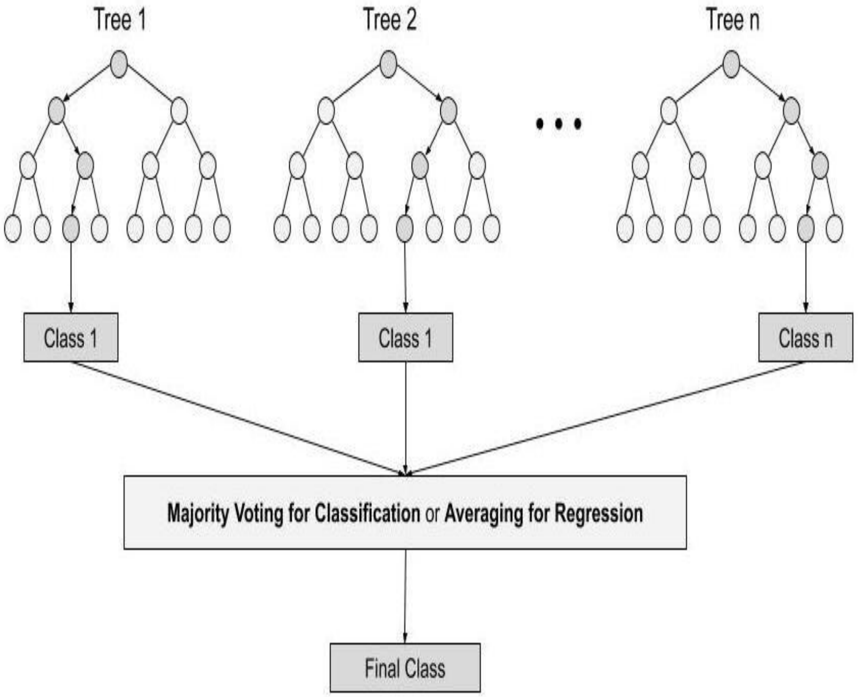
Steps of Random Forest Classifier:-

1.In Random forest n number of random records are taken from the data set having k number of records.

2. Individual decision trees are constructed for each sample.

3.Each decision tree will generate an output.

4.Final output is considered based on *Majority Voting or Averaging* for Classification and regression respectively.



Evaluation of Model

A **confusion matrix** is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known.

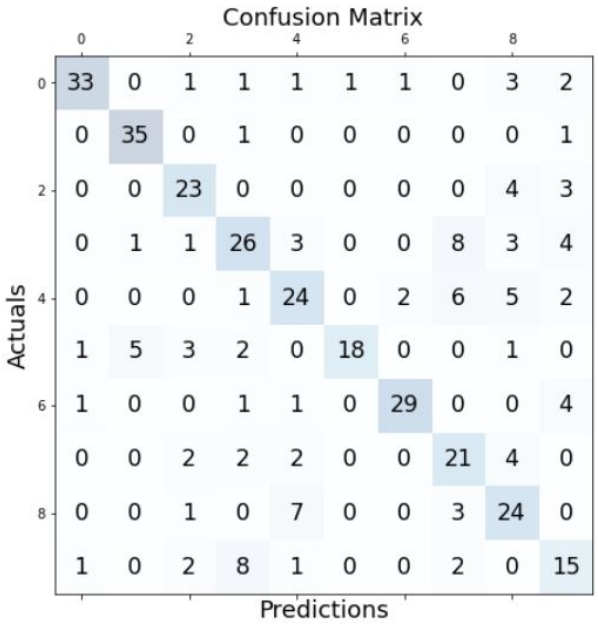
Example confusion matrix for a binary classifier

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Accuracy:(TP+TN)/total = (100+50)/165
Recall: TP/(TP+FN)=100/105
Precision:TP/(TP+FP) = 100/110
F Score: (2*recall*precision)/(recall+precision)
Where TP=**true positives**, TN=**true negatives**
FP=**false positives** (Type I error)
FN=**false negatives** (Type II error)

Evaluation of K-Nearest Neighbour

	precision	recall	f1-score	support
1	0.92	0.77	0.84	43
2	0.85	0.95	0.90	37
3	0.70	0.77	0.73	30
4	0.62	0.57	0.59	46
5	0.62	0.60	0.61	40
6	0.95	0.60	0.73	30
7	0.91	0.81	0.85	36
8	0.53	0.68	0.59	31
9	0.55	0.69	0.61	35
10	0.48	0.52	0.50	29
accuracy			0.69	357
macro avg	0.71	0.69	0.69	357
weighted avg	0.72	0.69	0.70	357



Evaluation of Random Forest Classifier

CLASSIFICATION REPORT

	precision	recall	f1-score	support
1	0.45	0.32	0.38	28
2	0.59	0.76	0.67	17
3	0.41	0.39	0.40	23
4	0.43	0.40	0.41	30
5	0.44	0.65	0.52	17
6	0.65	0.54	0.59	28
7	0.53	0.57	0.55	30
8	0.46	0.48	0.47	27
avg / total	0.50	0.49	0.49	200

What’s next

- Extract some more features from audio files.
- Perform EDA(Exploratory Data Analysis) to find relevant features.
- Training of model using Deep Learning approach such CNN, Transfer Learning to improve confusion matrix.

References

[1] J.W. Picone, Signal modeling techniques in speech recognition. Proc. IEEE 81, 1215–1247 (1993)

[2] J. Volkmann, S. Stevens, E. Newman, A scale for the measurement of the psychological magnitude pitch. J. Acoust. Soc. Am. 8, 185–190 (1937)

[3] Z. Fang, Z. Guoliang, S. Zhanjiang, Comparison of different implementations of MFCC. J. Comput. Sci. Technol. 16, 582–589 (2000)

[4] G.K.T. Ganchev, N. Fakotakis, Comparative evaluation of various MFCC implementations on the speaker verification task, in Proceedings of International Conference on Speech and Computer (SPECOM) (2005), pp. 191–194

[5] J.S. Mason, X. Zhang, Velocity and acceleration features in speaker recognition, in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) (1991), pp. 3673–3676