UNIVERSITY OF BURGUNDY

MASTERS IN COMPUTER VISION AND ROBOTICS

AUDIO GENRE CLASSIFICATION

A Project under the guidance of
Prof. Desire Sidibe

PROJECT TEAM

VAMSHI KODIPAKA

BHARGAV SHAH

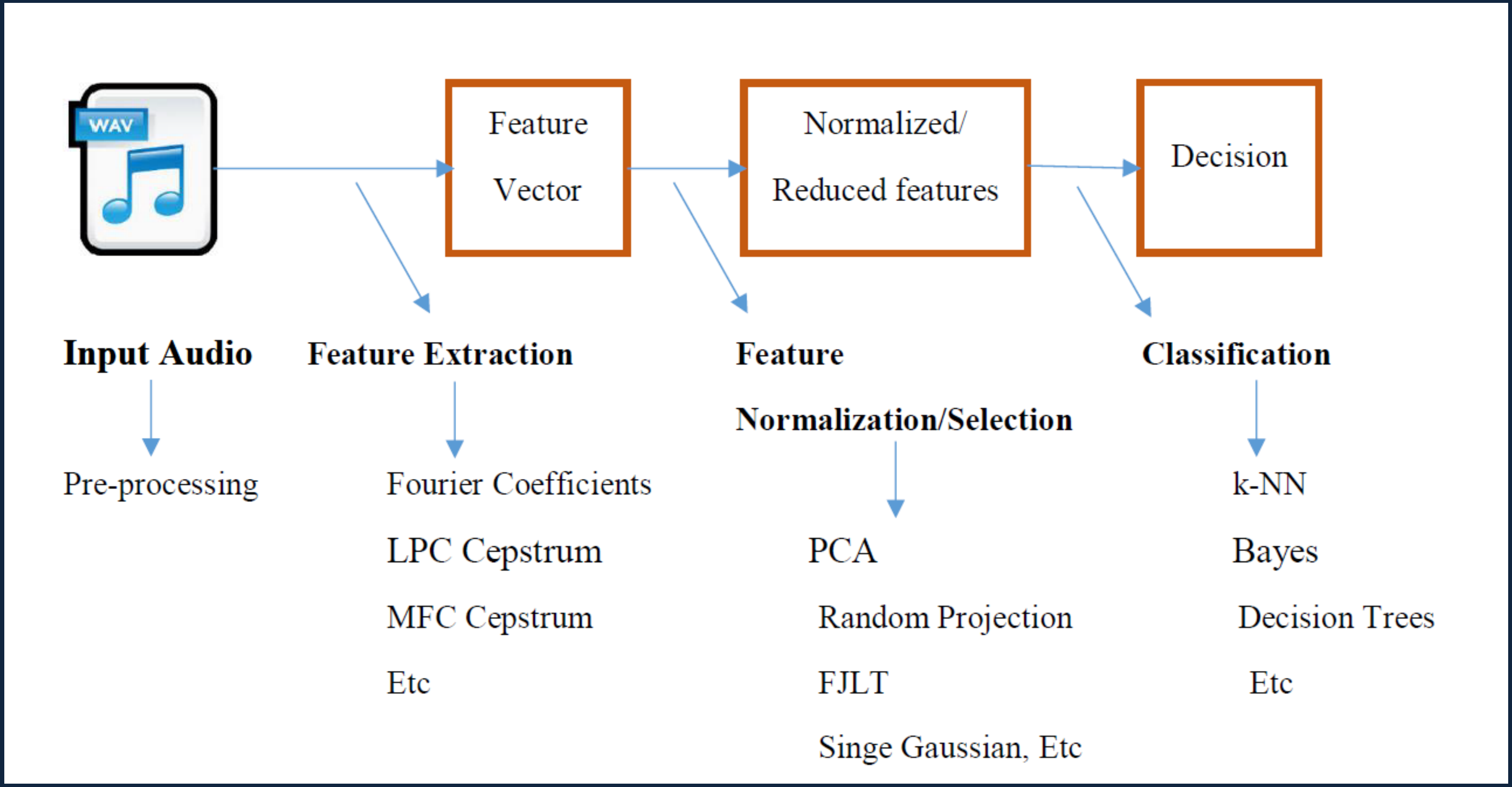
PARMAR HARDIKSINH

Overview

@Research Project for LAAS CNRS - 2018

- Introduction
  - AUDIO FILES
  - MUSIC DATA SET @ ISMIR -2004 (729 TRACKS)
- Methodology
  - Feature Extraction - MFCC
  - Feature Selection - PCA
  - Feature Classification – KNN
- Problems of Dimension Reductionality
- GTZAN Dataset -1000 au files (Quick implementation)
- Output/Results Achieved
- Future Work

Audio Understanding : Pipeline



AIM:

- Using Machine Learning Pipeline on Music Dataset we need to classify the dataset

Audio Access in MATLAB

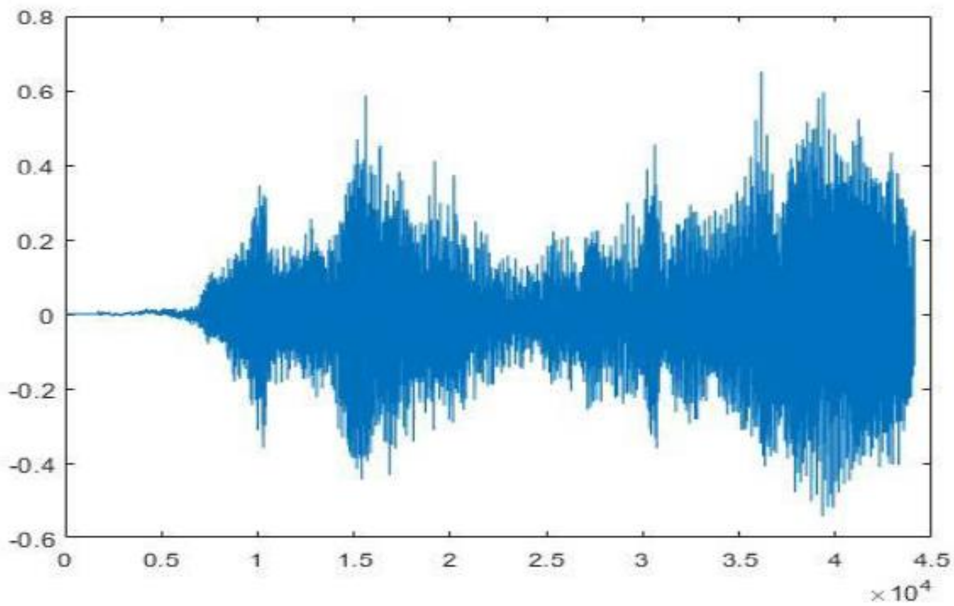
BASICS OF AUDIO PROCESSING:

1. <u>To load .mp3 file:</u> [y,Fs]=audioread('artist_1_album_1_track_1.mp3')

   Here, y gives amplitude and Fs gives frequency.

2. <u>Plot .mp3 with amplitude (vs) frequency:</u>

   plot(y(1:44100,1))



Amplitude vs Frequency Plot

Audio Access in MATLAB

4. To play .mp3 using sound command:

   sound(y(1:441000,1),Fs)

5. To play .mp3 using audioplayer object in MATLAB:

p =audioplayer(y,Fs)

play (p) :: plays loaded music fil

pause(p) :: pauses music file

stop(p) :: stops music file

start(p) :: starts from point of stop

clear(p) :: clears ‘p’ object’s music file.

Command Window

```
>> p=audioplayer(y,Fs)

p =

  audioplayer with properties:

        SampleRate: 44100
     BitsPerSample: 16
  NumberOfChannels: 2
          DeviceID: -1
     CurrentSample: 1
      TotalSamples: 1686000
           Running: 'off'
          StartFcn: []
           StopFcn: []
          TimerFcn: []
       TimerPeriod: 0.0500
               Tag: ''
          UserData: []
              Type: 'audioplayer'

>> play (p)
>> pause(p)
>> stop(p)
>>
```

Workspace

Name	Value
Fs	44100
y	1686000x2 double

```
>> info=audioinfo('artist_1_album_1_track_1.mp3')

info =

  struct with fields:

             Filename: 'D:\AudioGenreClassifier-master\AudioGenreClassifier-master\artist_1_album_1_track_1.mp3'
    CompressionMethod: 'MP3'
          NumChannels: 2
           SampleRate: 44100
         TotalSamples: 1687360
             Duration: 38.2621
                Title: []
              Comment: []
               Artist: []
              BitRate: 128
```

Mel refers to ‘Melody’.

Cepstrum means the IFT of the logarithm of the estimated spectrum of a signal.

Definition:

In sound processing,
the **Mel-Frequency Cepstrum** (**MFC**) is a representation of the short-term power spectrum of a sound, based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency

To retrieve info of music file:

Audioinfo(“filename”)

FEATURE EXTRACTION : MFCC

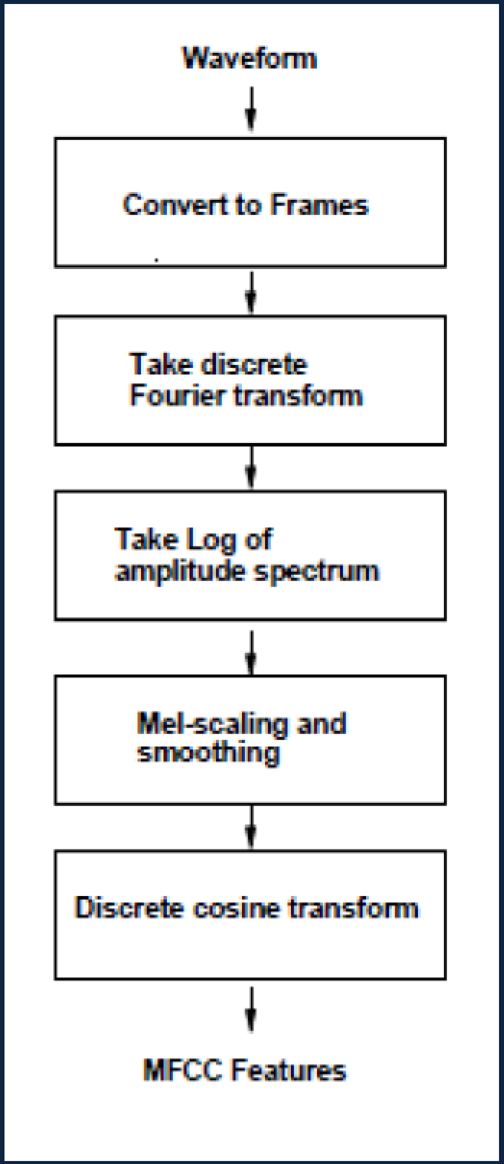
1. Take the Fourier transform of (a windowed excerpt of) a signal.

2. Map the powers of the spectrum obtained above onto the mel scale, using triangular overlapping windows.

3. Take the logs of the powers at each of the mel frequencies.

4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.

5. The MFCCs are the amplitudes of the resulting spectrum.



FEATURE EXTRACTION : TYPES

From conventional spectral analysis:

1. IFT

   a. Positive Cepstrum
   b. Negative Cepstrum

Linear Predictive Coding Cepstrum(LPC Cpestrum):

The LPC vector is defined by $[a_0,a_1,a_2,...a_p]$ and the CC vector is defined by $[c_0c_1c_2...c_p...c_{n-1}$

LPC Cepstrum (c_m)	
$c_0 = \log G^2$ $c_m = a_m + \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k}, \quad 1 \le m \le p$ $c_m = \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k}, \quad m > p$	$G = e^{c_0/2}$ $a_m = c_m - \sum_{k=1}^{m-1}\left(\frac{k}{m}\right)c_k a_{m-k}, \quad 1 \le m \le p$

MEL-FREQUENCY

Note: Mel-Scale is approximately linear for low-frequency (f<500Hz) and logarithmic for high frequencies

1. Noise Sensitivity
2. Use of MFCC
3. Pre-processing
4. Calculation:

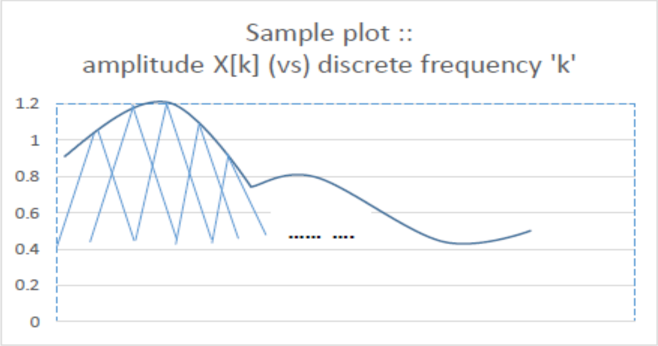
$$M(mel\_freq) = 1127 * \log(1 + f/700)$$

where f is frequency in linear scale
and M is frequency is mel scale.

This modulation of log acts as a weight vector like in
($y = w^T X - t$) in a classical regression model

Let x[n] is framed through the entire signal ,

has a window size fixed $\quad x[n] \xrightarrow{DFT} X[k]$ -- |X[k]|



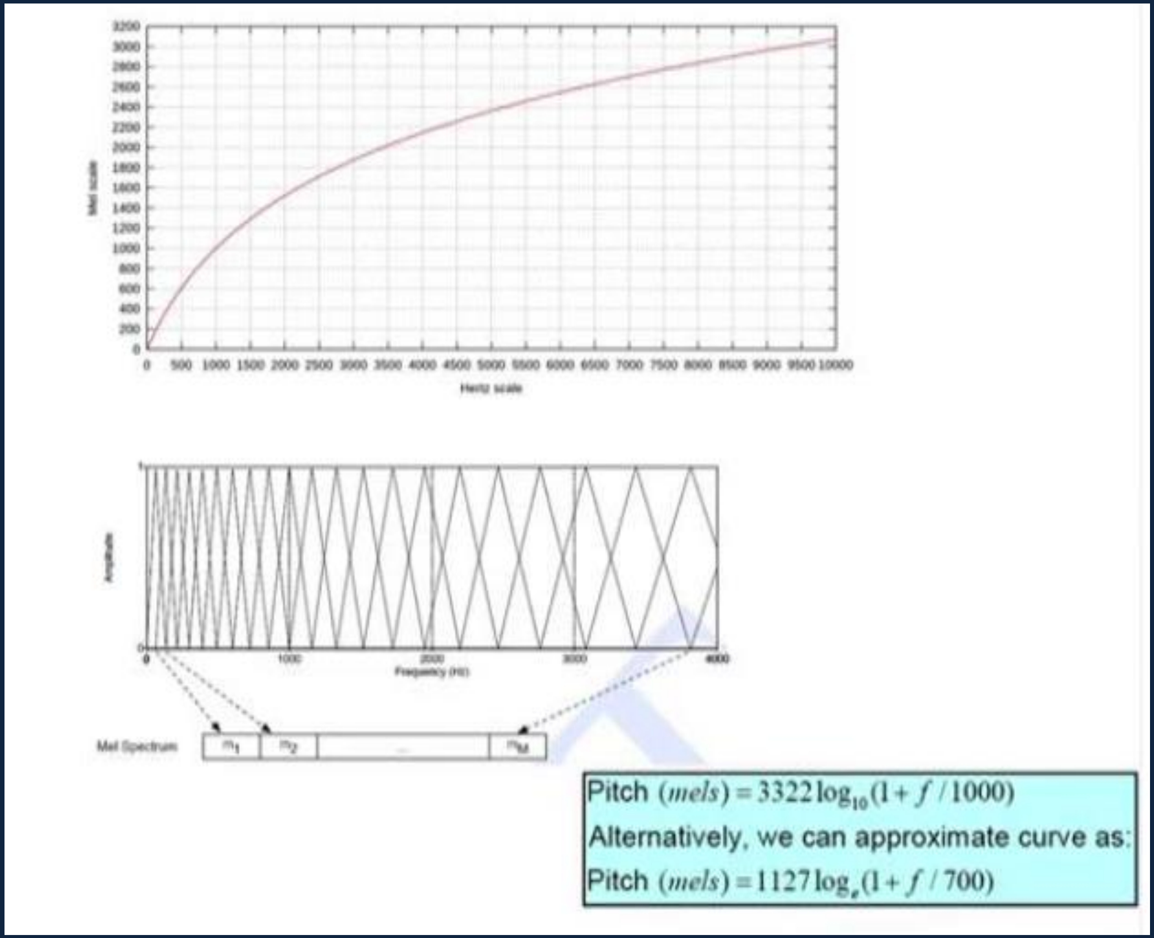
k= N/2; $\quad$ fmax > Fs/2 ; $\quad$ Fs = 8KHz; $\quad$ fs becomes 4KHz

If triangular train filter = 100Hz → Fs=4000/100 = 40points

Now suppose, this triangular filter is applied to 8KHz =80points will be generated.

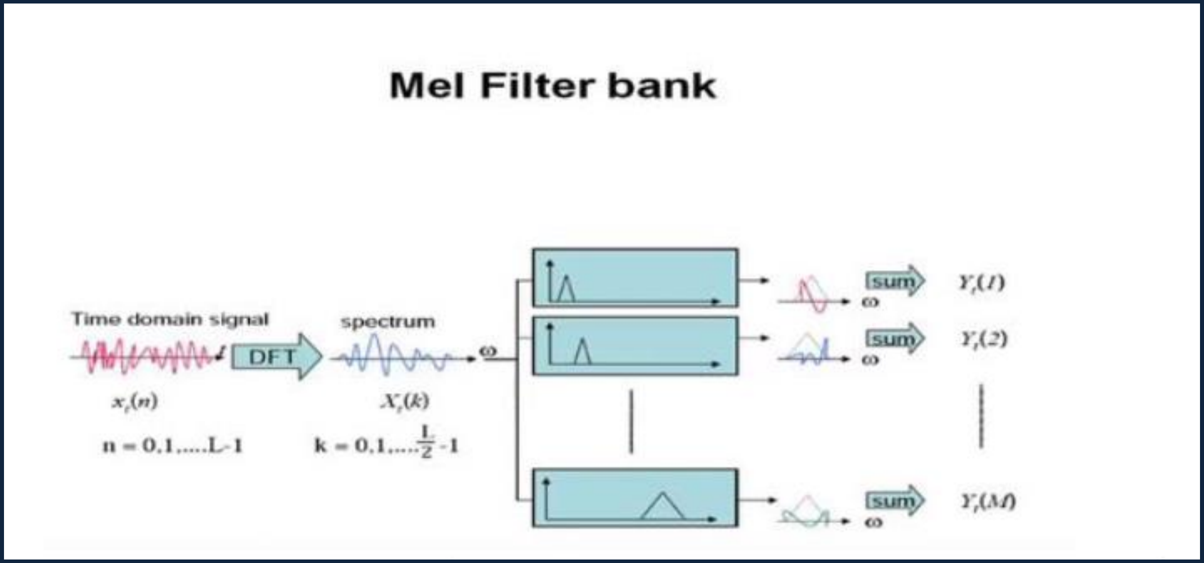
To convert into k to f, we have: $\quad f = 2\pi k/N$;

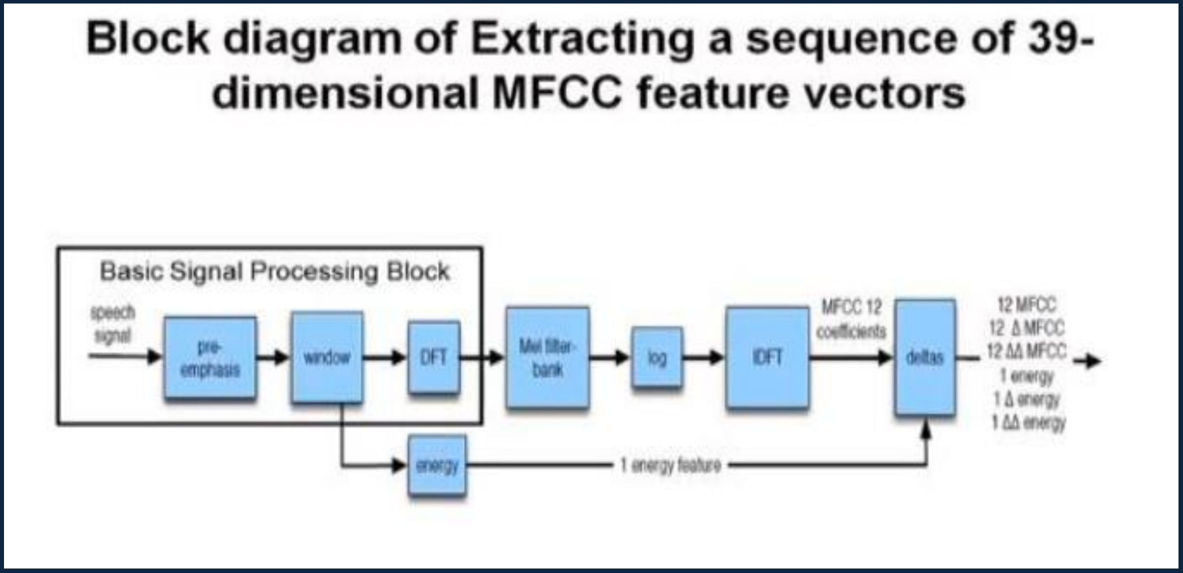
MEL-FREQUENCY



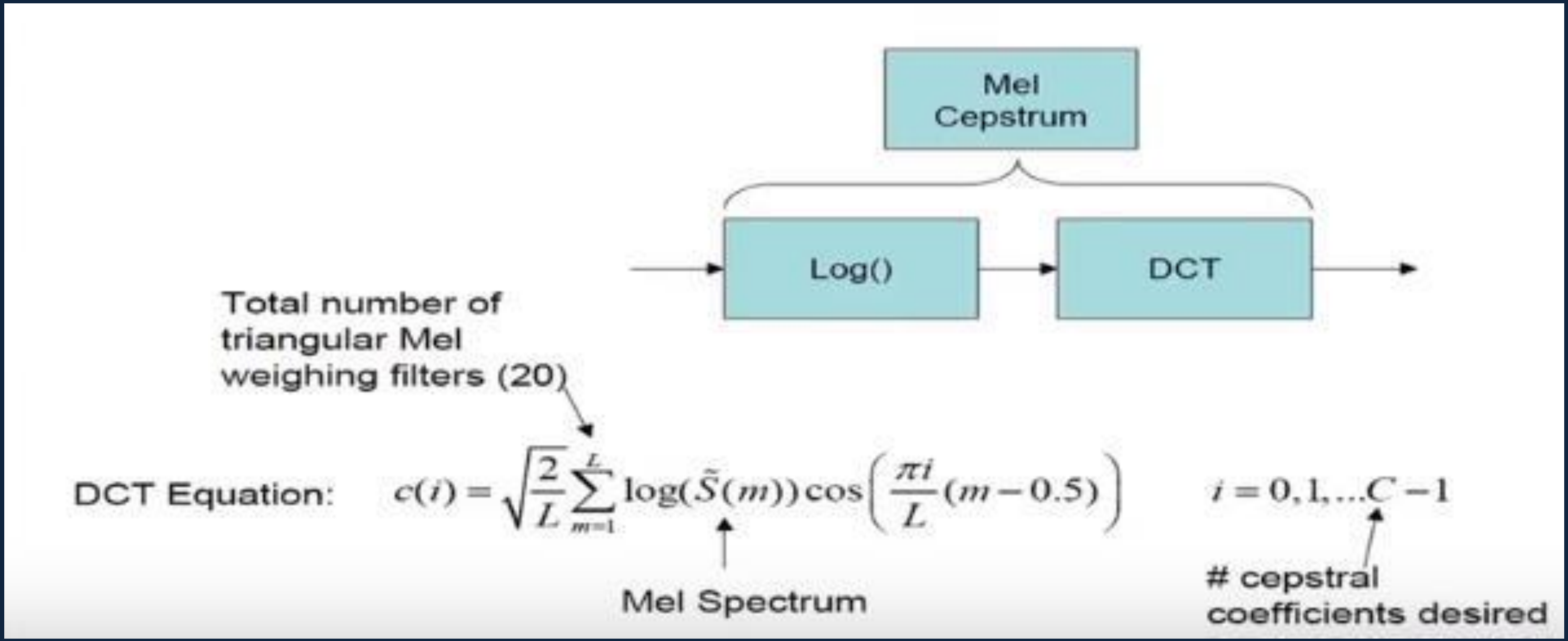
Log- Representation of MEL-FREQ

MFCC FEATURES





MFCC FEATURES



Instead of IFT we can take DCT: MFCC

MFCC FEATURE EXTRACTION: OUTPUTS

Workspace

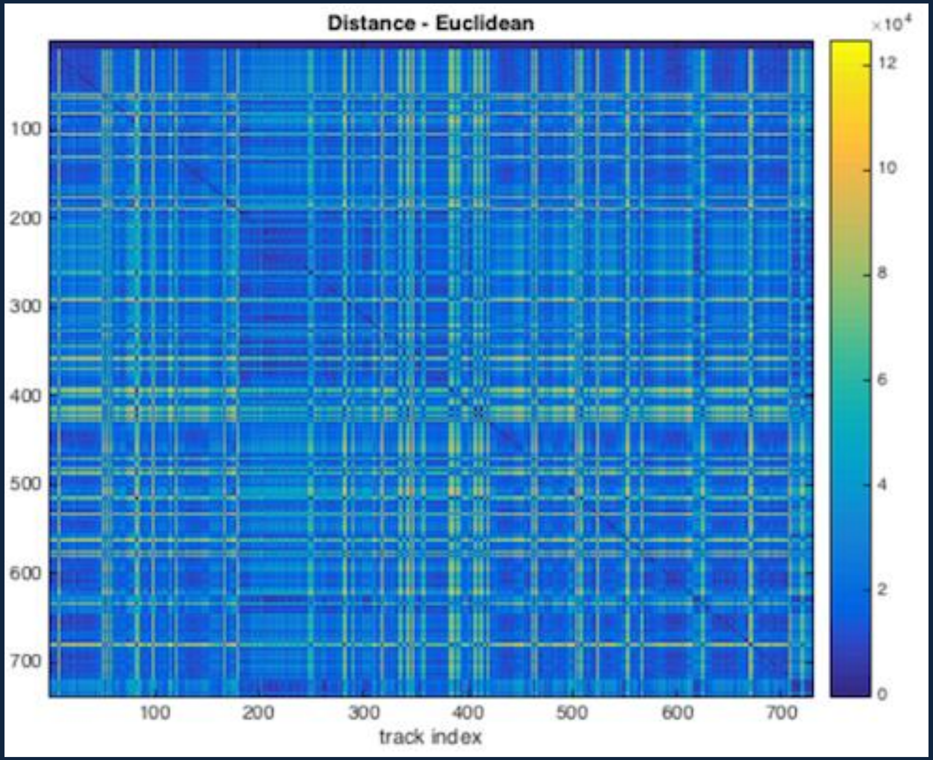
Name	Value
distance	743x729 double
euclideanDistance	1x79 double
features	729x79 double
filename	1x729 cell
files	729x1 struct
frobeniusNorm	0
GMModel	1x1 gmdistributio
groupNames	729x1 cell
groups	729x1 double
i	745
j	745
k	6
matrix1	79x79 double
matrix2	79x79 double
mfcc_artist_100_album_1_track_1	79x79 double
mfcc_artist_100_album_1_track_2	79x79 double
mfcc_artist_100_album_1_track_3	79x79 double
mfcc_artist_100_album_1_track_4	79x79 double
mfcc_artist_100_album_2_track_1	79x79 double
mfcc_artist_100_album_2_track_2	79x79 double
mfcc_artist_100_album_2_track_3	79x79 double
mfcc_artist_100_album_3_track_1	79x79 double
mfcc_artist_100_album_3_track_2	79x79 double
mfcc_artist_100_album_3_track_3	79x79 double
mfcc_artist_100_album_4_track_1	79x79 double
mfcc_artist_100_album_4_track_2	79x79 double
mfcc_artist_100_album_4_track_3	79x79 double
mfcc_artist_100_album_4_track_4	79x79 double
mfcc_artist_100_album_5_track_1	79x79 double
mfcc_artist_100_album_5_track_2	79x79 double
mfcc_artist_100_album_5_track_3	79x79 double
mfcc_artist_101_album_1_track_1	79x79 double
mfcc_artist_101_album_1_track_2	79x79 double
mfcc_artist_101_album_1_track_3	79x79 double
mfcc_artist_102_album_1_track_1	79x79 double
mfcc_artist_102_album_1_track_2	79x79 double

Workspace

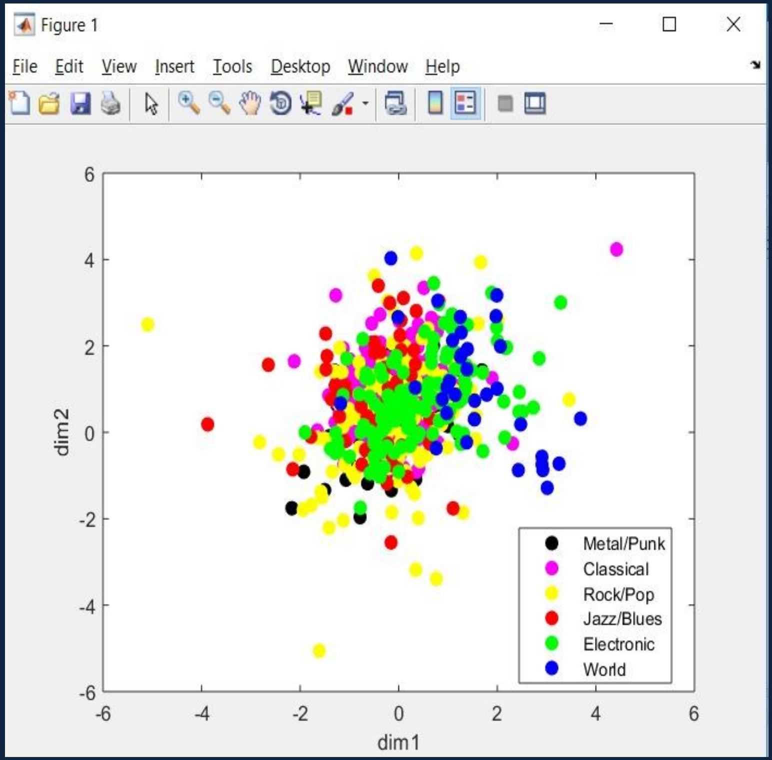
Name	Value
mfcc_artist_96_album_1_track_2	79x79 double
mfcc_artist_97_album_1_track_1	79x79 double
mfcc_artist_97_album_1_track_2	79x79 double
mfcc_artist_97_album_1_track_3	79x79 double
mfcc_artist_98_album_1_track_1	79x79 double
mfcc_artist_98_album_1_track_2	79x79 double
mfcc_artist_98_album_1_track_3	79x79 double
mfcc_artist_99_album_1_track_1	79x79 double
mfcc_artist_99_album_1_track_2	79x79 double
mfcc_artist_99_album_1_track_3	79x79 double
mfcc_artist_99_album_1_track_4	79x79 double
mfcc_artist_9_album_1_track_1	79x79 double
mfcc_artist_9_album_1_track_2	79x79 double
mfcc_artist_9_album_1_track_3	79x79 double
mfcc_artist_9_album_1_track_4	79x79 double
mfcc_artist_9_album_1_track_5	79x79 double
mfcc_artist_9_album_1_track_6	79x79 double
mfcc_artist_9_album_2_track_1	79x79 double
mfcc_artist_9_album_2_track_10	79x79 double
mfcc_artist_9_album_2_track_11	79x79 double
mfcc_artist_9_album_2_track_2	79x79 double
mfcc_artist_9_album_2_track_3	79x79 double
mfcc_artist_9_album_2_track_4	79x79 double
mfcc_artist_9_album_2_track_5	79x79 double
mfcc_artist_9_album_2_track_6	79x79 double
mfcc_artist_9_album_2_track_7	79x79 double
mfcc_artist_9_album_2_track_8	79x79 double
mfcc_artist_9_album_2_track_9	79x79 double
mfcc_artist_9_album_3_track_1	79x79 double
mfcc_artist_9_album_3_track_2	79x79 double
mfcc_artist_9_album_3_track_3	79x79 double
mfcc_artist_9_album_3_track_4	79x79 double
mfcc_artist_9_album_3_track_5	79x79 double
mfcc_artist_9_album_3_track_6	79x79 double
values	1x729 double
variables	745x1 cell

Set of feature Extracted : ISMIR2004

MFCC FEATURE EXTRACTION: OUTPUTS

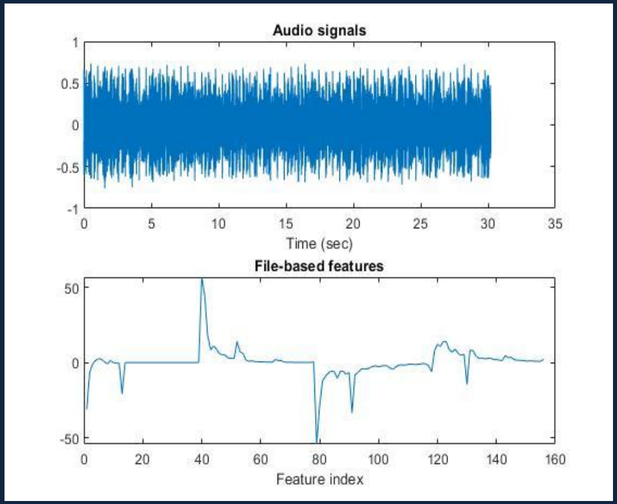


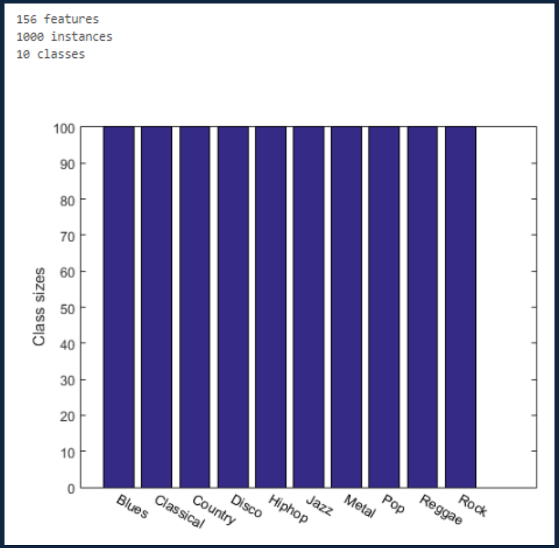
Calculation of Frobenius norm of the MFCC Euclidean Distance



Mean of MCC and GMM

WORKING ON GTZAN DATASET

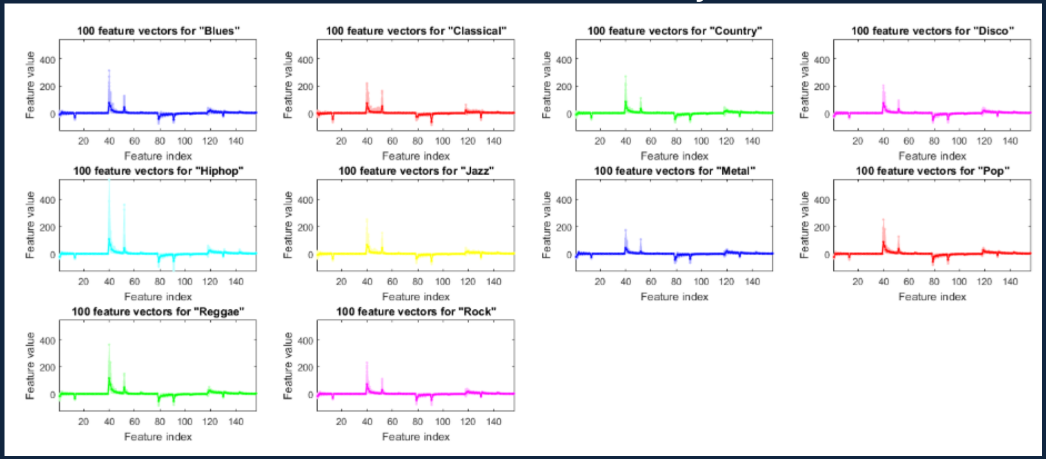




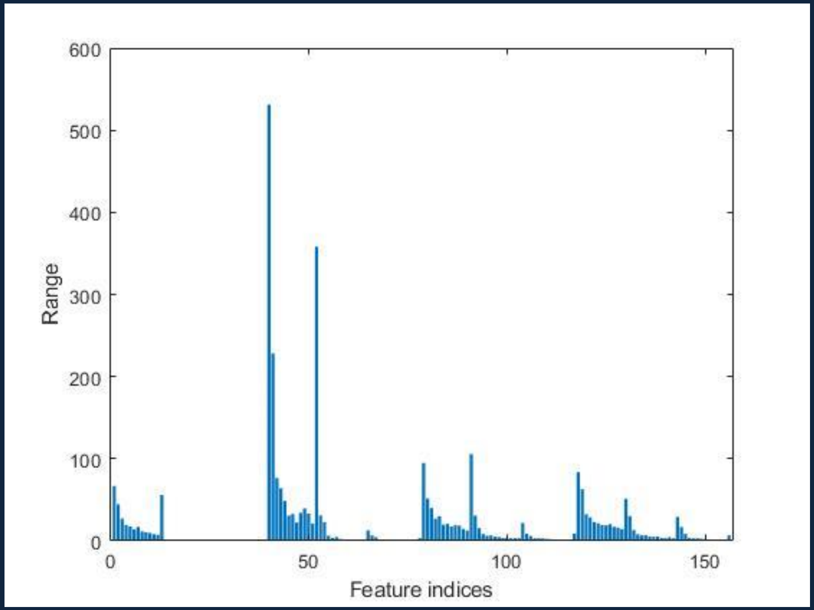
Feature extraction

Data Visualization

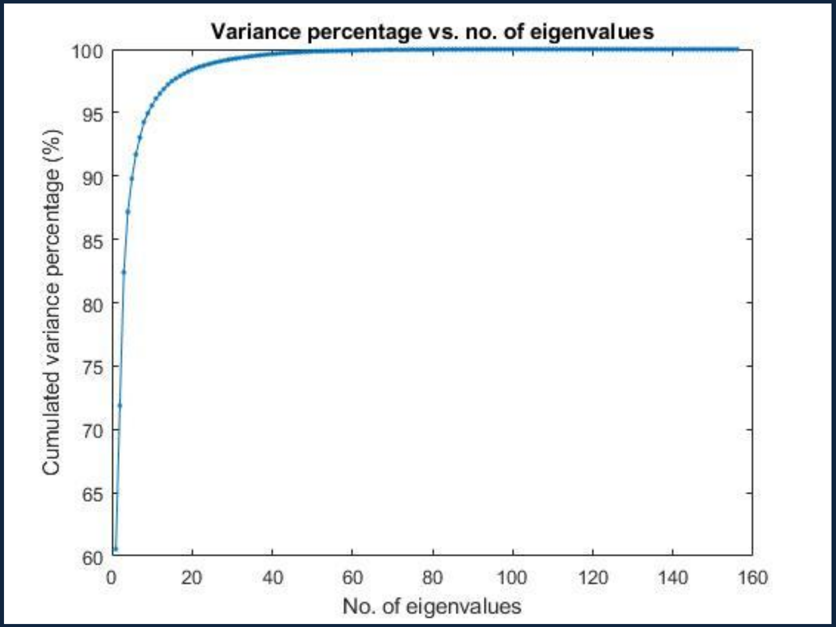
Class-wise Features Density



GTZAN

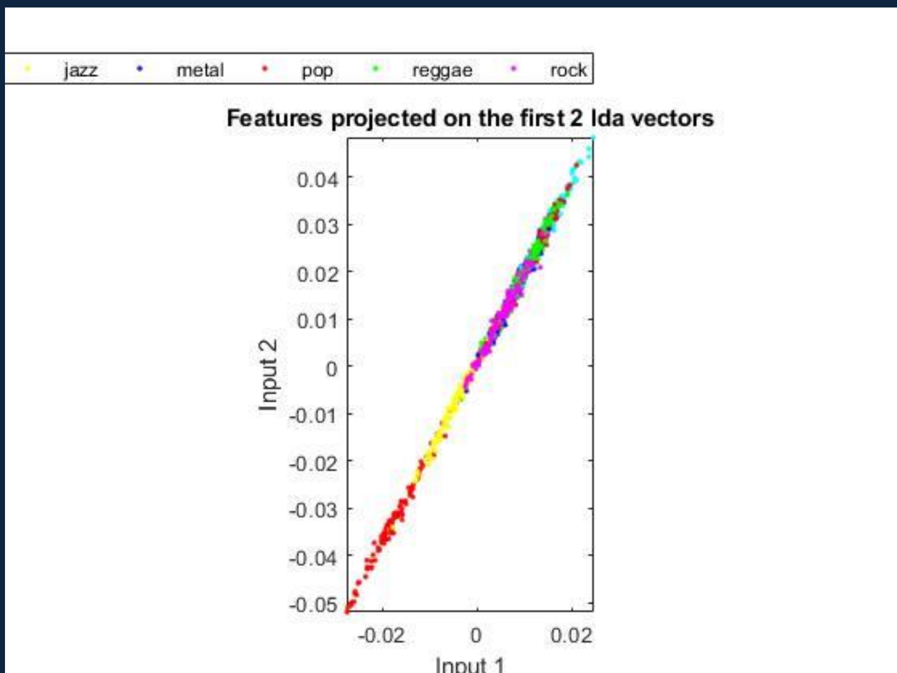


Feature Density

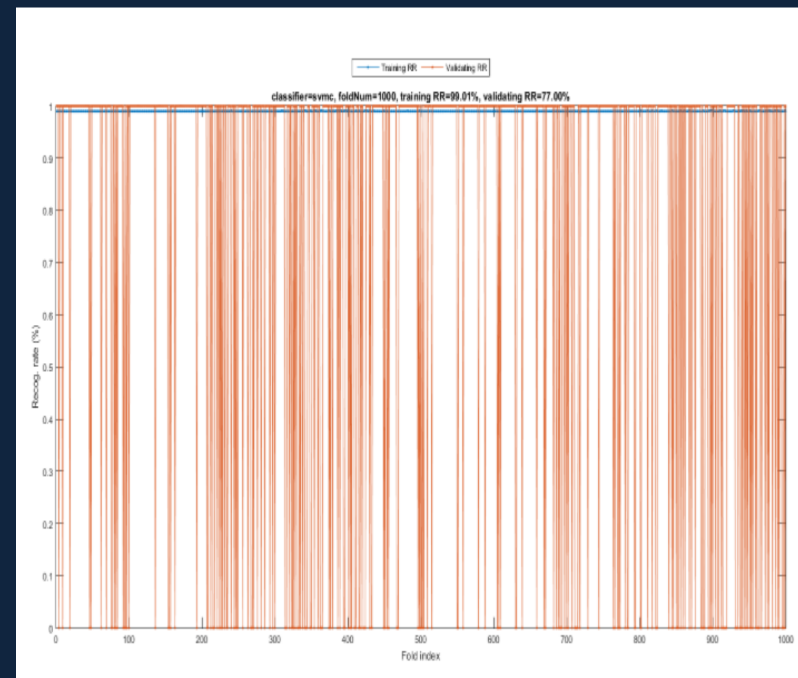


Dimension Reduction

GTZAN

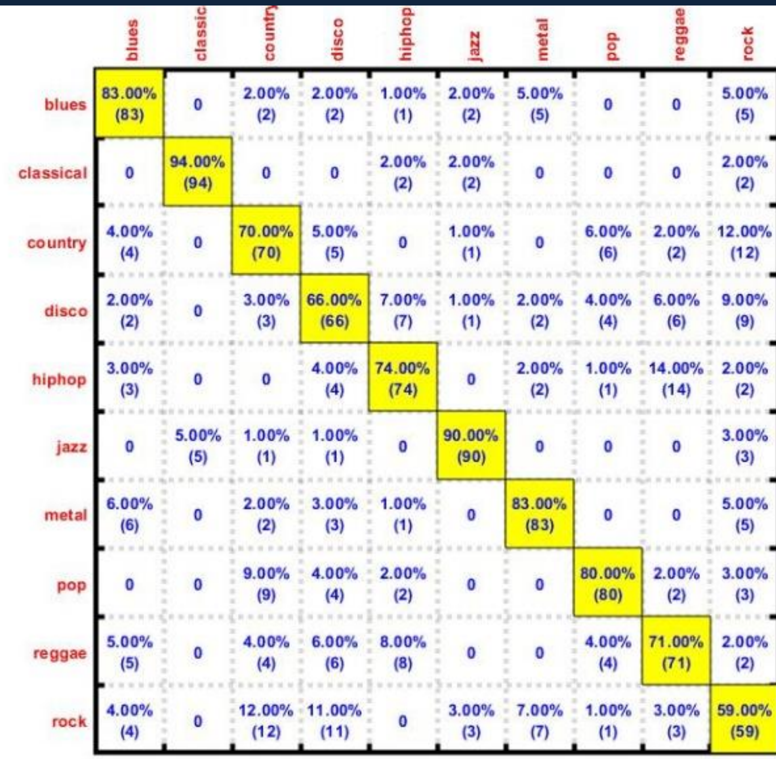


LDA Reduction



SVM Classifier

GTZAN



	blues	classic	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	83.00% (83)	0	2.00% (2)	2.00% (2)	1.00% (1)	2.00% (2)	5.00% (5)	0	0	5.00% (5)
classical	0	94.00% (94)	0	0	2.00% (2)	2.00% (2)	0	0	0	2.00% (2)
country	4.00% (4)	0	70.00% (70)	5.00% (5)	0	1.00% (1)	0	6.00% (6)	2.00% (2)	12.00% (12)
disco	2.00% (2)	0	3.00% (3)	66.00% (66)	7.00% (7)	1.00% (1)	2.00% (2)	4.00% (4)	6.00% (6)	9.00% (9)
hiphop	3.00% (3)	0	0	4.00% (4)	74.00% (74)	0	2.00% (2)	1.00% (1)	14.00% (14)	2.00% (2)
jazz	0	5.00% (5)	1.00% (1)	1.00% (1)	0	90.00% (90)	0	0	0	3.00% (3)
metal	6.00% (6)	0	2.00% (2)	3.00% (3)	1.00% (1)	0	83.00% (83)	0	0	5.00% (5)
pop	0	0	9.00% (9)	4.00% (4)	2.00% (2)	0	0	80.00% (80)	2.00% (2)	3.00% (3)
reggae	5.00% (5)	0	4.00% (4)	6.00% (6)	8.00% (8)	0	0	4.00% (4)	71.00% (71)	2.00% (2)
rock	4.00% (4)	0	12.00% (12)	11.00% (11)	0	3.00% (3)	7.00% (7)	1.00% (1)	3.00% (3)	59.00% (59)

Confusion Matrix

FUTURE WORK

- 1. Problem of Dimension Reduction – PCA
- 2. KNN Classifier
- 3. Testing and Classifying the Sample Audio

Thank you…