

# **Audio content-based Music Recommendation System**

**Submitted By:**

**Group 3**

Atul (2017032)

Prashant Jain (2018253)

Akanksha Pandey (MT20048)

Shivani Mishra (MT20062)

## **Steps for accessing/ using submission**

### **Requirements:**

- 1) Google Colab
- 2) Spyder
- 3) TKinter
- 4) ffmpeg
- 5) pandas

### **Dataset Link:**

[https://drive.google.com/drive/folders/14jb56ySUCGIBaBGAwzFcFSkkqWVvY\\_h5?usp=sharing](https://drive.google.com/drive/folders/14jb56ySUCGIBaBGAwzFcFSkkqWVvY_h5?usp=sharing)

### **Steps:**

- 1) Clone the repository  
git clone --branch final\_project  
[https://github.com/shivanirmishra/IR2021\\_A1\\_3/tree/final\\_project](https://github.com/shivanirmishra/IR2021_A1_3/tree/final_project)
- 2) To visualize the songs, run file :  
IRProject\_plots\_3.ipynb
- 3) To see all the models implemented by us for genre prediction and music recommendation, run file :  
IRProject\_AllModels\_3.ipynb

- 4) To see the recommendation of our system with frontend, run the following commands:
- i) pip install tk
  - ii) pip install tkintertable
  - iii) pip install ffmpeg
  - iv) Download exe file of ffmpeg using <https://www.ffmpeg.org/download.html> and set the path in the environment according to your system requirement.
  - v) After installation, restart your system.
- Run file : IRProject\_UI\_3.py

## **Contribution:**

We have divided the dataset into smaller chunks of 20s each using pydub which is further used to match the test song. The motivation behind this is to gather as much information possible from each song. The division of the dataset in chunks enables more overlapping of input songs with a song dataset.

For recommending songs we have also used DTW (dynamic time warping), in time series analysis, DTW is one of the algorithms for measuring similarity between two temporal sequences, which may vary in speed. DTW has been applied to temporal sequences of video, audio, and graphics data . DTW is a very efficient algorithm which can be used to calculate optimal scores between the songs, thus efficient recommendation.

Majority of papers that we read were using frequency-domain features for the any dataset but we are also combining time-domain features with frequency domain features for the same to intensify the performance or result for the data.