

**PROJECT: Create Your Own Spotify**

**“BeatBox”**

# **Course Name:**

Fundamental of Big-Data Analysis

**SUBMITTED BY:**

**Umaima Hashmi (22I-1984)**

**Nooran Ishtiaq (22I-2010)**

**M. Manhab Zafar (22I-1957)**

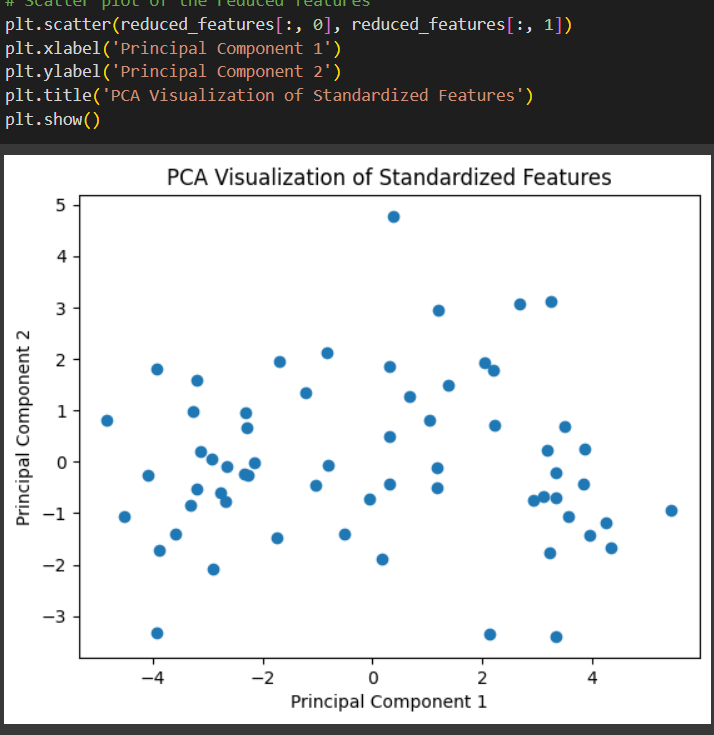
# **Introduction:**

In this project, we aimed to develop a streamlined alternative to Spotify, a popular digital music streaming service. Our project focused on implementing a music recommendation system, playback, and streaming capabilities, alongside real-time suggestions derived from user activity. The project was divided into three phases: Extract, Transform, Load (ETL) Pipeline, Music Recommendation Model and Deployment.

**Phase #1: Extract, Transform, Load (ETL) Pipeline:**

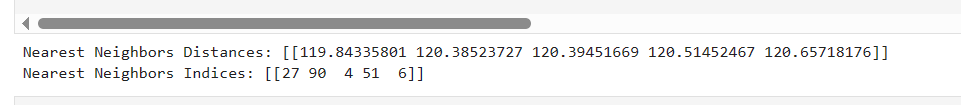
In the first phase, we created an ETL pipeline using the Free Music Archive (FMA) dataset. The dataset comprised 106,574 tracks spanning 161 unevenly distributed genres. We extracted important features from audio files using techniques like Mel-Frequency Cepstral Coefficients (MFCC), spectral centroid, and zero-crossing rate.

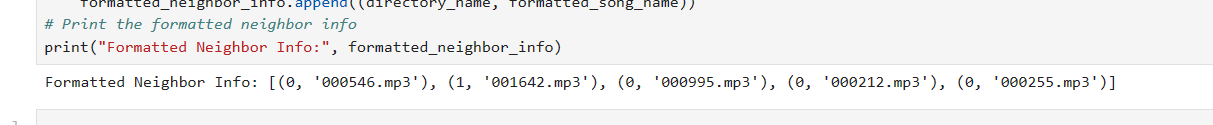
Additionally, we explored normalization, standardization, and dimensionality reduction techniques to enhance recommendation model accuracy. The transformed data was stored in MongoDB for scalability and accessibility.



# **Music Recommendation Model:**

The music recommendation Model is simple is understands the listening habit or the behavior of the user. When the user comes to the webpage he has a choice of the music to choose from he will choose from the available option and from there starts the work of our model the model will analiza the audio that the user has choosed and it will extract the features of the audio and then it will compare the audio with the vast data present and by applying the **NearestNeighbors** the model will figure out top 5 such songs with which the customers audio matches.

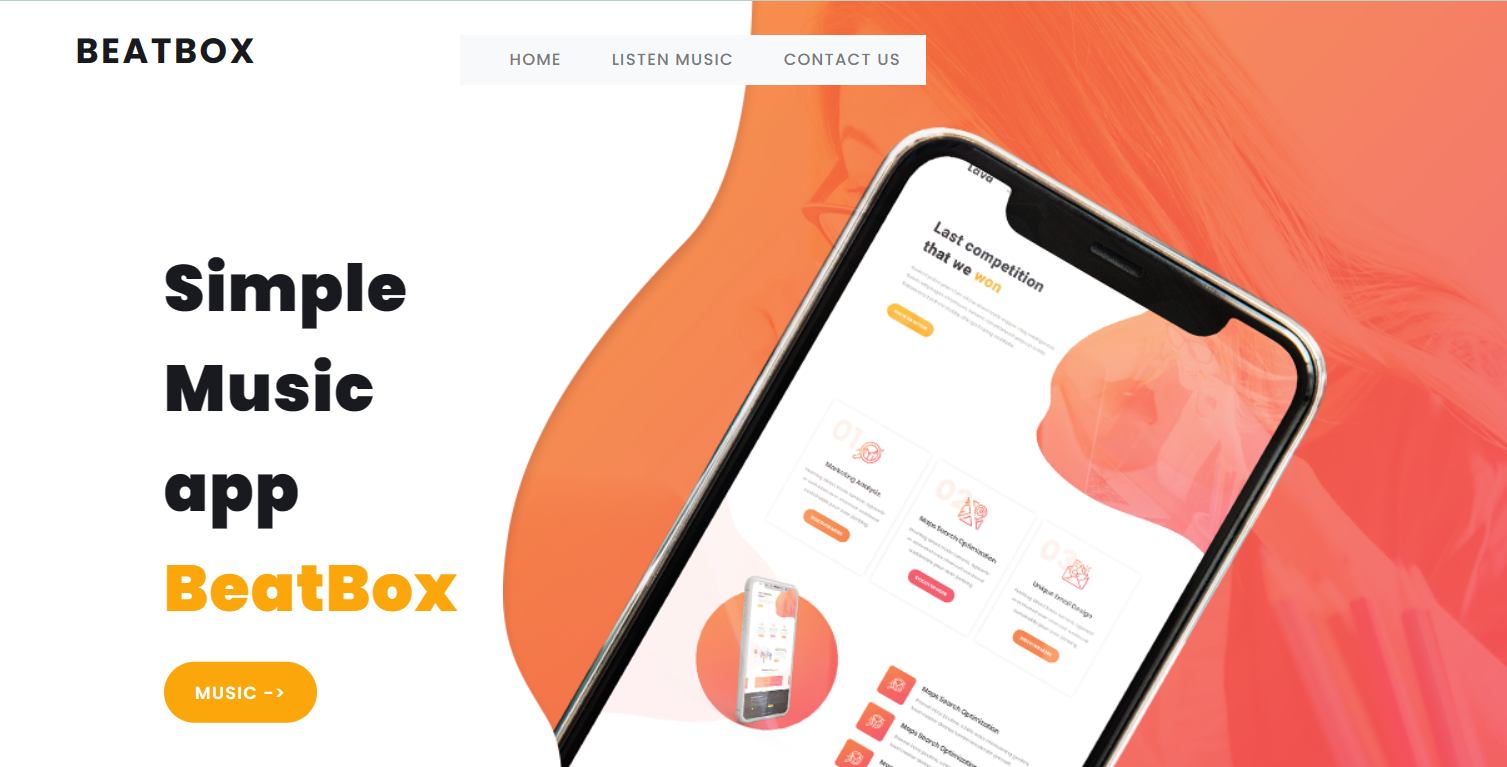




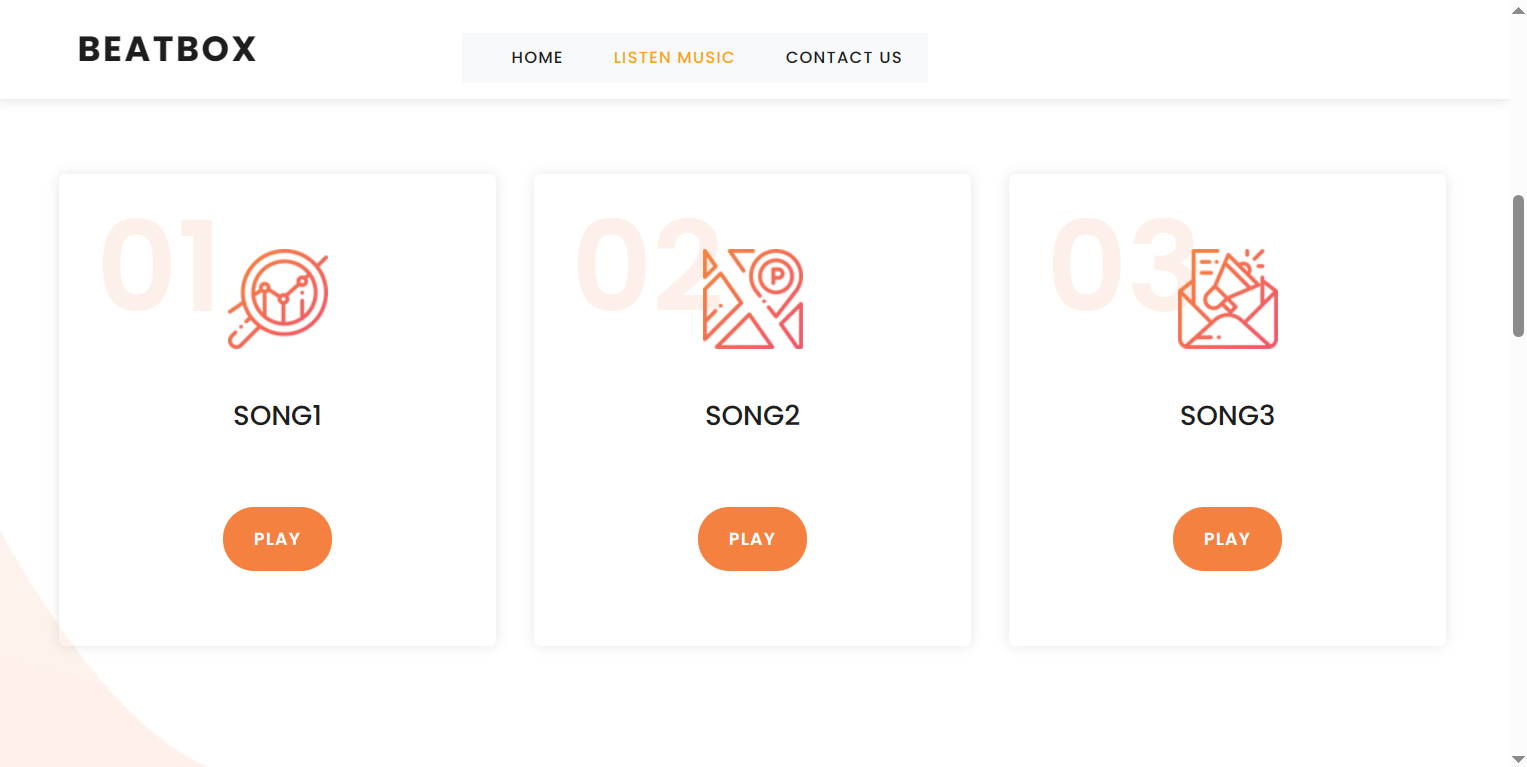
# **Webpage Through Flask:**

The final phase of the project was to deploy our Music Recommendation Model named “BeatBox” on a web page using the Flask so we did that by creating a user friendly interface for the user. Our web page is responsive. Also we have used vibrant colors for our webpage. We have created header, footer, contact us page, and also music section. When the user will select a song we will apply the mfcc mechanism on the audio that the user has selected and base on that we will show the user further audios and the process will continue.

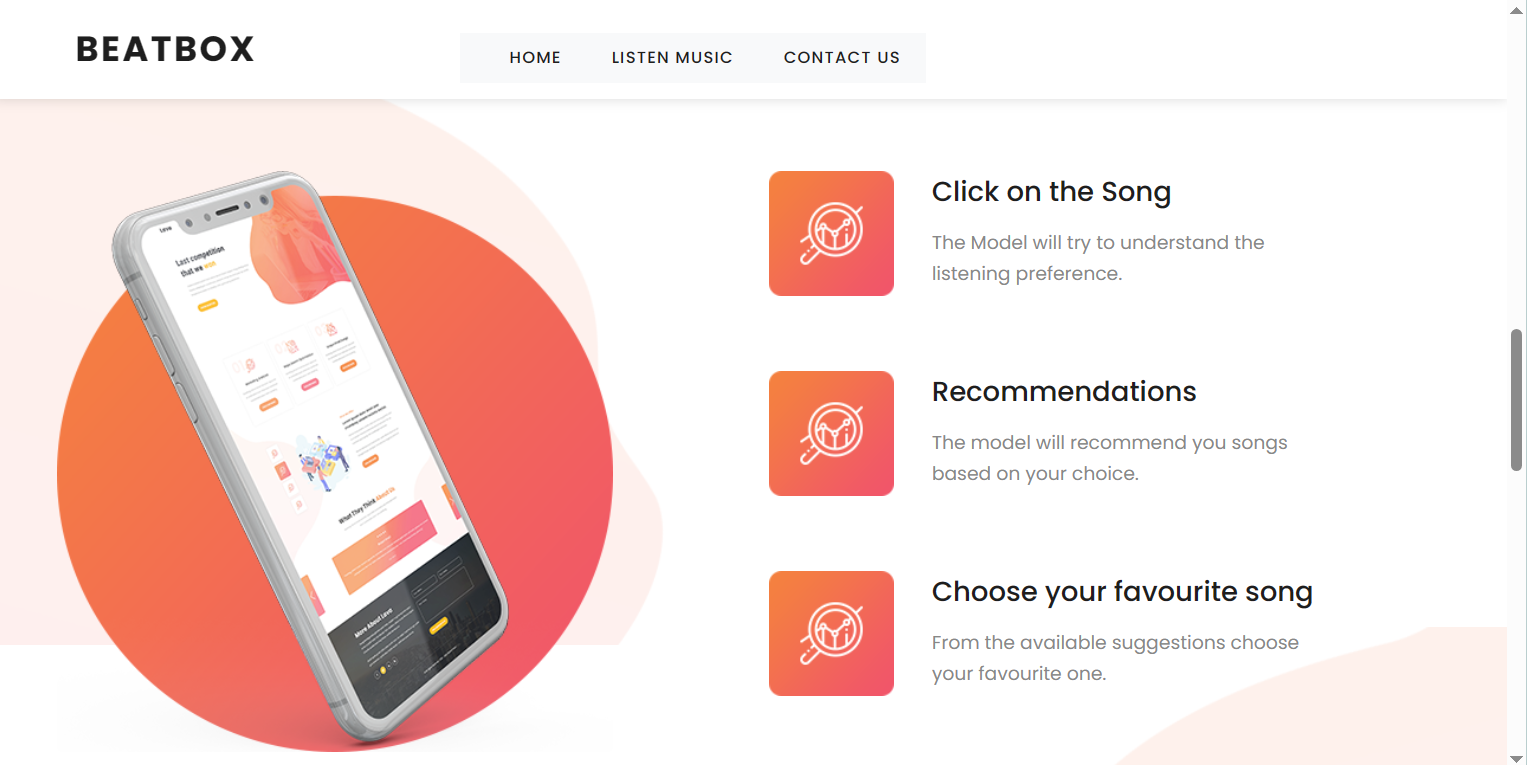
The interface of our website looks something like this.



The music section will pop when the user will click on the MUSIC-> button present in our hero section. Also for easy navigation we have mentioned LISTEN MUSIC option so when ever the usr will click on it he or she will be directed towards the music section of our Music Recommendation Model.



We have also attached a guide on how our model works within our website. The section which displays the whole system is as follows .



Lastly we have also incorporated the contact us page along with the footer of the web application.

