# Music Recommendation System

Project on

Data Science and Machine Learning

Submitted By

Ribesh Shrestha

BIM 8th Semester

Submitted to

Pawan Niroula

## Introduction:

The Music Recommendation System is designed to enhance the listening experience by providing personalized music suggestions based on user preferences and behaviors. By leveraging advanced algorithms and data analytics, the system aims to connect users with songs and artists they are likely to enjoy, thereby increasing user engagement and satisfaction. This documentation serves as a comprehensive guide for developers, data scientists, and end-users, detailing the system's architecture, functionality, and usage.

## System Overview:

The system overview of a music recommender system should include the architecture, key components, and data flow. It should detail how the Spotify interface integrates with Kaggle datasets for music data, describe the use of Jupyter Notebook for data preprocessing and analysis, and outline the recommendation algorithms employed to generate personalized music suggestions.

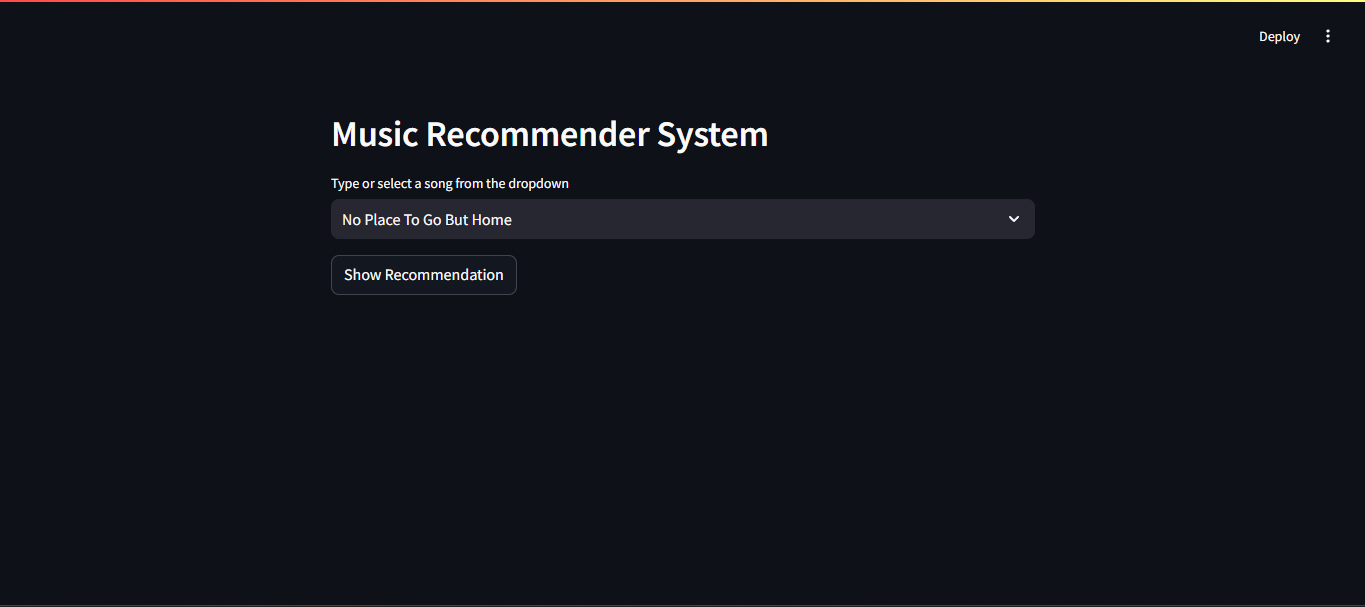
### Key Components:

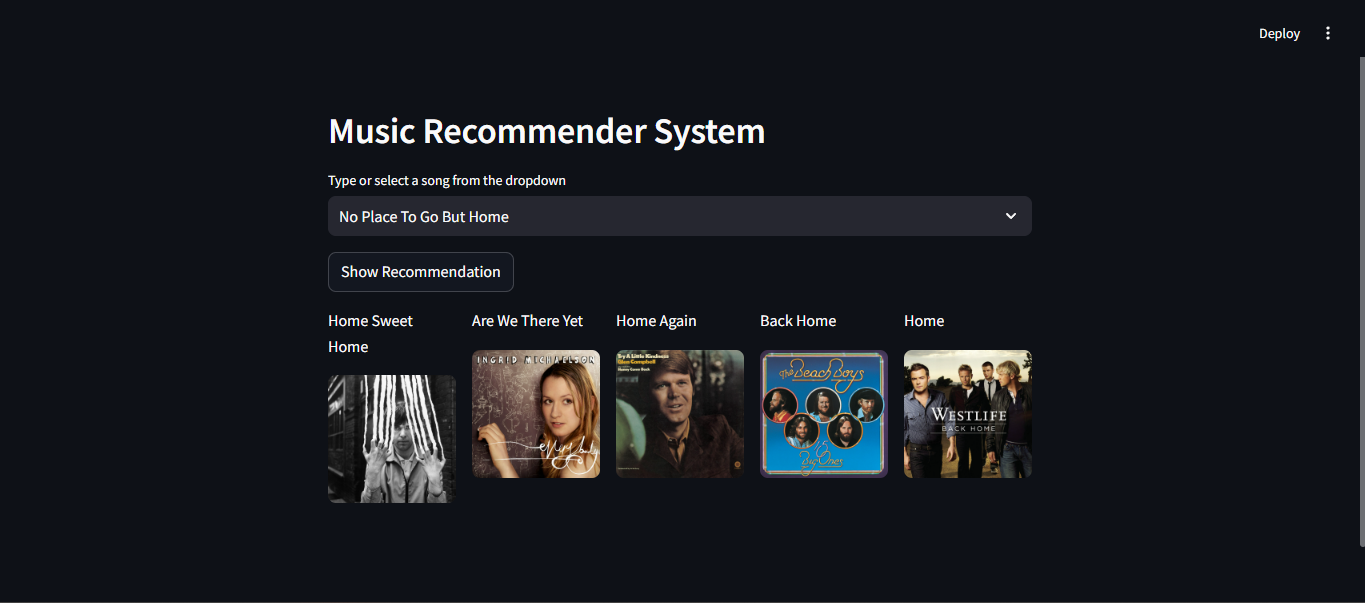
* Spotify Interface: The system integrates with the Spotify API to access a vast library of songs, playlists, and user data. This interface enables the retrieval of audio features and metadata, which are essential for generating accurate recommendations.
* Kaggle Datasets: The system utilizes datasets from Kaggle, such as the Million Playlist Dataset, which contains a rich collection of playlists and associated metadata. This data serves as the foundation for training the recommendation algorithms and enhancing the system's understanding of user preferences.
* Jupyter Notebook: Jupyter Notebook is employed for data exploration, preprocessing, and analysis. It allows data scientists to interactively clean and manipulate the datasets, perform exploratory data analysis, and visualize trends and patterns in the music data.

### Data Flow:

* Data Ingestion: The system collects data from the Spotify API and Kaggle datasets. This includes user interaction data, song features, and playlist information.
* Data Preprocessing: Using Jupyter Notebook, the raw data is cleaned and transformed. This step involves handling missing values, normalizing audio features, and creating feature vectors for songs.
* Recommendation Engine: The core of the system is the recommendation engine, which employs collaborative filtering and content-based filtering techniques. By analyzing user preferences and song attributes, the engine generates personalized music suggestions.
* User Interface: The user interface allows users to interact with the system, search for songs, and receive recommendations. Users can provide feedback on the suggestions, which helps refine the recommendation algorithms over time.

# Appendix:





# Conclusion:

This Music Recommendation System combines the power of the Spotify API, Kaggle datasets, and Jupyter Notebook to create a robust platform for music discovery. By integrating these components, the system aims to deliver a personalized and engaging experience for users, helping them discover new music that aligns with their tastes and preferences.