A Capstone project Overview

MUSIC RECOMMENDATION SYSTEM

Team:

MOHAMMED MUDASSIR HUSSAIN - 2103A52058
SYED KHAJA MUKARRAM AJAZ - 2103A52069
MOHAMMED NAVEED SHARIEF - 2103A52159

Guide:

Dr. Mamta Pandey,
Associate Professor, School of CS&AI

Introduction

The Music Recommendation System is a robust platform that utilizes Spotify's API and multiple machine learning models to deliver music recommendations based on detailed audio features, including attributes like danceability, energy, and tempo.

Problem Statement

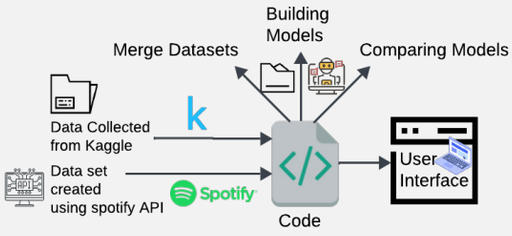
Music discovery is often challenging due to vast libraries and varying preferences. Users need personalized recommendations based on audio characteristics rather than just popularity or genre.

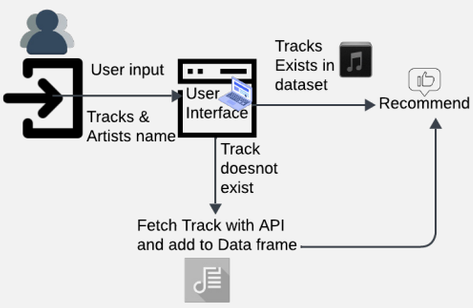
Objective:

- Provide personalized music recommendations using multiple algorithms
- Analyze audio features to find similar songs

System Architecture:

- Spotify API Integration
- Multiple ML Models (KNN, DBSCAN, Cosine Similarity, PCA)
- Python with pandas, scikit-learn
- Data Processing Pipeline
- Feature Engineering





System Architecture

Weeks 1–4: Research Analysis Activities:

- Integrated Spotify API for music data collection
- Gathered tracks from diverse playlists
- Collected audio features for thousands of songs
- Analyzed track characteristics and patterns
- Created data visualization pipelines

Outcomes:

- Comprehensive dataset with audio features
- Visual analysis of popular tracks
- Word cloud generation for track analysis
- Standardized data scaling implementation

Weeks 5–8: Developmen Phase

Frontend Development:

- Built the user interface using Streamlit for an interactive and user-friendly design.
- Designed and implemented reusable UI components such as track search forms, recommendation displays, and visualization tools (word clouds, bar charts).
- Enabled real-time interaction and seamless navigation for dynamic dataset updates and track recommendations.

Backend Development:

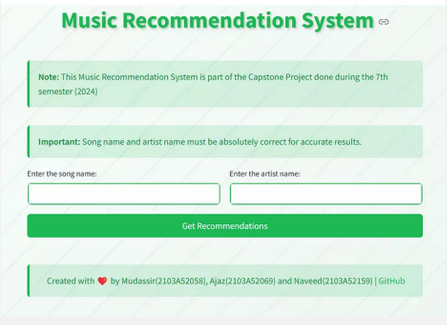
- Set up the backend using Python and integrated Spotify's API via the spotipy library for fetching metadata and audio features.
- Processed and handled user requests for track searches, data retrieval, and recommendation generation dynamically.
- Implemented efficient algorithms for KNN, DBSCAN, and Cosine Similarity to ensure robust recommendation logic.

APIs and Integration:

- IIntegrated Spotify Web API to fetch metadata and audio features dynamically.
- Established backend functions for real-time API communication.

Outcomes:

- Delivered a functional recommendation system with real-time interaction.
- Provided personalized recommendations with dynamic dataset updates.
- Integrated visualization tools for user insights.



Music Recommendation System App

Weeks 9–12: Testing and Quality Assurance

Unit Testing:

- Tested individual functions and Spotify API integration to ensure backend stability.
- Verified machine learning models (DBSCAN, Cosine Similarity) and data preprocessing for accuracy.

Integration Testing:

- Tested interaction between backend and Spotify API, ensuring seamless data retrieval and recommendations.
- Verified dynamic dataset updates and real-time API fetching functionalities.

End-to-End Testing:

- Simulated user workflows (e.g., fetching tracks, generating recommendations, and dataset updates).

Bug Fixing and Refinements:

- Optimized API calls and preprocessing workflows for improved performance.

Outcomes:

- Delivered a stable and functional recommendation system with tested core features.

Weeks 13–16: Deployment and Optimization

Deployment:

- Hosted the application locally and tested on Streamlit, ensuring seamless functionality during development.

Performance Optimization:

- Optimized data preprocessing steps to handle larger datasets efficiently.
- Improved machine learning model execution time for faster and accurate recommendations.
- Reduced API call frequency and implemented batch requests to enhance overall system performance.

Feedback Incorporation:

- Conducted user testing sessions to evaluate recommendation accuracy and usability.
- Incorporated feedback to enhance UI design and added features like improved visualizations.

Outcomes:

- Delivered a functional recommendation system with real-time Spotify API integration.
- Created a user-friendly interface with accurate and personalized recommendations.