SCHOOL OF COMPUTER SCIENCE

A

PROJECT REPORT

ON

Song Recommendation System

(FUNDAMENTALS OF DATA SCIENCE)

B.C.A - IV SEMESTER

JAN-MAY, 2025

Submitted to:

Dr. Surbhi Saraswat

Submitted by:

Harsh kumar singh

Table of Contents

1. Introduction

Song Recommendation System 1.2 Objective

Develop a web-based application that recommends songs based on user input by leveraging machine learning techniques and the Spotify API.

1.3 Purpose

Enhance music discovery by combining song feature analysis and artist similarity algorithms to provide personalized recommendations. This is inspired by advanced techniques described in research on music personalization.[1]

1.4 Scope

The project focuses on computing similarities using cosine and Jaccard similarity measures, integrating data from the Spotify API for rich metadata, and delivering recommendations via a responsive user interface with features like dark mode [2]

2. System Architecture

2.1 Overview

The system processes user input through a Flask backend. It computes similarity scores on preprocessed song features, retrieves additional metadata via the Spotify API, and then displays curated recommendations on an HTML/CSS-based frontend.

2.2 Technology Stack

- **Languages:** Python, HTML, CSS
- **Libraries:** Flask, scikit-learn, pandas, RapidFuzz, NumPy
- **APIs:** Spotify for Developers API
- **Frontend:** HTML with custom CSS styling using the DM Sans font

Algorithms and Techniques Used:

Technique	Purpose	Category
Cosine Similarity	Feature-based similarity between songs	Content-based filtering
Jaccard Similarity	Artist similarity	Collaborative hint
Fuzzy Matching	Flexible user input matching	UX Improvement
StandardScaler	Normalize features	Preprocessing
Hybrid Weighted Score	Balance between audio and artist similarity	Hybrid Filtering
Noise & Shuffling	Inject randomness and diversity	Novelty / Serendipity
Spotify API	Enrich UI with real-world metadata	External Integration

2.3 File Structure

- **app.py:** Contains backend logic including recommendation algorithms and API integration
- **index.html:** Manages the frontend interface including the display of recommendations
- **data.csv:** Dataset file with features like valence, tempo, popularity, etc.

3. Functional Components

3.1 Data Acquisition

Data is loaded from a CSV file that includes various numerical features (e.g., valence, tempo) and artist information. Preprocessing steps include:

- Parsing the artists field for list conversion
- Standardizing numerical data using StandardScaler This aligns with data preparation techniques found in music recommendation literature [3]

3.2 Backend Server

The Flask backend performs the following functions:

- Accepting user input via form submissions
- Employing fuzzy matching to find close song or artist names
- Computing cosine similarity on song features and Jaccard similarity for artist overlap
- Making API calls to Spotify to retrieve album cover images and track URLs These steps are based on common practices in modern recommendation systems [1].

3.3 Frontend UI

The user interface is built with HTML and CSS and offers:

- A sleek, responsive design compatible with mobile and desktop devices
- A dark mode toggle for enhanced user experience
- Display of song recommendations along with album artwork and direct Spotify links
  This design supports a user-friendly display of data-driven decisions [4].

---

4. Implementation

4.1 Setup

- **Dependencies:** Install necessary Python libraries (Flask, scikit-learn, pandas, etc.)

- **Configuration:** Set up Spotify API credentials (CLIENT_ID, CLIENT_SECRET)
- **Preprocessing:** Load and normalize data from data.csv This step form

4.2 Execution

Run the Flask application locally. The user enters a song or artist, and the backend processes the input, computes similarity scores, and returns recommendations. The Spotify API is queried in real time to provide current album metadata.

4.3 Challenges

Key challenges include:

- **Token Caching:** Efficiently reusing the Spotify API access token to reduce latency
- **Data Quality:** Managing noisy or ambiguous user input through fuzzy matching
- **Randomness:** Balancing diversity and recommendation relevance with controlled random noise
  Addressing these challenges is crucial for achieving a robust application [1], [3]

4.4 Reference Management with Mendeley

Citations in this documentation are managed using the Mendeley Cite add-in in Word. This tool automates in-text citation insertion and bibliography generation in your chosen style, ensuring consistency throughout the report.

---

5. Unique Features

- **Fuzzy Matching:** Enables flexible user input interpretation for both song titles and artist names
- **Dual Similarity Measures:** Combines cosine similarity (for song features) and Jaccard similarity (for artist overlap)
- **Enhanced Diversity:** Incorporates random noise to ensure varied recommendation outputs
- **User Interface Design:** Offers dark mode and responsive design for improved usability

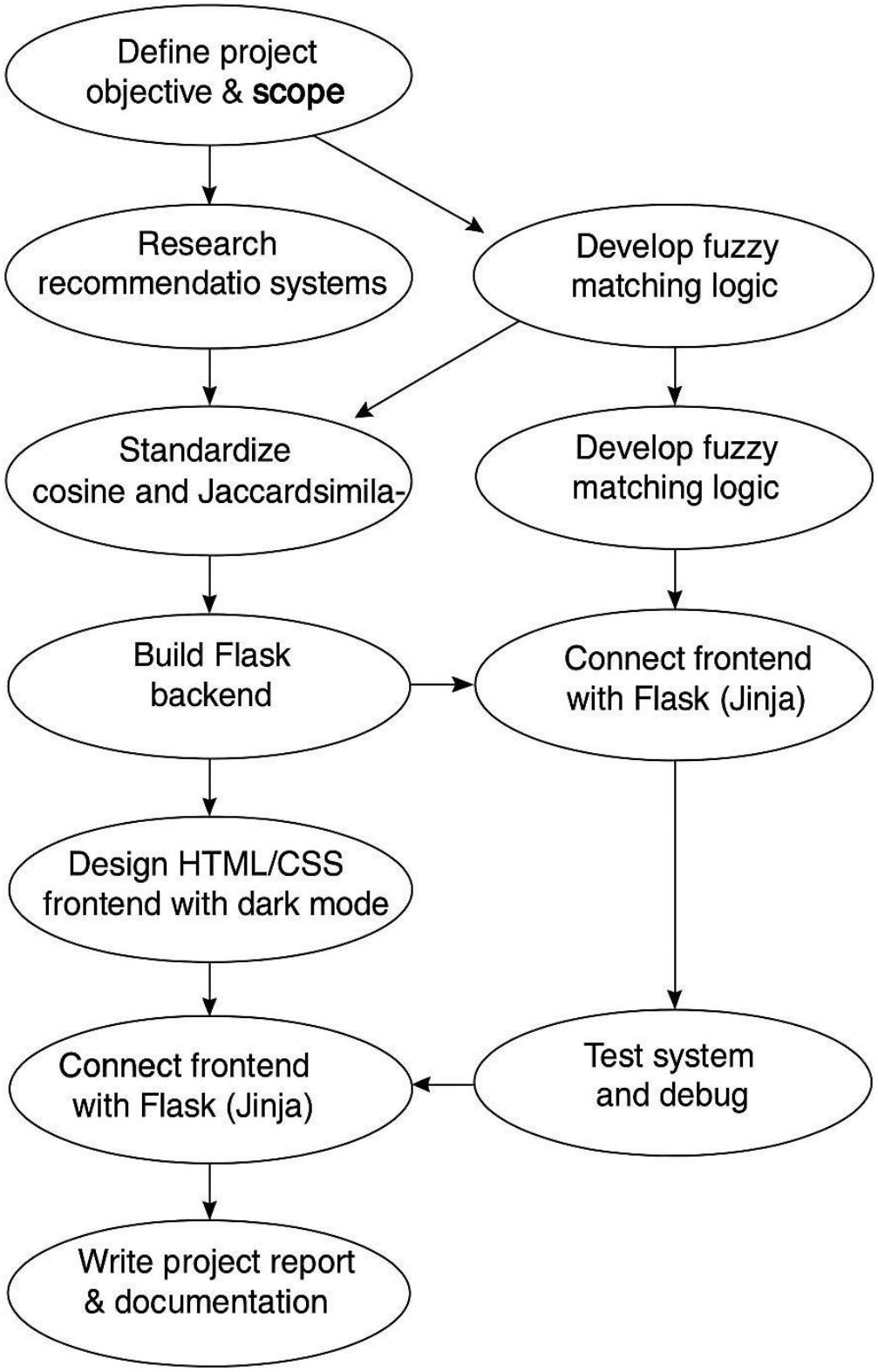
  [3]

6. Future Work

Potential enhancements include:

- **User Authentication:** Integrating Spotify login for personalized recommendations
- **Playlist Generation:** Automatically curating playlists based on user preferences
- **Advanced Filters:** Expanding feature-based filters such as mood or genre
- **Deep Learning:** Experimenting with neural network models for higher recommendation accuracy
  Research indicates that deep learning approaches can further refine recommendations [5], [6].

PERT CHART



PERT Chart for Recommendation System

7. Conclusion

The Spotify Song Recommendation System successfully integrates machine learning techniques with real-time data from the Spotify API. The project not only demonstrates effective use of cosine and Jaccard similarity measures for recommendation but also prioritizes a seamless user experience through its responsive and modern UI. These attributes form a scalable foundation for more advanced future enhancements [7].

---

8. Literature Review

8.1 Overview

Research on music recommendation systems emphasizes the importance of personalized content delivery in the era of digital streaming. This review covers foundational work on collaborative filtering, content-based filtering, and hybrid approaches, along with studies on the application of deep learning in music analysis.

8.2 Collaborative Filtering

Early methods focused on collaborative filtering by analyzing user-item interaction matrices to infer preferences (Sarwar et al., 2001). While effective, these methods can suffer from data sparsity and the cold-start problem.

8.3 Content-Based Filtering

Content-based techniques analyze the intrinsic attributes of songs (such as timbre, rhythm, and genre). Studies like those by Lops et al. (2011) demonstrate how leveraging detailed audio features can tailor recommendations to specific musical tastes. However, an overspecialization is exist when system diversity is compromised.

8.4 Hybrid Recommendation Systems

Hybrid approaches, such as combining collaborative and content-based filtering, have been shown to overcome many limitations inherent to singular techniques. Burke (2002) highlighted these methods as a means to improve both accuracy and the novelty of recommendations. Recent work by [6]further explores the potential of deep learning architectures to enhance these methods.

8.5 Deep Learning and Music Recommendation

Advances in deep learning have introduced new algorithms for music analysis. Van den Oord et al. (2013) and later works have employed convolutional neural networks and autoencoders for feature extraction from raw audio data. Such techniques, when applied to recommendation systems, offer improved personalization and robustness against noisy input [1].

8.6 Real-World Implementations

Spotify's own recommendation engine is a hybrid system that incorporates both user behavior and song metadata [1]. The use of live APIs and large-scale datasets, such as those found on Spotify and the Million Song Dataset, exemplifies the transition from academic research to practical applications [7].

8.7 Summary

The literature consistently advocates for a combination of methods to address the challenges in music recommendation. This project embraces both traditional statistical methods and modern machine learning techniques, reflecting the comprehensive strategies recommended in current research

---

9. References

[1] K. Jacobson, V. Murali, E. Newett, B. Whitman, and R. Yon, "Music Personalization at Spotify," pp. 373–373, Sep. 2016, doi: 10.1145/2959100.2959120.

[2] G. Björklund, M. Bohlin, E. Olander, J. Jansson, C. E. Walter, and M. Au-Yong-Oliveira, "An Exploratory Study on the Spotify Recommender System," *Lecture Notes in Networks and Systems*, vol. 469 LNNS, pp. 366–378, 2022, doi: 10.1007/978-3-031-04819-7_36.

[3] A. Umrani, V. Satpute, A. Chandsare, and Y. Umadi, "Music Recommendation System using Euclidean, Cosine Similarity, Correlation Distance Algorithm and Flask Web Application," 2023, Accessed: Apr. 15, 2025. [Online]. Available: https://www.academia.edu/download/105047227/IRJET_V10I714.pdf

[4] S. Al-Otaibi, N. Altwoijry, … A. A.-I. J. of, and undefined 2022, "Cosine similarity-based algorithm for social networking recommendation," *academia.edu*, Accessed: Apr. 15, 2025. [Online]. Available:

https://www.academia.edu/download/87477808/84_1570727168_25955_EM_8Sep_9jun_F.pdf

[5] E. Sarin, S. Vashishtha, S. K.-2021 4th International, and undefined 2022, "SentiSpotMusic: a music recommendation system based on sentiment analysis," *ieeexplore.ieee.org*, Accessed: Apr. 15, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9781862/

[6] X. Wang, Y. W.-P. of the 22nd A. international, and undefined 2014, "Improving contentbased and hybrid music recommendation using deep learning," *dl.acm.org*, pp. 627–636, Nov. 2014, doi: 10.1145/2647868.2654940.

[7] G. Björklund, M. Bohlin, E. Olander, … J. J.-… on I. S., and undefined 2022, "An exploratory study on the Spotify recommender system," *Springer*, Accessed: Apr. 15, 2025. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-04819-7_36