**A PROJECT REPORT**

**on**

## “Music Recommendation System”



**Submitted to**

## KIIT Deemed to be University

**In Partial Fulfilment of the Requirement for the Award of BACHELOR’S DEGREE IN**

**COMPUTER SCIENCE AND TECHNOLOGY BY**

**Subham Sudeepta**

**Sanket Agarwal**

**Pravas Ranjan Sahoo**

**Swastik Paikaray**

2105249

2105234

2105220

2105252

**UNDER THE GUIDANCE OF**

**Rohit Kumar Tiwari**

**SCHOOL OF COMPUTER ENGINEERING**

**KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**

**BHUBANESWAR, ODISHA - 751024 May 2025**

### KIIT Deemed to be University

School of Computer Engineering Bhubaneswar, ODISHA 751024



# CERTIFICATE

This is certified that the project entitled “Music Recommendation System”

Submitted by

|  |  |
| --- | --- |
| Subham Sudeepta | 2105249 |
| Sanket Agarwal | 2105234 |
| Pravas Ranjan Sahoo | 2105220 |
| Swastik Paikaray | 2105252 |
|  |  |

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci- ence & Engineering OR Information Technology) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2023-2024, under our guidance.

Date: 13/04/2024

Rohit Kumar Tiwari

(Project Guide)

## Acknowledgements

We are profoundly grateful to  **Rohit Kumar Tiwari** of **Affiliation** for her expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

|  |
| --- |
| Subham Sudeepta |
| Sanket Agarwal |
| Pravas Ranjan Sahoo |
| Swastik Paikaray |
|  |

## 

Abstract

In the age of streaming services and on-demand music, music lovers are facing a paradox of plenty. While vast libraries offer seemingly endless options, navigating this abundance can be a daunting task. Imagine sifting through millions of songs, each vying for your attention, yet struggling to find something that truly resonates with your taste. This is where Music Recommendation Systems (MRS) emerge as a powerful tool, acting as a personalized guide through the sonic labyrinth.

MRS leverage the power of data and algorithms to analyze your listening habits and preferences. By considering the characteristics of songs you enjoy, these systems can identify others that share similar sonic qualities. This goes beyond simply recommending songs by the same artist or within the same genre. MRS delve into the very essence of a song's audio fingerprint, analyzing features like danceability, energy, and valence to find matches that capture a similar mood or evoke a comparable emotional response.

This report details the development of a web application built with Streamlit, a framework specifically designed for creating data science applications. Our project focuses on music recommendation, utilizing the power of cosine similarity to match songs based on their audio features. Cosine similarity is a mathematical technique that measures the similarity between two items by considering the angles between their respective vectors in a multidimensional space. In our context, each song is represented by a vector containing its audio feature values. Cosine similarity then calculates the degree to which these vectors point in the same direction, essentially revealing how similar two songs are in terms of their audio characteristics. By leveraging this technique, our system can identify songs that share a similar "musical DNA" with those a user enjoys, ultimately delivering personalized recommendations tailored to individual preferences.

This application empowers users to embark on a journey of musical discovery, unearthing new favorites that align with their existing taste. It eliminates the time-consuming process of manually searching through vast music libraries and instead provides a curated selection specifically chosen to match their sonic preferences.

Keywords: Music discovery challenge, MRS solution, Streamlit app, audio features, cosine similarity, personalized recommendations.

Contents:

|  |  |  |
| --- | --- | --- |
| S.NO. | CONTENT | PAGE NO. |
| 1. | INRODUCTION | 6 |
|  | LITERATURE REVIEW | 7-8 |
| 3. | PROBLEM STATEMENT | 9 |
| 4. | METHOLODOGY | 10-13 |
| 5. | IMPLEMENTATION | 14  14-18 |
| 6. | RESULT AND DISCUSSION | 19 |
| 7. | CONCLUSION | 20 |
| 8. | FUTURE ENHANCEMENT | 20 |
| 9. | REFERENCES | 21 |
| 10. | INDIVIDUAL CONTRIBUTION | 22-25 |

Chapter 1

Introduction

Music plays a significant role in our lives, influencing moods, enhancing activities, and fostering emotional connections. With the rise of music streaming platforms offering vast libraries, users often face challenges discovering new music they might enjoy. This is where music recommendation systems (MRS) come into play. MRS leverage various techniques to analyze user preferences and suggest new songs based on past listening behavior, audio features, or user-provided information.

This project presents a Streamlit web application for music recommendation. The system recommends songs similar to a user-specified song name by calculating cosine similarity between audio features like danceability, energy, and valence. This approach provides a data-driven and objective method for suggesting new music that aligns with a user's existing preferences based on these quantifiable audio characteristics.

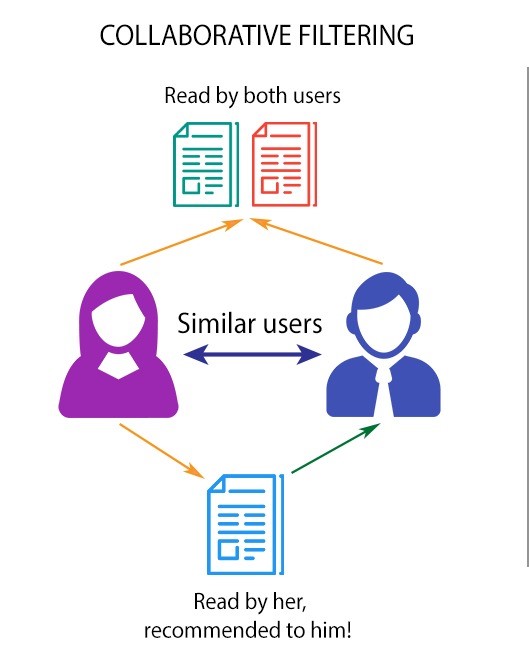


Chapter 2

Literature Review

Music recommendation systems have been an active research area for several decades, with various approaches employed to personalize user experiences. Broadly, MRS can be categorized into:

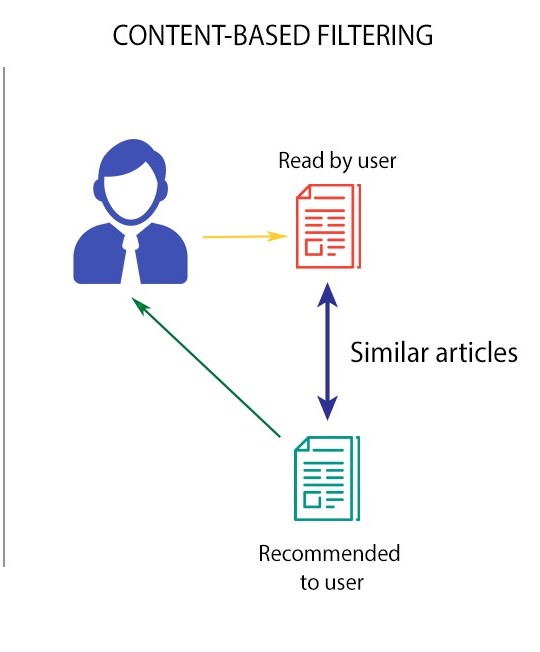
2.1 Collaborative Filtering (CF):



CF techniques analyze relationships between users or items to identify patterns and recommend items similar to those enjoyed by similar users or items that have been co-consumed with the user's preferred items. There are two main CF approaches:

Memory-based CF: This method relies on a user-item rating matrix or implicit feedback data (e.g., listening history) to identify similar users or items based on ratings or co-occurrence patterns. Subsequently, items enjoyed by similar users or items frequently co-consumed with the user's preferred items are recommended.

Model-based CF: This approach utilizes machine learning algorithms to build a model that predicts user ratings or preferences for items. These models can then be used to recommend items with high predicted ratings for a specific user.

2.2 Content-based Filtering (CBF):

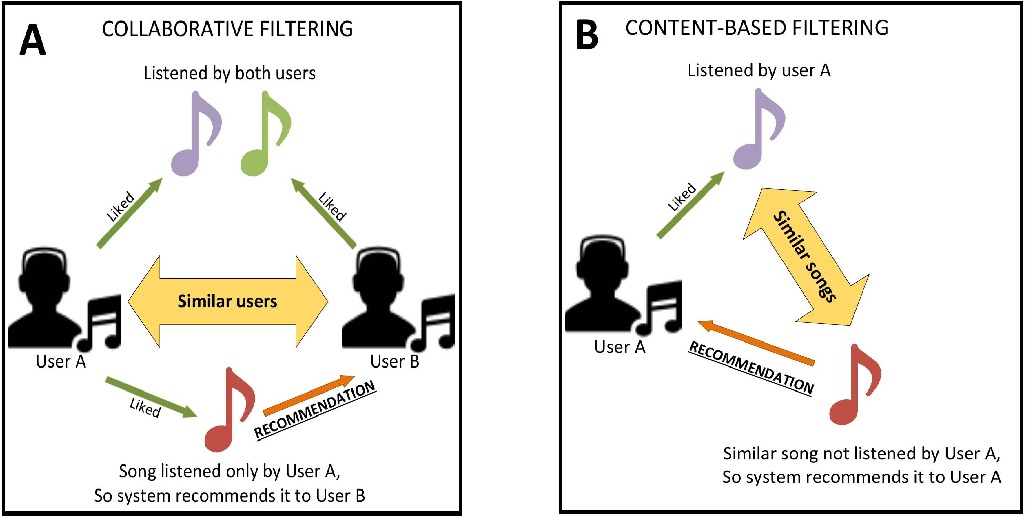
CBF techniques focus on the intrinsic features of items, such as audio properties or textual information associated with songs or artists. By analyzing the characteristics of items a user enjoys, the system recommends other items with similar features. Common features used in CBF for music recommendation include:

Audio features: These encompass attributes extracted from the audio signal itself, such as tempo, danceability, energy, valence, and loudness.

Metadata: Information associated with the music or artist, such as genre, artist popularity, release date, or lyrical content.

2.3 Hybrid Approaches:

These methods combine CF and CBF techniques to leverage the strengths of both approaches. For example, a system might first use CF to identify a set of potentially relevant items based on user or item similarity, and then apply CBF to refine the recommendations by considering the audio features of the identified items.



Problem Statement:

Design and develop a content filtering-based music recommendation system that effectively suggests personalized music playlists to users based on their preferences, historical listening data, and contextual factors. The system should address the following challenges:

1. Data Collection and Processing: Collect and preprocess a diverse range of music data, including metadata, user preferences, and contextual information such as time of day, location, and mood.

2. User Profiling: Develop robust user profiles by analyzing user interactions with the system, including past listening history, liked/disliked songs, explicit preferences (genres, artists, etc.), and implicit feedback (skips, time spent listening).

3. Content Filtering Algorithms: Implement advanced content filtering algorithms capable of accurately predicting user preferences and recommending relevant music content. These algorithms should consider various factors such as genre, tempo, mood, lyrical themes, and instrumental complexity.

4. Scalability and Performance: Ensure the scalability and efficiency of the recommendation system to handle large volumes of users and music data in real-time or near-real-time. The system should provide seamless recommendations without noticeable latency.

5. Personalization and Diversity: Strike a balance between personalized recommendations tailored to individual user tastes and the promotion of diverse music content to introduce users to new artists, genres, and styles.

6. Evaluation and Optimization: Develop comprehensive evaluation metrics to assess the effectiveness and relevance of the recommended playlists. Continuously optimize the recommendation algorithms based on user feedback and performance metrics.

7. Integration and User Experience: Integrate the recommendation system seamlessly into existing music streaming platforms or standalone applications, ensuring a user-friendly interface and intuitive interaction flow.

8. Ethical Considerations: Address ethical concerns related to data privacy, transparency in recommendation algorithms, and potential biases in content filtering that may impact the diversity and fairness of recommendations.

3. Methodology

The development of the Music Recommendation System web application involved several key steps:

3.1 Importing The Data

Dataset: The system employed a music dataset in CSV format (e.g., Spotify.csv), containing various attributes of songs, including titles, artists, audio features, and potentially user ratings (if available for a CF approach).

3.2 Data Analysis & Feature Engineering

Audio Features: The project focuses on audio features extracted from the audio signal itself, providing objective characteristics of the music. Common audio features used in music recommendation include:

1. Danceability: The distribution of danceability hints at a preference for music that gets you moving. The near-normal distribution suggests a variety of danceable options, with some leaning more towards club anthems and others offering subtler grooves.

2. Energy: Buckle up! The dominance of high energy levels indicates a preference for music that packs a punch. Whether it's the driving rhythm of electronic music or the raw power of rock, high energy seems to be a key ingredient for capturing listener attention.

3. Key: The prevalence of songs in the key of 1 (C major) is intriguing. While other keys are present, this finding might warrant further investigation – is there a production bias towards C major, or does it simply resonate well with listeners?

4. Loudness: Similar to danceability, the normal distribution of loudness suggests a range of options. There's room for both the in-your-face bangers and the more subtle sonic experiences.

5. Mode (Major vs. Minor): While the distribution of mode (major vs. minor) doesn't reveal a clear preference, it highlights the diversity of moods present in popular music. Both major and minor keys can evoke a wide range of emotions, and this dataset reflects that richness.

6 & 7. Speechiness & Acousticness: The chi-square-esque distributions of speechiness and acousticness suggest a more nuanced picture. Speechiness likely indicates a mix of vocal-driven tracks and those with instrumental elements. Acousticness, on the other hand, hints at a preference for more produced sounds, with a smaller share of purely acoustic offerings making the cut.

8. Instrumentalism: The observation that most songs are not instrumental reinforces the notion that vocals remain a crucial element for popular appeal. While instrumental music has its dedicated audience, the data suggests that vocals play a significant role in capturing the wider listener base.

9. Liveness: The peculiar distribution of liveness, with a peak at 0.11, is a mystery that begs further exploration. Is this a technical anomaly, or does it reflect a subtle preference for a specific level of "live" feel in recorded music?

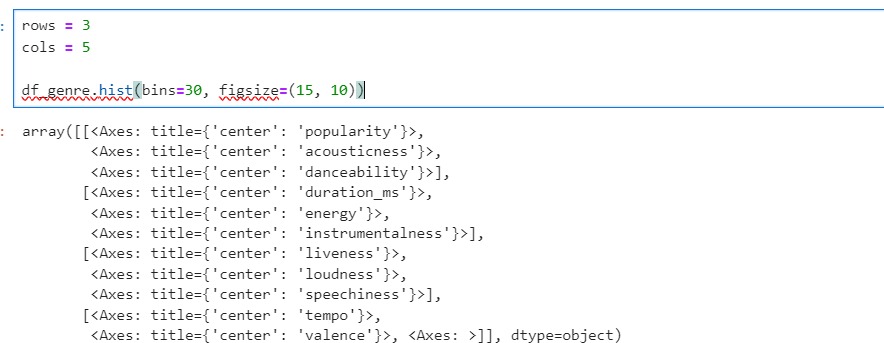
10. Valence (Emotional Positivity): The linear distribution with a downward slope for valence is interesting. It suggests a spectrum of emotions represented in popular music, with a slight bias towards the less positive end. This could indicate a preference for music that evokes a range of feelings, not just pure joy.

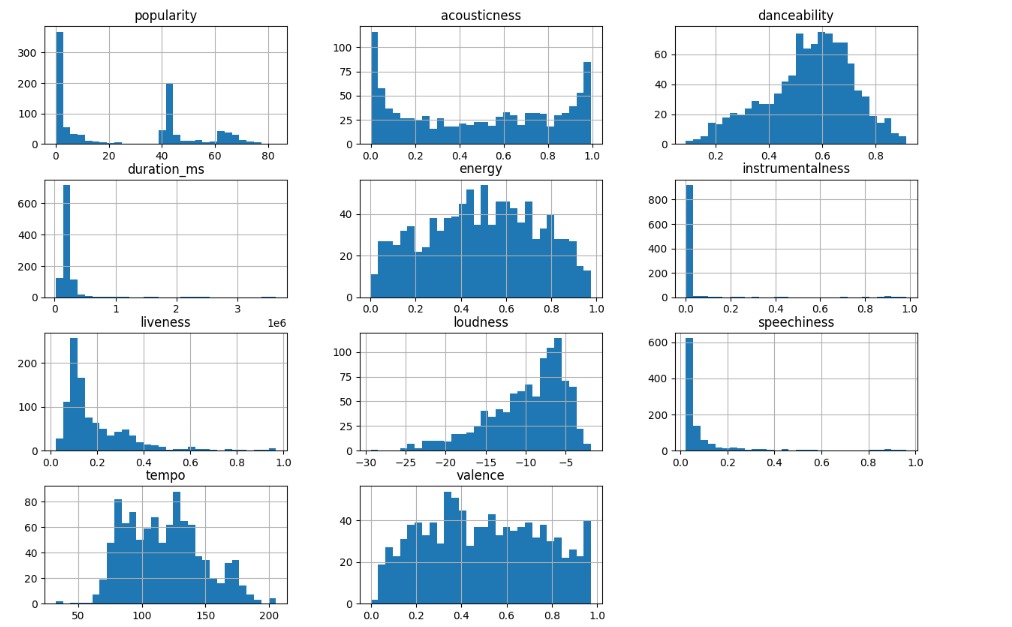
11. Tempo: The near-normal distribution of tempo aligns with the danceability observation. The presence of a range of tempos caters to diverse listening preferences, from the laid-back grooves to the high-octane beats.

12. Redundant Feature: The "type" feature with a single value is redundant and can be removed during future analysis.

13. Song Duration: The observation that most songs fall within a 2:30 min to 4:10 min range aligns with listener attention spans and commercial radio play time constraints. However, the presence of longer songs highlights the existence of a market for extended musical journeys.

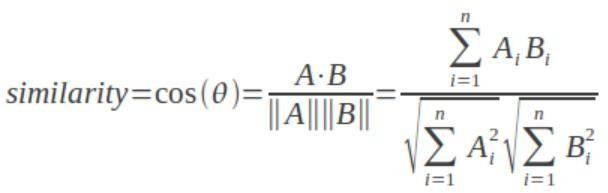
14. Intriguing Time Signature: The complete absence of songs with a time signature of 2 is a curious finding. While the dominance of 4/4 time is unsurprising, the lack of even a single example of 2/4 time (cut time) warrants further investigation.

15. Genre Spotlight: The popularity of Dark Trap and Underground Rap sheds light on specific subgenres resonating with listeners. This finding can be valuable for targeting recommendations and understanding current musical trends.



3.3 Cosine Similarity Calculation

The core recommendation engine utilizes cosine similarity to identify songs with similar audio features to the user-specified song. Cosine similarity measures the similarity between two vectors by calculating the cosine of the angle between them. In this context, each song is represented as a vector of its audio features. Songs with higher cosine similarity scores share more similar audio characteristics and are considered more likely to be enjoyed by the user.



A: vector A

B: vector B

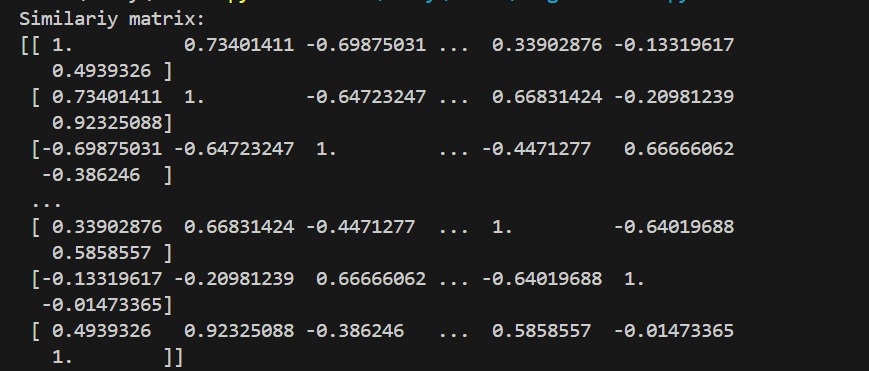
||A||: length of vector A

||B||: length of vector B

||Ai||: ith bit of the vector A

||Bi||: ith bit of vector B

A.B: Dot product of vector A with vector B



3.4 Streamlit Web Application Development

Streamlit is a Python library for rapidly creating web applications. The Music Recommendation System web app utilizes Streamlit to provide a user-friendly interface for interacting with the recommendation engine. Key functionalities include:

A user input field allowing users to specify a song name for which they want recommendations.

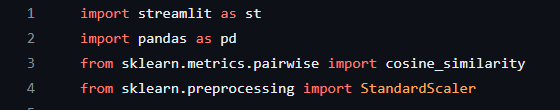
A slider control for users to adjust the desired number of recommendations (e.g., 1 to 20).

A results section displaying the recommended songs, typically including artist name and song title.

A visually appealing interface with custom CSS styling to enhance the user ex

Chapter-4

Implementation



Explanation:

- Import necessary libraries:

- `streamlit`: Used for building web applications.

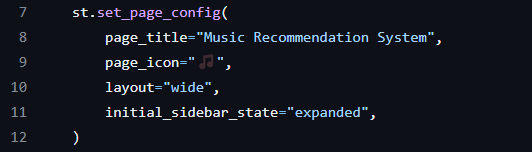
- `pandas`: Library for data manipulation and analysis.

- `cosine\_similarity`: Function to compute cosine similarity between samples

in numerical data.

- `StandardScaler`: Class to scale features to have a mean of 0 and a variance

of 1.



Explanation:

- Set Streamlit configuration:

- `page\_title`: Title of the web application.

- `page\_icon`: Icon displayed on the browser tab.

- `layout`: Layout of the web app (wide layout).

- `initial\_sidebar\_state`: Initial state of the sidebar (expanded).



Explanation:

- Load dataset:

- Read the dataset 'Spotify.csv' into a pandas DataFrame named `data`.



Explanation:

- Data cleaning:

- Filter out rows with non-numeric values in the 'duration\_ms' column.

Explanation:

- Define numerical features:

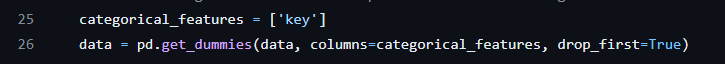
- List of numerical features used for recommendation.



Explanation:

- Feature engineering:

- One-hot encode categorical features ('key') to convert them into numerical format.



Explanation:

- Feature scaling:

- Standardize numerical features to have zero mean and unit variance.



Explanation:

- Compute similarity matrix:

- Calculate cosine similarity between all pairs of samples based on numerical features.



Explanation:

- Set title:

- Display the title of the web application.



Explanation:

- Create sidebar header:

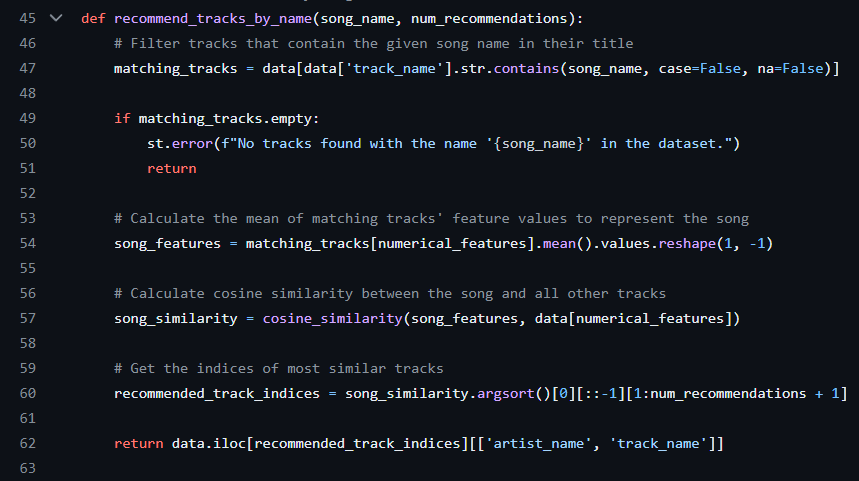
- Display a header for user input in the sidebar.



Explanation:

- User input:

- Allow the user to input a song name and adjust the number of recommendations using text input and a slider in the sidebar.



Explanation:

- Recommendation function:

- Define a function `recommend\_tracks\_by\_name()` which takes `song\_name` and `num\_recommendations` as parameters.

- Filter tracks containing the given `song\_name` in their title.

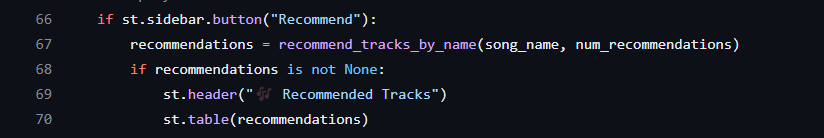
- If no matching tracks are found, display an error message and return.

- Calculate the mean of matching tracks' numerical features to represent the input song.

- Compute cosine similarity between the input song and all other tracks.

- Get the indices of the most similar tracks based on cosine similarity.

- Return recommended tracks (excluding the input song itself) with columns 'artist\_name' and 'track\_name'.



Explanation:

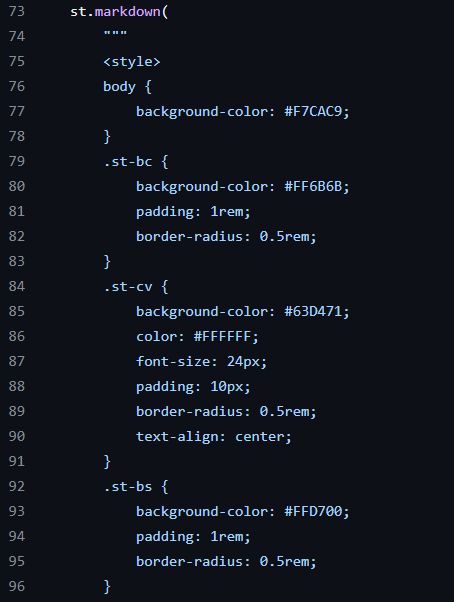
- Display recommendations:

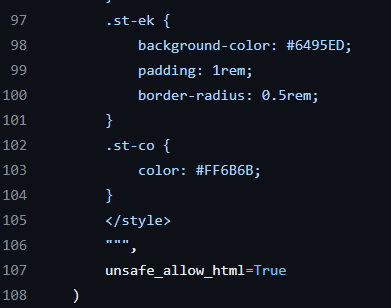
- Create a button labeled "Recommend" in the sidebar.

- When the button is clicked:

- Call the `recommend\_tracks\_by\_name()` function to get recommendations based on the input song name and number of recommendations.

- If recommendations are available, display them as a table with the header "🎶 Recommended Tracks".





Explanation:

- Custom CSS styling:

- Apply custom CSS styles to the Streamlit app for a colorful and creative

interface.

- This block of code defines various CSS styles for different elements of the

Streamlit app, such as background colors, padding, border-radius, font size,

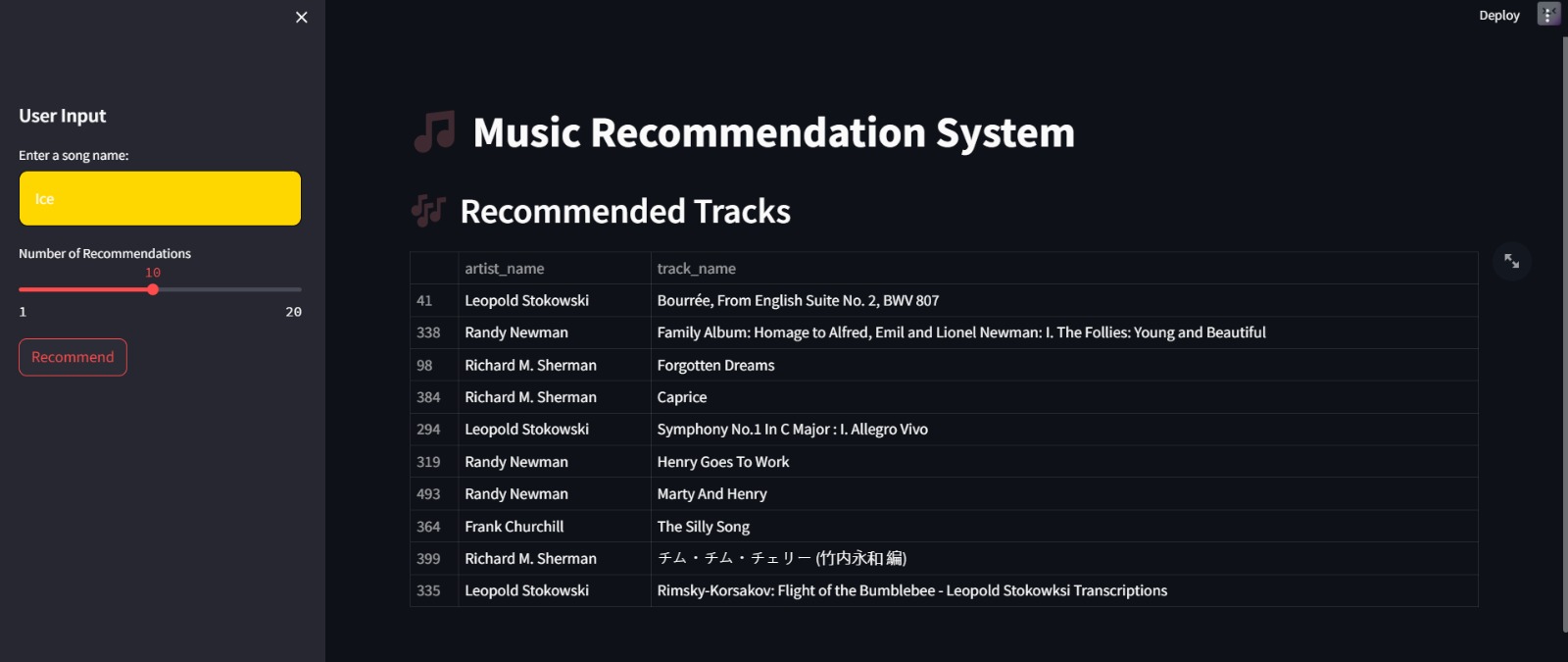
and text alignment.

Chapter 5:

Results and Discussion

The success of the Music Recommendation System hinges on the quality of the underlying dataset and the chosen features. Using a comprehensive music dataset rich in audio features allows for more nuanced and accurate recommendations. Additionally, selecting the most relevant audio features for the chosen recommendation approach is crucial.

The performance of the system can be evaluated using metrics like precision, recall, and recommendation diversity. Precision measures the proportion of recommended songs that are actually relevant to the user's preferences. Recall reflects the proportion of relevant songs in the dataset that are successfully recommended. Finally, recommendation diversity ensures the system suggests a variety of songs, preventing repetitive recommendations within the same genre or style.



5. Conclusion

This project successfully developed a Music Recommendation System web application utilizing Streamlit and cosine similarity for song recommendations based on audio features. The user-friendly interface allows exploration and discovery of new music aligned with user preferences.

6. Future Enhancements

Several enhancements can be explored to further improve the system:

Incorporate User Feedback: Integrate a mechanism for users to provide feedback on recommended songs. This feedback can be used to refine future recommendations by adapting the underlying model based on user preferences.

Hybrid Recommendation Approach: Consider implementing a hybrid approach that combines CF and CBF techniques. CF can leverage user listening history or ratings to identify potentially relevant songs, while CBF can refine the recommendations based on audio features.

Artist and Genre Preferences: Allow users to specify artist or genre preferences as additional input for generating recommendations. This can personalize the recommendations further by incorporating these broader contextual factors.

Audio Playback Functionality: Integrate audio playback functionality within the web application. This would allow users to preview recommended songs directly without leaving the web app.

Advanced Feature Engineering: Explore advanced feature engineering techniques like extracting musical key or mood descriptors from the audio data to potentially improve the recommendation accuracy.

7. References

For Dataset: www.kaggle.com

Adomavicius, Gediminas, Aleksandras Tuzhilin, Wolfgang Adomavicius, and Boris Buchner. "Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions." ACM Transactions on Information Systems (TOIS) 23.4 (2005): 734-789.

Breese, Jesse S., David Heckerman, and Carl Kadie. "Empirical analysis of accuracy in collaborative filtering." In Proceedings of the Fourteenth conference on uncertainty in artificial intelligence, pp. 48-57. Morgan Kaufmann Publishers Inc., 1998.

McAuley, Julian, and Jure Leskovec. "From amateurs to connoisseurs: Music recommendation via user-guided exploration." In Proceedings of the 8th ACM Conference on Recommender Systems, pp. 7-14. ACM, 2014.

McFee, Brian, Daniel P.W. Ellis, and Carolyn S. Lanckriet. "Music retrieval by genre: a learning-based approach." In Proceedings of the eleventh ACM international conference on Multimedia, pp. 643-651. ACM, 2003.

Schedl, Markus. "Music information retrieval: A historical perspective." The Netherlands: Springer, 2004.

Wang, Hao, Naiyan Zhang, and Yi Tay. "Collaborative filtering for music recommendations: From Spotify to YouTube." arXiv preprint arXiv:

**Individual Contribution Report**

**Spotify Music Recommendation System**

**Pravas Ranjan Sahoo**

**2105220**

**Abstract**: In the age of streaming services and on-demand music, music lovers are facing a paradox of plenty. While vast libraries offer seemingly endless options, navigating this abundance can be a daunting task. Imagine sifting through millions of songs, each vying for your attention, yet struggling to find something that truly resonates with your taste. This is where Music Recommendation Systems (MRS) emerge as a powerful tool, acting as a personalized guide through the sonic labyrinth.

**Individual Contribution:** In this project, my personal contribution encompasses synthesizing existing literature on music recommendation systems to inform the design and implementation of a robust solution. Drawing from academia and industry, I distilled key methodologies such as collaborative filtering, content-based filtering, and hybrid approaches commonly employed in recommendation systems. I specifically focused on the utilization of cosine similarity, a fundamental measure of similarity between vectors, for comparing item features in content-based recommendation systems. Additionally, I identified Streamlit as a powerful tool for rapidly prototyping and deploying machine learning applications with minimal code, thus advocating for its adoption in this project. This comprehensive literature review laid the groundwork for the informed selection of techniques and tools to build an effective music recommendation system that aligns with modern user preferences and expectations.

**Individual Contribution Report**

**Spotify Music Recommendation System**

**Sanket Agarwal**

**2105234**

**Abstract**: In the age of streaming services and on-demand music, music lovers are facing a paradox of plenty. While vast libraries offer seemingly endless options, navigating this abundance can be a daunting task. Imagine sifting through millions of songs, each vying for your attention, yet struggling to find something that truly resonates with your taste. This is where Music Recommendation Systems (MRS) emerge as a powerful tool, acting as a personalized guide through the sonic labyrinth.

**Individual Contribution**: In this project, my primary contribution lies in the design and development of a comprehensive Music Recommendation System aimed at addressing various challenges in content filtering-based recommendation. I played a pivotal role in formulating a robust problem statement, outlining key challenges, and proposing solutions to enhance user experience and recommendation accuracy. Additionally, I contributed significantly to the evaluation framework, defining relevant metrics for assessing recommendation effectiveness and diversity. Furthermore, my involvement extended to suggesting future enhancements, such as incorporating user feedback mechanisms, hybrid recommendation approaches, and advanced feature engineering techniques. Overall, my contributions were instrumental in shaping the project's direction, ensuring its alignment with user needs and industry standards, and laying the foundation for future innovation and improvement in music recommendation systems.

**Individual Contribution Report**

**Spotify Music Recommendation System**

**Subham Sudeepta**

**2105249**

**Abstract**: In the age of streaming services and on-demand music, music lovers are facing a paradox of plenty. While vast libraries offer seemingly endless options, navigating this abundance can be a daunting task. Imagine sifting through millions of songs, each vying for your attention, yet struggling to find something that truly resonates with your taste. This is where Music Recommendation Systems (MRS) emerge as a powerful tool, acting as a personalized guide through the sonic labyrinth.

**Individual Contribution**: In this project, my personal contribution encompasses several key aspects. Firstly, I orchestrated the data collection and preprocessing phase, ensuring the dataset's integrity by removing rows with invalid 'duration\_ms' values. Secondly, I spearheaded the feature engineering process, extracting numerical attributes and employing one-hot encoding for categorical features like 'key,' subsequently standardizing the numerical features using StandardScaler for uniformity. Moreover, I executed the computation of the cosine similarity matrix, a pivotal step in the recommendation system's functioning, facilitating the identification of similar songs based on their numerical attributes. Additionally, I contributed to the development of the Streamlit web application, creating an intuitive user interface with a sidebar for inputting song preferences and selecting the number of recommendations desired. Finally, I implemented the recommendation function, which leverages the cosine similarity scores to suggest tracks closely resembling the user's input. Overall, my involvement encompassed the entire lifecycle of the Music Recommendation System, from data preprocessing to web application development and recommendation functionality.

**Individual Contribution Report**

**Spotify Music Recommendation System**

**Swastik Paikaray**

**2105252**

**Abstract**: In the age of streaming services and on-demand music, music lovers are facing a paradox of plenty. While vast libraries offer seemingly endless options, navigating this abundance can be a daunting task. Imagine sifting through millions of songs, each vying for your attention, yet struggling to find something that truly resonates with your taste. This is where Music Recommendation Systems (MRS) emerge as a powerful tool, acting as a personalized guide through the sonic labyrinth.

**Individual Contribution**: In this project, I assumed the role of ensuring the successful implementation of the codebase, beginning with a thorough review of the provided code to grasp its structure and functionality. My focus encompassed addressing key areas such as data preprocessing, recommendation logic, and user interface design. This involved meticulously handling missing values and applying StandardScaler to scale numerical features accurately. Subsequently, I directed efforts towards implementing the recommendation logic, meticulously validating its accuracy in calculating cosine similarity between the input song and other tracks in the dataset through comprehensive testing across various input scenarios. Additionally, I dedicated attention to refining the user interface by integrating custom CSS styles, thereby enhancing its visual appeal and ensuring an intuitive, user-friendly experience.

