AI-Powered Music Recommender System With Collaborative Filtering

Runpeng Li
rli3@oxy.edu
Occidental College

1 Abstract

This paper serves as the proposal for my Occidental College Computer Science COMPS on developing a Music Recommendation System using Collaborative Filtering. The project explores the technical, evaluative, and ethical dimensions of building a system that can accurately suggest musical artists and tracks to users based on their historical preferences and listening setting. The proposal is structured into seven distinct sections: introduction, problem context, technical background, methods, evaluation, ethical considerations, and software documentation. It begins with an overview of the project's objectives and significance, following with the technical part of the work, last I end with the ethical concerns on this project. By dive into each of these sections, I plan the motivation for this work, ethical considerations, and how to produce this project.

2 Introduction

For my senior-year Occidental College Computer Science Comprehensive Project (COMPs), I will develop a Music Recommendation System (MRS) by integrating Collaborative and, researching with deep learning models for advanced feature extraction from music and weather data. The system will dynamically adapt recommendations based on real-time weather conditions and user feedback. This MRS goal is to provide more song options to music listeners and make listening music more enjoyable to users.

3 Problem Context

In the fast developing tech world, there is still challenges in crafting a personalized and contextually appropriate user experience. Current systems often overlook environmental factors such as weather, which can significantly influence a listener's mood and music preferences.

3.1 Personal experience

In my experience with digital music streaming, I've consistently bumped into an issue with the way Music Recommendation Systems (MRS) function. These platforms predominantly use algorithms rooted in past user interactions like song plays, ratings, and user-generated playlists. This approach, while functional on a basic level, fails to consider the dynamic and ever-changing landscape of a listener's life and mood. The result? A loop of recommendations that keeps repeating the back of my usual picks, which predominantly fall into the category of rap.

Now, as much as I love the hard-hitting lines and beats of rap, there are moments when it just doesn't fit the setting. Picture this: it's midnight, moonlight shines bright, and the rain is gently drizzling on the trees. At times like these, the soul wants for something a little softer, definitely not the aggressive bars of rap that are better matched for a sunny day by the beach or a workout session at the gym. This dissonance between what's being played and what's needed jars the environment, making the whole setting feel off.

Moreover, consider more intimate scenarios—like being on a date with someone special, in a setting that's carefully curated to be romantic and soft. The candles are lit, the atmosphere is just right, and then... boom, your playlist shuffles to a hardcore rap that talks about the struggles of street life and survival. Not exactly suitable for a romantic evening, right? It breaks the mood, pulling both of you out from a moment that should feel seamless and intuitively tuned to the vibes around you.

What I really need is a more dynamic playlist that not only respects my love for rap but also intelligently adapts to factors like the time of day and the weather, ensuring the music always complements the setting.

The problem is that most existing music recommendation systems lean heavily on static collaborative filtering techniques. While these methods are effective in mapping out general listening habits, they often overlook the nuanced changes in a listener's environment and mood. This oversight leads to suggestions that can feel out of place, diminishing overall user satisfaction and contributing to a lack of variety during streaming experience.

4 Technical Background

This section will focus on three primary aspects: enhancing traditional collaborative filtering (CF) with matrix factorization techniques to address the cold start problem and data sparsity, integrating context-aware capabilities to adapt music recommendations to the user's current environment and mood, and implementing a robust real-time data processing pipeline using Apache Kafka and Apache Flink. These elements are crucial for developing a dynamic MRS that not only understands user preferences but also responds intelligently to real-time changes in context, significantly enhancing the overall user experience.

4.1 Enhancing Collaborative Filtering with Matrix Factorization

My project will be developed using Python, a programming language favored for its powerful libraries and frameworks suited for data-driven applications. One of the core components of the system will be enhancing traditional collaborative filtering (CF). Collaborative filtering is a technique used in recommendation systems that predicts a user's preferences based on the preferences of other users with similar tastes. This method often encounters challenges with the cold start problem, where it struggles to make recommendations for new users or items that lack sufficient historical interaction data. Additionally, data sparsity, a condition where the majority of possible user-item interactions are unknown, can degrade the performance of CF systems. To address these issues, I will implement matrix factorization techniques. Matrix factorization is a mathematical tool that reduces a large matrix (in this case, user-item ratings) into a product of lower-dimensional matrices, revealing latent factors that represent underlying characteristics of both items and users. This approach, as demonstrated by Koren, Bell, and Volinsky [4], can uncover hidden patterns and relationships in user-item interactions that aren't immediately apparent, allowing for more accurate predictions of user preferences even with limited data.

4.2 Integrating Context-Aware Capabilities

To further personalize the music recommendations, my system will incorporate context-aware capabilities. These capabilities allow the system to adjust its operations based on dynamic information about its environment or the user's specific situation, such as time of day, location, or current activity. This approach is inspired by the research conducted by Rentfrow and Gosling in their study, The Do Re Mi's of Everyday Life [7]. They explored how different contexts such as emotional states, daily activities, and broader life situations significantly affect music preferences. Through a variety of psychological assessments and detailed music preference surveys, they demonstrated that individuals' music choices are closely linked to specific contexts and moods. This finding underscores the importance of integrating context-aware capabilities into the recommendation system to ensure that music suggestions not only reflect users' general tastes but also adapt to fit their current environmental and emotional circumstances, thereby enhancing user satisfaction and engagement. Therefore, when it comes to my project, I will follow a similar guideline for developing context-aware capabilities that dynamically adjust music recommendations based on real-time analysis of user context and environment.

4.3 Implementing Real-Time Data Processing with Apache Kafka

To manage the real-time data essential for adapting music recommendations based on contextual changes, I will implement a robust data pipeline using Apache Kafka. Kafka is a distributed streaming platform that excels at handling high volumes of data in real-time. It is designed to publish, subscribe to, store, and process streams of records in a fault-tolerant manner [6]. This capability is critical for my MRS, as it allows for the continuous intake and processing of live data streams—such as current weather conditions, time of day, or user interactions—which are integral to updating the music recommendations in real time. By integrating Kafka with Apache Flink, which excels at processing unbounded data streams quickly and reliably, the system can react to changes in context with minimal latency, ensuring that the music recommendations are always relevant and timely. Apache Flink is known for its ability to process streaming data at a large scale, providing precise control over time and state across a distributed system, which is essential for handling the complexities of real-time data feeds in context-aware systems.

5 Prior Work

5.1 Existing Music Recommendation Systems

While researching prior work to inform my project, I examined several key players in the music recommendation space. My initial focus was on Spotify's recommendation engine, which is known for its use of collaborative filtering combined with sophisticated audio analysis. According to Spotify's engineering blog [3], they leverage vast amounts of user data to create a personalized listening experience. While Spotify's system is robust, it lacks real-time context adaptability, primarily focusing on user interaction history and not the user's current environment or activities. This observation has led me to identify a critical area for improvement in my own project: enhancing the responsiveness of the system to real-time data.

Another significant system I looked into was Pandora's Music Genome Project. As described on their official website [2], Pandora categorizes songs based on hundreds of musical attributes but remains static in its recommendation approach without considering the listener's real-time context. This gap has inspired me to integrate adaptive context-aware features into my system, aiming to surpass the static nature of Pandora's recommendations.

5.2 Advanced Context-Aware and Real-Time Adaptive Systems

My research further led me to explore Moodagent. Moodagent's press kit [1] highlights how it uses mood sliders to adjust music playlists, which intrigued me because of the potential to expand this model by automatically detecting the user's mood through behavioral data.

Lastly, the technical backbone for managing real-time data in my project will be heavily inspired by Apache Kafka. As Narkhede, Neha and Shapira, Gwen and Palino, Todd detail in Kafka: The Definitive Guide [6], Kafka's capabilities to handle live data feeds efficiently will be crucial for my system. This will allow my music recommendation system not only to respond to historical preferences but also to adapt seamlessly to what the listener is experiencing at the moment—be it a change in weather, location, or activity. The details was mentioned in earlier sections.

6 Method

Although a detailed timeline is provided at the end of this paper, this section delves deeper into the approach I will be adopting for my project. Given the complexity and the dynamic nature of building a music recommendation system that adapts to real-time contextual data, I will adopt an approach that blends rigorous academic research with practical, hands-on system development.

6.1 Planning and Research

Over the summer, I will dedicate my efforts primarily to conducting thorough research on the existing technologies and methodologies used in music recommendation systems. Initially, I will explore the current landscape of such systems, focusing on their capabilities and limitations, especially in terms of handling real-time data and context-awareness. This phase is crucial because it will set the foundational knowledge needed to design a more responsive system.

Concurrently, I will investigate potential data sources that can be integrated into the system. This includes user behavioral data, environmental context like weather conditions through APIs, and possibly physiological data from wearable devices, assuming privacy and data protection laws are adhered to. The purpose of this dual-track research is to ensure that the system is not only technologically sound but also deeply integrated with the user's real-world context.

As I close the summer research phase, I will develop questionnaires and conduct preliminary surveys with potential users to gather data on their music preferences and the contextual factors that influence these preferences. These insights will guide the design of the recommendation algorithms that can adapt to changes in user context in real-time.

6.2 System Design and Implementation

With the onset of the fall semester, I will begin the active development phase of the project. This will start with the design of the system architecture, ensuring it can handle robust data processing and seamless user interactions. The system will be structured into three main layers:

- Data Layer: This layer will manage the ingestion and preprocessing of all incoming data. I will implement data normalization and encoding processes to prepare the data for analysis and ensure compatibility with machine learning models.
- Processing Layer: Here, I will develop the machine learning algorithms that will form the core of the recommendation system. Based on my summer research, I will experiment with both traditional models like collaborative filtering and more advanced models that incorporate neural networks to better handle the complexities of real-time, context-aware data.
- Presentation Layer: The user interface will be developed during this phase. It will be crucial to create an engaging and intuitive interface that allows users to interact easily with the system, providing feedback that can be used to further refine the recommendations.

6.3 Testing and Iteration

Once a functional prototype of the system is developed, I will begin a cycle of testing and iterations. This will involve:

- Unit Testing: To ensure each component of the system functions correctly independently.
- System Integration Testing: To verify that all system components work together as expected.
- User Testing: Initially, I will conduct informal testing with peers and colleagues to gather early feedback. This will be critical to understand how real users interact with the system and to identify any unforeseen issues or areas for improvement.

The feedback from these testing phases will be invaluable for refining the system. I plan to iterate on the design and functionality continuously, making incremental improvements based on user feedback and system performance.

7

As the end of the semester approaches, I will focus on consolidating all the work done. This will include final adjustments based on the testing phase,, ensuring the system is robust responsive, and user-friendly. And based on the testing phase, I will revise my comps paper accordingly. I will also begin compiling my findings and experiences to make the poster for presenting, detailing the development process, the technologies used, and the performance of the system against the initial objectives. Overall, I need to be on top of my schedule and take good notes. So I will have smooth process of constructing the paper, poster, and presentation with the experience I have with this project.

8 Evaluations

In order to measure the effectiveness and impact of the music recommendation system (MRS). I plan to evaluate it based on two key aspects: user satisfaction with the music recommendations and the system's responsiveness to changing contexts. These evaluations will help determine if the system meets the intended goals and will provide insights into areas for future improvement.

8.1 Music Recommendation Quality

The primary goal of this project is to enhance user experience by providing highly relevant and contextually appropriate music recommendations. To evaluate this aspect, I will employ a before-and-after questionnaire approach:

- Preliminary Questionnaire: Prior to using the system, participants will complete a questionnaire assessing their current satisfaction with existing music recommendation platforms and their specific preferences and pain points. This will establish a baseline for comparison.
- Post-Interaction Questionnaire: After interacting with the system for a period, the same participants will complete a similar questionnaire to evaluate changes in their satisfaction and perception of music recommendation quality. This will assess if the system has successfully tailored its suggestions to the users' tastes and situational contexts.

The questionnaires will be designed to ensure reliability, allowing for consistent responses across different times—excluding the change influenced by the interaction with the MRS. Key variables to control for will include music genre preferences, usual listening environments, and typical use cases (e.g., commuting, working out, studying).

8.2 System Responsiveness and Context Adaptation

Another crucial aspect of the MRS is its ability to adapt recommendations based on real-time contextual data. The evaluation of this component will involve:

- Context Simulation Tests: I will create various simulated environments to test how well the system adapts its music recommendations. For example, changing the simulated time of day or weather conditions and observing if the recommendations align with what would be contextually appropriate (e.g., more upbeat music on sunny days or calming music during late hours).
- User Feedback on Context Sensitivity: After using the system, users will provide feedback specifically about how appropriately the system responded to changes in their environment or situation. This will help identify any discrepancies between the system's context understanding and actual user expectations.

8.3 User Experience and Interface Usability

The overall user experience, including interface usability and interaction smoothness, will also be critically assessed:

- Usability Survey: Post-interaction, participants will complete a survey focusing on the usability of the interface. This will cover aspects like ease of use, aesthetic appeal, navigational intuitiveness, and any issues encountered during interactions.
- Focus Groups: Small focus group discussions may also be conducted to gain deeper insights into the user experience. These discussions can reveal subtle user sentiments and suggestions for enhancements that are not always captured in structured surveys.

8.4 Evaluation Participants

The participant group for these evaluations will primarily consist of college students (OXY's Studnets), who likely represent a diverse range of music preferences. This demographic will provide a robust dataset for assessing the MRS across different user types. Additionally, to ensure broader applicability, I will also invite staff and faculty members to participate, offering a wider perspective on the system's performance across different age groups and potentially different cultural backgrounds.

9 Ethical Consideration

While developing a music recommendation system (MRS) with the best intentions to enhance user experience and satisfaction, it is crucial to consider several ethical issues that may arise.

9.1 The Prevalence of Data Bias in Music Recommendation Systems

Data bias in AI-driven music recommendation systems significantly skews the musical landscape presented to users, favoring some genres and artists over others. This issue stems from several sources, including imperfect datasets that fail to represent the full spectrum of musical diversity. Milano et al. (2021) [5] extensively discuss the ramifications of such biases, noting how algorithmic favoritism towards popular genres such as pop and rock can marginalize less commercial styles like jazz or folk. They describe how initial training data, often derived from mainstream commercial tracks, leads to a feedback loop where popular music becomes more popular, further entrenching these biases. This not only perpetuates the popularity of already popular tracks but also sidelines niche genres that could potentially meet specific user preferences. Furthermore, Milano et al [5]. emphasize the subjective nature of musical taste, which complicates the issue further, as the algorithm's notion of what is 'good' or 'popular' might not resonate universally across diverse user bases. They argue that the challenge lies in crafting algorithms that acknowledge and adapt to the unique tastes of each listener, rather than conforming to the majority's preferences, which reinforces a feedback loop of popularity rather than true preference alignment. The authors provide an example of how such biases could lead to a homogenization of musical exposure, where new or independent artists find it increasingly difficult to break through the noise. This section of their work is a critical examination of how data bias in music recommendation systems not only affects the diversity of music that users are exposed to but also has broader implications for cultural representation and equity in the music industry [5].

10 Proposal Timeline

This timeline outlines a structured plan for the development and evaluation of the music recommendation system (MRS) leading up to the final comps paper due around 12/15. Each two-week period will focus on specific tasks critical to the project's progress.

- August - Early Semester Preparation
  - Wk 1-2: Finalize research on existing music recommendation systems and begin gathering initial data sources including APIs for real-time context data.
  - Wk 3-4: Develop initial user surveys and questionnaires to assess current user satisfaction with existing music recommendation platforms and to gather data on user preferences in various contexts.
- September - System Design and Initial Development
  - Wk 1-2: Set up data ingestion and preprocessing pipelines. Begin basic model development with collaborative filtering.
  - Wk 3-4: Continue with the integration of advanced machine learning models, focusing on incorporating real-time context awareness.
- October - Advanced Development and Initial Testing
  - Wk 1-2: Begin developing the frontend interface and integrating it with the backend system.
  - Wk 3-4: Conduct initial unit tests on individual components and start integration testing to ensure all parts of the system work together seamlessly.
- November - User Testing and System Refinement
  - Wk 1-2: Conduct informal user testing with peers to collect early feedback. Begin iterating on the system based on this feedback.
  - Wk 3-4: Expand testing to a broader audience for more comprehensive feedback. Start preparing the poster and make significant adjustments to the system based on user input.
- December - Final Adjustments and Project Consolidation
  - Wk 1-2: Finalize system adjustments. Finish compiling data and outcomes for the poster. Ensure all components of the MRS are fully integrated and functioning.
  - Wk 3-4: Conduct final user testing sessions. Complete the final comps paper, integrating all feedback and results from the tests. Finalize and submit the poster and paper.

References

[1] Adweek. *Moodagent for Android: Free App to Create Dynamic Song Playlists Using Mood Sliders*. 2024. URL: https://www.adweek.com/performance-marketing/moodagent-for-android-free-app-to-create-dynamic-song-playlists-using-mood-sliders/ (visited on 05/09/2024).

[2] Community, Pandora. *Artists & the Pandora Algorithm*. 2024. URL: https://community.pandora.com/t5/AMP-Support-Forum/Artists-the-Pandora-algorithm/td-p/138478 (visited on 05/09/2024).

[3] Engineering, Spotify. *How Spotify Uses ML to Create the Future of Personalization*. 2021. URL: https://engineering.atspotify.com/2021/12/how-spotify-uses-ml-to-create-the-future-of-personalization/ (visited on 05/09/2024).

[4] Koren, Yehuda, Bell, Robert, and Volinsky, Chris. "Matrix Factorization Techniques for Recommender Systems". In: *Computer* 42.8 (2009), pp. 30–37. DOI: 10.1109/MC.2009.263.

[5] Milano, Silvia, Taddeo, Mariarosaria, and Floridi, Luciano. "Ethical aspects of multi-stakeholder recommendation systems". In: *The Information Society* 37.1 (2021), pp. 35–45. DOI: 10.1080/01972243.2020.1832636.

[6] Narkhede, Neha, Shapira, Gwen, and Palino, Todd. *Kafka: The Definitive Guide Real-Time Data and Stream Processing at Scale*. 1st. O'Reilly Media, Inc., 2017. ISBN: 1491936169.

[7] Rentfrow, P. J. and Gosling, S. D. "The do re mi's of everyday life: the structure and personality correlates of music preferences." In: *Journal of Personality and Social Psychology* 84 (6 2003), pp. 1236–1256. DOI: 10.1037/0022-3514.84.6.1236.