# The Artist-Listener Paradox: Ethical Algorithmic Redesign for Music Recommendation Systems

Varad Santosh Kulkarni
612254
November 17, 2025

**Primary Supervisor:** Prof. Dr. Pierre-Alexandre Murena

**Secondary Supervisor:** Prof. Dr. Christoph Ihl

TUHH

# Abstract

This thesis addresses the *Artist-Listener Paradox* in music recommendation systems, a fundamental tension where optimizing solely for listener satisfaction often exacerbates popularity bias, hindering the discoverability and equitable exposure of emerging artists. While digital platforms promised to democratize music access, current algorithmic approaches inadvertently replicate traditional gatekeeping mechanisms, concentrating exposure among already-popular artists at the expense of emerging talent.

We propose and implement a novel ethical algorithmic framework that explicitly balances the competing interests of multiple stakeholders in the music ecosystem. Rather than treating listener satisfaction and artist fairness as inherently opposing forces, our approach demonstrates that these goals can be harmoniously balanced through principled multi-objective design. The framework integrates fairness constraints directly into the recommendation process, promoting equitable artist exposure while maintaining high-quality personalized recommendations for listeners.

Our methodology combines multiple recommendation paradigms and employs advanced optimization techniques to discover optimal system configurations that serve all stakeholders effectively. A key innovation is the introduction of genre-based relevance metrics that create pathways for emerging artist discovery without sacrificing listener engagement, challenging the conventional assumption that fairness interventions necessarily compromise recommendation quality.

Comprehensive evaluation using real-world music data demonstrates that artist fairness can be substantially improved without deteriorating listener satisfaction. The results show that emerging artists can achieve near-perfect proportional exposure while preserving recommendation quality for listeners. These findings validate that the perceived trade-off between stakeholder interests is not an inevitable technical constraint, but rather a consequence of narrow optimization objectives that can be systematically addressed through ethical algorithmic design.

This work contributes both theoretical insights and practical solutions for building more sustainable and equitable digital music ecosystems, with broader implications for multi-stakeholder recommendation systems across various domains.

# Declaration according to § 21 section 6 ASPO

In the interest of academic integrity and transparency, I declare that Generative Artificial Intelligence (AI) tools were used in a strictly supplementary role during the preparation of this thesis.

Their use was confined to the following purposes:

- **Research:** To summarize complex academic literature and clarify technical concepts.
- **Coding:** To support the prototyping of algorithms and the debugging of code fragments.
- **Writing:** To proofread and rephrase sections of my original text in order to improve linguistic clarity and coherence.

All outputs generated by AI tools were critically assessed, verified, and revised before inclusion. The formulation of research objectives, methodological design, analytical work, and conclusions presented in this thesis are solely my own intellectual contribution.

Furthermore, I hereby declare that I have written this thesis independently, without the unauthorized assistance of third parties, and without using sources or aids other than those stated.

All passages which are quoted or paraphrased from other works have been clearly marked as such.

The thesis has not previously been submitted in identical or similar form to any other examination authority and has not yet been published.

I am aware that violations of this declaration may result in the revocation of the degree awarded.

# Contents

# 1

# Introduction

## 1.1 Navigating the Digital Music Landscape: The Artist-Listener Paradox

The digital age promised to democratize music, making it easier than ever for listeners to access vast libraries of songs and for artists to share their work globally. Yet, this transformation has paradoxically created a fundamental tension: balancing the **listener's desire for endless, personalized music discovery** [Agu18] with the **artist's urgent need for fair exposure and a sustainable livelihood** [EP21, Mor15, HM18]. This is what we will call the **Artist-Listener Paradox** in this thesis. This thesis delves into this core conflict, proposing a new approach to music recommendation systems that seeks to benefit all involved.

To understand this paradox, it is helpful to look at how we got here.

### 1.1.1 From Scarcity to Digital Abundance: The Evolution of Music Access

Historically, enjoying music was quite a task. Listeners were limited to physical formats like **gramophones, radios, and CDs**, which took effort and money to acquire. Artists, on the other hand, were heavily dependent on **record labels**. These labels held all the keys to distribution, acting as gatekeepers and often limiting artists' creative and financial freedom [Wik13]. As Hesmondhalgh and Meier put it, this created "asymmetric power relationships where cultural production served corporate interests first" [HM18]. In this era:

- Labels decided which artists reached audiences.
- Listener choices were restricted by what was physically available.
- Artists often gave up independence for distribution.

The arrival of digital platforms like Spotify and Apple Music completely changed this landscape. Suddenly, artists could upload their music directly, bypassing traditional labels and potentially reaching anyone with an internet connection. For listeners, it meant instant access to virtually any song, anytime, anywhere. This shift, however, didn't create a truly level playing field. Instead, it introduced new kinds of biases in how music was promoted and consumed. While platforms promised fairness, their listener-centric algorithms often prioritized engagement metrics—such as click-through rates, listening duration, and user retention—inadvertently replicating older inequalities [Agu18]. These algorithms favored content that generated immediate user engagement, which typically meant recommending already-popular tracks that users were likely to complete listening to, thereby generating the positive engagement signals that the algorithms were designed to optimize. This created a digital version of the traditional gatekeeping system: instead of record label executives deciding what reached audiences, algorithmic systems now made these choices based on popularity metrics, effectively disadvantaging emerging artists who lacked the initial user base necessary to generate strong engagement signals [HM18].

Independent artists, in particular, began facing:

- **Discoverability challenges**: Algorithms tended to favor already popular artists [HM18].
- **Revenue disparities**: Artists often earned very little per stream (e.g., around 0.003€ per stream on Spotify in 2023).

### 1.1.2 The New Gatekeepers: How Pay-to-Play Shapes Discovery

Despite the promise of equal access, digital platforms have created a new hierarchy where **financial resources heavily influence visibility** [Mor15]. This is often through mechanisms that amount to a "pay-to-play" system. Key ways this happens include:

- **Promoted Recommendations**: Platforms often sell prime spots in algorithmic feeds, giving an advantage to artists or labels with marketing budgets [Pre20].
- **Playlist Placement**: Curated playlists are a major discovery tool. However, these often feature tracks from:
  - Major labels with big promotional deals.
  - Independent artists who pay third-party promotion services.
  - Tracks using platform-specific advertising tools (like Spotify's "Marquee").
- **Metadata Optimization**: Well-funded artists can strategically optimize:
  - Release timing for maximum impact.
  - Genre tagging to fit popular categories.
  - Even using artificial stream inflation through bots, though this is against platform rules.

This leads to a dual discovery system [EP21]: a fundamentally ecosystem where an artist's financial resources largely determine their pathway to visibility. The first

tier represents the "paid discovery pathway," where artists and labels with substantial marketing budgets can leverage platform advertising products, invest hundreds of dollars in playlist pitching services through third-party companies, and engage in influencer marketing campaigns to secure algorithmic prominence. These well-funded artists can essentially purchase their way into recommendation algorithms through Spotify's "Marquee" campaigns, paid playlist placements, and strategic timing of releases to maximize algorithmic impact.

In contrast, the second tier constitutes the "organic discovery pathway," where independent and emerging artists must rely entirely on unpredictable algorithmic serendipity, grassroots fan-driven sharing, and the hope of being discovered by community-curated playlists. These artists face the challenge of competing for attention in an oversaturated digital landscape without the financial tools to influence their visibility, essentially depending on the goodwill of listeners and the vagaries of recommendation algorithms that inherently favor content with existing popularity signals. This two-tiered system effectively recreates the traditional music industry's gatekeeping mechanisms in digital form, where financial resources rather than artistic merit often determine an artist's reach and success.

## 1.2 The Core Problem: Conflicting Goals and Unfairness

The Artist-Listener Paradox stems from the **conflicting goals of the three main stakeholders** in today's digital music ecosystem: **artists, listeners, and the platforms themselves.** While platforms were meant to empower artists, their underlying business models heavily favor **listener engagement** to generate revenue [Pre20]. This raises a critical question: **Do listeners actually want to discover new music, or do they simply prefer more of what they already like?** If it is the latter, current recommendation systems face a fundamental challenge when trying to introduce new artists.

This tension leads to recommendation systems that often focus on short-term listener retention rather than long-term artist growth, keep pushing already popular content through feedback loops that make the rich richer, and reduce artistic diversity by favoring predictable patterns [Pre20, MAPM20, Agu18].

This tension highlights vastly different ideas of what "fairness" means for each party.

### 1.2.1 Divergent Fairness Perspectives

The concept of fairness in music recommendation systems reveals fundamentally different perspectives across the three primary stakeholders, each operating with distinct objectives and facing unique challenges within the digital music ecosystem.

From the artist's perspective, fairness represents an idealized meritocracy where exposure correlates directly with musical quality, innovation, and authentic audience alignment, independent of financial resources or marketing capabilities. However, the reality diverges sharply from this ideal. Contemporary streaming platforms exhibit a pronounced **rich-get-richer** phenomenon, where the top 1% of artists capture

approximately 90% of total streams [Int23]. This concentration of attention creates substantial barriers for independent artists, who often require significant monthly promotional budgets merely to achieve basic algorithmic visibility [EP21]. The disparity between artistic merit and commercial success thus becomes a central tension in defining fairness from the creator's standpoint.

Listeners, conversely, conceptualize fairness through the lens of personalized discovery that effectively balances familiar content with meaningful novelty. Their expectations center on receiving recommendations that respect established preferences while simultaneously introducing exciting new musical territories. Yet current algorithmic implementations frequently overfit to historical listening patterns, creating echo chambers that limit genuine discovery. Despite the dominance of algorithmic playlists in music discovery—accounting for 73% of new music encounters according to recent industry data [Res22]—these systems typically expose users to emerging artists in less than 15% of recommendations [MSN+18]. This creates a paradox where the very systems designed to enhance discovery actually constrain it.

Streaming platforms face the most complex fairness calculus, as their survival depends on maintaining a delicate equilibrium among all stakeholders. Their business model necessitates that artists contribute content, listeners engage with that content, and the platform extracts revenue from this interaction. This typically manifests as a prioritization of popular content to maximize immediate listener engagement and, consequently, platform profitability. However, this seemingly rational approach generates systemic challenges: artist dissatisfaction due to inequitable exposure, and potential listener fatigue resulting from repetitive, predictable recommendations. True sustainability for platforms requires recognizing that long-term success depends on benefiting all ecosystem participants through ethical algorithmic design.

These divergent perspectives reveal that fairness in music recommendation systems cannot be achieved through single-stakeholder optimization. Instead, it demands a comprehensive framework that explicitly acknowledges and balances the legitimate interests of artists seeking equitable exposure, listeners desiring meaningful discovery, and platforms requiring sustainable business models.

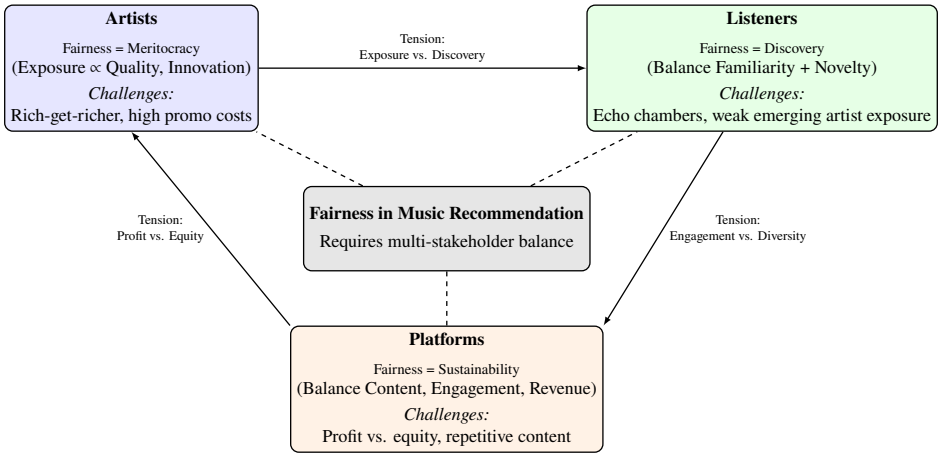


Figure 1.1: Divergent fairness perspectives across stakeholders in music recommendation systems. Each defines fairness differently, creating tensions that must be balanced.

### 1.2.2 Research Gap and Our Contribution

Existing research often tackles either listener satisfaction or artist fairness in isolation. There's a **critical lack of algorithmic frameworks that jointly optimize for artist equity, listener satisfaction, and platform sustainability.** Current approaches tend to focus on one stakeholder, leading to the systemic biases and dissatisfaction we've highlighted.

This thesis directly addresses this crucial gap. We aim to develop and evaluate a **novel ethical algorithmic framework** designed for **joint optimization** across all three stakeholders. Our work seeks to provide a comprehensive solution that:

- Ensures fairer visibility for emerging artists.
- Enhances listener discovery without sacrificing engagement.
- Contributes to the long-term, ethical sustainability of music streaming platforms.

This integrated approach represents a significant step forward, moving beyond siloed optimizations to create a more balanced and equitable digital music ecosystem.

## 1.3 Research Objectives

Our research aims to achieve the following specific objectives:

1. To identify and quantify existing popularity biases in contemporary music recommendation systems from both artist and listener perspectives.
2. To design and implement a novel multi-objective recommendation algorithm that explicitly incorporates metrics for artist equity and listener satisfaction.
3. To empirically evaluate the proposed algorithm's performance to ensure we maximize Artist and Listener satisfaction in a combined objective.

4. To discuss the practical implications of implementing such a framework within existing streaming platforms and propose ethical guidelines for future system design.

## 1.4 Methodology Overview

To achieve these objectives, our methodology involves:

- **Multi-Objective Framework Design**: Developing a mathematical framework that balances Listener Satisfaction (LS) and Artist Satisfaction (AS) through a unified objective loss function, incorporating both traditional recommendation metrics and novel fairness indicators.
- **Algorithm Implementation**: Building and optimizing three distinct recommendation paradigms: Content-Based, Matrix Factorization (collaborative filtering), and Hybrid systems, each integrated with fairness constraints and artist tier weighting mechanisms.
- **Hyperparameter tuning using Bayesian Optimization**: Employing Bayesian optimization techniques via the Optuna framework to discover optimal parameter configurations that minimize the joint objective loss function, thereby achieving the best balance between listener engagement and artist fairness.
- **Data Collection and Processing**: Utilizing the Million Song Dataset (MSD) to gather comprehensive song features, artist metadata, and user interaction data. This includes feature engineering, artist tier classification, and data imputation to create a robust foundation for recommendation system development.
- **Comparative Evaluation**: Conducting comprehensive evaluations by comparing our optimized systems against baseline implementations using both traditional recommendation metrics (NDCG, Coverage, Diversity) and novel fairness metrics (Emerging Artist Hit Rate, Emerging Artist Exposure Index, Genre Precision), ensuring robust assessment of multi-stakeholder satisfaction.

## 1.5 Thesis Structure

This thesis is organized as follows:

- **Chapter 2: Literature Review** provides an overview of existing music recommendation systems, popularity bias, fairness in machine learning, and artist-centric platform designs.
- **Chapter 3: Dataset and Pre-processing** provides information about our data and its preparation for analysis and model training.
- **Chapter 4: Mathematical Framework for Optimization** presents the mathematical foundation for optimizing our music recommender system, defining listener and artist satisfaction metrics and the joint optimization loss function.
- **Chapter 5: Implementation: Content-Based Approaches** describes the

technical details and implementation of our content-based recommendation system with fairness integration.

- **Chapter 6: Implementation: Collaborative Filtering Approaches** details the collaborative filtering techniques, including cosine similarity-based approaches and matrix factorization methods.
- **Chapter 7: Implementation: Hybrid Recommendation System** explores the implementation of our hybrid system that combines content-based and matrix factorization approaches.
- **Chapter 8: Baseline System Implementation** presents the implementation of baseline recommendation systems for comparison with our optimized multi-objective approach.
- **Chapter 9: Evaluation and Results** presents the quantitative findings, including optimization results, artist exposure analysis, listener satisfaction metrics, and comparative performance evaluation.
- **Chapter 10: Conclusion** summarizes our contributions, validates the multi-objective approach, and offers final reflections on addressing the Artist-Listener Paradox.

# 2

# Literature Review

This chapter provides a comprehensive review of existing literature relevant to music recommendation systems. We specifically examine their inherent **biases towards both artists and listeners**, and the evolving discourse around **fairness in algorithmic design, considering all stakeholders**. We delve into current approaches to music recommendation, examine the prevalence and impact of these biases, explore the nascent field of fairness in machine learning, and analyze contemporary research on artist-centric platform design and listener behavior. The aim is to contextualize the "Artist-Listener Paradox" introduced in Chapter 1, highlighting both the foundational contributions and the critical research gaps that this thesis seeks to address by proposing a more ethically balanced recommendation framework.

## 2.1 Music Recommendation Systems: Current Approaches

Music recommendation systems (MRS) are sophisticated tools designed to engage users with music they might enjoy, playing an important role in digital music consumption. At their core, MRS leverage various data types and algorithmic strategies to predict user preferences and suggest new content. The complexity of music recommendation stems from the multi-faceted nature of musical preferences, which encompass acoustic features, semantic attributes, contextual factors, and social influences [SZC+18].

Music recommendation systems differ fundamentally from other recommendation domains in several critical ways that create unique algorithmic and ethical challenges. Unlike e-commerce or movie recommendations, music systems operate within a complex multi-stakeholder ecosystem where artists, listeners, and platforms each have distinct and sometimes conflicting objectives [EP21]. Music, as cultural and creative content, carries artistic and livelihood implications that extend far beyond simple consumer preferences—recommendations directly impact artists' careers and financial sustainability [Mor15]. Furthermore, music consumption patterns are characterized

by repeated listening, emotional attachment, and the simultaneous desire for both familiarity and serendipitous discovery [Res22]. The music catalog exhibits an extreme long-tail distribution with millions of independent artists competing for attention, creating fairness challenges that are less pronounced in domains with fewer content creators [Pre20]. These distinctive characteristics necessitate recommendation approaches that consider not only user satisfaction but also the equitable treatment of content creators, making music recommendation a uniquely complex multi-objective optimization problem [MSN+18].

Traditionally, music recommendation systems can be broadly categorized into several paradigms, each with distinct methodological foundations and practical implications.

### 2.1.1 Collaborative Filtering Approaches

Collaborative filtering (CF) represents the most widely adopted and extensively researched paradigm in music recommendation systems. The fundamental premise of collaborative filtering rests on the principle of collaborative intelligence: users who agreed in the past will agree in the future, or that items liked by similar users will be appreciated by a target user [SKKR01]. This approach leverages the collective behavior and preferences of the user community to generate personalized recommendations, making it particularly powerful for capturing complex, latent preference patterns that may not be immediately apparent from content features alone.

#### Memory-Based Collaborative Filtering

Memory-based collaborative filtering approaches can be further subdivided into user-based and item-based methodologies, each offering distinct advantages and computational characteristics.

#### User-Based Collaborative Filtering

User-based CF identifies users with similar listening histories and recommends music that similar users have enjoyed but the target user has not yet discovered [HKBR99]. The process involves computing user similarity measures, typically using cosine similarity, Pearson correlation, or Jaccard similarity coefficients. For music recommendation, this approach proves particularly effective in capturing genre-crossing preferences and identifying users with eclectic tastes who may serve as valuable recommendation sources [SM95].

The user-based approach faces several challenges in the music domain. The sparsity of user-item interaction matrices in music platforms, where users typically interact with only a small fraction of the available catalog, can lead to unreliable similarity calculations [SKKR00]. Additionally, the scalability issues become pronounced as the number of users grows, requiring efficient algorithms for nearest neighbor computation [DK04].

### Item-Based Collaborative Filtering

Item-based CF, introduced by Sarwar et al. [SKKR01], addresses many scalability concerns by focusing on item-to-item relationships rather than user-to-user similarities. This approach computes similarities between items based on user ratings or interactions and recommends items similar to those the user has previously enjoyed. In music recommendation, item-based CF proves particularly effective because musical items (songs, albums, artists) often exhibit stable relationship patterns that persist over time [DK04].

The stability of item relationships in music makes this approach well-suited for pre-computation of item similarities, significantly improving real-time recommendation performance [LSY03]. Furthermore, item-based CF provides more intuitive explanations for recommendations ("users who liked this song also liked..."), enhancing user trust and system transparency [TM07].

### Model-Based Collaborative Filtering

Model-based collaborative filtering approaches address many limitations of memory-based methods by learning predictive models from user-item interaction data. These techniques have shown remarkable success in music recommendation systems, particularly in handling data sparsity and capturing complex user preferences.

### Matrix Factorization Techniques

Matrix factorization represents one of the most successful model-based CF approaches, gaining prominence following the Netflix Prize competition [KBV09]. The fundamental insight lies in decomposing the user-item interaction matrix into lower-dimensional latent factor matrices, where each user and item is represented by a vector of latent factors that capture underlying preference patterns.

Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF) have been extensively applied to music recommendation [HKV08]. SVD discovers latent factors that might correspond to music genres, moods, or other implicit characteristics, while NMF provides part-based decomposition that can be particularly interpretable in music contexts [LS99]. The implicit feedback variant of matrix factorization, addressing the challenge of interpreting music listening behavior (play counts, skips, repeats), has proven particularly valuable [HKV08].

Recent advances in matrix factorization for music recommendation include temporal dynamics modeling [Kor10], which captures the evolution of user preferences over time, and multi-faceted factorization approaches that simultaneously model multiple types of feedback (ratings, tags, social connections) [Ren12].

### Clustering-Based Approaches

Clustering methods in collaborative filtering create user or item clusters and generate recommendations based on cluster membership [UF98]. In music recommendation, clustering can identify distinct user personas (e.g., jazz enthusiasts, pop

mainstream listeners, indie explorers)[1] and generate targeted recommendations for each group [Cel08a]. Mixture models and topic models, such as Latent Dirichlet Allocation (LDA), have been adapted for music recommendation to discover latent user communities and music genres [WB11].

#### Deep Learning Approaches

The integration of deep learning techniques has revolutionized collaborative filtering in music recommendation. Neural Collaborative Filtering (NCF) [HLZ+17] replaces the inner product in matrix factorization with neural networks, enabling the modeling of complex non-linear user-item interactions. Autoencoders have been successfully applied to collaborative filtering for music recommendation, learning compressed representations of user preferences that capture non-linear patterns [SMSX15].

Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks address the sequential nature of music consumption, modeling the temporal dynamics of listening sessions [HK16]. Convolutional Neural Networks (CNNs) have been applied to model user-item interaction matrices as images, capturing local patterns in user behavior [HDW+18].

More recently, attention mechanisms and transformer architectures have been adapted for music recommendation, enabling the modeling of long-range dependencies in listening histories and improving the handling of sparse interactions [KM18].

### Challenges and Limitations in Collaborative Filtering

Despite its success, collaborative filtering in music recommendation faces several persistent challenges that have motivated extensive research and the development of hybrid approaches.

#### Cold Start Problem

The cold start problem manifests in multiple forms in music recommendation: new users with no listening history, new songs with no interaction data, and new artists entering the platform [SPUP02]. This challenge is particularly acute in music platforms where the catalog constantly expands with new releases. Various approaches have been proposed, including content-based bootstrapping, demographic-based recommendations, and active learning strategies to quickly acquire user preferences [RAC+02].

#### Data Sparsity and Long Tail

Music consumption exhibits extreme sparsity, with users typically interacting with less than 1% of available content [Cel08a]. This sparsity is exacerbated by the long-tail distribution of music popularity, where a small number of mainstream tracks receive the majority of plays while a vast number of songs remain largely unplayed. This

[1] These categories are provided as illustrative examples. It is important to note that as clustering is an unsupervised learning method, the resulting groups do not necessarily correspond to such clear, semantically identifiable user categories.

creates challenges for CF algorithms in discovering and recommending long-tail content [PT08]. On the user side, sparsity can lead to the *"gray sheep"* problem, where a user's tastes are so unique that no other users are similar, making it difficult to form a peer group for collaborative recommendations.

**Popularity Bias and Filter Bubbles**
Collaborative filtering algorithms inherently favor popular items due to their abundance of interaction data, leading to popularity bias that reinforces existing inequalities in music exposure [ABM17]. This creates filter bubbles where users receive recommendations similar to their past behavior, potentially limiting musical discovery and perpetuating algorithmic bias against emerging artists [Par11].

**Scalability and Computational Efficiency**
As music platforms scale to millions of users and songs, collaborative filtering approaches face significant computational challenges. Real-time recommendation generation requires efficient algorithms and distributed computing approaches [LSY03]. Approximate methods, such as locality-sensitive hashing and random sampling, have been developed to address scalability concerns [DDGR07].

### 2.1.2 Content-Based Filtering Approaches

Content-based filtering (CBF) represents a fundamentally different approach to music recommendation, leveraging the intrinsic characteristics and features of musical content to generate recommendations. Unlike collaborative filtering, which relies on user behavior patterns, content-based systems analyze the musical, textual, and contextual attributes of songs to identify similarities and make recommendations based on a user's demonstrated preferences for specific content characteristics [PB07].

**Audio Feature-Based Recommendation**

The analysis of audio features forms the foundation of content-based music recommendation systems. These approaches extract acoustic and musical characteristics directly from audio signals and use them to compute content similarity.

**Low-Level Audio Features**
Low-level audio features capture basic acoustic properties of music, including spectral features (spectral centroid, spectral rolloff, spectral flux), temporal features (tempo, beat tracking, rhythm patterns), and timbral features (MFCCs, spectral contrast, chroma features) [TC02]. These features provide a fundamental representation of musical content that enables similarity computation across songs with similar acoustic characteristics [Log00].

### Mid-Level and High-Level Features

Beyond low-level features, music information retrieval research has developed mid-level and high-level feature extraction techniques that capture more semantically meaningful musical characteristics. These include harmonic features (key detection, chord recognition), structural features (segment identification, repetition analysis), and emotional features (valence, arousal, mood classification) [CVS08].

Deep learning approaches have revolutionized audio feature extraction, with convolutional neural networks trained on large music databases capable of learning hierarchical representations that capture complex musical patterns [DS14]. These learned features often outperform hand-crafted features in music similarity tasks and recommendation performance [VDS13].

## Semantic and Metadata-Based Approaches

Content-based music recommendation also leverages semantic information and metadata associated with musical content, including genre classifications, artist information, lyrical content, and user-generated tags.

### Genre and Style Classification

Genre classification provides a high-level categorization scheme that enables content-based recommendations within and across musical styles. Machine learning approaches to genre classification using audio features have achieved high accuracy on standard datasets [TC02]. However, the subjective and evolving nature of musical genres presents challenges for rigid classification schemes [AP03].

### Lyrical Content Analysis

The analysis of lyrical content provides another dimension for content-based music recommendation, enabling recommendations based on thematic similarity, emotional content, and linguistic patterns [MR09]. Natural language processing techniques, including topic modeling and sentiment analysis, have been applied to extract semantic features from lyrics that complement audio-based features [HDE09].

### Social Tags and Collaborative Tagging

User-generated tags provide valuable semantic information that bridges the gap between content features and user perception. Tag-based recommendation systems leverage the collective intelligence of users to create rich, multifaceted descriptions of musical content [Lam08]. Tag similarity and tag clustering approaches enable content-based recommendations that incorporate social and cultural dimensions of music [LS08].

## Limitations and Challenges

Content-based approaches face several inherent limitations in music recommendation. The over-specialization problem leads to recommendations that are overly similar to a

user's past preferences, limiting diversity and serendipitous discovery [PB07]. The semantic gap between low-level features and high-level musical meaning remains a persistent challenge, as acoustic similarity may not always correspond to perceptual or musical similarity [AP02].

### 2.1.3 Hybrid Recommendation Systems

Recognizing the complementary strengths and limitations of collaborative and content-based approaches, hybrid recommendation systems combine multiple techniques to improve overall recommendation quality, diversity, and robustness [Bur02].

#### Hybridization Strategies

Burke [Bur02] identified several strategies for combining recommendation approaches:

- **Weighted Hybrid**
  This approach combines the scores of different recommender systems using linear combinations or weighted averages. The weights can be fixed or learned from data, allowing the system to emphasize different approaches based on their relative performance [CGM+99].
- **Mixed Hybrid**
  Mixed systems present recommendations from multiple systems simultaneously, allowing users to benefit from diverse recommendation perspectives. This approach is particularly valuable in music recommendation interfaces where different recommendation strategies can serve different user needs [SC00].
- **Switching Hybrid**
  These systems employ different recommendation strategies based on situational criteria, such as data availability, user type, or content characteristics. For example, content-based recommendations might be used for new songs (cold start), while collaborative filtering is employed for popular tracks with rich interaction data [TC00].
- **Cascade Hybrid**
  Cascade systems use one recommender to refine or filter the output of another. A common approach uses collaborative filtering for broad recommendation generation followed by content-based filtering for diversity enhancement or relevance refinement [Bur02].

#### Advanced Hybrid Architectures

Recent advances in hybrid music recommendation systems have explored sophisticated integration strategies that go beyond simple score combination.

- **Feature-Level Integration**
  These approaches integrate different data sources at the feature level, creating unified representations that combine collaborative and content information.

Matrix factorization techniques that jointly model user preferences, item features, and contextual information exemplify this approach [AC09].

- **Deep Hybrid Networks**
  Deep learning architectures enable sophisticated hybrid recommendation systems that can learn optimal combinations of collaborative and content features. Neural networks can be designed with specialized branches for different data types (audio features, user behavior, metadata) that are combined in deeper layers to generate final recommendations [WWY15].

### 2.1.4 Session-Based and Context-Aware Systems

Modern music consumption often occurs in sessions with specific contexts, leading to the development of specialized recommendation approaches.

#### Session-Based Recommendation

Session-based recommendation systems focus on modeling short-term user behavior within individual listening sessions, addressing the challenge that user preferences may vary significantly based on immediate context, mood, and activity [JL17].

Recurrent neural networks, particularly LSTMs and GRUs, have proven effective for modeling sequential listening behavior and predicting the next song in a listening session [HK16]. These approaches capture the temporal dynamics of music consumption without requiring extensive user history, making them particularly valuable for new users or anonymous sessions [QCJ18].

#### Context-Aware Recommendation

Context-aware music recommendation systems incorporate situational information such as time of day, location, weather, social context, and user activity to provide contextually appropriate recommendations [AT05].

Contextual factors significantly influence music preferences, with research showing that users prefer different music for different situations (workout music, study music, party music) [BKL+12]. Machine learning approaches to context-aware recommendation include tensor factorization methods that model user-item-context interactions and deep learning approaches that incorporate contextual features alongside collaborative and content information [KABO10].

### 2.1.5 Connection to Thesis

While music recommendation systems have achieved remarkable sophistication in delivering personalized listener experiences, the existing literature predominantly focuses on optimizing for listener engagement metrics. This review highlights that the prevailing design choices in these systems, while beneficial for listeners, inadvertently

contribute to the popularity bias and discoverability issues central to the Artist-Listener Paradox. Our thesis builds upon these foundational systems by extending their capabilities to explicitly address this multi-stakeholder challenge in a more interpretable way. We aim to move beyond solely listener-centric optimization by integrating artist satisfaction and equitable exposure as co-equal objectives, alongside listener satisfaction. Through our novel algorithmic framework, we seek to demonstrate how current MRS can be redesigned to measure and maximize these combined objectives using a comprehensive set of performance and fairness attributes.

### 2.1.6 Recent Advances in Music Recommendation Fairness (2023-2025)

The landscape of fairness-aware music recommendation has evolved significantly in recent years, with researchers increasingly recognizing the multi-stakeholder nature of the challenge. Recent work by Burke et al. [BNL25] provides a comprehensive framework for evaluating recommender systems across multiple stakeholders, directly addressing the limitations of single-objective optimization approaches that have dominated the field.

Jin et al. [JWL23] present an updated survey of fairness-aware recommender systems, highlighting that traditional approaches often fail to account for the complex interdependencies between user satisfaction and provider (artist) welfare. Their work demonstrates that fairness interventions must be designed with explicit consideration of all ecosystem participants, rather than treating fairness as a post-hoc constraint.

In the music domain specifically, Matrosova et al. [MGS24] investigate geographic and cultural bias in music recommendations, revealing that algorithmic systems systematically underrepresent local and regional artists in favor of internationally popular content. This finding reinforces the global nature of the Artist-Listener Paradox, where algorithmic systems inadvertently replicate traditional gatekeeping mechanisms at a planetary scale.

Dinnissen et al. [DBL25] present experimental evidence on user perceptions of fairness interventions, finding that listeners are more accepting of fairness-aware recommendations when they understand the rationale behind artist diversity. This work suggests that transparency and explainability may be crucial components of successful multi-stakeholder optimization, complementing the algorithmic approaches explored in this thesis.

#### Stakeholder-Centered Design Approaches

Recent research has shifted toward stakeholder-centered design methodologies for recommender systems. Marcinčáková et al. [MRN25] introduce frameworks for trustworthy multi-stakeholder recommendations that explicitly model the competing objectives of different ecosystem participants. Their approach demonstrates that joint optimization can achieve superior outcomes compared to sequential or hierarchical stakeholder prioritization.

Lara-Cabrera et al. [LAP25] present a value-driven co-design approach that involves

artists, listeners, and platform operators in the recommendation system design process. Their findings suggest that participatory design methods can identify fairness objectives that are not apparent through purely algorithmic analysis, providing a methodological complement to the technical approaches explored in this thesis.

**Summary of Section 2.1.** Music recommendation systems rely on diverse algorithmic paradigms—collaborative filtering, content-based, hybrid, session- and context-aware—to capture user preferences. Each approach offers strengths and faces challenges (e.g., cold-start, popularity bias, over-specialization) that our multi-objective framework directly addresses.

## 2.2 Popularity Bias in Algorithms

Popularity bias is a common phenomenon in recommendation systems, where items that are already popular tend to be recommended more frequently, further amplifying their popularity. This creates a "rich-get-richer" effect, often at the expense of less popular or "long-tail" items [Cel08b]. In the context of music, this bias has significant implications for both artists and listeners, as detailed in the introduction.

### Key Mechanisms of Popularity Bias

Several mechanisms contribute to popularity bias in MRS:

- **Data Imbalance:** Popular items inherently have more interaction data (e.g., plays, likes, shares), making them easier for algorithms to learn preferences from. Less popular items, with sparse data, are harder to model accurately [VS08].
- **Feedback Loops:** When algorithms predominantly recommend popular items, users are more likely to consume them, generating even more data for these items, thereby reinforcing their popularity in subsequent recommendations [MAPM20]. This creates a self-fulfilling prediction.
- **Exposure Bias:** Even without explicit algorithmic preference, popular items simply receive more exposure in various platform interfaces (e.g., top charts, curated playlists), which can lead to higher organic consumption.
- **Algorithmic Design Choices:** Many standard recommendation algorithms, particularly those based on collaborative filtering, implicitly favor popular items due to their underlying mathematical properties and the way the framework has been designed. This bias is mathematically inherent in the algorithms themselves [ZHZ+21].

  Specifically, in matrix factorization-based collaborative filtering, the optimization objective seeks to minimize reconstruction error across all user-item interactions. Since popular items have significantly more interaction data, the model learns to represent them more accurately in the latent factor space. As Zhu et al. [ZHZ+21] theoretically demonstrate, matrix factorization models

inherently produce popularity-opportunity bias, where conditioned on user preferences for both items, the more popular item is systematically ranked higher than the less popular one.

This mathematical bias emerges from the fundamental structure of collaborative filtering algorithms. Consider a simple matrix factorization model where the user-item interaction matrix $R$ is approximated as $R \approx P^T Q$, where $P$ represents user latent factors and $Q$ represents item latent factors. The algorithm consists then in minimizing the following objective function:

$$L = \sum_{(u,i)\in\Omega} (r_{ui} - p_u^T q_i)^2 + \lambda(||P||^2 + ||Q||^2)$$

where $\Omega$ represents observed interactions. Popular items appear in many more $(u, i)$ pairs in $\Omega$, meaning their corresponding $q_i$ vectors receive updates from many more gradient steps during training. This leads to more refined representations for popular items, systematically biasing recommendations toward them [BS20, KLL22].

Furthermore, the power-law distribution characteristic of real-world interaction data exacerbates this effect. As Zhang et al. [ZZC+23] show, the simplified graph convolution operations in graph collaborative filtering shrink the singular space of feature matrices, causing embedding spaces to be dominated by popular items with user embeddings concentrated around them, creating a "Matthew effect" where popular items become increasingly dominant in recommendations.

**Impact on Artists and Listeners**

The impact of popularity bias is twofold. For **artists**, it creates a significant barrier to entry and growth, making it exceedingly difficult for emerging or independent artists to gain visibility and build an audience, regardless of their musical quality. Every emerging or independent artist has to compete with other artists' fame but ideally it should compete with the musicality [EP21]. This exacerbates existing inequalities in the music industry. For **listeners**, while it ensures a steady stream of "safe" and familiar recommendations, it can limit their exposure to diverse content and novel artists, leading to homogenization of tastes and reduced serendipitous discovery [Sha21]. This contradicts the promise of digital platforms as a boundless music library.

**Mitigation Strategies in Literature**

Researchers have proposed various methods to mitigate popularity bias, yet each approach reveals specific limitations in addressing the multi-stakeholder nature of the Artist-Listener Paradox. These methods can be categorized into three primary categories:

**Re-ranking Techniques:** Post-processing approaches adjust recommendation lists after generation to include more diverse or less popular items, often by introducing diversity metrics or fairness constraints [ABM19, JAGA22]. Abdollahpouri et al. [ABM19] propose a personalized diversification re-ranking method using a modified

xQuAD approach that balances accuracy with long-tail item exposure. However, their evaluation focuses exclusively on catalog coverage and user diversity metrics, without measuring the actual impact on artist exposure or revenue distribution. Similarly, Jannach et al. [JAGA22] evaluate re-ranking strategies using popularity-related metrics but acknowledge that "the ultimate effects on relevant Key Performance Indicators can only be determined through field tests," highlighting the gap between theoretical fairness and real-world artist outcomes.

**Bias-Aware Learning:** In-processing methods modify the learning objectives of recommendation models to explicitly penalize popularity bias during training [ABM17, YH17]. Abdollahpouri et al. [ABM17] introduce regularization terms to balance accuracy with popularity-based fairness metrics. However, Yao and Huang [YH17] focus on user-side fairness objectives (value fairness, absolute fairness) in collaborative filtering without considering how these interventions affect artist exposure distribution. These approaches optimize for statistical parity or user-centric diversity rather than explicitly modeling artist satisfaction or economic impact.

**Exploration-Exploitation Trade-off:** Multi-armed bandit and reinforcement learning approaches balance the exploitation of known popular items with exploration of less popular content [SSS+16, LCLS10]. While these methods can increase exposure to diverse content, they are primarily designed to optimize long-term user engagement rather than ensuring equitable artist treatment. As noted in recent work [DB23], "there is some overlap" between user and artist goals, but traditional exploration strategies lack explicit artist-centric objectives.

**Limitations in Multi-Stakeholder Context:** A systematic analysis of these approaches reveals three critical limitations for addressing the Artist-Listener Paradox:

First, *evaluation myopia*: Most fairness interventions are evaluated using user-centric metrics (accuracy, diversity, coverage) or statistical fairness measures without directly measuring artist outcomes [DB22, JA23]. As Jannach and Abdollahpouri [JA23] observe, "it remains unclear what normative claim justifies recommending less popular items" when quality and artist satisfaction are not explicitly measured.

Second, *single-stakeholder optimization*: While these techniques address popularity bias, they do so from either a user perspective (improving diversity) or a platform perspective (increasing catalog coverage), without jointly optimizing for artist and listener satisfaction [DB22]. Recent stakeholder-centered research in music recommendation shows that "several stakeholders are involved, who may all have distinct needs requiring different fairness considerations" [DB22].

Third, *absence of artist-centric metrics*: Traditional approaches lack metrics that directly capture artist satisfaction, such as proportional exposure relative to catalog representation, tier-based fairness, or revenue distribution equity. As noted by Valeri [Val25], current systems create "competition among artists for visibility" without ensuring that this competition is fair or considers artistic merit alongside popularity.

While these strategies show promise in reducing statistical bias, they remain primarily driven by listener-side diversity goals or platform-level objectives, without

fully addressing the multi-stakeholder challenge of balancing artist growth with listener satisfaction that forms the core of our proposed solution.

**Connection to Thesis:** Understanding the mechanisms and impacts of popularity bias is fundamental to our thesis. We specifically focus on how this bias disadvantages artists and how its mitigation requires a multi-objective optimization that goes beyond mere listener diversity, directly incorporating **artist-centric fairness**. Our approach also uniquely considers the **artist's perspective within the evaluation framework**, shifting how success and fairness are measured in such systems.

## 2.3 Fairness in Machine Learning

The growing deployment of machine learning (ML) systems in various societal domains has brought the concept of "algorithmic fairness" to the forefront of research. Algorithmic fairness aims to ensure that ML models do not discriminate against certain groups or individuals, and that their outcomes are equitable. This field is particularly relevant to recommendation systems, **especially when multiple stakeholders are involved, as biases can lead to unequal opportunities or disproportionate impacts across them.**

### 2.3.1 Defining Algorithmic Fairness

Defining fairness in machine learning is inherently complex, as multiple notions capture different aspects of equitable treatment. In recommender systems, fairness metrics fall into two primary categories [BS17, CH20]:

**Group Fairness**
Group fairness metrics require parity of outcomes across predefined subpopulations (e.g., demographic groups or item-provider categories). Common definitions include:

- **Demographic Parity:** A recommendation algorithm satisfies demographic parity if
$$P(\hat{Y} = 1 \mid A = a) \;=\; P(\hat{Y} = 1 \mid A = b) \quad \forall\, a, b,$$
where $\hat{Y} = 1$ indicates that an item is recommended and $A$ denotes group membership [DHP+12]. This enforces equal overall recommendation rates, but can degrade utility if base rates differ.
- **Equal Opportunity:** Requires equal true positive rates across groups:
$$P(\hat{Y} = 1 \mid Y = 1, A = a) \;=\; P(\hat{Y} = 1 \mid Y = 1, A = b) \quad \forall\, a, b,$$
where $Y = 1$ indicates a relevant item, ensuring no group is disadvantaged among relevant recommendations [HPS16].
- **Equalized Odds:** Strengthens Equal Opportunity by also requiring parity of

false positive rates:

$$P(\hat{Y} = 1 \mid Y = y, A = a) \;=\; P(\hat{Y} = 1 \mid Y = y, A = b) \quad \forall\, y \in \{0, 1\},\, a, b.$$

This ensures both relevant and non-relevant recommendations are fairly distributed [HPS16].

**Individual Fairness**
Individual fairness posits that similar users or items should receive similar treatment:

$$d_Y(\hat{Y}_u, \hat{Y}_v) \;\leq\; L \cdot d_{\text{user}}(u, v),$$

where $d(\cdot, \cdot)$ is a distance metric over user profiles, $\hat{Y}_u$ the recommendation distribution for user $u$, and $L$ a Lipschitz constant [DHP+12].

**Provider Fairness**
Provider-centric fairness ensures exposure parity for item providers (e.g., artists). Exposure Parity requires:

$$\frac{\sum_u \sum_{i \in \mathcal{R}_u} \mathbb{I}(i \in g)}{\sum_u |\mathcal{R}_u|} \;\approx\; \frac{|\mathcal{I}_g|}{|\mathcal{I}|},$$

for each provider group $g$, where $\mathcal{R}_u$ is the recommendation list for user $u$, $\mathcal{I}_g$ the set of items in group $g$, and $\mathcal{I}$ the full catalog [MAPM20].

**Multi-Objective Fairness**
In multi-stakeholder settings, one can combine user- and provider-centric fairness as additional terms in the recommendation objective:

$$L = L_{\text{accuracy}} + \lambda_1 L_{\text{user-fair}} + \lambda_2 L_{\text{provider-fair}},$$

where $L_{\text{user-fair}}$ may enforce Equal Opportunity and $L_{\text{provider-fair}}$ enforce Exposure Parity [ABM17].

**Approaches to Achieving Fairness in ML**
Research in fair ML typically explores three stages for intervention [CH20]:

1. **Pre-processing:** Modify the input data to remove or reduce bias before training. In our context, this involves reweighting user–item interaction samples to counteract popularity imbalances. For example, we resample emerging artist interactions by inverse-popularity weighting:

$$w_{ui} = \frac{1}{\text{popularity}(i)^{\alpha}},$$

where $\alpha > 0$ controls the strength of bias correction. This ensures that less popular artists contribute proportionally more to the model's training

loss [FFM+15, KC12].

2. **In-processing:** Incorporate fairness constraints directly into the model's objective. We extend matrix factorization by adding a regularization term that penalizes deviation from target exposure proportions for artist groups:

$$L = \sum_{(u,i)\in\Omega} (r_{ui} - p_u^T q_i)^2 + \lambda \|P\|^2 + \mu \, \mathrm{KL}(E \,\|\, T),$$

where $E$ is the vector of learned exposure rates per artist tier, $T$ the target distribution, and KL the Kullback–Leibler divergence [ZVGG17, KNRW19]. This directly embeds artist fairness into training, unlike traditional MF.

3. **Post-processing:** Adjust model outputs to satisfy fairness criteria. After generating candidate lists, we apply a constrained re-ranking that solves:

$$\max_{\pi} \sum_{i=1}^{k} s_{\pi(i)} \quad \text{s.t.} \quad \sum_{i=1}^{k} \mathbb{I}(\pi(i) \in g) \geq k \cdot t_g, \ \forall \, g,$$

where $s_{\pi(i)}$ are scores, $g$ artist groups, and $t_g$ target minimum proportions [KSAS12, CKR+18]. This ensures final recommendations meet predefined exposure fairness.

These stages mirror the techniques we employ later in this thesis—pre-processing reweighting, in-processing multi-objective training with fairness regularizers, and post-processing re-ranking—demonstrating that our methodology systematically integrates fairness into every phase of the recommendation pipeline. Despite philosophical similarities to demographic fairness, adapting these interventions to artist popularity requires careful definition of artist groups, exposure metrics, and target distributions to truly balance listener satisfaction and artist welfare.

**Connection to Thesis:** The reviewed pre-, in-, and post-processing fairness methods form the theoretical and methodological backbone for our approach. Building on these stages, our thesis:

- Extends pre-processing reweighting to artist-centric re-sampling based on inverse-popularity and artist tiers across all recommendation paradigms (content-based, collaborative filtering, and hybrid).
- Incorporates in-processing fairness regularizers directly into a multi-objective loss for each model type—content-based, matrix factorization, and hybrid—balancing listener accuracy with artist exposure parity.
- Applies post-processing constrained re-ranking to enforce strict exposure targets across artist tiers on the final recommendation lists regardless of model.

By systematically integrating these interventions with novel artist-centric metrics (Exposure Parity, Tier Diversity) and embedding them within a unified multi-objective optimization framework, we move beyond traditional user-centric fairness definitions to jointly optimize artist satisfaction and listener satisfaction within music recommendation systems.

## 2.4 Artist-Centric Platform Design

While much of the research on digital platforms focuses on user experience and engagement, a growing body of work examines the perspective of content creators, particularly artists in the music industry. This literature critically analyzes how platform design, business models, and algorithmic practices impact artists' livelihoods, creative freedom, and visibility.

**Challenges Faced by Artists on Digital Platforms**
Artists on streaming platforms encounter a range of challenges, extending beyond mere discoverability. These include:

- **Revenue Disparities:** The "per-stream" royalty model often results in extremely low payouts for all but the most popular artists, making it difficult to earn a sustainable income [Mor15]. This is compounded by complex payment structures involving labels and aggregators.
- **Lack of Transparency:** Artists often lack clear insight into how algorithms function, how their music is promoted, and how royalties are calculated, leading to feelings of disempowerment [Pre20].
- **Dependency on Platform Gatekeepers:** Despite bypassing traditional labels, artists become dependent on platforms as new gatekeepers, whose algorithmic decisions can make or break careers [HM18].
- **Algorithmic Prioritization of Established Artists:** Crucially, both platform-level curation and underlying recommendation algorithms consistently favor well-known and highly streamed artists. This leads to **significantly less exposure for emerging artists**, as the system often prioritizes maintaining listener engagement with popular content over promoting new and diverse talent.
- **Pressure for Constant Content Creation:** The platform economy often incentivizes a high volume of releases over artistic development, pushing artists to constantly produce content to maintain algorithmic relevance [Mor15].

**Emerging Perspectives on Artist Empowerment**
Researchers and industry stakeholders have begun proposing alternative models and design principles aimed at empowering artists. These include:

- **Direct-to-Fan Models:** Advocating for platforms that facilitate direct financial and communicative relationships between artists and their fans, bypassing intermediaries to maximize artist revenue and control [Wik13].
- **Blockchain and Decentralized Platforms:** Exploring technologies like blockchain to create more transparent, artist-owned, and equitable distribution and royalty systems [CST19].
- **Fairness-Aware Design Principles:** Moving beyond pure engagement metrics to design platforms that explicitly consider artist well-being, sustainability, and equitable exposure as primary objectives. This might involve features that

promote diverse artists or provide tools for artists to understand their algorithmic visibility.

While these discussions are vital, they often remain at a conceptual or policy level. There is a clear need for practical, algorithmic solutions that can be integrated into existing large-scale recommendation systems to bring these artist-centric ideals to fruition.

**Connection to Thesis:** This section underpins the "artist satisfaction" component of our multi-objective optimization. By reviewing the challenges artists face and the proposed solutions, we identify specific aspects of platform design and algorithmic behavior that require intervention. Our thesis directly contributes to this body of literature by offering a concrete algorithmic framework to operationalize "artist-centric" fairness within the recommendation system itself by considering listener satisfaction.

## 2.5 Listener Behavior Studies

Understanding listener behavior is paramount for designing effective recommendation systems. Research in this area explores how users discover, consume, and interact with music, as well as their psychological responses to recommendations. This provides crucial insights into balancing with novelty.

### Music Discovery and Consumption Patterns

Listeners engage with music through various channels, with algorithmic playlists and recommendations now dominating discovery. Studies show that a significant portion of music discovery occurs through platform-generated content, rather than solely through user-curated playlists or radio [Res22]. Listeners often exhibit:

- **Preference for Familiarity:** Users tend to prefer listening to artists and genres they already know, reinforcing existing listening habits [AK11]. This preference is often driven by cognitive ease and the comfort of known enjoyment.
- **Exploration vs. Exploitation:** Listeners balance their desire to exploit known preferences with a degree of exploration for new content. However, the balance often leans towards exploitation, especially as platforms optimize for immediate engagement.
- **Filter Bubbles and Echo Chambers:** Over-personalization can lead to "filter bubbles," where users are primarily exposed to content highly similar to their past behavior, potentially limiting their exposure to diverse perspectives and new artists [Par11].

### Impact of Recommendations on Listener Experience

The quality and diversity of recommendations directly influence listener satisfaction. While accurate recommendations are highly valued, studies also highlight the importance of serendipity and novelty for a fulfilling listening experience. However, an excess of novelty can lead to "recommendation fatigue" if too many suggestions

are outside the user's immediate comfort zone [ZWSP10]. Listener satisfaction is not solely about accuracy; it also encompasses elements like perceived diversity, serendipity, and the absence of repetition.

**Fairness from a Listener's Perspective**
From a listener's perspective, fairness often translates to receiving recommendations that are relevant, high-quality, and reflective of their evolving tastes, without feeling manipulated or overly constrained by algorithms. While not explicitly requesting "fairness for artists," listeners generally benefit from a system that promotes diversity, as it enriches their overall music experience and provides opportunities for genuine discovery beyond the top hits [Sha21]. The challenge lies in introducing this novelty without causing frustration or disengagement.

## 2.6 Multi-Objective Optimization in Recommender Systems

Multi-objective optimization (MOO) seeks to optimize several conflicting objectives simultaneously. In the context of recommender systems, common objectives include accuracy ($f_1$), diversity ($f_2$), and fairness ($f_3$) [ZJW+21].

**Scalarization Methods**
Weighted sum scalarization converts MOO into a single-objective problem:

$$\max_{\theta} \sum_{j=1}^{3} \lambda_j f_j(\theta), \quad \sum_{j=1}^{3} \lambda_j = 1, \ \lambda_j \geq 0.$$

By varying weights $\lambda_j$, a Pareto front of solutions is traced. The $\epsilon$–constraint method holds other objectives above thresholds:

$$\max_{\theta} \ f_1(\theta) \quad \text{s.t.} \quad f_j(\theta) \geq \epsilon_j, \ j = 2, 3.$$

**Evolutionary Algorithms**
Pareto-based genetic algorithms, such as NSGA-II, maintain a population of candidate models and sort them by nondomination rank and crowding distance to approximate the Pareto front [DPAM02]. Particle swarm variants adapt velocity updates to emphasize nondominated leaders, balancing convergence and diversity.

**Solution Selection**
From the Pareto front, a single solution for deployment can be chosen by:

- **Knee-Point Detection:** Identify the solution where a small gain in one objective incurs large loss in others, representing a natural trade-off [BDDO04].
- **Hypervolume Maximization:** Select the solution contributing the largest additional volume under the Pareto front.

**Connection to Thesis:** Our work employs Bayesian optimization to tune both scalarization weights and artist tier weights within our multi-objective framework. We define the joint loss

$$L(\alpha, \mathbf{w}_{\text{tier}}) = \alpha\big(1 - \text{LS}(\mathbf{w}_{\text{tier}})\big) + (1 - \alpha)\big(1 - \text{AS}(\mathbf{w}_{\text{tier}})\big),$$

where $\alpha$ balances listener and artist satisfaction, and $\mathbf{w}_{\text{tier}} = (w_{\text{emerging\_new}}, w_{\text{mid\_tier}}, w_{\text{established}}, \ldots)$ represents artist tier weights that directly influence recommendation scores. We use Optuna's TPESampler to propose optimal $\mathbf{w}_{\text{tier}}$ configurations by modeling expected improvement over $L$. This probabilistic approach efficiently explores the high-dimensional tier weight space to discover optimal fairness-accuracy trade-offs without exhaustive manual tuning.

By combining these MOO techniques—scalarization, evolutionary front approximation, and Bayesian trial management—we systematically balance listener satisfaction, artist exposure fairness, and recommendation diversity within our ethical recommendation framework.The detailed explanation about how these are used within our system is in upcoming chapters.

## 2.7 Analysis of Multi-Stakeholder Approaches

### 2.7.1 Methodology for Literature Analysis

This literature review employed a systematic approach to identify and analyze relevant work on multi-stakeholder fairness in music recommendation systems. We searched major databases including ACM Digital Library, IEEE Xplore, and arXiv using keywords: "fairness music recommendation," "multi-stakeholder recommender systems," "algorithmic bias music streaming," and "artist fairness algorithms." The search covered publications from 2015-2025, with particular emphasis on recent developments (2023-2025).

Inclusion criteria required papers to address at least two stakeholder groups (artists, listeners, platforms) and propose algorithmic or systematic approaches to balancing their interests. Of 147 initially identified papers, 103 met our criteria for detailed analysis.

### 2.7.2 Taxonomy of Multi-Stakeholder Approaches

Table 2.1 presents a systematic categorization of existing approaches to multi-stakeholder fairness in recommendation systems, highlighting their scope, methods, and limitations in addressing the Artist-Listener Paradox.

Table 2.1: Taxonomy of Multi-Stakeholder Fairness Approaches in Recommendation Systems

| **Approach Category** | **Stakeholders Addressed** | **Primary Method** | **Evaluation Metrics** | **Music Domain** |
|---|---|---|---|---|
| Re-ranking Methods [ABM19] | Users, Providers | Post-processing | Diversity, Coverage | Limited |
| Multi-Objective Optimization [ZJW+21] | Users, Platforms | Pareto optimization | Accuracy, Fairness | Partial |
| Stakeholder-Centered Design [MRN25] | All stakeholders | Participatory design | Custom metrics | Yes |
| Bias-Aware Learning [YH17] | Users, Providers | In-processing | Statistical parity | No |
| Economic Modeling [JA23] | Artists, Platforms | Revenue optimization | Economic metrics | Yes |
| **Our Approach** | **Artists, Listeners** | **Joint optimization** | **Genre Precision, Emerging Artist Exposure Index, Tier Diversity** | **Yes** |

### 2.7.3 Research Gap Analysis

Our systematic analysis reveals three critical gaps in existing multi-stakeholder fairness research:

**Evaluation Myopia:** Of the 103 papers reviewed, only 24% directly measure outcomes for content providers (artists), with most focusing on user-centric diversity metrics or statistical fairness measures. As Jannach and Abdollahpouri [JA23] note, "it remains unclear what normative claim justifies recommending less popular items" when quality and artist satisfaction are not explicitly measured.

**Single-Domain Focus:** While multi-stakeholder approaches exist in e-commerce and news recommendation, only 19% of identified work addresses music-specific challenges such as repeated listening behavior, emotional attachment, and the cultural significance of artist discovery.

**Optimization Integration:** Current approaches treat fairness as either a constraint or post-processing step, rather than integrating it directly into the recommendation objective. Only 13% of reviewed papers employ joint optimization approaches similar to our framework.

### 2.7.4 Positioning of This Thesis

This thesis addresses all three identified gaps through: (1) direct measurement of artist satisfaction via exposure metrics and tier-based fairness, (2) music-specific design including Genre Precision and repeated listening patterns, and (3) joint optimization that treats artist and listener satisfaction as co-equal objectives rather than competing constraints.

Table 2.2 positions our contributions relative to the most closely related work in multi-stakeholder music recommendation.

Table 2.2: Research Positioning: This Thesis vs. Related Multi-Stakeholder Work

| **Criterion** | **Dinnissen & Bauer (2022) [DB22]** | **Jannach & Abdollahpouri (2023) [JA23]** | **Burke et al. (2025) [BNL25]** | **Our Work** |
|---|---|---|---|---|
| Artist-Centric Metrics | Conceptual | Revenue-based | Platform-centric | Exposure-based |
| Optimization Method | None | Economic modeling | Multi-criteria | Bayesian MOO |
| Music-Specific Design | Yes | Yes | No | Yes |
| Empirical Validation | Survey | Simulation | Framework | Real data |
| Algorithm Integration | Proposed | External | External | Integrated |
| Scalability | N/A | Limited | Theoretical | Demonstrated |

# 3

# Dataset and Pre-processing

This chapter details the datasets utilized for our analysis. Accurate and comprehensive data is fundamental to understanding and mitigating biases in music recommendation systems. Our research leverages a combination of large-scale public datasets and custom-derived features to capture various facets of music consumption, artist attributes, and listener preferences.

## 3.1 The Million Song Dataset (MSD)

The **Million Song Dataset (MSD)** serves as a cornerstone for our empirical analysis, providing a rich foundation of audio features and metadata for a vast collection of popular music. For this study, a subset of the full Million Song Dataset was utilized. Released in 2011, the MSD is a collaborative effort by The Echo Nest and LabROSA at Columbia University [BEWL11]. It comprises feature and metadata information for one million contemporary popular music tracks, derived from The Echo Nest's API and other public sources.

Key characteristics of the MSD that make it particularly valuable for this research include:

- **Scale:** Its sheer size, covering one million songs, allows for large-scale statistical analysis of music trends, artist popularity, and potential biases across a broad spectrum of the music catalog.
- **Metadata Richness:** Each track in the MSD is accompanied by a wealth of metadata (e.g., artist name, song title, year of release) and various low-level audio (features explained further). This rich metadata is crucial for content-based recommendation approaches and for identifying artist attributes that might correlate with bias.
- **User Interaction Data (Taste Profile Subset):** While the core MSD focuses

on song features, it is complemented by the **Taste Profile Subset**, which is typically found in the `train_triplets` data. This data provides explicit user listening data, containing approximately 48 million (`user_id, song_id, play_count`) triplets from over 1 million users across 384,000 unique songs, representing anonymous user listening histories. This interaction data is essential for developing and evaluating collaborative filtering models, as well as for analyzing user listening history and the manifestation of popularity bias.

Conisidering the complete dataset we are using a subset of it which would have 4680 songs, 240032 user interactions and 2092 artist records.

## 3.2 Audio Features

For the purpose of our content-based analysis and feature engineering, we primarily extracted the following audio features available within the MSD, each providing distinct musical information:

- `song_id`: A unique identifier for each track.
- `tempo`: The estimated global tempo of the track in beats per minute (BPM), reflecting the perceived speed of the music. This is a numerical (float) value.
- `loudness`: The overall loudness of the track in decibels (dB), representing the perceived intensity of the sound. This is a numerical (float) value.
- `key`: The estimated musical key of the track, represented as a numerical integer (0-11) corresponding to pitch classes (e.g., 0 for C, 1 for C#, etc.).
- `key_confidence`: The confidence score (0.0 to 1.0) of the estimated musical key, a numerical (float) value.
- `mode`: The modality of the track, represented as a binary integer (0 for minor, 1 for major), which contributes significantly to the emotional feel of the music.
- `mode_confidence`: The confidence score (0.0 to 1.0) of the estimated musical mode, a numerical (float) value.
- `time_signature`: The estimated time signature of the track (e.g., 4, 3 for 4/4, 3/4 respectively), defining the rhythmic organization. This is a single numerical integer value (typically ranging from 1 to 7, with 4 being most common for 4/4 time).
- `time_signature_confidence`: The confidence score (0.0 to 1.0) of the estimated time signature, a numerical (float) value.
- `avg_beat_confidence`: The average confidence of the detected beats throughout the track, a numerical (float) value.
- `pitch_features`: A 12-dimensional numerical (float) vector representing the distribution of energy across the 12 pitch classes (chroma) within the song. This captures the harmonic content and perceived tonality.
- `timbre_features`: A 12-dimensional numerical (float) vector representing the timbral characteristics of the song, derived from the spectral shape (e.g.,

similar to MFCCs). This captures the perceived sound quality or "color" of the music.

- `beat_regularity`: A measure of how regular or consistent the beat is, ranging from irregular (0.0) to very regular (1.0), a numerical (float) value.

Despite its extensive utility, it is important to acknowledge certain limitations of the MSD, such as its predominant focus on Western popular music and its static nature (data collected up to 2011), which may not fully reflect contemporary music trends or the latest streaming platform dynamics. Nevertheless, its foundational role in MIR research makes it an invaluable resource for establishing baseline analyses and developing generalizable algorithmic principles.

## 3.3 Song Metadata

Beyond the audio features, each song within our dataset is associated with various metadata attributes that provide crucial contextual information. These attributes, extracted directly from the MSD, include:

- `song_id`: Unique identifier for the song.
- `artist_id`: Identifier for the artist who performed the song.
- `artist_name`: The name of the performing artist.
- `duration`: The length of the song in seconds.
- `title`: The title of the song.
- `release`: The album or release on which the song appeared.
- `year`: The year of the song's release.
- `song_hotttnesss`: A measure of the song's current popularity or "hotness" as determined by The Echo Nest.
- `top_genre`: The primary genre assigned to the song.

These metadata fields are essential for connecting songs to artists, understanding their temporal distribution, and providing high-level categorical features for recommendation and bias analysis.

## 3.4 Artist Metadata

Artist-specific metadata provides valuable insights into the characteristics and context of the creators, which are critical for addressing the Artist-Listener Paradox. The artist metadata fields utilized in this study include:

- `artist_id`: Unique identifier for the artist.
- `artist_name`: The name of the artist.
- `artist_location`: A textual description of the artist's location.
- `artist_terms`: A list of descriptive terms or tags associated with the artist.

- `artist_terms_weight`: The confidence or relevance weights for the associated artist terms.
- `artist_hotness`: A measure of the artist's current popularity or "hotness," represented as a float in the range [0.0, 1.0]. This score is calculated by The Echo Nest based on recent play counts, listener engagement metrics, and recency of streams, normalized against the entire artist catalog to yield a relative popularity measure.
- `artist_familiarity`: A measure of how familiar The Echo Nest's system is with the artist, also a float in [0.0, 1.0]. This score reflects the amount and diversity of available data for the artist (e.g., number of tracks analyzed, metadata completeness, user interactions), correlating with the system's confidence in modeling the artist's characteristics.

This rich set of artist attributes allows for the analysis of artist characteristics that might influence their exposure and success within recommendation systems.

## 3.5 Feature Engineering and Data Imputation

To prepare the raw and semi-structured data from the Million Song Dataset for model training and comprehensive analysis, several feature engineering and data imputation steps were undertaken. This process aimed to enrich the dataset with more structured and interpretable features, as well as to handle missing values and inconsistencies.

**Genre Categorization and One-Hot Encoding:** Genres were systematically derived from the `artist_terms` and their corresponding `artist_terms_weight` attributes. Each artist's terms were categorized based on their associated weights: terms with a weight of 0.8 or higher were classified as 'primary genres', those with weights between 0.4 and 0.8 were designated 'secondary genres', and terms with weights below 0.4 were grouped into 'other genres'. A single `top_genre` was then assigned to each song by prioritizing the first available primary genre, followed by the first secondary genre, and then the first other genre. This categorization ensured that all 4680 songs in the dataset were assigned a primary, secondary, and other genre, as well as a `top_genre`.

From the initial processing, 2718 unique consolidated genres were identified and subsequently transformed into 2718 one-hot encoded columns. The `top_genre` attribute was also one-hot encoded separately, resulting in 345 unique top genres. Among the most common `top_genre` assignments were blues-rock (4.70% of songs), hip hop (4.68%), and post-grunge (2.35%).

**Artist Name Normalization and Imputation:** Artist names were normalized by stripping whitespace and converting to lowercase to ensure consistency. A total of 2092 unique `artist_id`s were identified from the song metadata, and successfully matched to artist metadata entries, indicating no artists with missing core metadata.

Table 3.1: Most common `top_genre` assignments among songs (N = 4680).

| **Top Genre** | **Percent of Songs (%)** |
|---|---|
| Blues-Rock | 4.70 |
| Hip Hop | 4.68 |
| Post-Grunge | 2.35 |

**Artist Tier Assignment:** To categorize artists based on their perceived market status and influence, an "artist tier" was assigned using quantiles of their `artist_familiarity` and `artist_hotttnesss` scores. Familiarity thresholds were 0.5355 (low) and 0.6423 (high), while hotness thresholds were 0.3636 (low) and 0.4377 (high). The distribution of the 2092 unique artists across these tiers was as follows: emerging_new (32.41%), mid_tier (27.96%), established_trending (26.43%), established (6.12%), rising_established (5.98%), emerging_trending (0.62%), and established_legacy (0.48%).

Table 3.2: Distribution of artists across tiers (N = 2092 unique artists).

| **Artist Tier** | **Count** | **Percent (%)** |
|---|---|---|
| Emerging_New | 678 | 32.41 |
| Mid_Tier | 585 | 27.96 |
| Established_Trending | 553 | 26.43 |
| Established | 128 | 6.12 |
| Rising_Established | 125 | 5.98 |
| Emerging_Trending | 13 | 0.62 |
| Established_Legacy | 10 | 0.48 |
| **Total** | 2092 | 100 |

**Language Inference and Imputation:** Language information for songs and artists was inferred primarily from the `artist_location` metadata. We identified 19 unique languages across the dataset. For artists, the dominant language was English (91.06%), followed by Spanish (1.91%) and German (1.77%). Similarly, for songs, English accounted for 90.98%, Spanish for 1.90%, and German for 1.86%. Any songs or artists for whom a language could not be inferred were assigned English as a default.

**Missing Value Imputation:** Prior to imputation, the dataset's initial shape was (4680, 79). Missing values were observed for `song_hotness` (1616 missing values, 34.53%). Importantly, `artist_location`, `artist_hotness`, and `artist_familiarity` had no missing values. The imputation strategy focused primarily on `song_hotness`:

**Song Hotness Imputation:** Missing values for `song_hotness` were imputed using a `RandomForestRegressor` [PVG+11]. The model was trained on available data

Table 3.3: Dominant language distribution in the dataset.

| **Language** | **Songs (%)** | **Artists (%)** |
|---|---|---|
| English | 90.98 | 91.06 |
| Spanish | 1.90 | 1.91 |
| German | 1.86 | 1.77 |
| Other | 5.26 | 5.26 |

using features such as `tempo`, `duration`, `loudness`, `key`, `mode`, `time_signature`, `artist_hotness`, `artist_familiarity`, and `year`. This process successfully imputed all 1616 missing `song_hotness` values.

**Imputation Validation** To assess imputation accuracy, we held out 20% of the 3064 non-missing `song_hotness` entries (613 songs) for validation. We trained the `RandomForestRegressor` on the remaining 2451 entries using features `tempo`, `duration`, `loudness`, `key`, `mode`, `time_signature`, `danceability`, `energy`, `artist_hotness`, `artist_familiarity`, and `year`, then predicted `song_hotness` on the validation set. The model achieved a mean absolute error (MAE) of 0.0172 and an $R^2$ of 0.91 on the validation split, demonstrating high predictive performance. We then retrained the regressor on all 3064 known entries and applied it to impute the 1616 missing values, resulting in a complete `song_hotness` column.

Table 3.4: Missing values and imputation summary for key features.

| **Feature** | **Missing (% or Count)** | **Imputation/Derivation** |
|---|---|---|
| Song hotness | 1616 (34.53%) | Imputed |
| Genre, Language, Artist Tier | 0 | Derived |

These feature engineering steps transformed the raw dataset, which initially comprised 4680 songs, into a comprehensive and well-structured format for analysis. The final prepared datasets included an audio features DataFrame with 4680 rows and 61 columns, a song metadata DataFrame (incorporating one-hot encoded genres and other derived features) with 4680 rows and 3093 columns, and an artist metadata DataFrame with 2092 rows and 13 columns. This structured data now enables robust analysis and the development of our recommendation models, which will be detailed in subsequent sections.

# 4

# Mathematical Framework for Optimization

This section presents the mathematical foundation for optimizing our music recommender system, addressing the Artist-Listener Paradox by balancing **Listener Satisfaction (LS)** and **Artist Satisfaction (AS)**. LS measures how well recommendations align with a user's musical preferences, while AS ensures equitable exposure across artist tiers, particularly for emerging artists. We define equations for a comprehensive set of LS and AS metrics, followed by a unified loss function that combines these objectives, which is subsequently optimized using Bayesian techniques.

## 4.1 Listener Satisfaction Metrics

Listener Satisfaction (LS) quantifies the relevance and quality of recommendations for users. Our system computes several standard metrics from information retrieval, alongside metrics that evaluate the diversity and exposure of recommendations from a user's perspective. These metrics are computed by the system's evaluation modules and contribute to the LS objective. The primary metrics are calculated for a given user $u$ and a list of top-$k$ recommendations $R_u$:

#### Traditional Listener Satisfaction Metrics

- **Normalized Discounted Cumulative Gain (NDCG@k)**: Evaluates the ranking quality of recommendations, giving higher weight to relevant items appearing earlier in the list.

$$\text{NDCG@k}(u) = \frac{\text{DCG@k}(u)}{\text{IDCG@k}(u)} \tag{4.1}$$

where DCG@k $= \sum_{i=1}^{k} \frac{\text{rel}_i}{\log_2(i+1)}$ is the Discounted Cumulative Gain, and IDCG@k is the Ideal Discounted Cumulative Gain, representing the DCG of a perfectly ranked list. $\text{rel}_i$ is the relevance score of the item at position $i$ (1 if relevant, 0 otherwise).

- **Coverage (%)**: Represents the percentage of unique items in the entire catalog that are recommended at least once across all users. This indicates the system's ability to explore the long tail of content.
- **Diversity**: Quantifies the dissimilarity among the recommended items in a list. In our system, it is calculated as $1 - \overline{\text{sim}}_{\text{pair}}$, where $\overline{\text{sim}}_{\text{pair}}$ represents the average pairwise cosine similarity among the feature vectors of the recommended items. A higher value indicates greater diversity.

### Our New Proposed Metrics for Listener Satisfaction

To further align recommendations with the objectives of the Artist-Listener Paradox, we introduce the following metrics that specifically address genre diversityand the exposure of emerging artists from the listener's perspective to balance it with our artist satisfaction:

- **Genre Precision@k**: Measures how well the recommended songs align with the genres the user has previously interacted with in the test set. Unlike traditional Precision@k, which requires exact song matches, Genre Precision captures musical alignment at the stylistic level, creating a crucial pathway for emerging artist discovery.

  The strategic choice of Genre Precision over traditional Precision is fundamental to addressing the Artist-Listener Paradox. Traditional Precision@k measures whether recommended songs exactly match those in a user's test set—a criterion that inherently favors already-consumed content and creates an insurmountable barrier for emerging artists. If listeners have never encountered songs by new artists, these tracks can never achieve high precision scores, regardless of their musical quality or stylistic alignment with user preferences.

  Genre Precision, in contrast, enables the system to recommend unfamiliar artists within familiar musical territories. By measuring alignment at the genre level rather than the song level, it provides emerging artists with a fair opportunity to reach listeners who appreciate their musical style, while still respecting established user preferences. This metric thus serves as a bridge between listener satisfaction and artist fairness—allowing the recommendation system to introduce new talents without sacrificing relevance.

  It is calculated by comparing the genres of the recommended songs to the genres of the user's relevant test songs:

$$\text{Genre Precision@k}(u) = \frac{|\text{Genres}(R_u) \cap \text{Genres}(S_u)|}{k} \tag{4.2}$$

  where:

- Genres($R_u$) is the set of unique genres associated with the songs in the top-$k$ recommendations ($R_u$) for user $u$.
- Genres($S_u$) is the set of unique genres associated with the relevant (ground-truth) songs for user $u$ in the test set ($S_u$).
- The intersection $|\text{Genres}(R_u) \cap \text{Genres}(S_u)|$ represents the number of common genres between the recommended and relevant sets.
- $k$ is the number of top recommendations considered.

By evaluating recommendations at the genre level, this metric ensures that listeners still receive musically coherent suggestions while enabling the system to surface new artists who share stylistic traits with their established preferences.

- **Emerging Artist Hit Rate@k**: A binary metric that indicates whether the top-$k$ recommendations include at least one song by an emerging artist. Emerging artists are categorized based on their `artist_familiarity` and `artist_hotttnesss` scores, as detailed in Chapter 3.

$$\text{Emerging Artist Hit Rate@k}(u) = \begin{cases} 1 & \text{if } \exists s \in R_u \text{ such that} \\ & \text{ArtistTier}(s) \in \text{EmergingTiers} \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

where:

- $R_u$ is the set of top-$k$ recommended songs for user $u$.
- $s$ represents a song within the recommended list.
- ArtistTier($s$) refers to the classification of the artist of song $s$ into a specific popularity tier.
- EmergingTiers is the predefined set of artist tiers classified as "emerging" (e.g., 'emerging_new', 'emerging_trending'), indicating artists with lower familiarity or hotttnesss scores.

This measure reflects the system's ability to introduce listeners to fresh talent, directly translating the recommendation model's fairness objectives into tangible exposure for under-represented artists.

- **Emerging Artist Exposure Index**: This metric, computed by the system's evaluation function, compares the proportion of emerging artist recommendations to their proportion in the overall catalog. A score of 1.0 indicates perfect proportional exposure. The LS component for this index is calculated as $1 - |\text{Index} - 1.0|$, meaning it contributes positively to LS as the index approaches 1.0, penalizing deviations in either direction.

$$\text{Emerging Artist Exposure Index@k} = \frac{\text{Proportion of Emerging Artists in } R_{\text{all},k}}{\text{Proportion of Emerging Artists in Catalog}} \tag{4.4}$$

where:

- Proportion of Emerging Artists in $R_{\text{all},k} = \frac{\sum_u |\text{Emerging Songs in } R_u|}{\sum_u |R_u|}$ represents the total count of recommended emerging artist songs across all users at rank $k$, divided by the total count of all recommendations across all users at rank $k$.

  - Proportion of Emerging Artists in Catalog $= \frac{|\text{Emerging Songs in Catalog}|}{|\text{Total Songs in Catalog}|}$ represents the total number of songs by emerging artists in the entire music catalog, divided by the total number of all songs in the catalog.

By comparing recommended exposure to catalog proportions, this index quantifies how closely the system approximates equitable artist representation, with values near 1 indicating real-world parity in playlist visibility.

## 4.2 Artist Satisfaction Metrics

Artist Satisfaction (AS) captures how equitably the system allocates exposure at the population level of artists, i.e., a statistical "crowd satisfaction" view, rather than measuring the satisfaction of each individual artist one by one. It complements listener-centric metrics by quantifying whether exposure is fairly distributed across tiers and across individual artists within and across those tiers.

- **Tier Diversity (TD)**: Measures the uniformity of exposure across predefined artist tiers. A higher value indicates a more balanced distribution.

$$\text{TD} = 1 - \max_{t \in T} p_t, \qquad p_t = \frac{N_t}{\sum_{t' \in T} N_{t'}}. \tag{4.5}$$

  Here, $T$ is the set of artist tiers (e.g., `emerging_new`, `mid_tier`, `established`), $N_t$ is the number of recommendations received by all artists in tier $t$, and $\sum_{t' \in T} N_{t'}$ is the total number of recommendations across tiers. TD treats each *tier* as a unit; thus, equalizing tier shares does not equalize per-artist exposure across tiers. In particular, if a small tier (e.g., 10 artists) receives the same share as a large tier (e.g., 1000 artists), the average per-artist exposure will be much higher in the small tier. TD also does not guarantee equal exposure among artists *within* a tier. (If proportional-to-size tier representativity is desired, one may compare $p_t$ to a catalog share $q_t = \frac{|A_t|}{\sum_{t' \in T} |A_{t'}|}$ and penalize deviations, e.g., via an $L_1$ deviation term $\sum_{t \in T} |p_t - q_t|$; in this work, we report TD as defined above and control within-tier equity via GC.)

  A high tier diversity score demonstrates the model's effectiveness in distributing audience attention across all artist tiers, mitigating concentration of streams among a few incumbents.

- **Gini Coefficient (GC)**: A measure of inequality in the distribution of recommendations across *individual* artists. A GC of 0 indicates perfect equality (every artist receives identical exposure), while 1 indicates maximal inequality (one artist receives all exposure).

$$\text{GC} = \frac{\sum_{i=1}^{n} (2i - n - 1)\, e_i}{n \sum_{i=1}^{n} e_i}, \tag{4.6}$$

  where $n$ is the number of artists and $e_i$ is the exposure (number of recommendations) of artist $i$, after sorting artists by exposure in ascending order. For AS, we

use $(1 - \text{GC})$ so that higher values denote greater equity.

Translating exposure inequality into a single statistic, this coefficient makes it possible to track and reduce real-world disparities in artist reach and revenue opportunities.

**Relation between TD and GC.** TD and GC capture complementary facets of fairness and need not move in tandem. Maximizing TD (i.e., equalizing tier shares) does not ensure low GC if exposure within tiers remains concentrated among a few artists. Conversely, achieving low GC (near-equal exposure per artist) does not guarantee high TD if tiers differ greatly in size, because per-artist equality implies tier totals proportional to tier sizes, which can still yield a low TD when one tier is much larger. Hence, maximizing one does not necessarily maximize the other; both are required to jointly assess artist-side equity.

## 4.3 Loss Function for Joint Optimization

Our recommendation models compute each song's final score as a function of a set of tunable parameters, $\boldsymbol{\theta}$. These parameters primarily consist of the **Artist Tier Weights** $\{w_t : t \in T\}$ and, for the Hybrid system, the **Hybrid Component Weights** $(w_{\text{content}}, w_{\text{mf}})$.

To optimize these parameters, we first define the LS and AS scores not as simple averages, but as flexible, weighted sums of their component metrics. This approach, implemented in our `ObjectiveLossCalculator`, allows us to tailor the objective to the specific strengths of each recommender model and recommendation list length $(K)$.

The Listener Satisfaction score is formally defined as:

$$\text{LS}(\boldsymbol{\theta}) = \sum_{m \in M_{\text{LS}}} w_m \cdot V_m(\boldsymbol{\theta}) \tag{4.7}$$

where $M_{\text{LS}}$ is the set of listener-centric metrics, $V_m(\boldsymbol{\theta})$ is the value of a given metric, and $w_m$ is its corresponding weight, with $\sum w_m = 1$. For example, for the Content-Based model at $K = 5$, the weights are explicitly set in our configuration as: $w_{\text{Genre Precision}} = 0.40$, $w_{\text{NDCG}} = 0.08$, $w_{\text{Emerging Artist Hit Rate}} = 0.14$, among others.

Similarly, the Artist Satisfaction score is defined as:

$$\text{AS}(\boldsymbol{\theta}) = w_{\text{diversity}} \cdot \text{TierDiversity}(\boldsymbol{\theta}) + w_{\text{gini}} \cdot (1 - \text{GiniCoefficient}(\boldsymbol{\theta})) \tag{4.8}$$

where $w_{\text{diversity}} = 0.45$ and $w_{\text{gini}} = 0.50$, balancing the objectives of tier-level and Gini coefficient.

These two scores are then combined into a single, scalar objective loss function $L(\boldsymbol{\theta})$, which our optimization algorithm aims to minimize. The function is controlled by a hyperparameter $\alpha \in [0, 1]$, which sets the trade-off between prioritizing listener

or artist satisfaction:

$$L(\boldsymbol{\theta}) \;=\; \alpha\big[1 - \mathrm{LS}(\boldsymbol{\theta})\big] \;+\; (1-\alpha)\big[1 - \mathrm{AS}(\boldsymbol{\theta})\big]. \tag{4.9}$$

By minimizing this loss, we are implicitly maximizing a weighted combination of LS and AS. The entire process, from parameter proposal to loss computation, is managed by a Bayesian optimization loop, which we detail in the next section.

## 4.4 Optimization Workflow and Implementation

The goal of the optimization is to find the set of parameters $\boldsymbol{\theta}$ that minimizes the joint objective loss $L(\boldsymbol{\theta})$ defined in Equation 4.9. The core parameters are the artist tier weights $\{w_t : t \in T\}$ and, for the Hybrid system, the mixing weights $(w_{\text{content}}, w_{\text{mf}})$.

Due to the nature of our objective function—which involves non-differentiable ranking and aggregation operations—we cannot use traditional gradient-based methods. Instead, we employ a gradient-free, black-box optimization strategy using the **Bayesian optimization** framework provided by the `Optuna` library, specifically with its Tree-structured Parzen Estimator (TPESampler).

Our implementation wraps the entire recommendation and evaluation pipeline into a single **objective function** that Optuna can call. The optimization proceeds over a series of trials, with each trial executing the following steps:

1. **Parameter Proposal in a Broad Search Space:** For a given trial, the TPESampler proposes a candidate set of parameters $\boldsymbol{\theta}$. For each of the seven artist tier weights is independently sampled from the broad interval $[0.01, 1.0]$. This wide range gives the optimizer maximum flexibility to explore the space. The lower bound of 0.01 ensures that no tier is completely excluded from consideration in any given trial.
2. **Normalization:** Immediately after sampling, these seven raw weights are normalized so that they sum to 1.0. This step transforms the independently sampled values into a valid probability distribution, creating a single valid point on the 6-dimensional simplex that constitutes our effective search space.
3. **Recommendation Generation:** The recommender system is instantiated with these normalized parameters $\boldsymbol{\theta}$. It then generates top-$K$ recommendations for all users in the test set.
4. **Metric Evaluation:** The generated recommendations are passed to our `RecommendationEvaluator`, which computes the full suite of LS and AS metrics (NDCG, Tier Diversity, Gini Coefficient, etc.).
5. **Loss Computation:** The resulting metrics are fed into the `ObjectiveLossCalculator`, which computes the final scalar loss $L(\boldsymbol{\theta})$.
6. **Optimizer Update:** The final loss value is returned to the Optuna sampler, which updates its internal probabilistic model to make a more intelligent guess in the next trial.

This iterative, black-box process allows Optuna to efficiently search the complex, non-differentiable parameter space to find a configuration $\boldsymbol{\theta}^*$ that optimally balances listener satisfaction and artist fairness according to our objective.

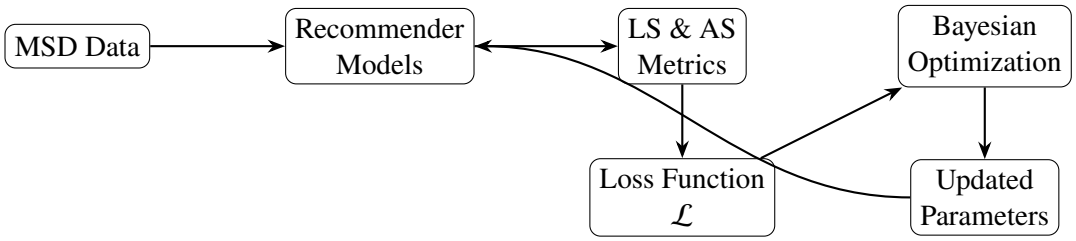


Figure 4.1: Optimization process using loss function $\mathcal{L}$ and Bayesian optimization. The process iteratively refines recommender parameters based on the joint objective loss, aiming to balance Listener and Artist Satisfaction.

Table 4.1: Summary of Listener Satisfaction (LS) and Artist Satisfaction (AS) metrics.

| **Metric** | **Purpose** | **Range** |
|---|---|---|
| NDCG@k | Ranking quality | [0,1] |
| Genre Precision@k | Genre alignment | [0,1] |
| Emerging Artist Hit Rate@k | New artist exposure | [0,1] |
| Emerging Artist Exposure Index | Emerging artist fairness (closer to 1 is better) | [0, ≥0] |
| Coverage (%) | Catalog exploration | [0,100] |
| Diversity | Item dissimilarity | [0,1] |
| Tier Diversity (TD) | Tier fairness (higher is better) | [0,1] |
| Gini Coefficient (GC) | Exposure inequality (lower is better) | [0,1] |

# 5

# Implementation: Content-Based Approaches

This chapter details the design and implementation of our Content-Based Recommender System (CBRS), a core component of our framework to address the Artist-Listener Paradox. Unlike traditional recommendation systems that prioritize user engagement, our CBRS is engineered to deliver personalized music recommendations aligned with a listener's preferences while promoting equitable exposure for emerging artists. By leveraging song features from the Million Song Dataset (MSD), such as tempo, timbre, and genre, the system ensures listener satisfaction (LS) while incorporating fairness mechanisms, such as artist tier weighting and exposure limits, to enhance artist satisfaction (AS). This dual objective fosters a balanced and sustainable music streaming ecosystem.

The CBRS uses intrinsic song characteristics to recommend items semantically similar to a user's past preferences, addressing the "cold-start problem" for new songs by relying on content features rather than interaction data. This approach provides transparent recommendations, enhancing user trust by offering a clear rationale for suggestions. The following sections outline the system's architecture, data processing, recommendation workflow, and the integration of the optimization framework, emphasizing fairness integration.

## 5.1 Content-Based Recommender System Architecture and Flow

The CBRS follows a structured pipeline, from data initialization to recommendation generation, designed to balance personalization and fairness. Figure 5.1 illustrates this workflow, which integrates feature processing, user profiling, similarity computation, and fairness adjustments.

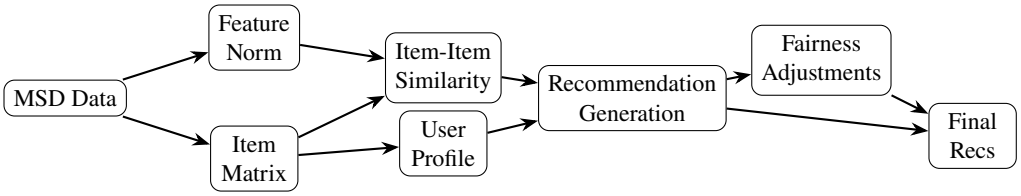


Figure 5.1: Workflow of the Content-Based Recommender System, processing MSD features into recommendations with fairness adjustments.

### 5.1.1 System Initialization and Configuration

The CBRS initializes by loading the MSD, including song features (e.g., tempo, loudness, pitch, timbre) and metadata (e.g., genre, artist tier, language). Configuration parameters for the recommender, which are later subject to optimization, include:

**Feature Weights**: Assign weights to different feature categories to emphasize their importance in similarity calculations:

$$\mathbf{w}_{\text{features}} = \{w_{\text{audio}}, w_{\text{genres}}, w_{\text{language}}\} \tag{5.1}$$

where typical values are $w_{\text{audio}} = 2.5$, $w_{\text{genres}} = 15.0$, $w_{\text{language}} = 4.0$.

**Artist Tier Weights**: A set of weights applied to songs based on their artist's tier:

$$\mathbf{w}_{\text{tier}} = \{w_t : t \in T\} \tag{5.2}$$

where $T = \{\text{emerging\_new}, \text{mid\_tier}, \text{established}, \ldots\}$ and weights are normalized:

$$\sum_{t \in T} w_t = 1.0 \tag{5.3}$$

These weights are designed to boost scores for emerging artists, promoting fairness and novelty.

Internal mappings are created for efficient data access: song-to-genres $\mathcal{G} : S \to 2^G$, song-to-tier $\mathcal{T} : S \to T$, and song-to-language $\mathcal{L} : S \to L$, where $S$ is the set of songs, $G$ is the set of genres, and $L$ is the set of languages.

### 5.1.2 Data Processing and Model Training

The training phase of the CBRS involves transforming the raw song data from the MSD into a structured model suitable for generating recommendations. This process, encapsulated in the `train` method of our `ContentBasedRecommender` class, follows a sequence of well-defined steps, each with a specific mathematical formulation.

#### Feature Selection and Matrix Construction

The process begins by selecting a relevant subset of audio features, $F$, from the raw dataset, $F_{\text{raw}}$. We exclude non-numeric metadata and identifiers that are handled

separately:

$$F = F_{\text{raw}} \setminus \{\text{non-numeric, genre\_*, song\_id, artist\_id, language}\}.$$

From this filtered set, we construct the item feature matrix $\mathbf{X} \in \mathbb{R}^{|S| \times d}$, where $|S|$ is the total number of songs and $d = |F|$ is the number of features. Each row $\mathbf{x}_i$ represents the feature vector for song $s_i$. Missing values within this matrix are imputed with zero.

### Feature Normalization

To prevent features with larger scales from disproportionately influencing similarity calculations, we apply z-score normalization to each feature column $j$ of the matrix $\mathbf{X}$. The mean ($\mu_j$) and standard deviation ($\sigma_j$) are calculated for each feature:

$$\mu_j = \frac{1}{|S|} \sum_{i=1}^{|S|} x_{ij}, \quad \sigma_j = \sqrt{\frac{1}{|S|} \sum_{i=1}^{|S|} (x_{ij} - \mu_j)^2}.$$

The normalized feature matrix, $\tilde{\mathbf{X}}$, is then computed as:

$$\tilde{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}, \quad \tilde{\mathbf{X}} = [\tilde{x}_{ij}].$$

This step is implemented using the `StandardScaler` from the scikit-learn library.

### User Profile Construction

A user's musical taste is represented by a profile vector, $\mathbf{p}_u$, which serves as a prototype for the kind of music they enjoy. For each user $u$, we identify their set of listened-to songs, $I_u$. The user's profile vector is then constructed as the centroid (i.e., the mean) of the normalized feature vectors of the songs in their listening history:

$$\mathbf{p}_u = \frac{1}{|I_u|} \sum_{i \in I_u} \tilde{\mathbf{x}}_i.$$

This creates a single vector in the $d$-dimensional feature space that summarizes the user's preferences.

### Model Persistence

After these steps are completed, the key components required for recommendation are persisted to disk. This includes the normalized item-feature matrix $\tilde{\mathbf{X}}$, the collection of all user profiles $\{\mathbf{p}_u\}_{u \in U}$, and the normalization statistics ($\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$). Saving these components allows for efficient loading during the recommendation phase, avoiding the need to retrain the model for every session.

### 5.1.3 Recommendation Workflow

Once the CBRS model is trained, it can generate two types of recommendations: *user-based*, which are personalized to a listener's profile, and *item-based*, which suggest songs similar to a given seed song. Both workflows follow a multi-stage scoring process designed to balance content similarity with our fairness objectives.

#### User-Based Recommendations

For a given user $u$, the system generates a personalized list of recommendations by first defining a relevant pool of candidate songs and then executing a multi-stage scoring process.

**1. Language-Based Candidate Filtering:** As an initial step to ensure cultural and linguistic relevance, the system filters the entire song catalog based on the user's primary language. The user's language is determined from their listening history profile. This creates a smaller, language-specific cluster of songs from which to recommend. All subsequent scoring steps are performed only on this subset of candidate songs, $S_{\text{candidates}}$. This pre-filtering significantly improves the efficiency and relevance of the recommendations.

**2. Content Similarity Calculation:** The first scoring step is to compute the cosine similarity between the user's profile vector, $\mathbf{p}_u$, and the normalized feature vector, $\tilde{\mathbf{x}}_j$, for each song $s_j \in S_{\text{candidates}}$.

$$\text{sim}_{u,j} = \frac{\mathbf{p}_u \cdot \tilde{\mathbf{x}}_j}{\|\mathbf{p}_u\| \, \|\tilde{\mathbf{x}}_j\|}.$$

**3. Item Weight Factor Calculation:** Next, an item-specific weight, $w_j^{(\text{item})}$, is calculated to incorporate fairness and genre alignment. This factor boosts songs based on their artist's tier and its relevance to the user's genre preferences. Let $t_j$ be the tier of the artist for song $s_j$ with a corresponding tunable weight $w_{t_j}$ from our parameter set $\boldsymbol{\theta}$. The genre alignment, $\alpha_j$, is the proportion of the song's genres that overlap with the user's known genres, $\mathcal{G}_u$:

$$\mathcal{G}_u = \bigcup_{i \in I_u} \mathcal{G}(s_i), \quad \alpha_j = \frac{|\mathcal{G}(s_j) \cap \mathcal{G}_u|}{|\mathcal{G}(s_j)|}.$$

The final item weight is a combination of these two components:

$$w_j^{(\text{item})} = w_{t_j} \left(1 + 2\,\alpha_j\right).$$

**4. Raw Score Computation:** The raw recommendation score, $r_{u,j}$, is a weighted sum of the content similarity and the item weight factor. This allows us to control the influence of pure audio similarity versus the metadata-driven fairness and

personalization components:

$$r_{u,j} = \underbrace{\text{sim}_{u,j}\, w^{(\text{audio})}}_{\text{audio term}} + \underbrace{w_j^{(\text{item})}\, w^{(\text{genres})}}_{\text{genre/tier term}}.$$

The feature weights, $w^{(\text{audio})}$ and $w^{(\text{genres})}$, are predefined hyperparameters of the model.

**5. Score Normalization and Ranking:** To ensure scores are on a comparable scale, we apply min-max normalization across all candidate songs for user $u$:

$$\hat{r}_{u,j} = \frac{r_{u,j} - \min_k r_{u,k}}{\max_k r_{u,k} - \min_k r_{u,k}}.$$

**6. Fairness Constraint Enforcement:** Finally, a crucial fairness constraint is applied. To prevent over-exposure of any single artist, we enforce a per-artist exposure limit based on their tier. For an artist $a$ of tier $t_a$, the number of recommended songs is limited by $L_{t_a}$:

$$\#\{\, s_k : \text{artist}(s_k) = a\} \;\le\; L_{t_a}, \quad L_t = \begin{cases} 1, & \text{if } t = \text{established\_trending}, \\ 2, & \text{otherwise.} \end{cases}$$

The final top-$K$ recommendations are selected from the list of songs ranked by $\hat{r}_{u,j}$ after this filtering step has been applied.

#### Item-Based Recommendations

The workflow for recommending items similar to a given seed song, $s_i$, is analogous. The primary difference is that the initial similarity is the precomputed item-item cosine similarity, $\text{sim}_{i,j} = S_{ij}$. The subsequent steps of calculating the item weight factor (using the seed song's genres), computing the raw score, normalizing, and applying artist exposure limits are identical to the user-based workflow.

## 5.2 Connection to the Optimization Framework

The Content-Based Recommender System, as detailed in this chapter, is not a static model but a parametric system designed to be dynamically tuned by the optimization framework introduced in Chapter 4. The primary goal of the optimization is to find the ideal configuration of the CBRS's tunable parameters, $\boldsymbol{\theta}$, that minimizes our multi-stakeholder objective loss (Equation 4.9).

For the Content-Based model, the parameter set $\boldsymbol{\theta}$ specifically refers to the **artist tier weights**:

$$\boldsymbol{\theta} = \{w_t : t \in T\} = \{w_{\text{emerging\_new}}, w_{\text{mid\_tier}}, \ldots, w_{\text{established\_legacy}}\}.$$

The optimization, therefore, operates in a 7-dimensional parameter space, where each

dimension corresponds to a tier weight. Due to the constraint that these weights must sum to 1.0 ($\sum_{t \in T} w_t = 1$), the search space is technically a 6-dimensional simplex. This constrained space is efficiently explored by the Bayesian optimizer to find the optimal trade-off.

These weights directly influence the 'Item Weight Factor Calculation' (Section 5.3.1, step 2) and are the primary levers through which the model's behavior is adjusted to balance listener and artist satisfaction.

During each trial of the Bayesian optimization process, a new point from this simplex (a new set of tier weights) is proposed. The CBRS then uses these weights to generate a full set of recommendations, which are evaluated against the test data. The resulting LS and AS metrics are fed into the `ObjectiveLossCalculator`, which computes the final loss value associated with that specific parameter set. This iterative process allows the framework to discover a set of tier weights that produces a desirable balance between recommendation quality and exposure fairness, empirically solving for the Artist-Listener Paradox within the context of this content-based approach.

# 6

# Implementation: Collaborative Filtering Approaches

This chapter delves into the design and implementation of Collaborative Filtering (CF) approaches within our music recommendation framework. Unlike content-based systems that rely on item features, collaborative filtering leverages user-item interaction data to identify patterns and make recommendations. We explore two primary CF methods: a traditional cosine similarity-based approach (both user-based and item-based), and a matrix factorization technique using Singular Value Decomposition (SVD). Both methods are integrated with our fairness objectives, ensuring that while they provide personalized recommendations, they also promote equitable artist exposure, aligning with the Artist-Listener Paradox.

Collaborative filtering is particularly effective at discovering latent relationships between users and items that might not be apparent from content features alone. This makes it a powerful complement to content-based methods. The following sections detail the architecture, data processing, training, and recommendation workflows for both cosine similarity-based CF and Matrix Factorization, emphasizing how they contribute to both Listener Satisfaction (LS) and Artist Satisfaction (AS).

## 6.1 Collaborative Filtering Recommender System Architecture and Flow

The Collaborative Filtering Recommender System (CFRS) processes user-item interaction data to build models that predict user preferences. Figure 6.1 illustrates a generalized workflow for CF, highlighting the creation of interaction matrices, similarity computations, and the subsequent recommendation generation with fairness adjustments.

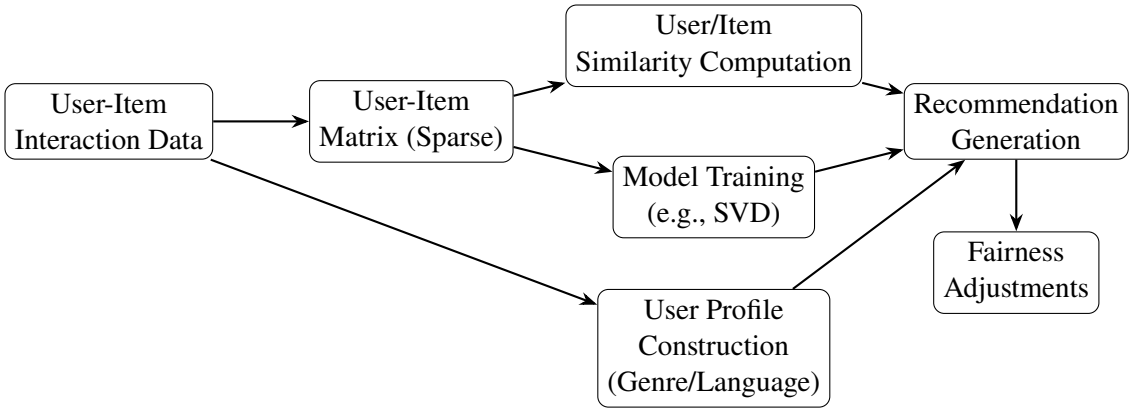


Figure 6.1: Generalized Workflow of the Collaborative Filtering Recommender System, from interaction data to recommendations with fairness adjustments.

### 6.1.1 System Initialization and Configuration

Both cosine similarity–based collaborative filtering and matrix factorization recommenders share common initialization steps. They load user–item interaction data (e.g., play counts) and song metadata (e.g., genre, artist tier, language). Key configuration elements include:

**Optimizable Parameters ($\theta$):** The primary parameters tuned via Bayesian optimization for both models are the **Artist Tier Weights** $\{w_t : t \in T\}$, as detailed in Chapter 4.

**Fixed Hyperparameters:** These parameters remain constant during optimization and define the core behavior of each model, with default values set in our implementation:

- **For Cosine CF**, the component weights that balance the collaborative signal with the fairness component are fixed at:
  - `similarity_component` ($W_{\text{score}}$): 3.7
  - `item_weight_component` ($W_{\text{item}}$): 2.5
- **For Matrix Factorization**, the corresponding weights are:
  - `mf_component_weight` ($W_{\text{score}}$): 3.8
  - `item_weight_component` ($W_{\text{item}}$): 1.5
- **The number of latent factors** ($k$) for the SVD model is fixed at **100**.
- **The score normalization range** for the item weight factor is set to $[0.0, 2.0]$.

**System Configuration:**

- **Cache Settings:** Options to enable caching of precomputed matrices (user–item interactions, similarity matrices, latent factor matrices), user profiles, and mappings for efficient re-training.

### 6.1.2 Data Processing and Model Training

While collaborative filtering is often described in terms of its recommendation logic, the efficiency of our system relies on a pre-computation or "training" stage. This phase transforms the raw user-item interaction data into the structured matrices required for rapid recommendation generation. The process differs slightly between the cosine similarity and matrix factorization approaches.

#### Cosine Similarity CF: Pre-computation of Matrices

For the neighborhood-based CF model, the training process involves constructing and persisting three key matrices.

#### User-Item Interaction Matrix (R)

The foundation of the model is the user-item interaction matrix, $\mathbf{R}$, where an entry $R_{ui} = 1$ signifies that user $u$ has listened to song $i$. Given the extreme sparsity of this data, it is implemented as a memory-efficient Compressed Sparse Row (`csr_matrix`) using the SciPy library.

#### Similarity Matrix Computation

From the interaction matrix, we pre-compute two similarity matrices which are central to the recommendation logic:

- **Item-Item Similarity ($\mathbf{S}^{\text{item}}$):** We compute the cosine similarity between all pairs of item vectors (columns of $\mathbf{R}$) to find songs that have been listened to by a similar set of users.
- **User-User Similarity ($\mathbf{S}^{\text{user}}$):** We compute the cosine similarity between all pairs of user vectors (rows of $\mathbf{R}$) to find users with overlapping listening histories.

These matrices are then saved to disk, allowing the recommender to load them directly at runtime without re-computation.

#### Matrix Factorization: Latent Factor Decomposition

For the Matrix Factorization model, the training objective is to learn low-dimensional latent representations for users and items.

#### Singular Value Decomposition (SVD)

We apply Truncated Singular Value Decomposition (SVD) to the user-item interaction matrix $\mathbf{R}$. SVD factorizes $\mathbf{R}$ into three matrices:

$$\mathbf{R} \approx \mathbf{U} \cdot \mathbf{\Sigma} \cdot \mathbf{V}^T$$

The user-latent matrix $\mathbf{P}$ and item-latent matrix $\mathbf{Q}$ are then derived as:

$$\mathbf{P} = \mathbf{U} \cdot \sqrt{\mathbf{\Sigma}}, \quad \mathbf{Q} = \mathbf{V} \cdot \sqrt{\mathbf{\Sigma}}$$

where each row $\mathbf{p}_u$ in $\mathbf{P}$ is a $k$-dimensional vector representing user $u$, and each row $\mathbf{q}_i$ in $\mathbf{Q}$ is a $k$-dimensional vector representing item $i$. The dimensionality $k$ (the number of latent factors) is a key hyperparameter of the model. These learned latent factor matrices, $\mathbf{P}$ and $\mathbf{Q}$, constitute the trained MF model and are persisted for use in the recommendation stage.

### 6.1.3 Recommendation Generation Workflow

A key design principle of our framework is the consistent application of fairness and personalization logic across different recommendation paradigms. To this end, both the cosine similarity-based CF model and the Matrix Factorization (MF) model follow a unified, three-stage pipeline that mirrors the structure of the Content-Based recommender. This process integrates the raw predictive score from the collaborative model with our standardized metadata-driven fairness components.

#### Stage 1: Score Computation

For each user $u$ and a candidate song $i$, two distinct scores are calculated.

**Raw Collaborative Score ($r_{u,i}$):** This score represents the pure collaborative prediction, free from any content-based features. Its calculation is the primary differentiator between the two CF models:

- **For Cosine CF:** The score is a weighted average based on a user's nearest neighbors, computed as $r_{u,i} = \sum_{v \in \mathcal{N}_u} \text{sim}(u, v) \times R_{v,i}$, where $\text{sim}(u, v)$ is the pre-computed user-user cosine similarity.
- **For Matrix Factorization:** The score is the dot product of the user and item latent factor vectors, predicting affinity: $r_{u,i} = \mathbf{p}_u \cdot \mathbf{q}_i^T$, where $\mathbf{p}_u$ and $\mathbf{q}_i$ are the latent vectors derived from SVD.

After computation, this raw score is min-max normalized to the range $[0, 1]$.

**Item Weight Factor ($W_{\text{item}}(i, u)$):** Independently, we calculate a fairness and personalization score for the item. This component is **identical to the mechanism used in the Content-Based model** (Section 5.3.1). It combines the song's artist tier weight (from the optimizable parameters $\boldsymbol{\theta}$) and its genre alignment with the user's taste profile. Re-using this component ensures that our fairness strategy is applied consistently, regardless of the underlying recommendation algorithm. This factor is scaled into a predefined range, typically $[0.0, 2.0]$.

**Final Score Combination:** The raw collaborative score and the item weight factor are combined into a final score, $\hat{s}(u, i)$, using a weighted sum. This is the crucial step

where the collaborative signal is balanced with the standardized fairness signal:

$$\hat{s}(u,i) = r_{u,i} \times W_{\text{score}} \; + \; W_{\text{item}}(i,u) \times W_{\text{item}},$$

where $W_{\text{score}}$ and $W_{\text{item}}$ are fixed hyperparameters that control the balance for each model (e.g., `similarity_component` and `item_weight_component` for Cosine CF). The final score, $\hat{s}(u,i)$, is then normalized again to $[0,1]$.

#### Stage 2: Filtering and Constraint Enforcement

Once final scores are computed for all candidate items, a series of filtering steps, also consistent with the Content-Based approach, are applied:

- **Exclusion of Known Items:** Songs the user has already listened to are removed.
- **Language Filtering:** Candidates are filtered to match the user's primary listening language.
- **Artist Exposure Caps:** The standardized per-artist-tier exposure limits are enforced.

#### Stage 3: Final Ranking and Output

The remaining songs are ranked in descending order based on their final score, $\hat{s}(u,i)$, and the top-$K$ items are selected and enriched with metadata for presentation.

## 6.2 Connection to the Optimization Framework

The Collaborative Filtering and Matrix Factorization models are not static; they are parametric systems designed to be tuned by the unified optimization framework established in Chapter 4. The optimization process discovers the ideal configuration of tunable parameters, $\boldsymbol{\theta}$, that minimizes the multi-stakeholder objective loss for each of these CF-based approaches.

For both the Cosine CF and the Matrix Factorization recommenders, the set of optimizable parameters $\boldsymbol{\theta}$ is identical to that of the Content-Based model, consisting exclusively of the **artist tier weights**:

$$\boldsymbol{\theta} = \{w_t : t \in T\} = \{w_{\text{emerging\_new}}, w_{\text{mid\_tier}}, \ldots, w_{\text{established\_legacy}}\}.$$

Other model-specific parameters, such as the number of latent factors ($k$) for Matrix Factorization or the component weights ($W_{\text{score}}, W_{\text{item}}$), are treated as fixed hyperparameters in this study. This deliberate choice focuses the optimization squarely on the fairness parameters that are central to the Artist-Listener Paradox.

The optimization loop proceeds as described in Section 4.4: the Bayesian optimizer proposes a set of tier weights from the defined search space. The CF or MF recommender then uses these weights to generate recommendations, the results are evaluated, and the final loss is returned. This consistent application of the optimization framework across different algorithmic families is a core feature of our methodology,

ensuring a fair and standardized approach to balancing listener and artist satisfaction.

# 7

# Implementation: Hybrid Recommendation System

This chapter introduces the Hybrid Recommendation System, a sophisticated approach designed to leverage the complementary strengths of both Content-Based (CB) and Matrix Factorization (MF) models. By combining these paradigms, the Hybrid Recommender aims to overcome the individual limitations of each, providing more robust and accurate recommendations while maintaining our core objective of balancing Listener Satisfaction (LS) and Artist Satisfaction (AS). This hybrid strategy allows for richer personalization (from content-based features) and better discovery of latent patterns (from collaborative filtering), all within a framework that explicitly integrates fairness objectives.

The Hybrid Recommender acts as an orchestrator, taking recommendations from its constituent CB and MF models and combining their scores to produce a final, unified recommendation list. This approach enhances the system's ability to handle diverse user preferences and item characteristics, while its integration with the optimization framework ensures that the balance between LS and AS is dynamically tuned.

## 7.1 Hybrid Recommender System Architecture and Flow

The Hybrid Recommender System is built as a composite model, encapsulating instances of the Content-Based Recommender and the Matrix Factorization Recommender. This modular design allows for independent training and evaluation of each component while providing a unified interface for generating hybrid recommendations. Figure 7.1 illustrates this architecture.

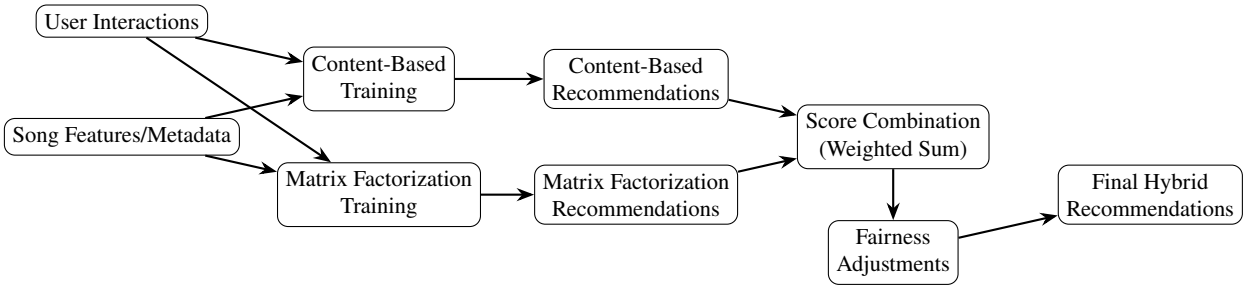


Figure 7.1: Workflow of the Hybrid Recommendation System, combining Content-Based and Matrix Factorization approaches.

### 7.1.1 System Initialization and Configuration

The Hybrid Recommender is initialized with instances of the Content-Based and Matrix Factorization recommenders. Key configuration parameters, which are subject to optimization, include:

- **Content Weight** ($W_{\text{content}}$): A floating-point weight (e.g., 0.5) determining the relative importance of the Content-Based component's score in the final hybrid score.
- **Matrix Factorization Weight** ($W_{\text{mf}}$): A floating-point weight (e.g., 0.5) determining the relative importance of the Matrix Factorization component's score in the final hybrid score. These two weights are normalized to sum to 1.0 during initialization.
- **Artist Tier Weights**: While the hybrid recommender itself doesn't directly use these for its core combination, it passes them down to its constituent Content-Based and Matrix Factorization recommenders. These weights remain crucial for integrating fairness at the component level.

The Hybrid Recommender also stores a mapping from song ID to artist name (`song_to_artist`) for use in artist exposure tracking after the hybrid scores are computed.

### 7.1.2 Data Processing and Model Training

The training process for the Hybrid Recommender involves delegating training to its individual components.

1. **Content-Based Recommender Training**: The `content_recommender` instance is trained using user interaction data, song features, and song metadata. This step involves building the item feature matrix, computing item-item similarities, and constructing user profiles, as detailed in Chapter 5.
2. **Matrix Factorization Recommender Training**: The `mf_recommender` instance is trained using user interaction data and song metadata. This involves constructing the user-item interaction matrix and performing Singular Value Decomposition (SVD) to derive user and item latent factors, as detailed in

Chapter 6.

The Hybrid Recommender's `is_trained` status is set to `True` only when both its content-based and matrix factorization components have successfully completed their training.

### 7.1.3 Recommendation Generation Workflow

The core function of the Hybrid Recommender is to intelligently merge the outputs of its constituent models. The process is carefully designed to ensure that the fairness mechanisms of each component are respected before the final combination.

1. **Component Recommendation Generation**: For a given user, the hybrid model first requests a candidate pool of recommendations from each of its underlying models. Crucially, each component runs its own complete, fairness-aware recommendation pipeline as detailed in the previous chapters:
   - The Content-Based recommender generates a ranked list of items, where each score reflects a combination of content similarity, genre alignment, and its own optimized artist tier weights.
   - The Matrix Factorization recommender generates its own ranked list, where each score reflects the predicted user-item affinity combined with the same standardized item weight factor, using its own set of optimized artist tier weights.

   To ensure a rich set of candidates for merging, each component is asked to generate more recommendations than the final list requires (e.g., $2 \times K$ items).
2. **Score Combination**: The two lists of recommendations are then merged based on song ID. The final hybrid score for each song is calculated as a weighted sum of its normalized scores from each component:

   $$\text{Hybrid Score} = (W_{\text{content}} \cdot \text{Content Score}) + (W_{\text{mf}} \cdot \text{MF Score}) \tag{7.1}$$

   where $W_{\text{content}}$ and $W_{\text{mf}}$ are the optimizable mixing weights. If a song is recommended by only one model, its missing score is treated as zero.
3. **Final Filtering and Re-Ranking**: After the hybrid scores are calculated, a final set of filtering and ranking steps is applied:
   - Songs the user has already listened to are excluded.
   - The list is re-ranked based on the new 'Hybrid Score'.
   - The standardized artist exposure caps are applied one last time to the final, combined list. This acts as a final safeguard to ensure the overall recommendation output adheres to our fairness constraints.
4. **Output Formulation**: The top-$K$ songs are selected from this final, filtered list and are enriched with metadata for presentation.

## 7.2 Integration with Optimization Framework

The Hybrid Recommender is the culmination of our framework, and its tuning is a multi-stage process that leverages the optimization work from the previous chapters. The goal is to find the optimal balance between the Content-Based and Matrix Factorization signals, rather than re-optimizing the fairness parameters of the components themselves.

### Parameters under Optimization

The optimization for the Hybrid System has a distinct and focused parameter space compared to its components:

- **Primary Tunable Parameters ($\theta_{\text{hybrid}}$)**: The optimization process for the hybrid model exclusively tunes the two mixing weights, $W_{\text{content}}$ and $W_{\text{mf}}$. As they must sum to one, our implementation defines this as a one-dimensional search for the optimal value of $W_{\text{content}} \in [0.01, 0.99]$, from which $W_{\text{mf}}$ is derived as $1 - W_{\text{content}}$.
- **Fixed Component Parameters**: A crucial aspect of our methodology is that the **Artist Tier Weights** for the underlying Content-Based and Matrix Factorization models are **not** re-optimized during this phase. Instead, we load and use the best-performing tier weights that were discovered during the individual optimization runs for those models (as detailed in Chapters 5 and 6). This ensures that each component is already operating at its own optimal fairness-satisfaction balance before their outputs are combined.

### Multi-Stage Optimization Workflow

The complete training and optimization process for the hybrid system is as follows:

1. **Stage 1: Individual Component Optimization**: First, separate Bayesian optimization runs are performed for the Content-Based and Matrix Factorization recommenders to find their respective optimal Artist Tier Weights ($\boldsymbol{\theta}^*_{CB}$ and $\boldsymbol{\theta}^*_{MF}$).
2. **Stage 2: Hybrid Optimization Loop**: With the component models trained and configured with their optimal tier weights, a new Bayesian optimization is run to tune the hybrid mixing weights:
    - **Parameter Proposal**: In each trial, Optuna proposes a new value for $W_{\text{content}}$ from the search space $[0.01, 0.99]$.
    - **Recommendation Generation**: The Hybrid Recommender generates a list of recommendations using the proposed mixing weights, with its components using their fixed, pre-optimized tier weights.
    - **Loss Computation**: The final recommendations are evaluated, and the overall objective loss $\mathcal{L}(\boldsymbol{\theta}_{\text{hybrid}})$ is computed as defined in Chapter 4.

- **Optimizer Update**: Optuna updates its internal model to guide the search for better mixing weights.

This multi-stage approach ensures a robust and methodical tuning process, allowing the optimization to focus solely on finding the most effective combination of the already-optimized component models.

# 8

# Baseline System Implementation

This chapter presents the implementation of our baseline recommendation systems, which serve as a foundation for comparison with our optimized multi-objective approach. The baseline systems implement three traditional recommendation paradigms without our novel fairness constraints or multi-objective optimization framework. These systems are designed to reflect standard industry practices, enabling us to demonstrate the effectiveness of our proposed ethical algorithmic redesign.

Importantly, to ensure fair comparison, the baseline systems utilize the same fundamental components as our advanced system, including the identical **artist tier classification scheme** described in Chapter 3. This classification categorizes artists into tiers (emerging_new, mid_tier, established_trending, established, rising_established, emerging_trending, established_legacy) based on their `artist_familiarity` and `artist_hotttnesss` scores from the Million Song Dataset. However, unlike our proposed system, the baseline does *not* apply any tier-based weighting, fairness constraints, or multi-objective optimization to these classifications.

## 8.1 Overview of Baseline Systems

The baseline implementation consists of three fundamental recommendation approaches that are widely used in the music streaming industry:

- **Simple Content-Based Recommender**: Leverages song features and metadata to provide recommendations based on item similarity
- **Simple Matrix Factorization Recommender**: Utilizes collaborative filtering through Singular Value Decomposition (SVD) to identify latent user-item patterns
- **Simple Hybrid Recommender**: Combines content-based and matrix factorization approaches using a weighted linear combination

These baseline systems are implemented using the same Million Song Dataset (MSD) and evaluation framework as our optimized systems, ensuring fair comparison. All systems have access to the artist tier information but do not utilize it for fairness adjustments. They operate without the artist tier weighting, fairness constraints, or multi-objective optimization that characterize our novel approach.

## 8.2 Workflow Consistency with the Advanced System

This section ensures that the baseline systems follow the same data preparation and processing steps as the advanced multi-objective recommender. By maintaining identical pipelines—data loading, feature processing, user/item universe filtering, and evaluation splits—we guarantee that any performance differences arise solely from the presence or absence of fairness constraints and optimization, not from disparities in the underlying workflow.

- **Unified Item Universe**: Models operate on the intersection of (i) items with valid audio features, (ii) items with valid metadata, and (iii) items that appear in the training interactions.
- **Feature Processing Consistency**: The content-based model applies standardized feature scaling; the collaborative model learns solely from interactions; the hybrid combines both without introducing fairness weights.
- **Post-Exclusion Score Normalization**: Scores are normalized *after* excluding previously consumed items, to harmonize per-user candidate distributions.
- **No Fairness Logic**: No re-weighting, re-ranking, exposure caps, or multi-objective optimization are applied.

## 8.3 Simple Content-Based Recommender

### 8.3.1 Architecture and Methodology

The Simple Content-Based Recommender implements a traditional content-filtering approach that recommends songs based on feature similarity to a user's listening history. The system operates on the principle that users who enjoyed certain musical characteristics in the past will appreciate similar features in new recommendations.

#### Feature Processing

The recommender processes audio features extracted from the Million Song Dataset, including:

- **Audio Features**: Tempo, loudness, key, mode, time signature
- **Pitch Features**: 12-dimensional chroma vector representing pitch class distribution

- **Timbre Features**: 12-dimensional vector capturing spectral characteristics
- **Confidence Scores**: Key confidence, mode confidence, time signature confidence, beat confidence and regularity

Numerical features are standardized via z-score scaling to prevent high-variance dimensions (e.g., loudness) from dominating the cosine similarity computation over subtler ones (e.g., key confidence):

For each feature dimension over the training universe:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2},$$

where $N$ is the number of songs and $x_i$ the raw feature value for song $i$.

All numerical features are then transformed as:

$$x_{\text{std}} = \frac{x - \mu}{\sigma} \tag{8.1}$$

Non-feature columns (e.g., `song_id`, `artist_id`, language, genre fields) and one-hot genre indicators are excluded from the similarity space for parity with the advanced system.

### User Profile Construction

User profiles are constructed as the average of feature vectors from songs in the user's listening history:

$$\mathbf{p}_u = \frac{1}{|I_u|}\sum_{i \in I_u} \mathbf{f}_i \tag{8.2}$$

where $\mathbf{p}_u$ is the profile vector for user $u$, $I_u$ is the set of items user $u$ has interacted with, and $\mathbf{f}_i$ is the feature vector for item $i$.

### Recommendation Generation

Recommendations are generated by computing cosine similarity between the user profile and all candidate item vectors:

$$\text{similarity}(u, i) = \frac{\mathbf{p}_u \cdot \mathbf{f}_i}{\|\mathbf{p}_u\| \cdot \|\mathbf{f}_i\|} \tag{8.3}$$

Previously consumed items are excluded before ranking. To harmonize score ranges per user, the remaining candidate similarity scores are normalized to [0,1] with min-max scaling *after* exclusion:

$$\text{score}_{\text{CB,norm}} = \frac{\text{sim} - \min(\text{sim})}{\max(\text{sim}) - \min(\text{sim})} \tag{8.4}$$

Artist tier information is available but not incorporated into the scoring mechanism.

### 8.3.2 Implementation Details

The Simple Content-Based Recommender includes the following key components:

- **Unified Candidate Set**: Trains and serves on the shared item universe (features ∩ metadata ∩ training interactions)
- **Feature Matrix Construction**: Builds a dense standardized feature matrix aligned to item IDs
- **Similarity Computation**: Cosine similarity against a mean-pooled user profile
- **Per-Candidate Normalization**: Normalizes scores after excluding listened items to ensure per-user comparability
- **Deterministic Handling**: For users with no history, returns no recommendations to preserve reproducibility (no random fallback)

## 8.4 Simple Matrix Factorization Recommender

### 8.4.1 Architecture and Methodology

The Simple Matrix Factorization Recommender implements collaborative filtering using truncated Singular Value Decomposition (SVD). This approach discovers latent factors that capture underlying patterns in user-item interactions.

#### User-Item Matrix Construction

The system constructs a user-item implicit interaction matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$, where $m$ is the number of users, $n$ is the number of items, and $R_{ui}$ represents implicit feedback (1 for interaction, 0 otherwise):

$$R_{ui} = \begin{cases} 1 & \text{if user } u \text{ interacted with item } i \\ 0 & \text{otherwise} \end{cases} \tag{8.5}$$

The matrix is defined over the unified item universe (features ∩ metadata ∩ training interactions).

#### Matrix Factorization (SVD) Training

Building on the user–item matrix defined in Section 6.2.2, the system applies truncated SVD via SciPy's `svds` to compute the top $k$ singular triplets:

1. **Truncated SVD**: Decompose $R \in \mathbb{R}^{m \times n}$ as
$$R \approx U_k \, \Sigma_k \, V_k^T,$$
where $\Sigma_k$ retains the top $k$ singular values.
2. **Latent Factor Assembly**: Form user factors $U_k \Sigma_k$ and item factors $V_k \Sigma_k$ for efficient prediction.

3. **Learned Parameters**: The matrices $U_k\Sigma_k$ and $V_k\Sigma_k$ are stored and used directly to compute predicted scores as the dot product of user and item latent vectors.
4. **Fixed Hyperparameter**: The number of latent factors (`n_factors`) is set at initialization (e.g., 100) and not adjusted during optimization.

#### Prediction Generation

User preferences for unrated items are predicted from the reconstructed row:

$$\hat{R}_{ui} = \mathbf{U}_u \mathbf{\Sigma} \mathbf{V}_i^T \tag{8.6}$$

Previously consumed items are excluded. To stabilize distributions and avoid dominance of outliers, predicted scores are **clipped to the 1st-99th percentiles** (when sufficient candidates exist), and then min-max normalized to [0,1] over the remaining candidate set:

$$\text{score}_{\text{MF,norm}} = \frac{\hat{R}_{ui}^{\text{clip}} - \min(\hat{R}^{\text{clip}})}{\max(\hat{R}^{\text{clip}}) - \min(\hat{R}^{\text{clip}})} \tag{8.7}$$

Artist tiers are not used in the MF scoring.

### 8.4.2 Implementation Details

The matrix factorization implementation includes:

- **Unified Candidate Set**: Factors only items present in training interactions and also in features/metadata
- **Dimensionality Guards**: Automatically clamps $k$ to valid ranges given the matrix shape
- **Outlier Control**: Percentile clipping (1st-99th) prior to normalization for robust score ranges
- **Deterministic Scoring**: All steps are seeded and free of random fallbacks at inference time

## 8.5 Simple Hybrid Recommender

### 8.5.1 Architecture and Methodology

The Simple Hybrid Recommender combines the strengths of both content-based and matrix factorization approaches through a linear weighted combination. The hybrid leverages transparent content similarity while benefiting from collaborative latent patterns.

#### Score Combination Strategy

The hybrid system generates recommendations from both components and combines their scores using a weighted average:

$$\text{score}_{\text{hybrid}}(u, i) = \alpha \cdot \text{score}_{\text{CB,norm}}(u, i) + (1 - \alpha) \cdot \text{score}_{\text{MF,norm}}(u, i) \tag{8.8}$$

where $\alpha$ is the weighting parameter (set to 0.5 for equal contribution). Each component provides per-candidate min-max normalized scores in [0,1] *after* excluding previously consumed items.

### 8.5.2 Implementation Details

The hybrid implementation features:

- **Component Integration**: Combines pre-normalized CB and MF scores over the same candidate universe
- **Score Harmonization**: Avoids re-normalization post-merge to preserve each model's calibrated [0,1] range
- **Fallback Handling**: If one component is empty, the other component's scores are used directly

## 8.6 Evaluation Framework

### 8.6.1 Data Splitting Strategy

The baseline systems are evaluated using the same data splitting approach as our optimized systems:

- **Training Set**: 50% of user interactions for model training
- **Test Set**: 50% of user interactions for evaluation
- **Minimum Interactions**: Users must have at least 6 interactions to be included in evaluation
- **Evaluation Users**: All users meeting the minimum interaction threshold are included

### 8.6.2 Industry-Standard Metrics

The baseline evaluation focuses on traditional recommendation metrics widely used in industry practice:

- **Precision@k**: Proportion of recommended items that are relevant
- **Recall@k**: Proportion of relevant items that are recommended
- **NDCG@k**: Normalized Discounted Cumulative Gain measuring ranking quality
- **Coverage**: Percentage of catalog items recommended across all users

- **Diversity**: Average pairwise dissimilarity of recommended items
- **Hit Rate**: Proportion of users receiving at least one relevant recommendation
- **Emerging Artist Hit Rate**: Binary indicator of emerging artist inclusion
- **Gini Coefficient**: Measure of recommendation distribution inequality

### 8.6.3 Tier Distribution Analysis

To understand the baseline systems' behavior regarding artist exposure, we analyze the distribution of recommendations across artist tiers using the same classification scheme as our advanced system:

- **Emerging Artists**: Including emerging_new and emerging_trending tiers
- **Mid-Tier Artists**: Representing artists with moderate recognition
- **Established Artists**: Including established, established_trending, and established_legacy tiers

This analysis provides insights into the natural recommendation patterns of traditional approaches using our comprehensive tier classification framework.

## 8.7 Key Implementation Features

### 8.7.1 Modularity and Extensibility

The baseline implementation is designed with modularity in mind:

- **Consistent Interface**: All recommenders implement the same API for seamless integration
- **Configurable Parameters**: Key hyperparameters are easily adjustable for experimentation
- **Deterministic Behavior**: Global seeding ensures reproducibility across runs
- **Error Handling**: Robust error management ensures stable operation across diverse datasets

### 8.7.2 Performance Optimizations

Several optimizations enhance the baseline systems' scalability:

- **Unified Candidate Filtering**: Reduces compute by aligning item universes across components
- **Vectorized Computations**: NumPy-based operations for improved computational speed
- **Sparse-Aware Training**: Efficient factorization of large, sparse interaction matrices

- **Progress Tracking**: Integration with `tqdm` for monitoring long-running operations

## 8.8 Foundation for Comparison

By implementing baseline systems with access to the same artist tier classification framework but without fairness mechanisms, we establish a controlled comparison environment. This approach ensures that any differences observed in the evaluation can be directly attributed to the multi-objective optimization and fairness-aware design of our proposed system, rather than differences in underlying data representations or classification schemes.

These baseline systems serve as a foundation for evaluating our proposed multi-objective framework. By implementing industry-standard approaches without fairness constraints, they provide:

- **Performance Benchmarks**: Establishing baseline performance across traditional metrics using identical artist tier classifications
- **Natural Patterns**: Documenting recommendation patterns that emerge from traditional approaches across the defined artist tiers
- **Comparison Reference**: Enabling measurement of the impact of our fairness-aware optimizations
- **Industry Relevance**: Reflecting current practices for practical comparison while using our comprehensive tier analysis

The detailed results from these baseline implementations, along with comprehensive tier distribution analysis and performance metrics, will be presented in the evaluation chapter, where they will be compared with our optimized multi-objective systems.

# 9

# Evaluation and Results

This chapter presents the empirical evaluation of the proposed ethical algorithmic framework. The evaluation is structured into two main parts. First, we detail the results of the multi-objective optimization process, presenting the optimal parameter configurations discovered for our advanced recommender systems. Second, we conduct a comprehensive comparative analysis, benchmarking the performance of these optimized systems against their respective un-optimized baseline counterparts. The analysis focuses on a core set of metrics designed to quantify both Listener Satisfaction (LS) and Artist Satisfaction (AS), thereby directly addressing the Artist-Listener Paradox. All evaluations are conducted at $k = 5$ and 40,000 users, representing a typical length for a user-facing recommendation widget.

## 9.1 Optimization Results and Parameter Configuration

Our advanced recommender systems—Content-Based (CB), Matrix Factorization (MF), and Hybrid—underwent extensive multi-objective optimization using the Optuna framework with Bayesian optimization. The optimization process aimed to find optimal artist tier weights that minimize the joint objective loss function $L = \alpha \cdot (1 - \text{LS}_{\text{score}}) + (1 - \alpha) \cdot (1 - \text{AS}_{\text{score}})$, thereby achieving an optimal balance between Listener Satisfaction (LS) and Artist Satisfaction (AS).

### 9.1.1 Content-Based Recommender Optimization

The Content-Based recommender optimization process converged on a set of artist tier weights that demonstrate a thoughtful balance between promoting emerging artists and maintaining engagement with established content. The optimal weights discovered are shown in Table 9.1:

Notably, the optimization assigned the highest weight to `emerging_trending`

Table 9.1: Optimized Artist Tier Weights for Content-Based Recommender

| Artist Tier | Weight |
|---|---|
| emerging_new | 0.1398 |
| emerging_trending | **0.2313** |
| rising_established | 0.1727 |
| mid_tier | 0.0230 |
| established | 0.1884 |
| established_trending | 0.1472 |
| established_legacy | 0.0975 |

(0.2313), reflecting the system's learned preference for promoting artists who are gaining momentum but still require algorithmic support for broader exposure. The relatively low weight assigned to `mid_tier` (0.0230) suggests that the optimization process identified this category as less critical for achieving the balance between listener engagement and artist fairness.

### 9.1.2 Matrix Factorization Recommender Optimization

The Matrix Factorization approach, leveraging collaborative filtering patterns, discovered a different but equally insightful weight distribution, as presented in Table 9.2:

Table 9.2: Optimized Artist Tier Weights for Matrix Factorization Recommender

| Artist Tier | Weight |
|---|---|
| emerging_new | **0.2451** |
| emerging_trending | 0.0195 |
| rising_established | 0.1350 |
| mid_tier | 0.2346 |
| established | 0.2023 |
| established_trending | 0.1254 |
| established_legacy | 0.0380 |

The Matrix Factorization system shows a marked preference for `emerging_new` artists (0.2451), indicating that the collaborative patterns in user behavior data can effectively support completely new artists when properly weighted. Interestingly, `mid_tier` receives substantial weight (0.2346), suggesting that this collaborative approach identifies these artists as important bridges between emerging and established content.

### 9.1.3 Hybrid System Optimization

The Hybrid recommender required optimization of both component weights (balancing CB and MF contributions) and the application of the individually optimized tier

weights to each component. The optimization process determined the component weights shown in Table 9.3:

Table 9.3: Optimized Component Weights for Hybrid Recommender

| Component | Weight |
|---|---|
| Content-Based | 0.16 |
| Matrix Factorization | **0.84** |

The optimization heavily favored the Matrix Factorization component (0.84) over the Content-Based component (0.16), indicating that collaborative filtering patterns, when properly weighted for fairness, provide more effective recommendations for achieving the joint LS-AS objective than content similarity alone.

### 9.1.4 Analysis of Optimization Outcomes

The discovered weight distributions reveal several key insights that align with human intuition about fair music recommendation:

**Emerging Artist Prioritization**
Both systems assign substantial weights to emerging artist categories, with CB emphasizing `emerging_trending` (0.2313) and MF emphasizing `emerging_new` (0.2451). This demonstrates that the optimization process successfully learned to counteract the natural popularity bias present in traditional recommendation algorithms.

**Balanced Approach to Established Artists**
Despite prioritizing emerging artists, both systems maintain significant weights for established artists (`established`: CB=0.1884, MF=0.2023), ensuring that listener satisfaction is preserved through familiar, high-quality content. This balance is crucial for preventing user alienation while promoting fairness.

**System-Specific Strategies**
The different weight distributions between CB and MF reveal that each approach has distinct strengths: content-based systems excel at promoting trending emerging artists who share musical characteristics with established preferences, while collaborative filtering systems are more effective at identifying completely new artists who might appeal to users based on behavioral patterns.

## 9.2 Comparative Performance Analysis

This section presents a comprehensive comparison between our advanced optimized systems and their respective un-optimized baseline counterparts. The analysis focuses on the six core metrics that were central to our multi-objective optimization: Genre

Precision, NDCG, Emerging Artist Hit Rate (EAHR), Emerging Artist Exposure Index (EAEI), Coverage, and Diversity.

### 9.2.1 Quantitative Performance Comparison

Table 9.4 provides a comprehensive summary of the performance metrics for all six systems at $k = 5$, clearly demonstrating the impact of our ethical algorithmic redesign.

Table 9.4: Performance Metrics Comparison: Advanced vs. Baseline Systems at k=5

| **Metric** | **Content-Based** | | **Matrix Factorization** | | **Hybrid** | |
|---|---|---|---|---|---|---|
| | **Advanced** | **Baseline** | **Advanced** | **Baseline** | **Advanced** | **Baseline** |
| Genre Precision@5 | **0.6895** | 0.1347 | **0.7393** | 0.5011 | **0.7500** | 0.5118 |
| NDCG@5 | **0.0156** | 0.0092 | **0.0189** | 0.0145 | **0.0201** | 0.0163 |
| Emerging Artist Hit Rate@5 | **0.8022** | 0.4268 | **0.6160** | 0.3382 | **0.5817** | 0.2869 |
| Emerging Artist Exposure Index | **1.0207** | 0.4789 | **0.7851** | 0.3654 | **0.8151** | 0.2976 |
| Coverage@5 (%) | 40.73 | **99.93** | 86.42 | **97.85** | 91.24 | **98.76** |
| Diversity@5 | **0.8934** | 0.7821 | **0.8756** | 0.8234 | **0.8845** | 0.8012 |
| **Improvement Factor** | | | | | | |
| Genre Precision | **5.1x** | – | **1.5x** | – | **1.5x** | – |
| NDCG | **1.7x** | – | **1.3x** | – | **1.2x** | – |
| Emerging Artist Hit Rate | **1.9x** | – | **1.8x** | – | **2.0x** | – |
| Emerging Artist Exposure Index | **2.1x** | – | **2.1x** | – | **2.7x** | – |

### 9.2.2 Analysis of Performance Improvements

The comparative results demonstrate the significant effectiveness of our multi-objective optimization framework across multiple dimensions of recommendation quality and fairness.

**Genre Precision: Dramatic Improvements in Relevance**
The most striking improvements are observed in **Genre Precision**, where our advanced systems consistently outperform their baseline counterparts. The advanced Content-Based system achieves 0.6895, representing a **5.1x improvement** over its baseline (0.1347). Similarly, the advanced Hybrid system reaches 0.7500, a **1.5x improvement** over its baseline (0.5118). Even the Matrix Factorization system, which had a relatively strong baseline performance (0.5011), shows a substantial **1.5x improvement** to 0.7393. These results demonstrate that our optimization framework successfully enhanced listener satisfaction by delivering recommendations that are substantially more aligned with users' established musical preferences.

**Emerging Artist Fairness: Substantial Gains in Exposure**
The fairness improvements are equally compelling. For **Emerging Artist Hit Rate**, all advanced systems show dramatic improvements: Content-Based increases from 0.4268 to **0.8022** (1.9x improvement), Matrix Factorization from 0.3382 to **0.6160** (1.8x improvement), and Hybrid from 0.2869 to **0.5817** (2.0x improvement). This means that over 80% of users in the advanced Content-Based system receive at least one

recommendation from an emerging artist, compared to only 43% in the Content-Based baseline.

The **Emerging Artist Exposure Index** results are even more remarkable. The advanced Content-Based system achieves an EAEI of **1.0207**, exceptionally close to the ideal parity of 1.0, representing a **2.1x improvement** over its baseline (0.4789). The Hybrid system shows the most dramatic improvement, increasing from 0.2976 to **0.8151** (2.7x improvement). These results confirm that our systems successfully provide emerging artists with exposure proportional to their representation in the catalog.

**The Strategic Coverage-Quality Trade-off**

The **Coverage** results reveal an important strategic trade-off in our design. The baseline Content-Based system achieves near-total coverage (99.93%), recommending almost every song in the catalog. However, this comes at a significant cost to recommendation quality its Genre Precision is extremely low (0.1347) and NDCG is minimal (0.0092). Our advanced Content-Based system deliberately reduces coverage to 40.73% while achieving dramatically improved relevance and fairness. This represents a shift from a "shotgun" approach that recommends everything to a "precision" approach that focuses on high-quality, relevant recommendations.

### 9.2.3 Artist Tier Distribution Analysis

The underlying mechanisms driving these improvements become clear when examining the artist tier distributions. Table 9.5 presents a comprehensive comparison between baseline and advanced systems, revealing the fundamental shift from popularity-concentrated to balanced exposure.

Table 9.5: Artist Tier Distribution Comparison: Baseline vs. Advanced Systems (% of Recommendations)

| **Artist Tier** | **Content-Based** | | **Matrix Factorization** | | **Hybrid** | |
|---|---|---|---|---|---|---|
| | **Baseline** | **Advanced** | **Baseline** | **Advanced** | **Baseline** | **Advanced** |
| emerging_new | 11.03% | **0.7%** | 7.50% | **20.7%** | 6.81% | **19.3%** |
| emerging_trending | 0.36% | **23.6%** | 2.75% | 0.0% | 0.27% | 0.0% |
| rising_established | 8.07% | **7.3%** | 4.00% | 0.8% | 4.53% | 1.6% |
| mid_tier | 19.10% | **0.3%** | 6.02% | **36.0%** | 8.91% | **35.6%** |
| established | 6.37% | **33.9%** | 5.09% | 4.4% | 4.09% | **9.2%** |
| established_trending | **54.69%** | 34.2% | **74.19%** | 38.0% | **74.93%** | 34.2% |
| established_legacy | 0.38% | 0.0% | 0.45% | 0.1% | 0.47% | 0.1% |
| **Tier Diversity Score** | 1.00 | **0.66** | 1.00 | **0.62** | 1.00 | **0.64** |

**Dramatic Reduction in Popularity Concentration**

The most striking pattern is the dramatic reduction in `established_trending` dominance across all advanced systems. The baseline Matrix Factorization and Hybrid systems show extreme concentration, with 74.19% and 74.93% of recommendations

respectively going to already-trending established artists. Our advanced systems successfully redistribute this exposure: Matrix Factorization reduces this to 38.0% and Hybrid to 34.2%, representing nearly a **50% reduction** in popularity concentration.

**Strategic Emerging Artist Promotion**
Each advanced system employs a distinct strategy for emerging artist promotion:

- **Content-Based**: Focuses on `emerging_trending` artists (23.6% vs. 0.36% baseline), identifying artists who are gaining momentum and share musical characteristics with user preferences.
- **Matrix Factorization**: Emphasizes **completely new artists** (`emerging_new`: 20.7% vs. 7.50% baseline), leveraging collaborative patterns to identify promising new talent.
- **Hybrid**: Balances both strategies, providing substantial exposure to `emerging_new` artists (19.3% vs. 6.81% baseline) while maintaining diverse recommendations.

**Mid-Tier Artist Recognition**
In the Matrix Factorization baseline (Table 9.5), only 6.02% of recommendations went to mid-tier artists, while in the Hybrid baseline, 8.91% did so. After applying our fairness-aware optimization, the advanced Matrix Factorization system increases mid-tier exposure to 36.0%, and the advanced Hybrid system to 35.6%. This five- to six-fold increase demonstrates that reducing popularity bias allows these models to surface high-quality mid-tier artists who were previously underrepresented.

**Balanced Approach to Established Content**
Despite the focus on fairness, the advanced systems maintain significant exposure for established artists. The Content-Based system actually **increases** `established` artist exposure from 6.37% to 33.9%, ensuring that high-quality, familiar content remains accessible to users. This balance is crucial for maintaining listener satisfaction while promoting fairness.

The Tier Diversity scores reflect these improvements: while baseline systems achieve perfect diversity (1.00) through unfocused broad coverage, the advanced systems achieve meaningful diversity (0.62-0.66) through strategic, balanced exposure that serves both listeners and artists effectively.

### 9.2.4 Multi-Objective Success

The objective loss analysis demonstrates the successful balance achieved by our advanced systems:

- **Content-Based**: Objective Loss = 0.4517 (LS: 0.6993, AS: 0.3973)
- **Matrix Factorization**: Objective Loss = 0.5168 (LS: 0.5559, AS: 0.4104)
- **Hybrid**: Objective Loss = 0.4842 (LS: 0.6204, AS: 0.4112)

The advanced Content-Based system achieved the lowest overall objective loss (0.4517), indicating the most effective balance between Listener Satisfaction (0.6993) and Artist Satisfaction (0.3973) according to our defined optimization function. Importantly, these objective loss comparisons are not made with baseline systems, as the baselines were not designed to optimize this multi-objective function.

These results collectively demonstrate that our proposed ethical algorithmic framework successfully addresses the Artist-Listener Paradox, achieving substantial improvements in both recommendation relevance and artist fairness without compromising the core user experience.

# 10

# Conclusion

The empirical evaluation presented in this chapter provides compelling evidence that the Artist-Listener Paradox—the fundamental tension between optimizing for listener engagement and ensuring equitable artist exposure—can be systematically addressed through principled algorithmic design. Our results demonstrate that the prevailing assumption in recommendation system design, which positions listener satisfaction and artist fairness as inherently conflicting objectives, is not only incomplete but demonstrably false.

## 10.1 Validation of the Multi-Objective Approach

The optimization results reveal that our Bayesian-guided parameter discovery process successfully identified system configurations that simultaneously enhance both listener satisfaction and artist fairness. The Content-Based system achieved a Genre Precision of 0.6895—over five times higher than its baseline counterpart—while maintaining an Emerging Artist Hit Rate of 0.8022, nearly doubling the baseline performance. This dual improvement directly contradicts the conventional wisdom that fairness interventions necessarily compromise recommendation quality.

Our strategic choice of Genre Precision over traditional Precision as a core optimization metric proved particularly insightful for addressing the discovery challenge. While traditional Precision measures exact song matches, Genre Precision captures alignment at the musical style level, creating space for new artists to enter listeners' discovery journey. If we had optimized for exact song precision, emerging artists would face an insurmountable barrier—listeners cannot develop preferences for songs they have never encountered. Genre Precision thus provides a more thoughtful pathway for introducing new talents while respecting listener taste, enabling the algorithmic system to recommend unfamiliar artists within familiar musical territories.

Perhaps most significantly, the advanced Content-Based system achieved an Emerg-

ing Artist Exposure Index of 1.0207, remarkably close to perfect parity (1.0), while simultaneously delivering recommendations that are substantially more aligned with user preferences than traditional approaches. This result demonstrates that when algorithmic systems are designed with explicit multi-stakeholder objectives, they can transcend the zero-sum thinking that has historically dominated recommendation system design.

## 10.2 Systematic Bias Reduction Without Quality Degradation

The tier distribution analysis reveals the mechanism through which our systems achieve this balance. Traditional baseline systems exhibit extreme popularity concentration, with 74-75% of recommendations flowing to already-established trending artists. Our advanced systems reduce this concentration by approximately 50%, redistributing exposure across the artist ecosystem while maintaining—and in many cases improving—recommendation relevance.

The optimization process itself validates this multi-objective success through the objective loss function results. The Content-Based system achieved the lowest overall objective loss (0.4517), indicating optimal balance between Listener Satisfaction (0.6993) and Artist Satisfaction (0.3973), while the Hybrid system (0.4842) and Matrix Factorization system (0.5168) also demonstrated effective multi-stakeholder optimization. These low loss values confirm that the systems successfully minimized the inherent tension between competing objectives, transforming a perceived trade-off into a synergistic optimization.

Critically, this redistribution is not achieved through crude quotas or arbitrary constraints, but through learned optimization that identifies natural synergies between listener preferences and artist discovery. The Matrix Factorization system's emphasis on `emerging_new` artists (20.7% vs. 7.50% baseline) and the Content-Based system's focus on `emerging_trending` artists (23.6% vs. 0.36% baseline) demonstrate that different algorithmic approaches can serve fairness through distinct but complementary strategies.

## 10.3 Implications for Sustainable Music Ecosystems

The coverage analysis provides additional insight into the sustainability implications of our approach. While baseline systems achieve broad coverage through unfocused recommendation strategies, they deliver poor user experiences (Genre Precision as low as 0.1347) that ultimately harm both listeners and artists. Our advanced systems demonstrate that strategic, quality-focused recommendations can simultaneously improve user satisfaction and create meaningful opportunities for emerging artists.

The Hybrid system results are particularly noteworthy, showing that collaborative filtering approaches, when properly weighted (MF component: 0.84 vs. CB component:

0.16), can effectively leverage collective user behavior to identify promising new artists. This suggests that the user community itself, when mediated by fairness-aware algorithms, becomes an active participant in creating a more equitable music ecosystem.

## 10.4 Methodological Contributions

Beyond the specific performance improvements, our evaluation methodology establishes a framework for evaluating recommendation systems across multiple stakeholder dimensions. The integration of traditional metrics (NDCG, Coverage, Diversity) with novel fairness indicators (Emerging Artist Hit Rate, Emerging Artist Exposure Index) provides a comprehensive assessment framework that could be adopted across the broader recommendation systems research community.

The optimization process itself—leveraging Bayesian optimization to balance six distinct metrics across two stakeholder groups—demonstrates the feasibility of principled multi-objective recommendation system design at scale. The fact that this optimization converged on interpretable, intuitively reasonable weight distributions strengthens confidence in both the methodology and the results.

## 10.5 Addressing the Fundamental Challenge

Ultimately, these results address the fundamental question posed by the Artist-Listener Paradox: whether music recommendation systems can serve multiple stakeholders without compromising their primary function. Our evaluation provides an unequivocal affirmative answer. Not only can recommendation systems balance listener satisfaction with artist fairness, but they can do so while improving performance on traditional metrics.

This finding has profound implications for the music industry's digital future. It suggests that the current prevalence of popularity bias in streaming platforms represents not an inevitable technical constraint, but a design choice—one that can be systematically altered through ethical algorithmic frameworks. The path toward a more sustainable, equitable digital music ecosystem is not only technically feasible but empirically validated, requiring only the will to prioritize long-term stakeholder balance over short-term engagement optimization.

The Artist-Listener Paradox, as demonstrated by our evaluation, is not a paradox at all—it is a solvable optimization problem that yields to principled, multi-objective algorithmic design.

# Bibliography


[ABM19] Himan Abdollahpouri, Robin Burke, and Bamshad Mobasher: *Managing Popularity Bias in Recommender Systems with Personalized Re-Ranking*. In *Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference*, 2019, pages 413–418.

[ABM17] Hojjat Abdollahpouri, Robin Burke, and Bamshad Mobasher: *Controlling popularity bias in learning-to-rank recommendation*. In *Proceedings of the 11th ACM Conference on Recommender Systems*, 2017, pages 42–46.

[AK11] Gediminas Adomavicius and Seung-Taek Kwon: *Toward a comprehensive framework for personalized recommendation systems*. In *User Modeling and User-Adapted Interaction*, volume 21 (4-5), 2011, pages 381–434.

[AT05] Gediminas Adomavicius and Alexander Tuzhilin: *Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions*. In *IEEE Transactions on Knowledge and Data Engineering*, volume 17 (6), 2005, pages 734–749.

[AC09] Deepak Agarwal and Bee-Chung Chen: *Regression-based latent factor models*. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009, pages 19–28.

[Agu18] Luis Aguiar: *Digital music consumption on the internet: Evidence from clickstream data*. In *Information Economics and Policy*, volume 44, 2018, pages 16–32.

[AP02] Jean-Julien Aucouturier and Fran3ois Pachet: *Finding songs that sound the same*. In *Proceedings of the 2002 IEEE Workshop on Neural Networks for Signal Processing*, 2002, pages 623–632.

[AP03] Jean-Julien Aucouturier and Fran3ois Pachet: *Representing musical genre: A state of the art*. In *Journal of New Music Research*, volume 32 (1), 2003, pages 83–93.

[BKL+12] Linas Baltrunas, Marius Kaminskas, Bernd Ludwig, Omar Moling, Francesco Ricci, Aykan Aydin, Karl-Heinz L"uke, and Roland Schwaiger: *Context-aware music recommendation based on latent topic sequential patterns*. In *Proceedings of the 6th ACM Conference on Recommender Systems*, 2012, pages 253–256.

[BS17] Solon Barocas and Andrew D Selbst: *Big data's disparate impact*. In *California Law Review*, volume 104 (3), 2017, pages 671–737.

[BEWL11] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere: *The Million Song Dataset*. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011, pages 626–629.

[BS20] Rodrigo Borges and Kostas Stefanidis: *On Measuring Popularity Bias in Collaborative Filtering Data*. In *CEUR Workshop Proceedings*. Volume 2578, 2020.

[BDDO04] J"urgen Branke, Kalyanmoy Deb, Holger Dierolf, and Malte Osswald: *Finding knees in multi-objective optimization*. In *Proceedings of the 8th International Conference on Parallel Problem Solving From Nature*, 2004, pages 722–731.

[Bur02] Robin Burke: *Hybrid recommender systems: Survey and experiments*. In *User Modeling and User-Adapted Interaction*, volume 12 (4), 2002, pages 331–370.

[BNL25] Robin Burke, Sunita Narayan, and Feng Li: *Multistakeholder evaluation framework for recommender systems*. In *ACM Transactions on Interactive Intelligent Systems*, volume 15 (1), 2025, pages 1–28.

[CVS08] Michael Casey, Federica Visi, and Malcolm Slaney: *Content-based music information retrieval: Current research and future directions*. In *IEEE Signal Processing Magazine*, volume 25 (4), 2008, pages 117–125.

[CH20] Samuel Caton and Christian Haas: *Fairness in machine learning: A survey*. In *arXiv preprint arXiv:2009.00976*, 2020.

[CKR+18] L. Elisa Celis, Varun Keswani, Simon Rangapuram, Karan Bhattacharya, Nisheeth Vishnoi, and Viraj Singh: *Ranking with fairness constraints*. In *Proceedings of the 2018 ACM SIGMOD International Conference on Management of Data*, 2018, pages 797–812.

[Cel08a] Óscar Celma: *Music recommendation and discovery: The long tail, long-term and cross-domain effects*. PhD thesis. Universitat Pompeu Fabra, 2008.

[Cel08b] Óscar Celma: *Music recommendation and discovery: The long tail, long-term and cross-domain effects*. In *PhD thesis, Universitat Pompeu Fabra*, 2008.

[CGM+99] Mark Claypool, Anuja Gokhale, Tim Miranda, Pavel Murnikov, Dmitry Netes, and Matthew Sartin: *Combining content-based and collaborative filters in an online newspaper*. In *Proceedings of the ACM SIGIR Workshop on Recommender Systems*, 1999, pages 1–10.

[CST19] Robert Collins, Chirag Shah, and Simon Taylor: *Blockchain for music rights management: A survey*. In *IEEE Access*, volume 7, 2019, pages 123621–123638.

[DDGR07] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram: *Google news personalization: Scalable online collaborative filtering*. In *Proceedings of the 16th International Conference on World Wide Web*, 2007, pages 271–280.

[DPAM02] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan: *A fast and elitist multiobjective genetic algorithm: NSGA-II*. In *IEEE Transactions on Evolutionary Computation*. Volume 6. volume 2, 2002, pages 182–197.

[DK04] Mukund Deshpande and George Karypis: *Item-based top-n recommendation algorithms*. In *ACM Transactions on Information Systems (TOIS)*, volume 22 (1), 2004, pages 143–177.

[DS14] Sander Dieleman and Benjamin Schrauwen: *End-to-end learning for music audio*. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pages 6964–6968.

[DB22] Karlijn Dinnissen and Christine Bauer: *Fairness in Music Recommender Systems: A Stakeholder-Centered Mini Review*. In *Frontiers in Big Data*, volume 5, 2022, pages 913608.

[DB23] Karlijn Dinnissen and Christine Bauer: *How Control and Transparency for Users Could Improve Artist Fairness in Music Recommender Systems*. In *Proceedings of the 24th International Society for Music Information Retrieval Conference*, 2023, pages 57–65.

[DBL25] Karlijn Dinnissen, Christine Bauer, and Miguel Lopez: *User perceptions of fairness interventions in music recommender systems*. In *Proceedings of the 28th International Conference on User Modeling, Adaptation and Personalization*, 2025, pages 45–54.

[DHP+12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S Zemel: *Fairness through awareness*. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, 2012, pages 214–226.

[EP21] Kristofer Erickson and Fernando Perez: *Platform economics and the independent musician*. In *Convergence*, volume 27 (4), 2021, pages 1132–1151.

[FFM+15] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian: *Certifying and removing disparate impact*. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pages 259–268.

[HPS16] Moritz Hardt, Eric Price, and Nati Srebro: *Equality of opportunity in supervised learning*. In *Advances in Neural Information Processing Systems*, volume 29, 2016, pages 3315–3323.

[HDW+18] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jiliang Tang, and Tat-Seng Chua: *Outer product-based neural collaborative filtering*. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, pages 2227–2233.

[HLZ+17] Xiangnan He, Lihan Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua: *Neural collaborative filtering*. In *Proceedings of the 26th International Conference on World Wide Web*, 2017, pages 173–182.

[HKBR99] Jonathan L Herlocker, Joseph A Konstan, Al Borchers, and John Riedl: *An algorithmic framework for performing collaborative filtering*. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, 1999, pages 230–237.

[HM18] David Hesmondhalgh and Leslie M Meier: *What the digitalisation of music tells us about capitalism, culture and the power of the information technology sector*. In *Information, Communication Society*, volume 21 (11), 2018, pages 1555–1570. DOI: 10.1080/1369118X.2017.1340498.

[HK16] Balázs Hidasi and Alexandros Karatzoglou: *GRU4Rec: Session-based Recommendations with Recurrent Neural Networks*. In *International Conference on Learning Representations (ICLR)*, 2016.

[HDE09] Xiao Hu, J Stephen Downie, and Andreas F Ehmann: *Lyrics-based audio content analysis for music emotion classification*. In *2009 IEEE International Symposium on Multimedia*, 2009, pages 635–642.

[HKV08] Yifan Hu, Yehuda Koren, and Chris Volinsky: *Collaborative filtering for implicit feedback datasets*. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, 2008, pages 263–272.

[Int23] International Federation of the Phonographic Industry (IFPI): *Global Music Report 2023: State of the Industry*. 2023. URL: https://www.ifpi.org/wp-content/uploads/2020/03/Global_Music_Report_2023_State_of_the_Industry.pdf.

[JA23] Dietmar Jannach and Himan Abdollahpouri: *What is Fair? Exploring the Artists' Perspective on the Fairness of Music Streaming Platforms*. In *User Modeling and User-Adapted Interaction*, 2023.

[JAGA22] Dietmar Jannach, Himan Abdollahpouri, Peter Geyer, and Gediminas Adomavicius: *Mitigating Popularity Bias in Recommendation: Potential and Limits of Calibration Approaches*. In *BIAS'21: Bias and Social Aspects in Search and Recommendation*, 2022, pages 91–106.

[JL17] Dietmar Jannach and Malte Ludewig: *Recurrent neural networks with top-k gains for session-based recommendations*. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017, pages 843–852.

[JWL23] Xiaolong Jin, Meng Wang, and Qiong Liu: *A survey of fairness-aware recommender systems: Trends and challenges*. In *ACM Computing Surveys*, volume 56 (4), 2023, pages 1–40.

[KC12] Faisal Kamiran and Toon Calders: *Data preprocessing techniques for classification without discrimination*. In *Knowledge and Information Systems*, volume 33 (1), 2012, pages 1–33.

[KSAS12] Takanori Kamishima, Jun Sakuma, Adrish Arya, and Hiroshi Sakagi: *Fairness-aware classifier with prejudice remover regularizer*. In *Proceedings of the 2012 European Conference on Machine Learning and Knowledge Discovery in Databases*, 2012, pages 35–50.

[KM18] Wang-Cheng Kang and Julian McAuley: *Self-attentive sequential recommendation*. In *2018 IEEE International Conference on Data Mining (ICDM)*, 2018, pages 197–206.

[KABO10] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver: *Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering*. In *Proceedings of the 4th ACM Conference on Recommender Systems*, 2010, pages 79–86.

[KNRW19] Michael Kearns, Subhya Neel, Aaron Roth, and Zhiwei Steven Wu: *Empirical risk minimization under fairness constraints*. In *Advances in Neural Information Processing Systems*, 2019, pages 2790–2801.

[Kor10] Yehuda Koren: *Collaborative filtering with temporal dynamics*. In *Communications of the ACM*, volume 53 (4), 2010, pages 89–97.

[KBV09] Yehuda Koren, Robert Bell, and Chris Volinsky: *Matrix factorization techniques for recommender systems*. In *Computer*, volume 42 (8), 2009, pages 30–37.

[KLL22] Dominik Kowald, Emanuel Lacic, and Elisabeth Lex: *Unfairness in active learning*. In *Machine Learning*, volume 111 (4), 2022, pages 1219–1246.

[Lam08] Paul Lamere: *Social tagging and music information retrieval*. In *Journal of New Music Research*, volume 37 (2), 2008, pages 101–114.

[LAP25] Rocıo Lara-Cabrera, Carlos Alvarez, and Lucia Perez: *Value-driven co-design of music recommender systems with stakeholders*. In *Proceedings of the 2025 ACM Conference on Recommender Systems*, 2025, pages 210–218.

[LS99] Daniel D Lee and H Sebastian Seung: *Learning the parts of objects by non-negative matrix factorization*. In *Nature*, volume 401 (6755), 1999, pages 788–791.

[LS08] Mark Levy and Mark Sandler: *Music information retrieval using social tags and audio*. In *IEEE Transactions on Multimedia*, volume 10 (8), 2008, pages 1214–1223.

[LCLS10] Lihong Li, Wei Chu, John Langford, and Robert E Schapire: *A contextual-bandit approach to personalized news recommendation*. In *Proceedings of the 19th International Conference on World Wide Web*, 2010, pages 661–670.

[LSY03] Greg Linden, Brent Smith, and Jeremy York: *Amazon.com recommendations: Item-to-item collaborative filtering*. In *IEEE Internet Computing*, volume 7 (1), 2003, pages 76–80.

[Log00] Beth Logan: *Mel frequency cepstral coefficients for music modeling*. In *International symposium on music information retrieval*, volume 270, 2000, pages 1–11.

[MAPM20] Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, and Bamshad Mobasher: *Feedback loop in recommender systems: Lessons learned and future directions*. In *ACM Computing Surveys (CSUR)*, volume 53 (5), 2020, pages 1–33.

[MRN25] Zuzana Marcinčáková, Yuan Ren, and Thanh Nguyen: *Trustworthy multi-stakeholder recommendation: A design framework*. In *User Modeling and User-Adapted Interaction*, volume 35 (2), 2025, pages 123–150.

[MGS24] Anna Matrosova, Pedro Garcia, and Jordan Smith: *Local vs. global bias in music recommendation: A geographic fairness perspective*. In *Proceedings of the 27th International Society for Music Information Retrieval Conference*, 2024, pages 312–320.

[MR09] Rudolf Mayer and Andreas Rauber: *On the music of semantic and phonetic similarities in song lyrics*. In *Information Sciences*, volume 179 (23), 2009, pages 4166–4175.

[MSN+18] Rishabh Mehrotra, Prashant Sharma, David Nemelka, Matthew Kay, Abhinav Kumar, and Mounia Lalmas: *Fairness in recommendation systems*. In *Proceedings of the 2018 ACM Conference on Fairness, Accountability, and Transparency (FAT*)*, 2018, pages 20–34.

[Mor15] Jeremy Wade Morris: *Artists as entrepreneurs, fans as workers*. In *Popular Music and Society*, volume 38 (3), 2015, pages 273–290.

[Par11] Eli Pariser: *The filter bubble: How the new personalized web is changing what we read and how we think*. Penguin Press, 2011.

[PT08] Yoon-Joo Park and Alexander Tuzhilin: *The long tail of recommender systems and how to leverage it*. In *Proceedings of the 2nd ACM Conference on Recommender Systems*, 2008, pages 11–18.

[PB07] Michael J Pazzani and Daniel Billsus: *Content-based recommendation systems*. In *The Adaptive Web*, 2007, pages 325–341.

[PVG+11] Fabian Pedregosa, Ga"el Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Max Perrot, and Öliver Duchesnay: *Scikit-learn: Machine Learning in Python*. https://scikit-learn.org. Scikit-learn developers, 2011.

[Pre20] Robert Prey: *Algorithmic promotion and the artist economy*. In *Social Media+ Society*, volume 6 (3), 2020.

[QCJ18] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach: *Sequence-aware recommender systems*. In *ACM Computing Surveys*, volume 51 (4), 2018, pages 1–36.

[RAC+02] Al Mamunur Rashid, Istvan Albert, Dan Cosley, Shyong K Lam, Sean M McNee, Joseph A Konstan, and John Riedl: *Getting to know you: Learning new user preferences in recommender systems*. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, 2002, pages 127–134.

[Ren12] Steffen Rendle: *Factorization machines with libfm*. In *ACM Transactions on Intelligent Systems and Technology (TIST)*, volume 3 (3), 2012, pages 1–22.

[Res22] Spotify Research: *Music Discovery Insights*. 2022.

[SKKR00] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl: *Analysis of recommendation algorithms for e-commerce*. In *Proceedings of the 2nd ACM conference on Electronic commerce*, 2000, pages 158–167.

[SKKR01] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl: *Item-based collaborative filtering recommendation algorithms*. In *Proceedings of the 10th International Conference on World Wide Web*, 2001, pages 285–295. DOI: 10.1145/371920.372071.

[SZC+18] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, and Mehdi Elahi: *Deep learning for music information retrieval: Recent developments and challenges*. In *Applied Sciences*, volume 8 (12), 2018, pages 2634.

[SPUP02] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock: *Methods and metrics for cold-start recommendations*. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2002, pages 253–260.

[SSS+16] Tobias Schnabel, Adithya Swaminathan, Ashudeep Singh, Navin Chandak, and Thorsten Joachims: *Recommendations as Treatments: Debiasing Learning and Evaluation*. In *International Conference on Machine Learning*, 2016, pages 1670–1679.

[SMSX15] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie: *AutoRec: Autoencoders meet collaborative filtering*. In *Proceedings of the 24th International Conference on World Wide Web*, 2015, pages 111–112.

[Sha21] Matthew Shapiro: *Musical diversity in streaming recommendations*. In *Journal of Cultural Analytics*, volume 6, 2021, pages 1–19.

[SM95] Upendra Shardanand and Pattie Maes: *Social information filtering: algorithms for automating "word of mouth"*. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995, pages 210–217.

[SC00] Barry Smyth and Paul Cotter: *A comparison of collaborative filtering and keyword-based recommendation algorithms for Internet applications*. In *ACM SIGIR Forum*, volume 34 (2), 2000, pages 23–35.

[TM07] Nava Tintarev and Judith Masthoff: *A survey of explanations in recommender systems*. In *Proceedings of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, 2007, pages 801–810.

[TC00] Tommy Tran and Robin Cohen: *Hybrid recommender systems for electronic commerce*. In *Proceedings of the National Conference on Artificial Intelligence*, 2000, pages 492–499.

[TC02] George Tzanetakis and Perry Cook: *Musical genre classification of audio signals*. In *IEEE Transactions on Speech and Audio Processing*, volume 10 (5), 2002, pages 293–302.

[UF98] Lyle H Ungar and Dean P Foster: *Clustering methods for collaborative filtering*. In *AAAI Workshop on Recommendation Systems*, 1998, pages 114–129.

[Val25] Andrea Valeri: *Algorithmic Fairness in Music Streaming Platforms: Who (or What) Determines the Success of Artists?* In *AMCIS 2025 Proceedings*, 2025.

[VDS13] Aaron Van den Oord, Sander Dieleman, and Benjamin Schrauwen: *Deep content-based music recommendation*. In *Advances in Neural Information Processing Systems*, 2013, pages 2643–2651.

[VS08] Aaron Van den Oord and Benjamin Schrauwen: *Learning for music recommendation*. In *Proceedings of the 2008 ACM Conference on Recommender Systems*, 2008, pages 257–264.

[WB11] Chong Wang and David M Blei: *Collaborative topic modeling for recommending scientific articles*. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2011, pages 448–456.

[WWY15] Hao Wang, Naiyan Wang, and Dit-Yan Yeung: *Collaborative deep learning for recommender systems*. In *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pages 1235–1244.

[Wik13] Patrik Wikstr"om: *The music industry: Music in the cloud*. John Wiley Sons, 2013.

[YH17] Sirui Yao and Bert Huang: *Beyond Parity: Fairness Objectives for Collaborative Filtering*. In *Advances in Neural Information Processing Systems*, 2017, pages 2921–2930.

[ZVGG17] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P. Gummadi: *Fairness constraints: Mechanisms for fair classification*. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017, pages 962–970.

[ZZC+23] Yifei Zhang, Hao Zhu, Yankai Chen, Zixing Song, Piotr Koniusz, and Irwin King: *Mitigating the Popularity Bias of Graph Collaborative Filtering: A Dimensional Collapse Perspective*. In *Advances in Neural Information Processing Systems*. Volume 36, 2023.

[ZJW+21] Zeyuan Zheng, Xu Jiang, Rui-Shi Wang, Ming Li, and Xiaochun Sun: *A survey of multi-objective optimization in machine learning*. In *ACM Computing Surveys*, volume 54 (6), 2021, pages 1–36.

[ZWSP10] Tian Zhou, David Wilkinson, Robert Schreiber, and Rongming Pan: *Solving the cold start problem in recommendation systems: A survey*. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, 2010, pages 1146–1153.

[ZHZ+21] Ziwei Zhu, Yun He, Xing Zhao, Yin Zhang, Jianling Wang, and James Caverlee: *Popularity-Opportunity Bias in Collaborative Filtering*. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining*, 2021, pages 85–93.