POLITECNICO DI TORINO

MASTER's Degree in Mechatronic Engineering

Fairness in Music Recommendation Systems

Supervisors
Prof. Cristina Emma Margherita ROTTONDI
Prof. Massimiliano ZANONI

Candidate
Aliasghar ERTEGHAEIAN

July 2026

Abstract

The advent of online streaming services has revolutionized the entertainment consumption, in more ways than one. The availability of thousands of hours of entertainment, however, came with caveats, as it created an overwhelming experience for the user to navigate through and discover the most relevant content. Thus, the significant shift from analog to digital has made the presence of recommender algorithms an urgent aspect of the modern media landscape. Online entertainment platforms were forced to adopt automated mechanisms to offer customized curation to users, in the most efficient manner. From online TV and movie providers such as Netflix and Amazon Prime to video sharing services such as YouTube, to online music services on world-leading platforms such as Spotify and Apple Music, the use of recommender tools has become an industry standard.

Since the recommendation methodologies can vary widely across different entertainment sectors, this study aims specifically at the recommendation practice in a music streaming service. The rise in implementation of recommender algorithms in the music sphere brings an ethical discourse into studying the underlying mechanisms of the recommender tool. While the ideal scenario would be to lead the user towards the recommendations in the most "organic" manner, in reality, the user cannot be the only benefactor of the streaming platform, as there are also artists that demand exposure to a wider audience, as well as the platform itself that seeks to ensure it sustains a steady revenue under a given recommendation model.

This study aims to explore the disparity of interests by offering the concept of fairness in recommendation. While avoiding the philosophical interpretation of fairness as a topic of debate, the goal is to categorize and represent factors that contribute to the fairness of a music recommender system. In particular, three diffusion-based recommender models and one hybrid model were studied, and their performance analyzed in terms of the influence of user feature data in recommendation behavior. The analysis was done using Python language, and conducting XAI analysis through Shapely (SHAP).

Further research could provide the missing argument of ethical reasoning, and an in-depth philosophical debate over the feasibility of approaching fairness given the provided analysis.

Table of Contents

Acronyms

AI	Artificial Intelligence
ALS	Alternating Least-Squares
BPR	Bayesian Personalized Ranking
CARS	Context-Aware Recommender System
CBF	Content-Based Filtering
CDiff4Rec	Collaborative Diffusion model for Recommender [System]
DiffNet	[Influence] Diffusion neural network
DM	Diffusion Model
DR	Diffusion Recommender
EDiffuRec	Enhanced Diffusion Model [for Sequential] Recommendation
FFN	Feed Forward Network
GAN	Generative Adversarial Network
GCN	Graph Convolutional Networks
GDPR	General Data Protection Regulation
HR	Hit Rate

ItemKNN	Item K-Nearest Neighbors
LR	Learning Rate
MF	Matrix Factorization
Mostra	Multi-Objective Set Transformer
MRS	Music Recommender System
MSE	Mean Square Error
VAE	Variational Autoencoders
NDCG	Normalized Discounted Cumulative Gain
POP	Popular Items
ReLU	Rectified Linear Unit
SLIM	Sparse Linear Models
SVD	Singular Value Decomposition
TF-IDF	Term Frequency–Inverse Document Frequency
VAE	Variational Autoencoders
XAI	Explainable Artificial Intelligence

Chapter 1

Introduction

1.1 Context and Motivation

During their short history of nearly two decades, recommender systems have experienced a rapid rise. Their emergence in the mid-2000s was primarily centered around collaborative and content-based filtering. Later, methods to fuse the mentioned filtering means with early forms of deep learning were the main focus of the studies. Deep neural network models were introduced in the late 2010s, enabling a more personalized and context-sensitive recommender model. Later, integration was expanded to merge in large language models in early 2020s to leverage real-time user feedback. [1]

The widespread adoption of recommendation models in recent years introduced their utility into sensitive and high stakes areas. Consequently, this latest evolution necessitates the presence of constant reporting on model performance. In an attempt to offer explainability to a black-box model for neural networks, explaining AI models can be done either through transparency design or post-hoc explanation. According to [2]:

> "The transparency design reveals how a model functions, in the view of developers. It tries to (a) understand model structure, e.g., the construction of a decision tree; (b) understand single components, e.g., a parameter in logistic regression; (c) understand training algorithms, e.g., solution seeking in a convex optimization. The post-hoc explanation explains why a result is

> inferred, in the view of users. It tries to (d) give analytic statements, e.g. why a good is recommended in a shopping website; (e) give visualizations, e.g. saliency map is used to show pixel importance in a result of object classification; (f) give explanations, e.g. K-nearest-neighbors in historical dataset are used to support current results."

As a recipient of AI's decisions, it's necessary to know how those decisions are made. It could be also grounds for a developer to enhance the AI algorithm.

The emergence of Industry 4.0 coupled with the growing emphasis on explainability has brought Explainable AI, highlighting the increasing importance of transparency [1]. Evidenced by deep investments by DARPA and publishing of GDPR by the European parliament, the attention on Explainable AI has been increasing over the past years. [2]

1.2 Problem Definition

There's always potential for the recommender system to lean towards certain biases, be it the result of a specific design of the system, or the way the system's parameters are adjusted. It's an analytical challenge to explain the black-box model to uncover the system's "intentions" in offering a recommendation. Specifically, this study aims at the issue of fairness in music recommenders, from a feature selectivity standpoint. The problem of the dataset's framing, and how the sparsity of the features can affect the system output is also addressed.

1.3 Objectives of the Thesis

The emergence of recommender systems in the context of music streaming services requires a deeper understanding of the underlying mechanisms of the recommender algorithm, as means to assess its utility in this sphere. This objective

is achieved by perusing the previous research done in this area, as well as a real-time analysis of three state-of-the-art recommendation practices using diffusion models, and a hybrid method. It's the ultimate goal to unravel the issue from multiple aspects, to enable a philosophical discourse over the critical steps to be taken, and to analyze the instances involving presence of "fairness" as an ethical standard, in music recommender systems.

Chapter 2

Background and Related Studies

2.1 Overview of Fairness in Music Recommender Algorithms

In the context of a recommender, it's largely agreed that three stakeholders are distinguishable: the users, the platforms, and the item providers. A music platform aims at maximizing the engagement of the users and artists (or labels), while also benefitting from retaining the parties mentioned, i.e. a successful match. Further, the platform needs to ensure it considers the rules and regulations by the governments and interest groups. Hence, there is a need for so-called "multi-stakeholder" research done to address the different stakeholders' interests. [3]

On average, online music consumers are shown to have a more diversified taste in the music they listen to, compared to radio and TV audience, thanks to the implementation of personalized recommendations. However, it's shown that users who have a more diversified music taste tend to follow the algorithm less often. The negative effect is further demonstrated by considering the unfavorable public opinion towards systems when compared to a human judge. The inability to provide diverse recommendations may harm the introduction of new artists to users. [4][5]

Music domain in general is characterized by an inclination towards popularity. While it cannot be necessarily deemed as detrimental as it signifies commonality of interests, it can negatively affect a number of minority user groups when the same logic is adopted by the algorithm. Users having a more "mainstream" taste would

get more relevant recommendations, whereas a user with a more specific taste in music or in a less commonly spoken language would receive lower quality recommendation. A similar treatment is applied to item providers as well, where larger group size leads to receiving more exposure to the artist. Artists from less represented groups, e.g. female artists in a platform's database, experience lower exposure, whereas the ones from more represented groups are highlighted. [3]

However, the imbalance in stream share doesn't seem to stem from a position of gender suppression against female artists. There hasn't been a major difference reported between the male/female or different age users' share of streaming female artists. Moreover, being insufficiently represented doesn't only come from gender inequality, since there's also the size of artist's catalogue and their country of origin that affect their exposure. [4]

Fairness goals are mainly set for users and not artists in literature [3]. In limited instances, e.g. where artists are interviewed on the issue, there's a demand for nurturing diversity to promote female artists to a higher extent [5]. Another instance of artists being unfairly treated comes from the widespread use of artificial intelligence to generate musical works. Musicians argue that the role of an AI musician is relegated to a music editor rather than composer, merely shaping the already existing body of works by human artists into a new form [6].

Artists have limited interaction with the platform they work with, yet they are widely affected by the platform's design. Platforms are largely seen as the party responsible for advocating new and upcoming artists, with power to harm the music culture when said promotion is done ineffectively. In particular, artists demand being adequately represented by the system. Another system downside mentioned by the artists is not being represented by their latest work, but rather their most played album/song, which artists view as a downside. [5]

In some cases, playlists can be detrimental to a music track when taken out of its context, and placed in a way which may be particularly harmful to or undesired by the creator. Artists may find their work on a shuffled session alongside artists that they don't share the same ideology with. Also, new and emerging artists face an inherent cold-start disadvantage, where their work isn't recommended as relevant material to a broader audience. The power should be given to the parties involved in a curation scenario to veto the decisions made, in case such issues arise. [5][6]

Some believe that the artist's catalogue size may be an indicator of how prolific the artist is, with more work providing more ample opportunities for the artist to be recommended by the platform. Others, however, view providing this opportunity coming from a misunderstanding of an artist's merits, which shouldn't be considered as grounds for elevated exposure. They claim that there exist numerous cases where a prominent and influential artist produced limited musical content throughout their career, who would be wrongfully cast aside as less relevant in this type of assessment.

Moreover, being from a region with smaller user numbers can too, put the artist at an inherent disadvantage. Artists from these lower populated regions would be recommended less frequently, compared to artists from regions with larger user size. Artists argue that putting a higher weight on recommending the users from their pool of local artists may alleviate the issue. [5]

Methods such as re-ranking and gradually increasing artist exposure for minority groups have been suggested to overcome the underrepresentation issues [3]. For gender, it's advised to include ones outside of the traditional binary male/female assignment, and discourage the system from leaning towards majority gender groups. Several debiasing approaches in favor of minority groups have been proposed in literature, the main ones being rebalancing, regularization, counterfactual intervention, and adversarial training. The approach may be a pre-

or post-processing step, with a common pre-processing practice being the relabeling of training data to reach equal number of relevant group-wide labels, while post-processing being concerned with restructuring the output list to reach fairness standards. Also, regulatory intervention needs to be incorporated to ensure diversity of content and sources, in light of biases and mis or under-representation in some areas of content. [6][7]

On a higher level, the balance is struck between the user and item provider satisfaction through requiring a tradeoff. There are a number of multi-stakeholder tradeoff schemes presented in literature, such as counterfactual estimation [8] framework to balance fairness with user relevance and contextual bandits [9] to fairly optimize multiple objectives. It's also possible to enhance the data structure to prevent discrimination towards certain user and artist demographics. In terms of data availability, there aren't many accessible rich public music datasets, and the available ones don't provide descriptive data beyond age, (binary) gender, or ethnicity. While there's a valid concern to account for the user privacy in sharing data, to achieve meaningful fairness improvements, there's a need for wider access to more descriptive and detailed music datasets. [3]

In particular, to set a fair tradeoff balance, the study by Bugliarello et.al. [10] proposes Multi-Objective Set Transformer model (Mostra). It enables system designers to balance multiple objectives and dynamically control a singular objective to satisfy others, in a multi-objective music sequencing scheme. Mostra benefits from both input and output set-awareness. It encodes a set of music tracks simultaneously, and decodes them while ensuring that the goal of multi-objective diversity is satisfied. There are four objectives considered to be the focal point of this multi-dimensional study:

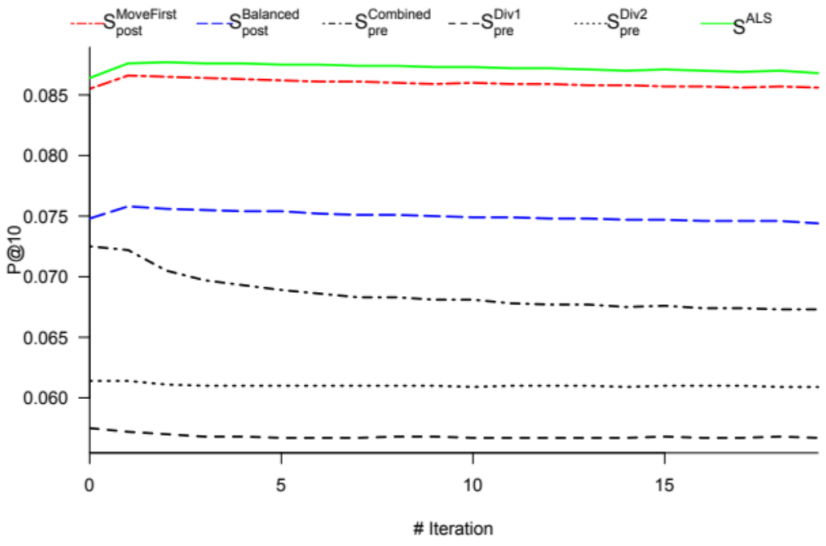
I. **Short-term satisfaction:** Is a metric evaluated from the user interaction with a certain track, whether they skip or stream it. It serves as a proxy for user satisfaction, and is denoted as SAT.
II. **Music discovery:** Involves mainly the platform's role in suggesting new music to the user. It's proven to be the main motivator for the user to continue their subscription. However, it's also statistically shown that increasing the number of discovery tracks would reduce user satisfaction, so there needs to be a tradeoff.
III. **Emerging artist exposure:** Explores the platform's capability in offering the work of emerging, lesser-known artists to users when making recommendation decisions.
IV. **Content Boosting:** Consists of every other instance in which the platform boosts content, e.g. because of a real-world event taking place, or to celebrate and/or honor a certain artist.

Because of the existence of competing interests when aggregating objectives, set awareness is deployed, which adaptively searches for better solutions based on objective composition of specific input sets. Mostra's multivariate scoring captures the local context and cross-track interaction effects, that goes against the traditional simple item-level neural architecture. Moreover, counterfactual scoring gives Mostra the ability to control the tradeoff across objectives. [10]

It should be noted that different criteria showcase the quality of recommendation in unique ways. In algorithmic sense, fidelity of the results to the input serves as the quality metric, e.g. when searching for the term "Kendrick", a high-quality algorithm shows specifically the tracks, albums, and overall discography of Kendrick Lamar, rather than, say, Anna Kendrick's song for the Trolls soundtrack. A user-centric recommender, on the other hand, prioritizes the user's enjoyment as the point of focus, showing results that reach the best balance between new and familiar content to the user, to match their moods or emotions. [6]

2.2 Advantages and Challenges of Feature-Based Recommendation

Although modern recommender systems overwhelmingly outperform old methods of curation, they are still prone to similar shortcomings. An MRS may amplify the existing discrepancies, in a phenomenon referred to as “compounding imbalances”. However, by applying debiasing methods, it has the potential to improve recommendation fairness [7]. In a study by Bauer et.al [11], using post-processing strategies for ALS algorithm, it’s shown that it is possible to positively affect reaching gender balance by re-ordering the items in favor of minority gender groups. It enables steady featuring of minority groups in higher positions in recommendation. However, this comes at a slightly lower performance cost for a number of performance metrics like accuracy, as can be seen in Fig. 2.1 for $S_{post}^{\text{Move First}}$ which positions top-ranking recommendation of a woman as first.



[continued]

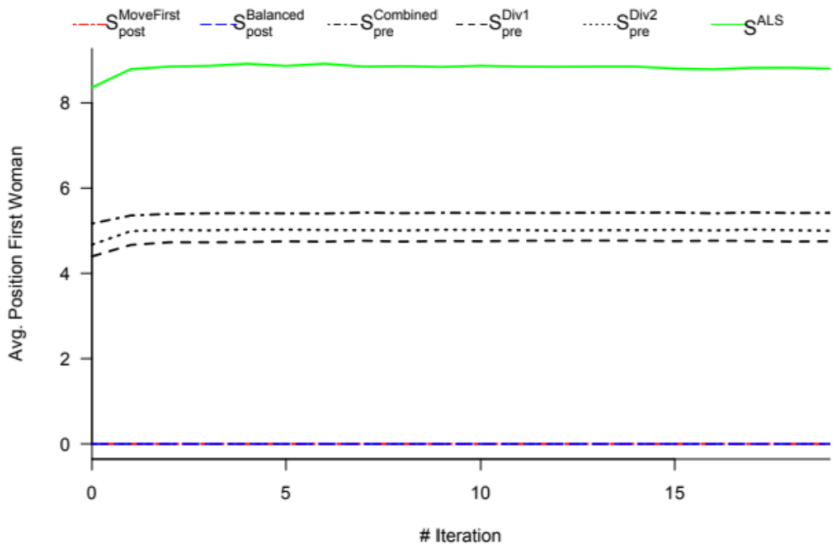


Figure 2.1 Top 10 recommendations precision plot for all artists (top), Average position of the highest-ranking item representing a woman (bottom) [11]

This study makes use of a small dataset and follows a simple-choice model as opposed to the complexity of people's choices in real life. Still, it can successfully demonstrate how it's possible to factor in a higher variety in recommendation, with a marginal impact on performance. [11]

Several models directly benefit from feature-based recommendations. CBF systems rely on content descriptors such as tempo, rhythm patterns, or genre information to identify items similar to that the user already listened to. CARS relies on contextual information such as time, location, or activity of the user to serve as basis for recommendations. There exist also hybrid models, which aggregate or fuse two or more individual approaches into a unitary solution. Models have also been developed to address specific points of bias in literature. In particular, fairness-aware algorithm points out neural networks' gender bias towards male in retrieval results. [7]

As already discussed, while it's possible for the recommender to offer compensation to less represented groups, it can also amplify the existing biases. To investigate this, The study by Ferraro et.al. [12] demonstrates how popular artists and tags (including genre, mood, and era) are recommended more than what the user already consumes, in collaborative filtering-based recommenders. Another study suggests that in view of users, an inclination towards long-tail items can harm the quality of recommendation received, in terms of accuracy and calibration [13]. The creation of a feedback loop can further exacerbate the issue. By dropping the coverage of a majority of the songs, the system relies on the user's history to recommend a lower percentage of the whole songs count. User's choice from the limited batch serves as the input to the next medley of recommendations, leaving the system in a state of perpetually recommending increasingly similar items. [14]

2.3 Explainability in MRS

It's essential to know the distinct characteristics of the music recommendation compared to recommendation in other contexts. Firstly, music can be consumed in shorter sessions typically in form of songs, compared to movies or books. Secondly, the context is shown to strongly affect the user's consumption habits. Depending on the user's mood, location, time of day, or being alone or with friends, their music preferences may vary widely. Lastly, music tracks are typically listened to in succession in form of playlists or listening sessions, which shape users' short and long-term preferences. [15]

Unsurprisingly, music preferences in individuals may be sought after in their emotions and moods. Music consumption is shown to have a strong social element, with research suggesting music recommendation system to closely replicate the kind of recommendation that a friend might provide. Moreover, the discovery of new music among listeners is tied to providing opportunities for individuals to

bond and form new friendships. Peers can also directly shape a user's distaste for, or inclination towards certain music. A user may decide on taking a recommendation on the aesthetic aspects, song title/lyrics, or their previous experience with the artist/song/genre. Also, the reception might be based on spontaneous decisions such as a 10-20 second snippet of the song, or the user's proclivity towards taking a recommendation at the moment of receiving it.

Users believe that some context might be lost in recommendation, so that an RS might not know an artist's troubled behavior outside of the music sphere or their toxic fanbase. Users have expressed interest in understanding why a certain recommendation was made to them, while embracing the possibility to customize and change parameters to receive more personalized recommendations. [16]

One challenge to explainability in MRS domain is the absence of any ground truth. This can cause issues in assessing the explanation quality, detecting misinterpretations, and comparing different XAI systems. The concept of "incompleteness" was proposed to justify the use of an explanation system as the missing piece. Under this concept, the explanations may not be needed if the issue of incompleteness doesn't exist. Accounting for a target demographic, Understandability is the notion that simply bridges the gap between a select XAI system and this new concept of incompleteness. [15]

Two types of explanation can be considered for an MRS to be studied: a local and a global explanation [17]. The local explanation aims at uncovering the reason behind a certain recommendation to a user, which in accordance with GDPR*, provides the user's "Right to Explanation". On the other hand, the global explanation clarifies the system's intentions in a broad sense, showing the system's biases and highlighting its performance on a large scale. There could be links established between the two, and the global explanation can be constructed from several instances of local explanations. [15][18]

*Parliament, E. U. (2016). Regulation (eu) 2016/679.

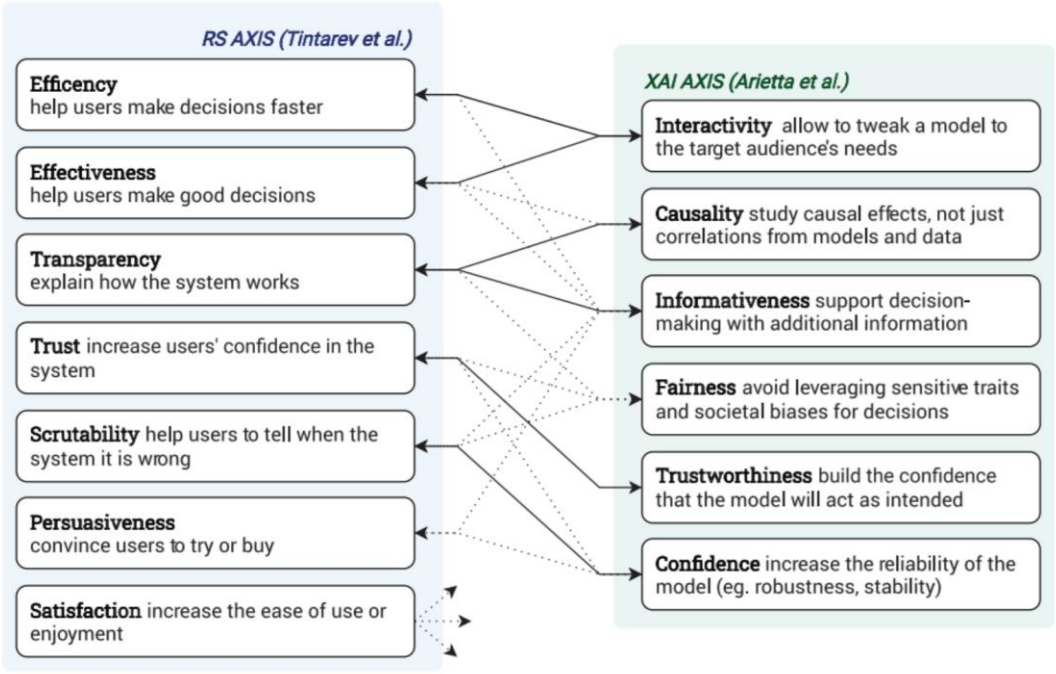


Figure 2.2 Links between RS and XAI goals are shown as a solid line (strong correspondence) and dotted lines (weak link that depends on context), with satisfaction being linked to all [15]

The explanation can be done on the inner workings of the system, which should be embedded into the system in advance. Being inherent to the model means the explanation coincides with the recommender system, whereas post-hoc analysis treats the recommender as a black-box model. It investigates the specific designation of the features and how that led to the formation of a set of recommendations by the system. Both are valid methods for explaining the system, although each choice introduces its own advantages and drawbacks. While the post-hoc analysis doesn't clarify a system's inner processing on par with intrinsic method, it can be applied to any existing recommender system without the need for prior integration. However, it requires extra safety measures to ensure faithfulness of the explanation to the model.

An MRS explanation is conducted by relying on various data sources, with existent attempts at leveraging information sources such as audio attributes, users' social information such as their friends, and tags describing items or users. Features are categorized as multi-modal information, which render the personalized recommendation possible. Since an extensive set of features may be available, it's essential that we identify a compact subset of features that are the most relevant, when creating the recommendations. Moreover, interpretability of the feature set determines the relevancy of a given feature-based explanation. [15]

In terms of increasing the understandability of the results from explanations, visual [19] or social (sentence-based) [20] explanations can be exercised. There should be a balance in providing details, as an information overload may have a negative impact on the explanation understandability. Furthermore, in case of a user study, it is important to determine the type of measurement that is employed, with the common types in MRS being interaction count (i.e. number of songs played), surveys, or semi-structured interviews [21]. The choice depends on which specific explanation goal is concerned, e.g. if interactivity and efficiency are concerned, play count is more relevant. From the perspective of engineers and data analysts, or technical stakeholders in general, there are considerations for interpretability that minimize misinterpretations. These considerations include robustness to small changes [22], consistency across similar models [23], and sparsity [24].

2.4 LastFM-360k

This music dataset, which is used throughout this study, consists of <user, artist, plays> tuples collected from Last.fm API* using the user.getTopArtists() method [25]. It contains 17,559,530 entries and 359,347 unique users. The dataset also includes some information that is irrelevant to the current study and needs to be stripped from the set, such as user signup date. Also, there exist logs that might lack one or

*http://www.last.fm/api

more relevant user or item information. As the base filtration process, these data points are eliminated from the set. Any additional preprocessing step is carried out depending on the specific model and its input requirements.

2.5 Optuna Optimizer

This hyperparameter optimization software [26] follows define-by-run principle. It lets users to dynamically construct parameter search space, and allows optimization to be carried out without having to meticulously plan everything in advance about the optimization strategy. Various tasks are supported, ranging from lightweight experiments to distributed heavy-weight computations.

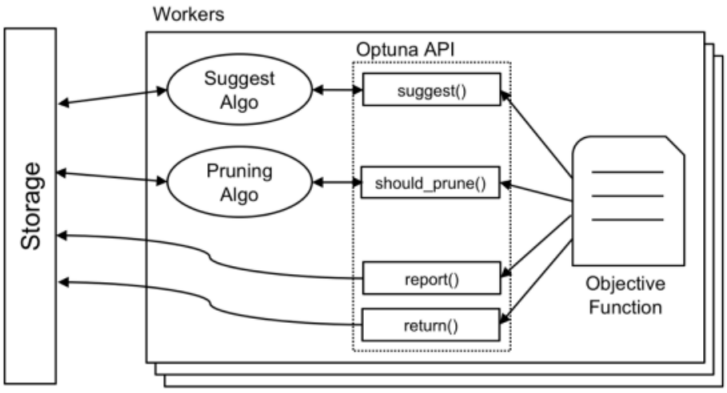


Figure 2.3 Overview of Optuna's system design [26]

Jobs are delegated to "workers", each executing an instance of objective function. They report their result of the study to the storage. Each trial is run by Optuna APIs, that retrieve information from previous studies upon invocation. The cost-effectiveness of the process is guaranteed by a combination of efficient search and pruning algorithms. It will be used on instances in the study to optimize the model parameters and search for the best possible combination.

Chapter 3

Recommender Models

3.1 LightFM

To address recommendation difficulties in absence of ample interaction data, LightFM [27] aims to fill the sparsity gap in a cold-start scenario. Similar to a collaborative filtering model, the users and items are realized as latent vectors. However, the novelty in LightFM's approach is that it integrates the advantages of content-based and collaborative recommenders. The model has the capability to use both user and item metadata, which enables it to operate under both item and user cold-start scenarios.

Interactions are either interpreted as negative S^- or positive S^+, together unionizing to form the set of interaction pairs $(u, i) \in U \times I$. Each user u or item i is defined by a set of features, $f_u \subset F^U$ and $f_i \subset F^I$ respectively. For each feature f, the model is parametrized in terms of d-dimensional user and item feature embeddings e_f^U and e_f^I. The latent representation of each user u and each item i is determined by the sum of its features' latent vectors:

$$q_u = \sum_{j \in f_u} e_j^U, \qquad p_i = \sum_{j \in f_i} e_j^I \tag{3.1}$$

Which would form the prediction for user u and item i, that is the dot product of user and item representations, adjusted by their feature biases:

$$\hat{r}_{ui} = f(q_u \cdot p_i + b_u + b_i) \tag{3.2}$$

Since binary data is considered in the prediction, sigmoid function is used as $f(\cdot)$ in this study. As optimization objective, the likelihood of the data conditional on the parameters needs to be maximized, with likelihood defined as:

$$L(e^U, e^I, b^U, b^I) = \prod_{(u,i)\in S^+} \hat{r}_{ui} \times \prod_{(u,i)\in S^-} (1 - \hat{r}_{ui}) \tag{3.3}$$

The feature embeddings in LightFM are estimated by factorizing the collaborative interaction matrix; Which is dissimilar to content-based systems, that factorize pure content co-occurrence matrices when dimensionality reduction is used. The adaptability of the model to both ends of the data density spectrum is a promising indication that it has the capability to perform well at different data sparsity levels.

If there exist only indicator features in the feature set, LightFM's performance resembles a standard Matrix Factorization model. In case of feature sets also containing shared metadata features by more than one user or item, LightFM expands the MF model through feature latent factors explaining in part the structure of user interactions. Simulation results demonstrate LightFM's performance is promising in cold-start scenarios; It's on par with pure content-based models on a low-density dataset, and in presence of abundant collaborative data and availability of user metadata, it outcompetes the existing content-based and hybrid models.

3.2 Diffusion Models: A Novel Approach

Several algorithms have been proposed in literature in the context of music recommendation. Some common algorithms based on the user-item Boolean interaction matrix are:

- **POP**: Recommends the top popular items to all users

- **ItemKNN**: Recommends new items based on item similarity to previously interacted items
- **ALS:** Approximates the original user-item matrix through dot product of the user-item embeddings
- **BPR:** Focusing on the items that are interacted with, it ranks them according to user preference
- **SLIM:** By sparsely aggregating past user interactions, it aims at predicting the user's recommended items

There are numerous performance metrics which help with evaluating the performance of the algorithms mentioned. Recall@k quantifies the ability to retrieve relevant items, and NDCG@k in addition, also assesses the ability to rank the mentioned items. Coverage@k defines the fraction of tracks that appear in the top-k list of at least one user. Typically, different numbers for k top recommendations considered, include 5, 10, 50, and 100 tracks, with 10 being of particular interest as it's the number of top recommendations displayed by default to a user on last.fm (which is the source for dataset of choice in the current study).

Analyses on the algorithms mentioned show SLIM being the most unfair, although performing superior on accuracy and diversity metrics. It can be also seen as a persistent trend, with accuracy having an inverse relationship with fairness across all of the algorithms. Thus, increasing the gap by improving the majority group is seen as a way to gain higher performance. In general, except POP, the majority of algorithms compound existing biases. In addition to debiasing methods, there's the possibility to explore other types of algorithms. [7]

One interesting alternative solution is to use generative models such as variational autoencoders (VAEs) and generative adversarial networks (GANs). Compared to traditional methodologies such as collaborative filtering, they mitigate the adversarial effects of noisy data, weak latent representations, and inadequate

collaborative signals. By learning non-linear probabilistic latent space, self-supervising aspect helps to capture precisely the ever-evolving preferences of users, which is a desired behavior in the context of music recommendation.

Diffusion models, as a recent class of generative algorithms, offer distinguished generative capability, superior representation learning, and flexible backbone structure. They achieve close alignment with the distribution of training data, by leveraging a denoising framework that works to reverse the multi-step noising process to generate synthetic data. However, a drawback associated with these models may be poor explainability because of their black-box nature, and stochastic sampling of diffusion-denoising process. Also, since these models are mainly trained on copyrighted materials, there's a risk of infringement if not asked for permission. [28]

To explore the current diffusion model landscape, the next sections will discuss three diffusion model examples used in the context of recommendation. Further, a thorough assessment will be made to determine their viability, by exploring their algorithms and evaluating their performance in the context of music recommendation.

3.2a Enhanced Diffusion for Sequential Recommendation

This model being proposed by Lee et.al. [29] is an innovative sequential recommendation diffusion model designed for real-life-oriented online commerce. Also referred to as EDiffuRec, it integrates a number of key improvements to the non-enhanced model, including an altered transformer structure derived from the Macaron Net, a normalized loss function, a novel noise assumption based on the Weibull distribution, and making use of warmup techniques.

Sequential recommendations that are able to reflect the ever-changing preferences and current interests of users over time can generate personalized recommendations that elevate user satisfaction. Generative recommendation models are able to generate probabilities for items that lack interactions, when interaction information is missing and it's possible to obtain useful information by learning latent features in case of noisy data. This can be interpreted as a multi-class classification problem, since a target item is inferred from a number of items. Hence, the cross-entropy loss is utilized as follows:

$$\hat{y} = \frac{\exp(\|\hat{x}_0\|.e_{t+1})}{\sum_{i\epsilon I}\exp(\|\hat{x}_0\|.e_i)}, \qquad \mathcal{L}_{CE} = \frac{1}{|U|}\sum_{i\epsilon U} - \log \hat{y}_i \tag{3.4}$$

Where set of items and set of users are denoted by "I" and "U", respectively. When L2-normalization to $\hat{x}_0$ is applied (denoted as $\|\hat{x}_0\|$), it results in the vectors having the same range. This makes the comparison more straightforward, as a more consistent assessment is possible when the recommendation scores are based on the directionality of the embedding values. This would lead to higher stability and performance of the model.

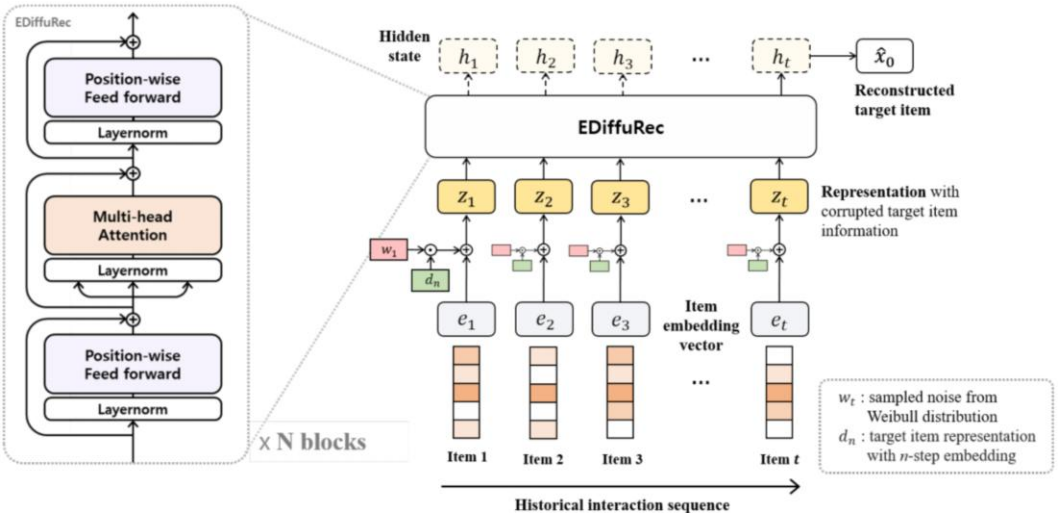


Figure 3.1 Architecture of the proposed EDiffuRec [29]

The training process is shown in Fig. 3.1, showing representation z_t as input to the approximator being calculated by adding up embedded item vector e_t with element-wise multiplication of Weibull distribution-driven noise sample w_t and d_n which is obtained by summing up positional embeddings and x_n at step n. The asymmetric nature of Weibull distribution as the noise, mitigates the impact of noise or outliers. Therefore, the use of Weibull distribution provides higher accuracy in modeling user behavior characteristics, enabling personalized recommendations, and is well-adjusted to noisy or asymmetric real-world data.

By incorporating L2 norm in the loss, the proposed model starts off the training with a substantially lower loss. Stable learning results in EDiffuRec converging in fewer epochs, making comparable net training time and enhanced final performance. The improvement is evidenced by “enhancing performance across all metrics for the Tools and Video datasets”. By using sampled noise from the Weibull distribution, the model excels “in the NDCG metric for the Beauty dataset and all metrics for the Tools dataset”. EDiffuRec manages to outperform baselines across all datasets and metrics, by effectively combining these components.

Using noise distribution, which has higher degrees of freedom, can generally lead to performance improvements. However, warmup duration can cause performance to fluctuate, which necessitates careful selection of duration to suit the data characteristics. Also, contrary to the traditional transformer encoder structure, adding an extra FFN layer leads to a thorough learning of user behavior patterns rooted in sequential information.

The numerical instability of the training weights is compensated when the starting phases of training are done with a sufficiently low learning rate, enabling the model to achieve a lower loss by ruling out local optima and converging towards a global optimum.

3.2b Neural Influence Diffusion for Social Recommendation

In a social platform, scientists have agreed on social influence being a natural process for users to disseminate their preferences to the followers in the network, so that the status (preference, interest) of a user changes with the influence from his/her trusted network. The first diffusion iteration starts off the social influence propagation process, with the corresponding latent embedding of each user being affected by the early embedding of their trusted connections.

Existing solutions discarded the iterative social diffusion process for social recommendation. However, the study by Wu et.al. [30] highlights recursive nature of the social influence instead of static, with each user being progressively influenced by the social connections.

Labelled as DiffNet, the model "is an influence diffusion neural network-based model". It facilitates user and item embedding modeling in social recommendation by triggering the recursive social influence propagation process (it models the evolution of users' latent embeddings with the social diffusion process).
DiffNet contains four main parts:

I. **Embedding layer:** Outputs free users and items embeddings

II. **Fusion layer:** Generates a hybrid user (item) embedding by fusing both a user's (an item's) free embedding and the associated features. The input to the fusion layer for each user a consists of free embeddings p_a and associated feature vector x_a. In turn, it produces user fusion embedding h_a^0 as output, that seizes the user's initial preferences from various forms of input data. The fusion layer is modelled as a single-layer wholly connected neural network in Eq. 3.5:

$$h_a^0 = g(W^0 \times [x_a, p_a]) \tag{3.5}$$

With W^0 being a transformation matrix, and g(x) as a non-linear function.

III. **Layer-wise influence diffusion layers:** Built with a layer-wise structure to model the recursive social diffusion process in the social network, which is the key idea of DiffNet. As information diffusion occurs in the social network, the multi-layer framework of influence diffusion segment is analogously constructed.

Specifically, two steps construct the updated embedding h_a^{k+1}:

A. Diffusion influence aggregation (AGG) remodels all the social trusted users' influences into a fixed length vector h_{Sa}^{k+1} from a's trusted users from the k-th layer:

$$h_{Sa}^{k+1} = Pool\left(h_b^k \middle| b \epsilon S_a\right) \tag{3.6}$$

where the Pool function's possibility is to be the average of all the trusted users' latent embeddings at the k-th layer. The a-th updated embedding h_a^{k+1} is a fusion of user's k-th layer latent embedding h_a^k and aggregation of influence diffusion embedding h_{Sa}^{k+1} from her trusted users.

B. Since it is unclear how each user balances these two parts, a non-linear neural network is used to replicate the combination as:

$$h_a^{k+1} = s^{(k+1)}(W^k \times [h_{Sa}^{k+1}, h_a^k]) \tag{3.7}$$

with $s^k(x)$ being the non-linear transformation function.

IV. **prediction layer:** After the influence diffusion process stabilizes, the output layer forms a user-item pair's ultimate predicted preference. Eq. 3.8 shows user a to item i's predicted preference as:

$$u_a = h_a^K + \sum_{i \epsilon R_a} \frac{v_i}{|R_a|}, \quad \hat{r}_{ai} = v_i^T u_a \tag{3.8}$$

Where R_a denotes a's liked set of items, and v_i is each item i's fusion vector. Each user's final latent representation u_a in the equation (3.8) is composed of two parts:

A. The first term shows h_a^k as embeddings from the output of the social diffusion layers. It exploits recursive social diffusion process to capture the user's interests from the social network structure.

B. The second term forms preferences from user's historical behaviors as $\sum_{i \epsilon R_a} \frac{v_i}{|R_a|}$. It mirrors the SVD++ model that highlighted the historical feedback of the user to tackle the data sparsity of classical CF models.

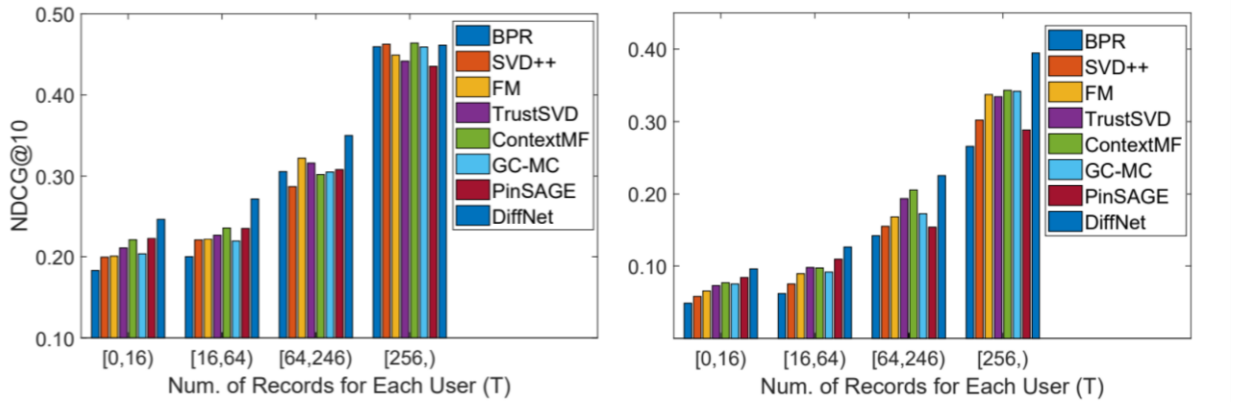


Figure 3.2 Performance under different sparsity for Yelp (left) and Flickr (right) [30]

In terms of performance, Fig. 3.2 shows DiffNet model's superior performance overall, surpassing all the baselines (BPR, SVD++, FM, TrustSVD, ...) for Flickr dataset under various sparsity scenarios. With recursive social diffusion process, Diffnet adopts the late advances of graph convolutional networks (GCN). GCNs have proven

to be successful in stretching the convolution operation from the regular Euclidean to non-Euclidean graph domains.

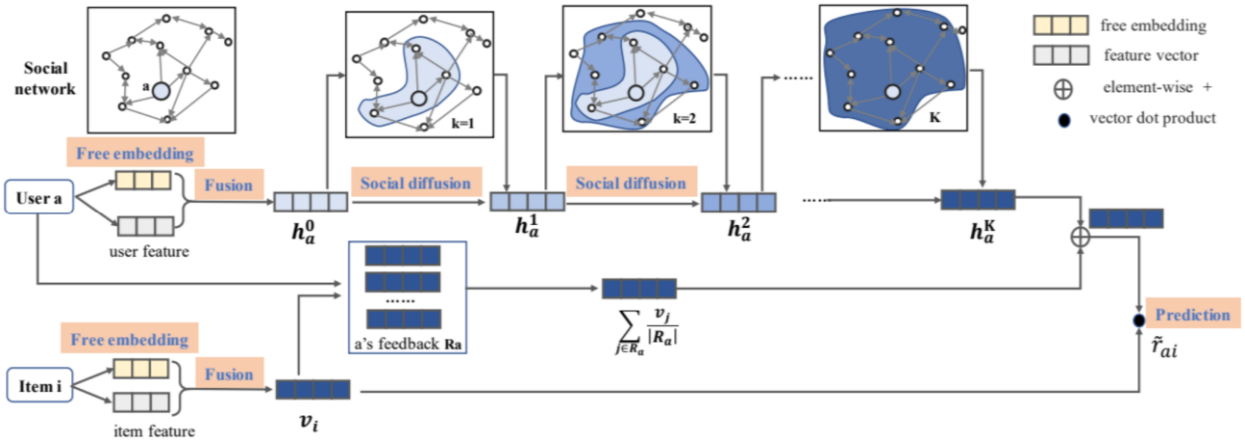


Figure 3.3 The overall architecture of DiffNet model. The four parts of DiffNet are shown with orange background [30]

As can be seen in Fig. 3.3, the main premise of GCNs is to create node embeddings of a graph by passing message or diffusing information. These GCN based models could be used in real-world graphs with much cheaper computation than former spectral based models. The advantages of DiffNet include being efficient on time and storage, and flexibility in the absence of user and item attributes. The model's space complexity is the same as classical embedding models, and the additional time complexity is acceptable, making the proposed DiffNet model applicable to real-world social recommender systems.

3.2c Collaborative Diffusion for Recommender Systems

This simple diffusion model proposed by Lee et.al. [31] aims at addressing issues in existing models. As it stands, diffusion recommenders are presented with two major limitations:

i. Striking a trade-off between enhancing generative capability via noise addition, and preserving the loss of personalized information.

ii. The limited usage of item-side abundance of information.

Excessive noise-injection processes convert inputs into pure noise, leading to significant loss of personalized information and complicating user preference reconstruction, while inadequate noise restricts generative capabilities.

To address these challenges, the proposed model, namely Collaborative Diffusion model for Recommendation (CDiff4Rec), turns item features into pseudo users. It uses behavioral likeness to detect and leverage collaborative signals from both real and pseudo personalized neighbors, which results in effective reconstruction of nuanced user preferences.
CDiff4Rec consists of three stages:

i. Creating pseudo-users from item review logs

ii. Detecting two forms of personalized top-K neighbors from real and pseudo-users

iii. Using those neighbors to aggregate collaborative signals, by addition of their preference data during the diffusion operation

The effective compensation for the loss of personalized information in user-oriented diffusion recommenders is made possible by the model's strategic leveraging of item features. Specifically, the generation of pseudo-users treats each feature (i.e., a review word by the system's native description) as a distinct user with a unique interaction history. Advantages of leveraging item features as pseudo-users include:

- The possibility to flawlessly incorporate m_{pu} (a pseudo-user's normalized interaction vector) into the existing CF framework based on inferred feedback without the need for specialized loss functions for item features.
- The mutual advantages of introducing pseudo users into DR: selectively leveraging the most prominent item content is made possible by DR's capability in diminishing noise through its advanced denoising potency.

Objective Function: the final objective function of CDiff4Rec is calculated as follows:

$$\mathcal{L}_t = \mathbb{E}_{q(x_t|x_0)}[C\|\hat{r}'_u - r_u\|_2^2] \tag{3.9}$$

Where $r_u \in \{0,1\}^{|I|}$ is the interaction history of a given user u, and r'_u denotes a fine-grained representation of user preferences. However, user's own interactions aren't the only factor that guide the recommendation, since the neighbors are also taken into consideration. As can be seen in Fig. 3.4, there are three adjustable values denoted as α, β, and γ (with $\alpha + \beta + \gamma = 1$), which are hyperparameters that control the weights of the user's own, real neighbors, and pseudo neighbors, resulting in fine-grained representation of user preferences as:

$$\hat{r}'_u = \alpha\hat{r}_u + \beta \sum_{ru_i \epsilon \mathcal{U}^u_{ru}} a_{ru_i}\hat{r}_{ru_i} + \gamma \sum_{pu_j \epsilon \mathcal{U}^u_{pu}} a_{pu_j}\hat{m}_{pu_j} \tag{3.10}$$

With $\hat{r}_{ru_i} \epsilon \mathbb{R}^{|I|}$ and $\hat{m}_{pu_j} \epsilon \mathbb{R}^{|I|}$ representing the predictions for a real and a pseudo neighbor, alongside their corresponding attention scores a_{ru_i} and a_{pu_j}.

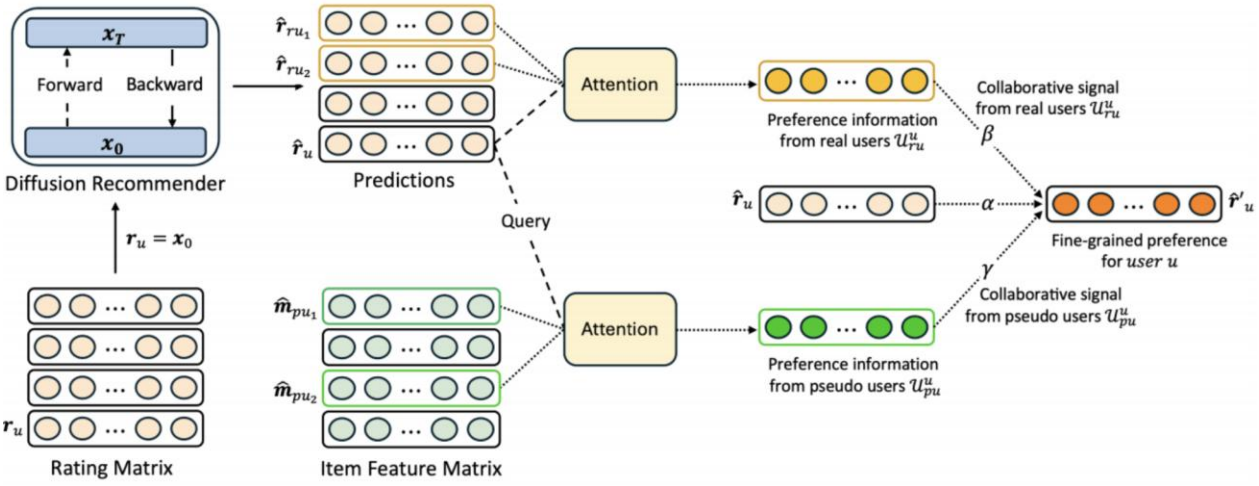


Figure 3.4 The overview of the proposed CDiff4Rec model [31]

CDiff4Rec scores higher than all the baselines. Values inside red boxes in Table 3.1 show superior performance of CDiff4Rec over DiffRec, measured by two metrics on different numbers of top-k recommendations, for almost the same wall time for all models. Thus, overall test results suggest that CDiff4Rec outperforms DiffRec by striking a better balance between accuracy and efficiency.

Dataset	Model	R@10	R@50	R@100	N@10	N@50	N@100	Wall time
Yelp	DiffRec	0.0643	0.1882	0.2818	0.0437	0.0779	0.0982	0:37:47
	CDiff4Rec	0.0722	0.2027	0.2966	0.0490	0.0853	0.1057	0:41:45
AM-Game	DiffRec	0.1365	0.3542	0.4689	0.0771	0.1279	0.1478	0:25:13
	CDiff4Rec	0.1442	0.3577	0.4813	0.0806	0.1299	0.1514	0:25:54
Citeulike-t	DiffRec	0.1175	0.2270	0.2826	0.0748	0.1008	0.1109	3:00:25
	CDiff4Rec	0.1177	0.2282	0.2838	0.0760	0.1019	0.1120	2:55:51

Table 3.1 Performance of CDiff4Rec and DiffRec over Recall and NDCG metrics for 3 datasets [31]

Chapter 4

Preprocessing Dataset for Models

4.1 Preprocessing for LightFM

LightFM model [27] is tailored to accommodate large data sizes effortlessly. Therefore, there weren't any filters applied to the last.fm dataset, beyond purging half of the user set to manage run-time memory allocation. The splitting of the users was done randomly, with the select users appended to their interaction history for conducting the analysis.

Total	Male	Female	< 30 yrs	≥ 30 yrs	North American	European	Rest of the World
133692	99019	34673	107222	26470	29098	73828	30766
-	74%	26%	80%	20%	22%	55%	23%

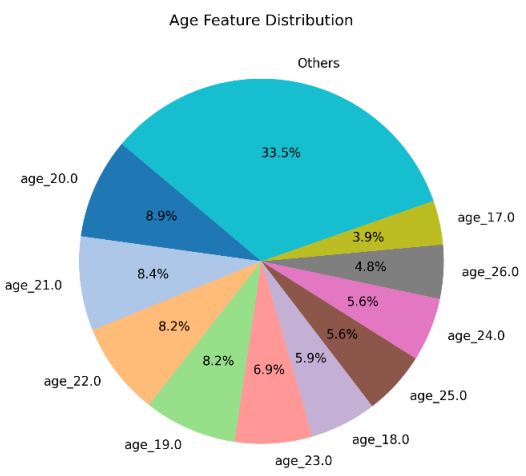
Table 4.1 Distribution of users in the filtered set

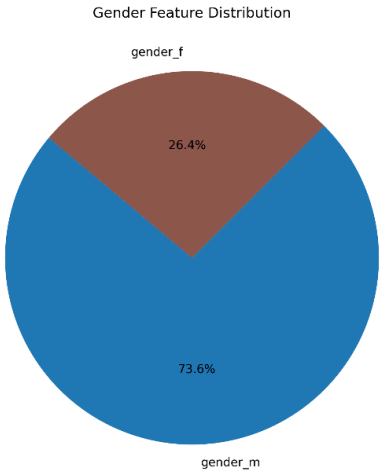
The filtered data in Table 4.1 shows a 3 to 1 male to female ratio in user count. Also, a majority of the user set consists of individuals aged below 30 years, which comes as no surprise, as online casual music listeners tend to be more among the younger audience. Also, North America (most notably the US) and Europe are among the highest user counts, being a result of higher quality infrastructure for a reliable online connection, as well as regular occurrence of shows and concerts, that keeps a greater percentage of population in the music listening habit.

4.2 Preprocessing for EDiffuRec

After being sufficiently filtered and trimmed down, last.fm dataset was pre-processed using EDiffRec's own provided tool. Since the dataset contains more than 17 million entries, it was first cut to its initial 500,000 entries and then filtered to include only the users whose age, country, and gender data are available. Then the users that have appeared on the list less than 5 times were eliminated, as well as logs that contained less than 5 plays per artist.

There were no filters applied to the artists on the list, for two reasons: first, to avoid eliminating the more "niche" artists and include artists with a low count of appearances on the dataset. Second, since only a portion of the entries are considered, artists with minimal appearance count on the current batch might otherwise be dominant on the full scale of the dataset. So, the risk of eliminating more mainstream artists is also avoided.





[continued]

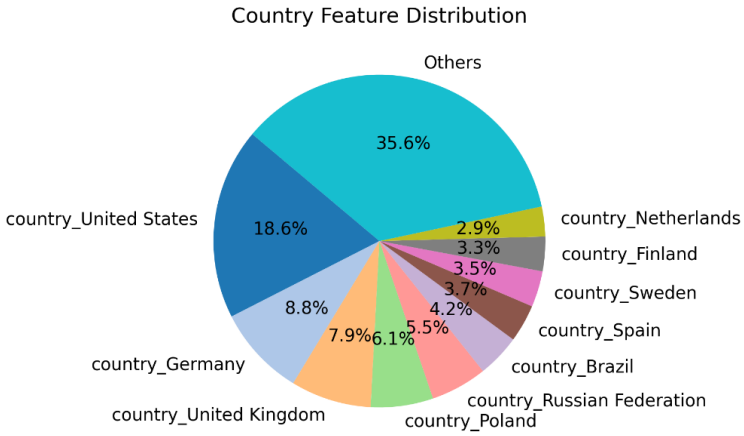


Figure 4.1 Pie charts showing feature distributions across feature categories

Since our "lfm-360k" music dataset doesn't exhibit the exact similar characteristics to the style of input required by the program, the pre-processing tool needs to undergo a key modification to accommodate the new dataset: sorting of the dataset using a recency preference based on the review timestamps, was replaced by sorting based on the total play-count value. The greater number of plays for any given artist, the higher they are placed on the list and (possibly) more likely to be recommended by the algorithm.

The remaining filtered dataset comprises of 5332 artists and 7434 users, which have 210 distinctive features overall. Fig. 4.1 shows how each feature category shares a distribution of features, with top-10 features being shown for age and country. The distributions corroborate with the presumption that the main online music consumers are located in more affluent nations with a lower age range. For gender, there's no diversity provided beyond the binary classification, which is assumed to provide little distinctive quality for a recommender system. It highlights the male user base dominating the female users count, which is a typically expected trait.

4.3 Preprocessing for DiffNet

For the last.fm dataset, firstly, the number of users was limited to 10000, to have a manageable portion of the dataset for this analysis. Then, a limit was imposed to leave out logs with less than 20 plays. Following the method provided by the authors [30], "we filtered out users that have less than 2 rating records and 2 social links, and discarded the items which have been rated less than 2 times." The latter was applied by filtering out artists with less than 2 appearances on the list. Finally, 10000 users and 52790 artists, with 242 distinct features are considered to serve as the inputs to the DiffNet algorithm.

For the user vector data, the model normally lists users with their associated feature values. For "Yelp" dataset, embedding of the words is used for each review, and for "Flickr", it's the image feature representations. For both datasets, the item vector is obtained by averaging that item over its every associated user's feature values. The same is done for the last.fm dataset, with the user vectors being defined as one-hot lists, having zeros equal to the number of all available features (all ages, all genders, and all countries), while for every matching feature for a user, 0 is changed to 1. For gender, feature value changes to 1 in case of a female user.

Dataset	Flickr	Yelp	Lastfm
Users	8358	17237	10000
Items	82120	38342	52790
Total Links	187273	143765	276600
Interactions	314809	204448	665861
Link Density	0.268%	0.048%	0.2766%
Int. Density	0.046%	0.031%	0.1261%

Table 4.2 Comparison of datasets on the paper [30] with Lastfm

Dividing the dataset into train, test, and validation, also follows the instructions given by the paper. Hence, out of 665861 total interactions, it's divided into 539348 for training, 59927 for validating, and 66586 for testing. Table 4.2 shows Last.fm dataset's attributes being compared to datasets from the study [30]. The links between users were established between a pair of users with the same artist and the same play count (within a $\pm 5\%$ margin of error)

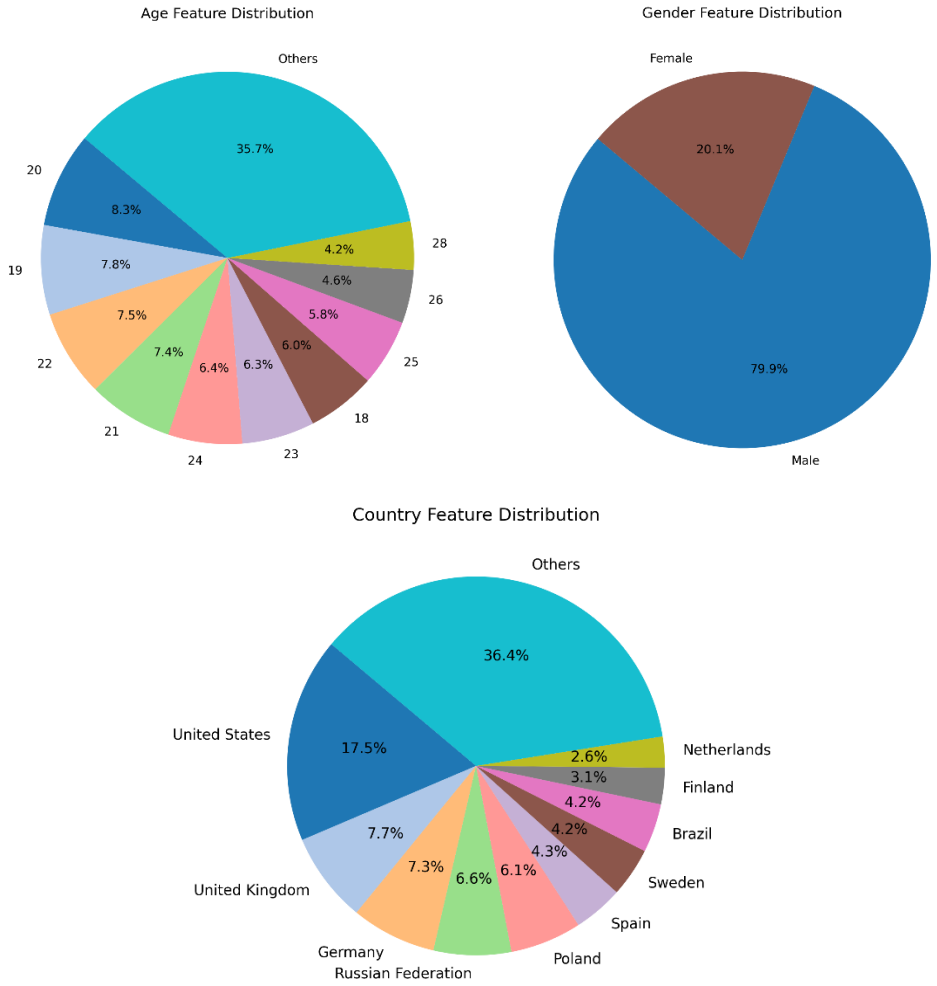


Figure 4.2 Pie charts representing feature distribution across users

As expected, the dominant features are similar to the last filtering of the “lfm-360” dataset for EDiffuRec model. Fig. 4.2 shows the male still surpassing female population, and users below 30 from western countries dominating the charts.

4.4 Preprocessing for CDiff4Rec

A similar method to DiffNet is used for preprocessing, where a limited number of data points from last.fm dataset is used for creating the input set. A filtering criterion is set to eliminate users with less than 5 entries, artists with less than 5 appearances, and datapoints with fewer than 5 play counts. The difference from DiffNet however, is that CDiff4Rec introduces pseudo users, to “mitigate the loss of personalized information in user-oriented diffusion-based recommender systems by treating each feature (i.e., a review word) as an individual user with a unique interaction history.” [31]

However, for the purpose of this study, the features are considered as the subsets of three main categories of age, gender, and country of the user set. Thus, each pseudo-user is conceived as a unique combination of these three main features that are present among the users in the dataset. From this preprocessing script, we record the interaction history for each user in terms of the artists that they have interacted with, to later exploit this data when conducting the XAI analysis.

Dataset	AM-Game	Yelp	Citeulike-t	last.fm
Users	2343	26695	7947	14408
Items	1700	20220	25975	54816
Interactions	39263	942328	132275	960432
Sparsity	99.01%	99.83%	99.94%	99.88%

Table 4.3 Comparison of datasets on the paper [31] with last.fm

Table 4.3 shows how the dataset is structured, to fit as input into the CDiff4Rec's algorithm, compared to datasets from the study. After dividing the set into validation, train, and test, a minimum of 3, 42, and 6 interactions per user exists in each batch respectively, while ensuring that each artist appears at least once in every batch.

Total count	Train	Test	Validation	Items (in training set)
2533	1773	506	254	49041

Table 4.4 Pseudo users' structure

The pseudo-user dataset is structured with the division shown in Table 4.4, to correlate each item in the training set, to the pseudo-user(s) that it has an interaction record with, from their associated real user(s). Following the paper's methodology for creating the set, "we apply TF-IDF to compute the importance of the feature for each item. Then, we perform Min-Max normalization to scale values into [0, 1]." [31]

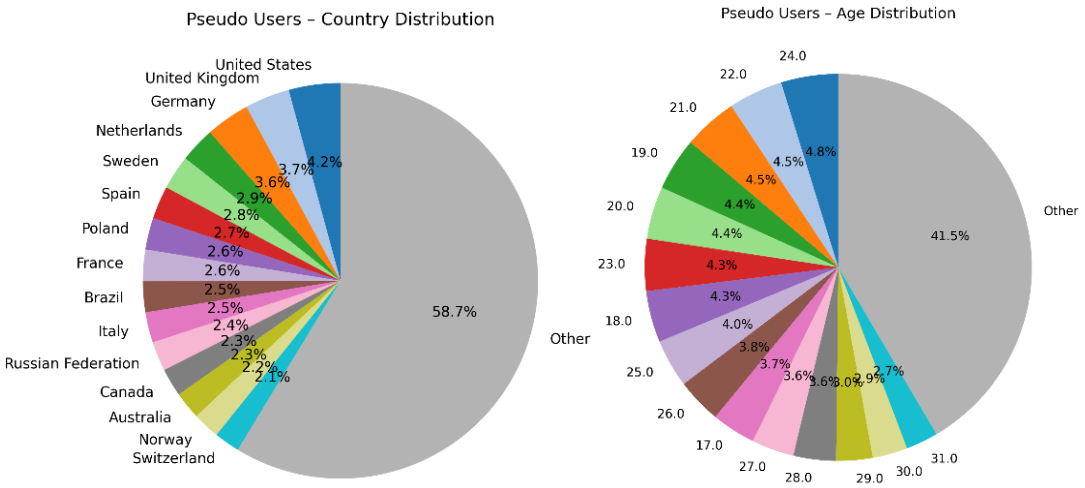


Figure 4.3 Pie charts showing how features are distributed across pseudo users

The distributions in Fig. 4.3 pie charts show country and age distributions, with the gender distribution being almost 2:1 for male to female ratio. Since the exact same methodology is used here for creating the real-user set from the raw last.fm input as for the DiffNet model (except for the minimum play count of 5 instead of 20), the feature distributions are almost identical to the previous set, and thus not discussed in detail.

Chapter 5

Simulation and Result Analysis

5.1 Simulation Results for LightFM

LightFM's novel approach is promising to yield highly reliable outcomes. Comparing LightFM to related models, it takes a feature's effect in predicting an interaction in context of all features, instead of linearly combining feature vector of interacted items. Moreover, it boasts a simplified unitary optimization objective to factorize the user-item matrix. This opposes other approaches that minimize a weighted sum of reproduction loss. Also, unlike other approaches that fail at effectively modelling user features (even considering attempts at incorporating other user interactions on the same item in a cold-start scenario), LightFM jointly factorizes the user-item, item-feature, and user-feature matrices. [27]

To assess the model's efficacy in a music recommendation scenario, a tiny subset of last.fm dataset is used to discover the best combination of model parameters. After running optimizer on last.fm dataset, the best parameter set turned out as learning rate equal to 0.0075, learning schedule being 'adagrad', and number of components as 32. The model is tested in both warm and cold-start scenarios, in the manner described by the paper [27]: "... a warm-start setting: 20% of all interaction pairs are randomly assigned to the test set, but all items and users are represented in the training set. The second is an item cold-start scenario: all interactions pertaining to

20% of items are removed from the training set and added to the test set." The study was conducted using an Nvidia A40 GPU and 150 Gigabytes of RAM.

dataset	Warm	Cold
Last.fm	0.9492	0.9588
CrossValidated	0.695	0.696

Table 5.1 AUC Performance comparison to the dataset in study [27]

As demonstrated in Table 5.1, the model scores substantially high for the 'Area Under the Curve' metric on the last.fm dataset. This metric assesses the model's performance in terms of constantly ranking a randomly chosen positive item higher than any randomly chosen negative item (in other words, it assesses how effectively it can avoid rank inversion). The model decisively scores higher on last.fm than the "CrossValidated" dataset, and interestingly, the cold-start scenario is the case in which both datasets score slightly better.

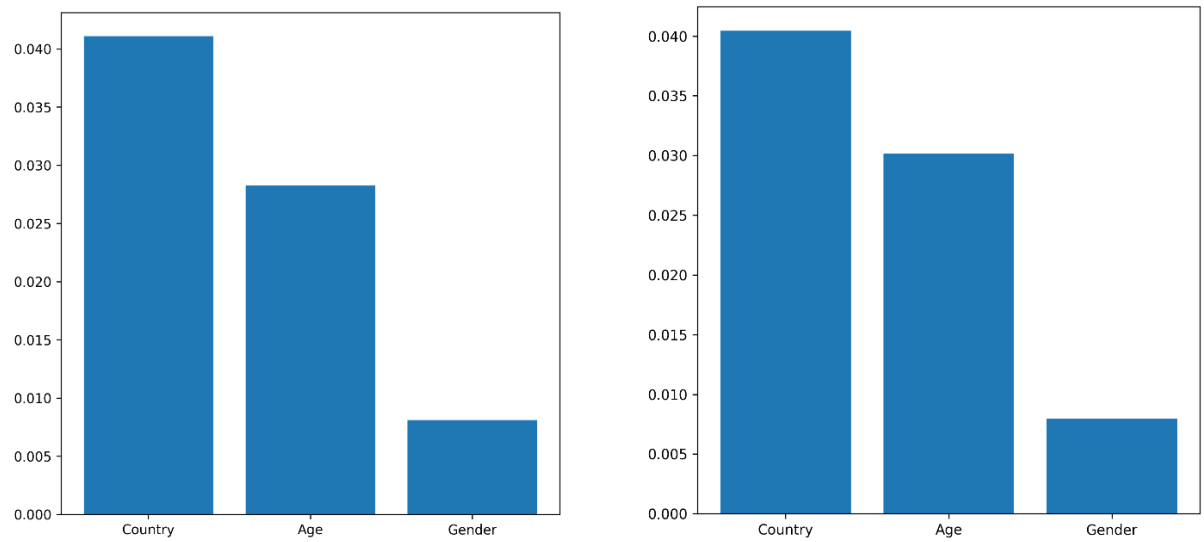


Figure 5.1 Mean global SHAP values for the warm-start (left) and cold-start (right) scenarios

Performing the SHAP analysis for both cold and warm-start scenarios in Fig. 5.1 shows the model ends up prioritizing country as the main contributing feature across the three major categories. Since gender has fewer representative counts than the other two categories, being restricted to only male or female, its lower score when averaged over all of the users is unsurprising (even though the whole dataset in the filtered crop contain gender information). Since there's little difference between the cold and warm-start scenarios in terms of recommendation quality, the rest of the study will be solely based on the cold-start scenario, where the model performs statistically better (Table 5.1).

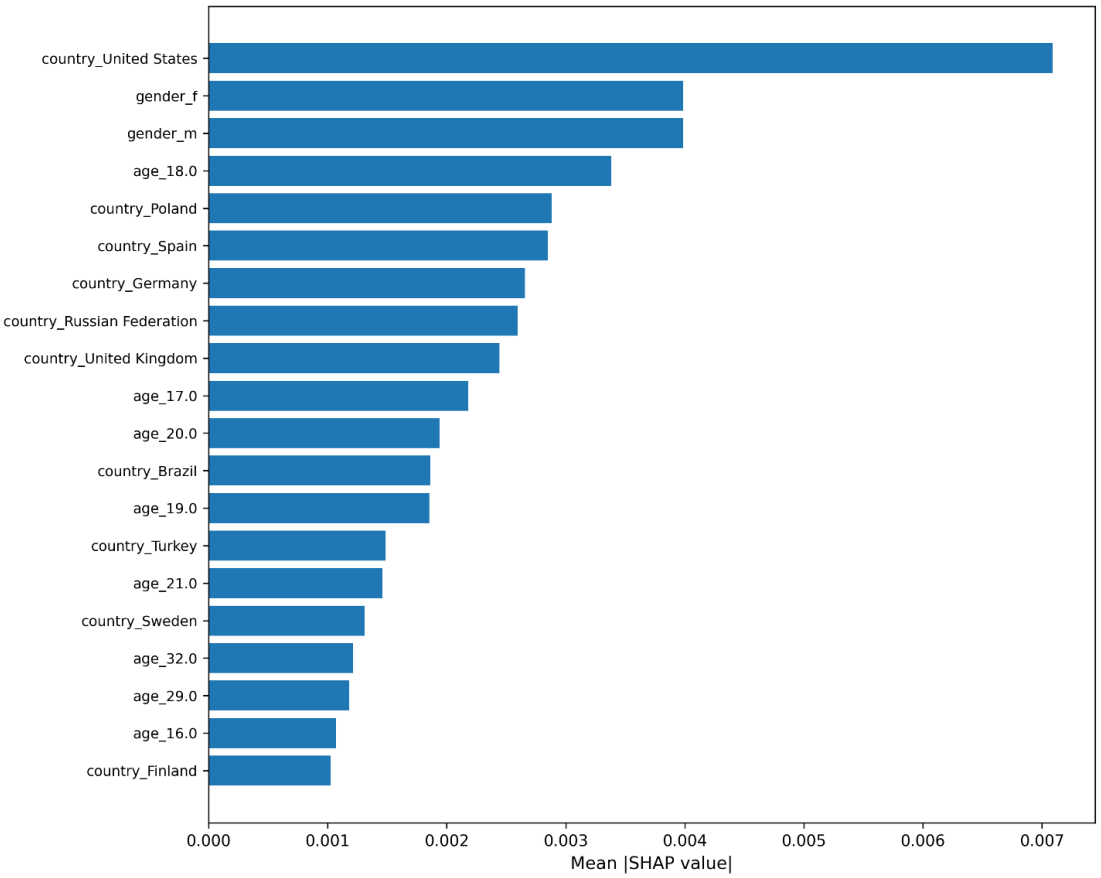


Figure 5.2 Bar plot for the averaged absolute SHAP values on top-20 features

Top features shown in Fig. 5.2 fall in line with the majority of users being labelled with those features. The model would pull the recommendation items featuring plurality of opinions to increase likelihood of the item being accepted. Genders can effectively be interpreted as male and non-male (or vice versa), which explains their scores being the same. Only top categories for age in the top-10 features are ages 17 and 18, that justifies age coming at the 2nd place amongst the main categories (Fig. 5.1).

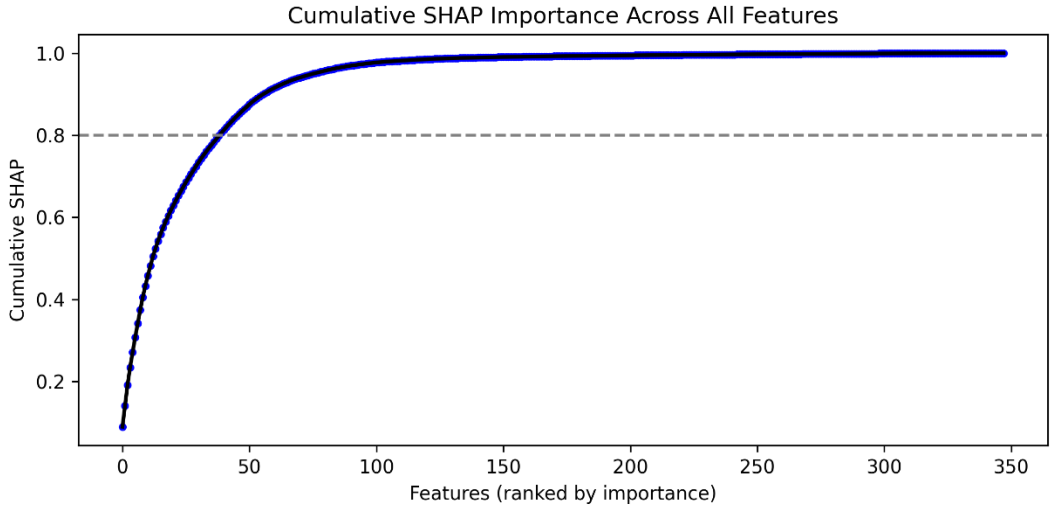


Figure 5.3 Cumulative SHAP graph across all features

To assess what fraction of features are responsible for the cumulative SHAP score, a cumulative graph in Fig. 5.3 shows a rapid convergence towards a score of 1, with 40 to 50 features being responsible for a 0.8 cumulative score. The model fully converges relying on roughly half of the features in the feature set.
Overall, LightFM handles huge data sizes efficiently, which is a desired trait. The simulations show a highly promising performance for LightFM in the music recommendation scenario. Given that the cold-start scenario is a commonplace reality in online recommendation scene, the model's capability in filling the sparsity gap makes it a highly effective tool for mitigating recommendation inequality.

5.2 Simulation Results for EDiffuRec

As the study suggests [29], EDiffuRec is “tailored for real-life-oriented online commerce”. Based on that premise, the model normally uses the information from customer reviews on an e-commerce website and forms a list of recommendations based on that input. Using a Macaron-Net inspired transformer structure and incorporating the Weibull distribution for the noise assumption, are mentioned as the core improvements over the base DiffuRec model.

Based on the results from the paper, using 50 epochs provides a reasonable decrease in training loss, with 100 epochs being the optimal value, and higher epochs providing minimal improvements. To guarantee a reasonable performance for this analysis, 100 epochs were implemented, with the rest of the settings as recommended by the program “with a batch size of 512, a dropout rate of 0.1, and dimensions of embeddings and hidden layers set to 128. We limit the maximum sequence length to 50. The initial LR for warmup is set to 10−5, and after a duration of 20, it is set to 10−3. Additionally, the parameters k and λ for the Weibull distribution are set to 2 and 0.5, respectively.” [29] Run was carried out using an Nvidia A100 GPU, 128 Gigabytes of RAM for the main study, and 200 Gigabytes of RAM for the XAI evaluation.

Dataset	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
LastFM	1.7991	3.2695	6.0231	1.1157	1.5347	2.2089
Amazon Tools	3.2993	4.6449	6.9201	2.3881	2.8196	3.3616

Table 5.2 Performance results from simulating with EDiffuRec compared to results from paper [29]

To evaluate performance of the model on last.fm dataset that is considered in this study, the best results across six metrics are put against the “Amazon Tools” dataset results from the paper in Table 5.2. Although lower performances can be observed

on all accounts, they may be justifiable given the fewer number of meaningful correlations in user-item relationships in the last.fm dataset, compared to the review-based datasets considered in the study. Despite measures taken in preprocessing last.fm to eliminate a portion of the noise by taking out inadequate interactions from the dataset, playing any given song is still signaling a weak intent compared to leaving a product review. The evaluated epoch loss of 8.071 can be considered decent, being equal to the same loss for the Amazon tools after 50 epochs.

A list of the top 100 recommendations for each user was recorded, to perform a post-XAI analysis on the model. The way to incorporate XAI into the analysis was through SHAP, which was calculated as the global SHAP importance per feature (i.e. all ages, all countries, and two genders), taken as average over recommendations list. To fit to the model, logistic regression for the multi-output classifier was considered, with “liblinear” as the solver and 300 maximum iterations. For SHAP, linear explainer with “interventional” feature perturbation is chosen to calculate and store SHAP values for every feature.

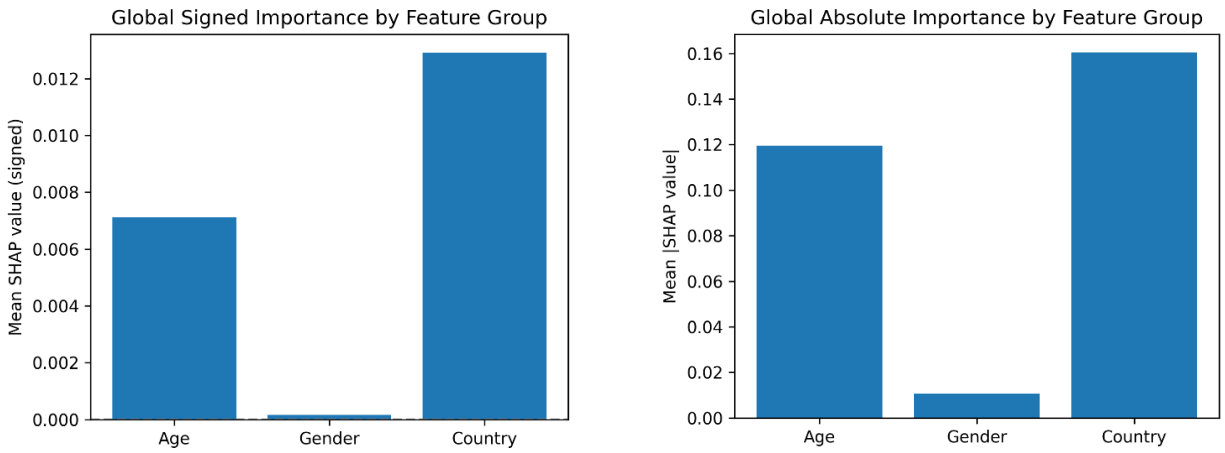


Figure 5.4 mean values across each major feature category for actual and absolute SHAP scores

As a first assessment of the SHAP values, the averages were obtained by grouping them into main categories. The mean values were calculated, to see how every feature category on average contributes to the formation of any given recommendation. Fig. 5.4 shows the country being the most influential category with the highest SHAP value, while age being second most influential, and gender having little effect in comparison, on the final recommendation. This can be mainly attributed to the inadequate information being available on users' gender data with two categories, being only limited to male and female.

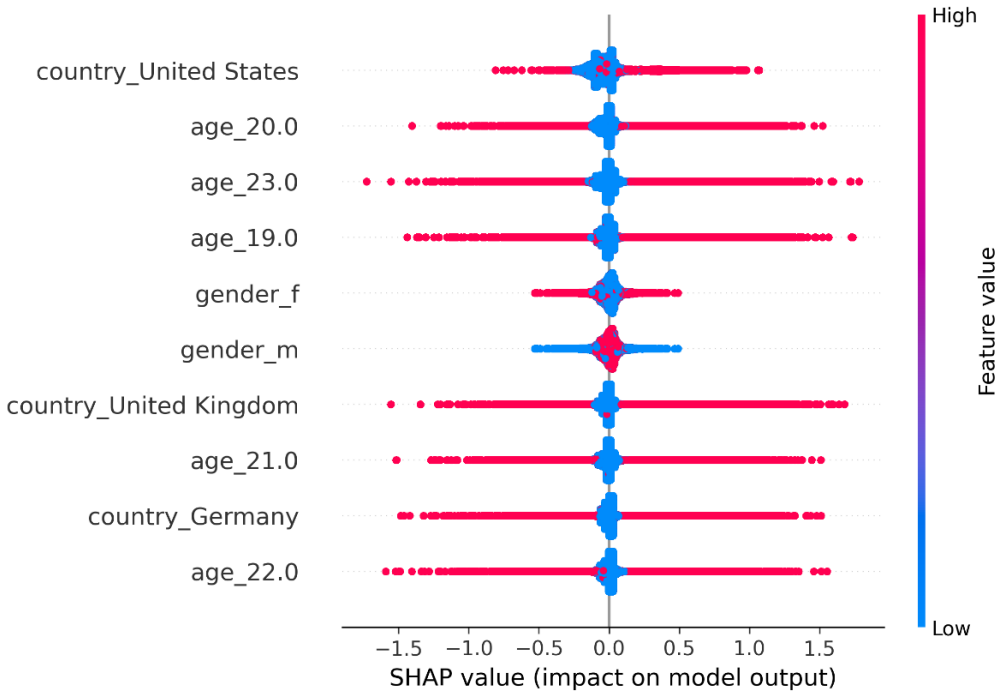


Figure 5.5 Beeswarm plot for SHAP values

To inspect which categories the features with top SHAP values belong to, a Beeswarm plot is considered in this comparison which can be found in fig. 5.5. It shows United States as the most influential feature, with UK and Germany being

among the highest influential geographical features in shaping the user's recommendations list. Like location information, the age of more engaging younger listeners can be among the main decisive factors, with a high influence on the curation. This effect can be broadly seen among the top-10 list, showing how age of users being 23 and under can drastically affect the recommendations list.

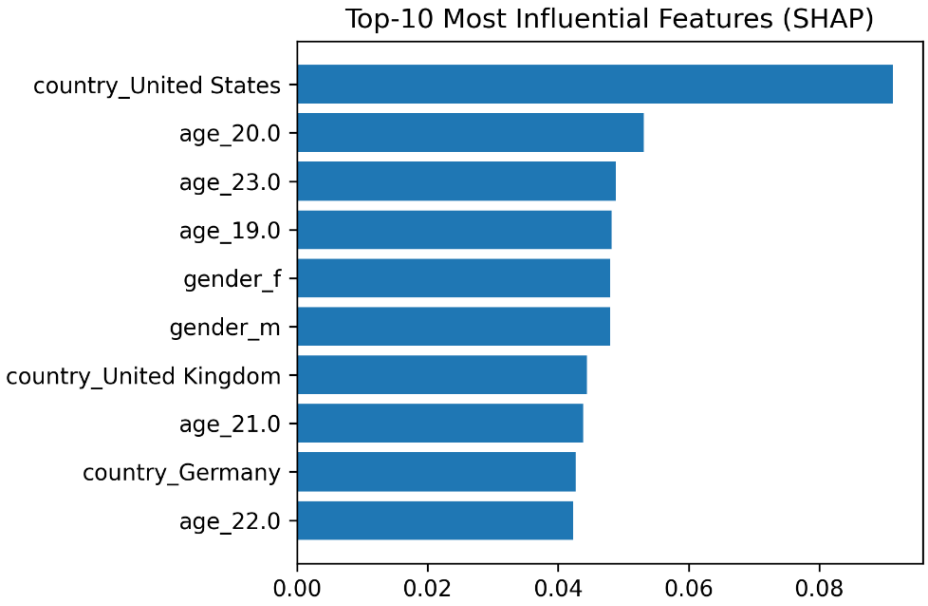


Figure 5.6 Bar plot for the averaged absolute SHAP values on top-10 features

Despite gender variation sparsity, it can still be remarkably influential in shaping a recommendation. Fig. 5.6 depicts gender being ranked as the fifth most influential feature, showing how emphasis is placed by gender in creating the list, which can be generally viewed as a negative trait. Although it cannot inherently point to the system's gender bias (as the feature information isn't explicitly being reported to the system as input), it can show how gender identity may broadly end up as a diversifying metric in any given recommendation.

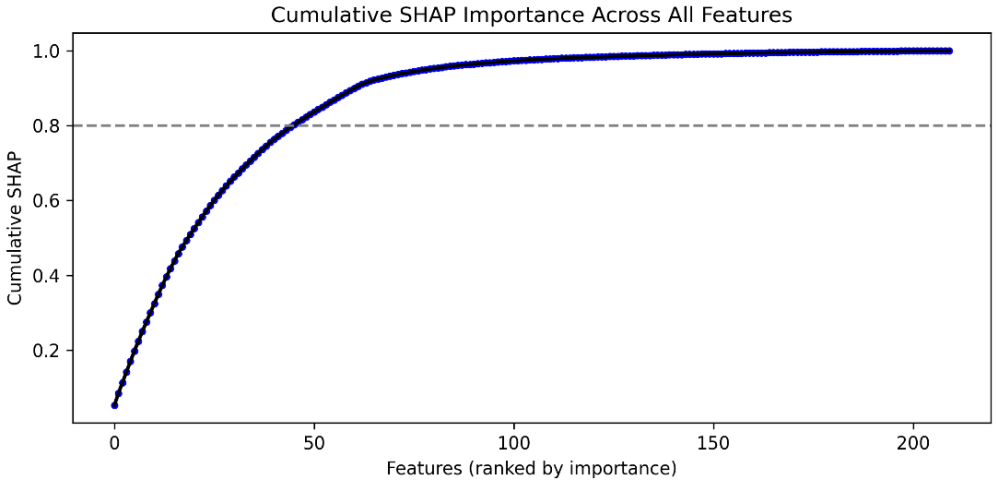


Figure 5.7 Cumulative SHAP graph across all features

It can be concluded that overall, the model's prioritization ends up putting location tokens over other feature metadata in forming a list of recommendations. To know how efficient the features are in explaining system's behavior in forming a recommendation, the cumulative SHAP importance is plotted in Fig. 5.7. It shows approximately 40 features are needed to meet an 80% cutoff, highlighting roughly a fifth of the features being linked to the majority of the system's movement towards the final list. This can further signify an attribute of the system pointed out earlier, that it manages to pair inclusiveness with prevalence of items for picking a recommendation.

5.3 Simulation Results for DiffNet

Diffnet model [30] relies on a layer-wise design for influence diffusion layers, to replicate social diffusion in a social network. It surpasses embedding-based recommendation models in terms of optimizing time and storage. The model bases its social influence premise on the users following each other and their inter-relations. The influence aspect with the last.fm dataset is implemented non-directly and is based on the shared interests, as previously discussed, between users with the same listening habits. The study describes the social influence as a dynamic process with evolving social connections, where in the context of a music recommender, this can be replaced with the evolution of algorithmic recommendations.

The predefined values from the source code are set for the last.fm dataset, and as will be evidenced, the performance was superior to the tests carried out by the study. Diffnet makes use of ADAM as optimizing method, which is a gradient-based method, with an initial learning rate which is $\lambda = 0.001$. Depth parameter is set to $K = 2$, “similar to many GCN models”, and dimension of each layer’s output is set to the user and item free embedding size $D = 32$. The code assigns a batch size of 512 to training and 2560 for evaluation. An Nvidia A100 GPU, 100 Gigabytes of RAM for the DiffNet itself, and 200 Gigabytes of RAM for SHAP analysis were used to generate results.

Dataset		scores
Yelp	HR@10	0.3477
	NDCG@10	0.2121
Lastfm	HR@10	0.5744
	NDCG@10	0.5580

Table 5.3 Performance scores comparison to results from the paper [30]

As the study points out [30], “HR measures the number of items that the user likes in the test data that has been successfully predicted in the top-N ranking list. Further, NDCG considers the hit positions of the items and gives a higher score if the hit item is in the top positions”. With 300 epochs and top-10 recommendations generated for every user, a comparison is shown in Table 5.3 over two indicators. Diffnet scores on last.fm show satisfactory results, surpassing the Yelp dataset on both accounts. This further establishes the role of sparsity (Table 4.2), and presence of a rich dataset with a high number of interactions. Moreover, these results corroborate the findings from the paper, where higher interactions density leads to performance improvements.

Input list of users containing metadata was processed by SHAP, alongside the list of top-10 recommendations generated by DiffNet. The same considerations are applied to the SHAP analysis as for the EDiffuRec model, with fitting done through “liblinear” solver with 300 maximum iterations, and “interventional” chosen as the feature perturbation mode of the explainer.

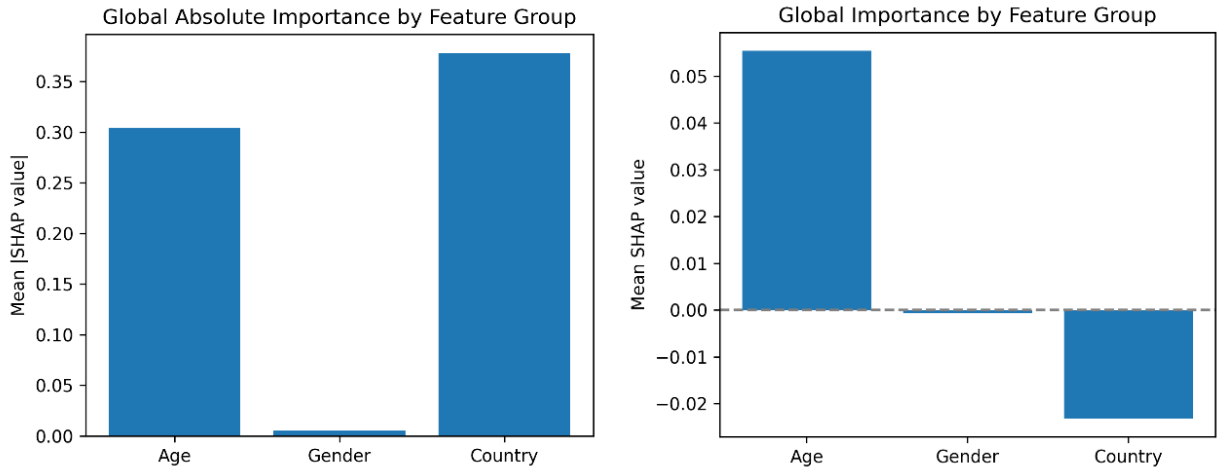
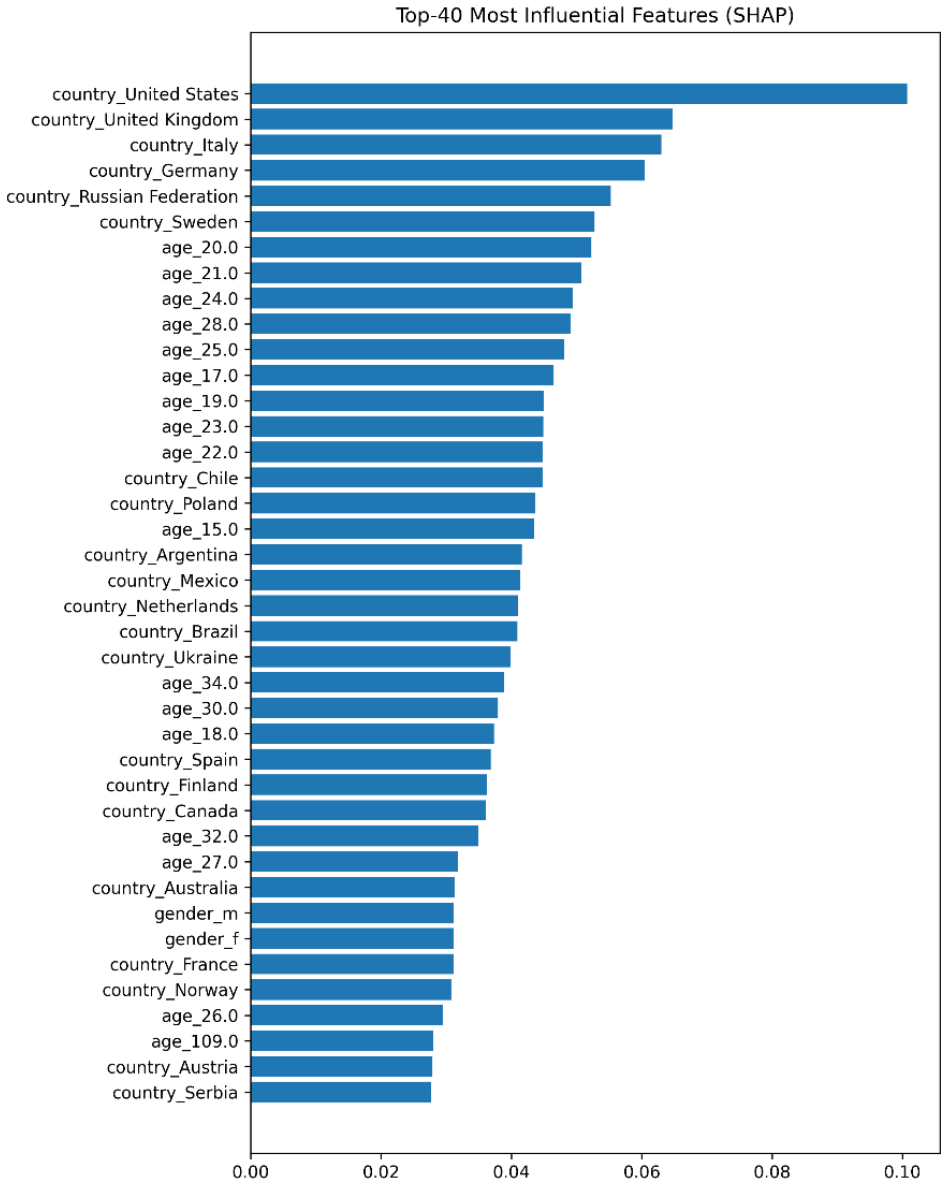


Figure 5.8 Mean values across each major feature category for actual and absolute SHAP scores

The average scores across the main categories in Fig. 5.8 show country being the overall dominating contributor to the recommender's momentum, with the average of its absolute values surpassing age category by a slight margin.



[continued]

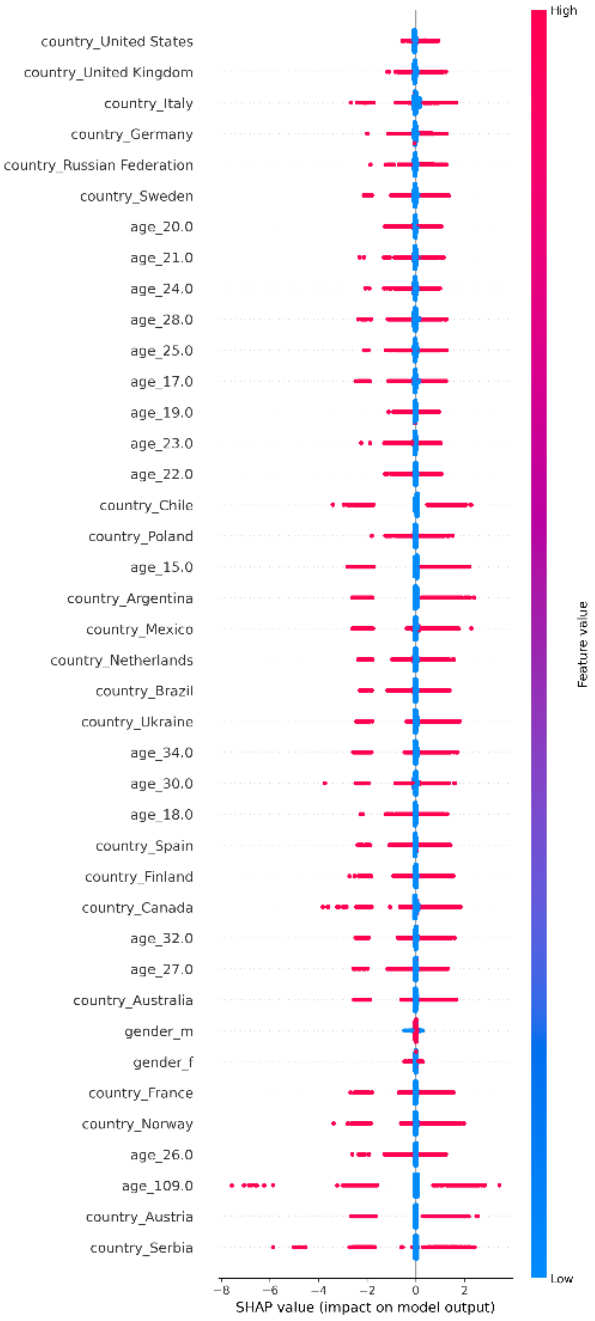


Figure 5.9 Bar plot (last page) and beeswarm plot (current page) for SHAP values on top-40 features

On an individual level, as can be seen in Fig. 5.9, prevalent features among users, e.g. country being United States or age 20, follow the popularity pattern seen in Fig. 4.2. There is a major exception however, where 3rd most influential feature (country being Italy) isn't seen among the most prevalent in the data. The reason may be evident from the beeswarm plot, as the majority of instances for Italy are positive-leaning and concentrated close to zero, and fewer instances of negative scores near the median-left. Genders can be seen taking the 33rd spot, which can be interpreted as a decent aspect of the system for not having an emphasized gender bias, that can be attributed to a high density of scores at zero and their balanced distribution, as seen in beeswarm plot.

Moreover, the higher number of 21 countries in Fig. 5.9 graphs in comparison to 17 age feature appearances among 40 top features in bar plot, also shows the geographical feature outweighing age and gender. However, unlike EDiffuRec model, age feature comes close in second place and as discussed, causes the largest shift to an average recommendation.

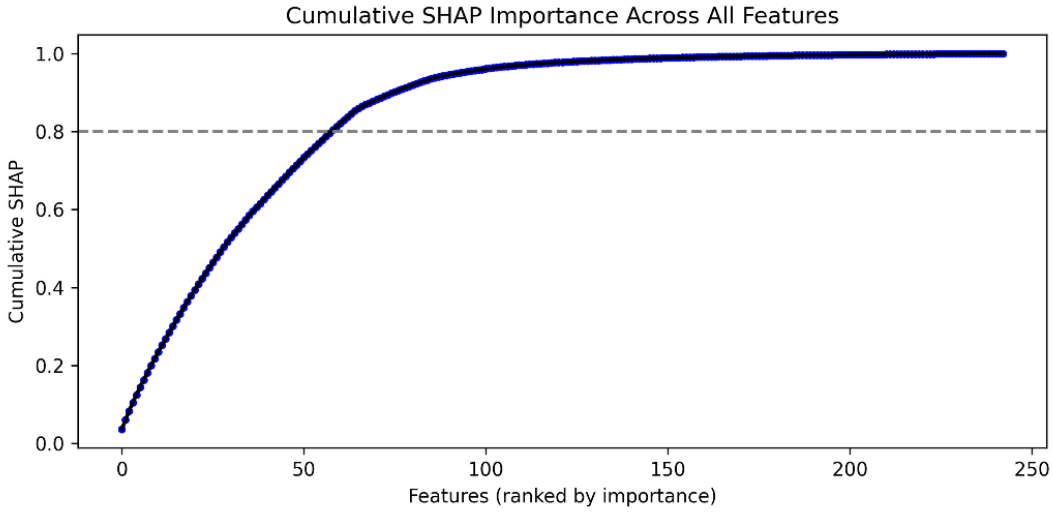


Figure 5.10 Cumulative SHAP graph across all features

The cumulative SHAP graph in Fig. 5.10 shows approximately a quarter of the features being responsible for 80% of the decision-making, and with nearly half the features sufficient for reaching cumulative SHAP value equal to one. This demonstrates how incorporating features as input to the model in DiffNet in comparison to EDiffuRec, leads to the inclusion of a higher portion of the features in recommendations list. The model also performs as tested here, faster and more memory-efficient than EDiffuRec, which renders the model more suitable for a real-time implementation scenario.

5.4 Simulation Results for CDiff4Rec

After the creation of pseudo- and real- user data, we establish two types of top-k neighbors based on the user type. Although the model [31] may run only on real user data, by introducing pseudo-users, which is the most distinctive feature of CDiff4Rec, it results in efficient reformation of nuanced user preferences. By tuning hyperparameters alpha, beta, and gamma, the weight ratio on user's own, real neighbors, and pseudo neighbors can be set, to personalize the recommendation behavior based on a certain performance goal or a specific dataset's characteristics.

Dataset	#Top-K	Recall	NDCG
Lastfm (unoptimized)	**@10**	0.1227	0.1247
	@20	0.1874	0.1564
	@50	0.3004	0.2001
	@100	0.4057	0.2335
Lastfm (optimized)	**@10**	0.1246	0.1839
	@20	0.1877	0.1971
	@50	0.3015	0.251
	@100	0.4094	0.2929
AM-Game	**@10**	0.1442	0.0806
	@20	0.2255	0.1052
	@50	0.3577	0.1299
	@100	0.4813	0.1514

Table 5.4 Performance scores comparison to results from the paper [31]

As a first assignment, the hyperparameters were set to $\alpha = 0.3,\ \beta = 0.3,$ and $\gamma = 0.4$ to have an almost equal representation of every neighbor type (with a slight emphasis on the pseudo neighbors, as it is the distinctive characteristic of the recommender). Second assignment was done by optimizing the parameters using Optuna for all Recall and NDCG metrics, which resulted in $\alpha = 0.6,\ \beta = 0.38,$ and

$\gamma = 0.02$. Learning rate was set to 0.0001 (the default value), the number of similar interest users top-k neighbors was set to 50, batch size to 400, and top-k pseudo-users to 1773 (equal to the number of users in the pseudo train set).

A maximum limit of 1000 epochs is set, and the model would exit the training early in case the best result is already achieved and the performance stagnates in the following epochs. Table 5.4 shows the performance on the last.fm dataset that is achieved after 53 epochs for the unoptimized and 64 epochs for the optimized model, which is comparable to the best performing dataset being "AM-Game", from the paper [31]. However, this goal is reached in 90 minutes and 11 seconds wall time for the unoptimized model and 104 minutes and 15 seconds for the optimized, compared to 25 minutes and 54 seconds for AM-Game. This can be attributed to the significantly higher number of data points on last.fm compared to "AM-Game" (Table 4.3). Since the optimized model performs better on all metrics, especially NDCG which is a crucial aspect to consider here, the optimized model is utilized in this study despite its slightly longer wall time. The run was carried out using an Nvidia A40 GPU, and 100 Gigabytes of RAM.

Since the model produces an output in which latent dimensions contribute to artists, binary artist vectors can be achieved by projecting through input layer's weight array. Hence the SHAP analysis needs to be done through kernel instead of linear SHAP, for which the explainer model has to be simplified enough to output results within a reasonable timeframe. Thus, a subsample of 50 items was chosen from all artists, containing 17 popular items, 16 medium (niche) items, and 17 from long tail (noisy embeddings). Also, the countries were bucketed into the general categorization of continents and ages into age groups, while genders were considered as-is. The kernel explainer was defined using a background of k-means, with k=50. 200 users are being explained for, while the SHAP explainer used 300 samples (as perturbations per user to estimate contributions).

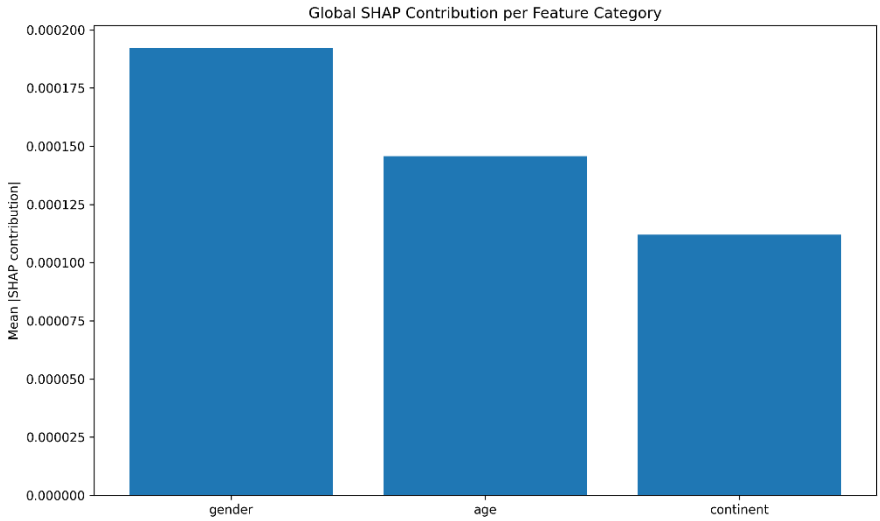


Figure 5.11 Mean values across each major feature category for absolute SHAP scores

Previously, in a linear SHAP analysis, with the user preferences being already known, a weak signal such as gender could offer negligible explanatory power to the recommendation (as it was also averaged out over users). Unlike those linear SHAP analyses, here the latent space allows us to examine how the user's preferences are skewed towards a certain recommendation, through a reconstruction of the user profile.

Hence, as can be seen in Fig. 5.11, gender has the most pronounced presence, contrary to the previous analyses. Age category surpasses continent importance as the more influential contributor, with continent scoring the lowest amongst the main categories. A decision plot can be found in Fig. 5.12, in which the order that the features appear slightly differs from the global importance ranking, with an instance-level contribution. It shows major shifts occurring in output values at every instance, which highlights how feature contributions exponentially grow in influence when bucketed into subcategories. A beeswarm plot in Fig. 5.12 further clarifies how bucketing features lead to a high polarization in feature contributions.

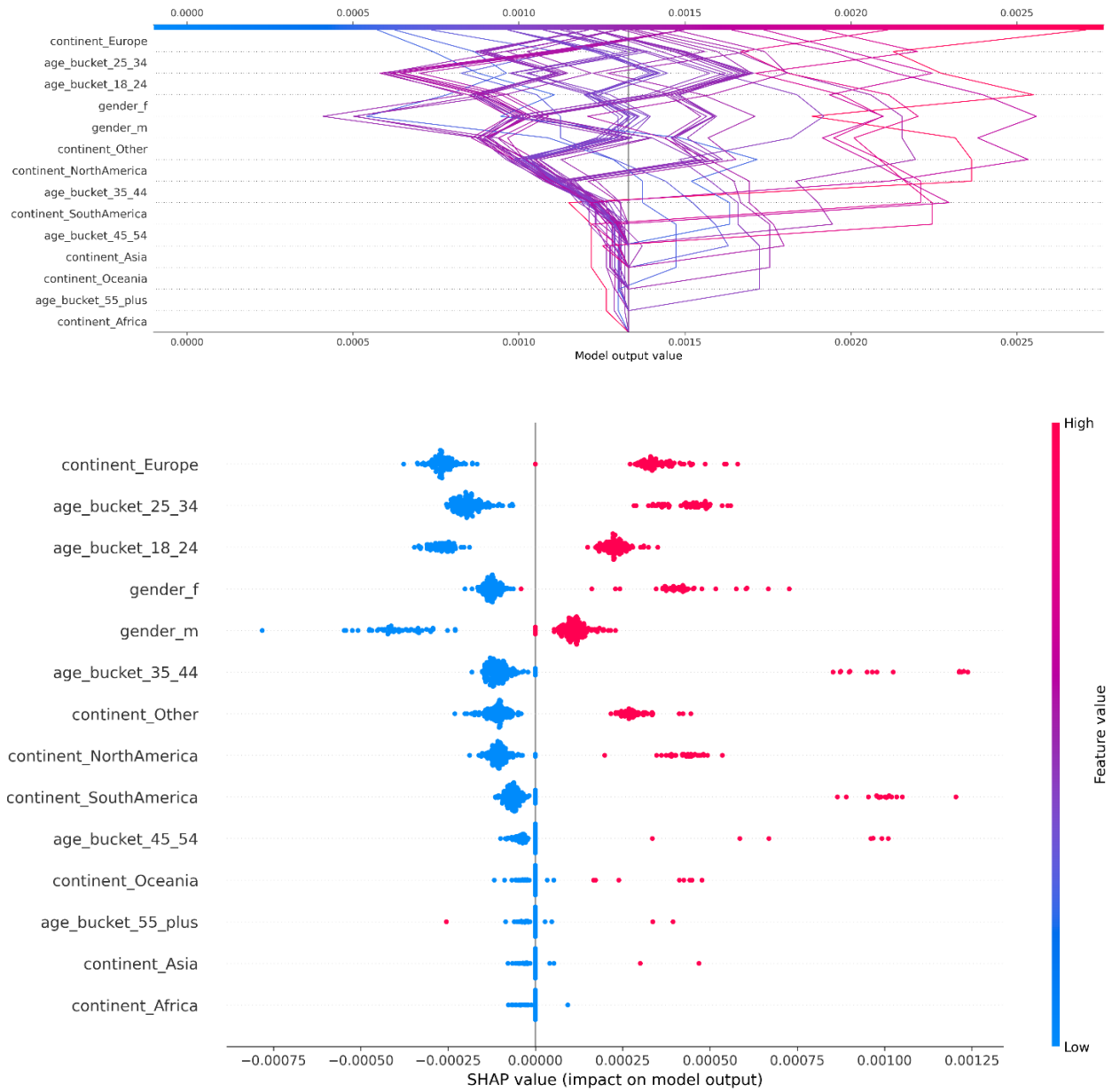


Figure 5.12 Decision plot over bucketed features and genders (top), Beeswarm plot over bucketed features and genders (bottom)

Although users aged 18-24 are dominating the age demographics (Fig. 4.3), they are placed lower in feature contribution than ages 25-34. This can be attributed to a shift in scores further to the right of the graph in Fig. 5.12 for this category, whereas ages 18 to 24 is characterized by a more evenly distributed set of scores. The same holds true for the features ranked fourth and fifth, with female gender having a right-leaning balance of SHAP scores. This shows that although the recommender system may pull more from the majority representation globally (as seen in previous models), it isn't a likely system trait to realize a body of users having a singular preference towards certain majority population, as evidenced when reconstructing profiles.

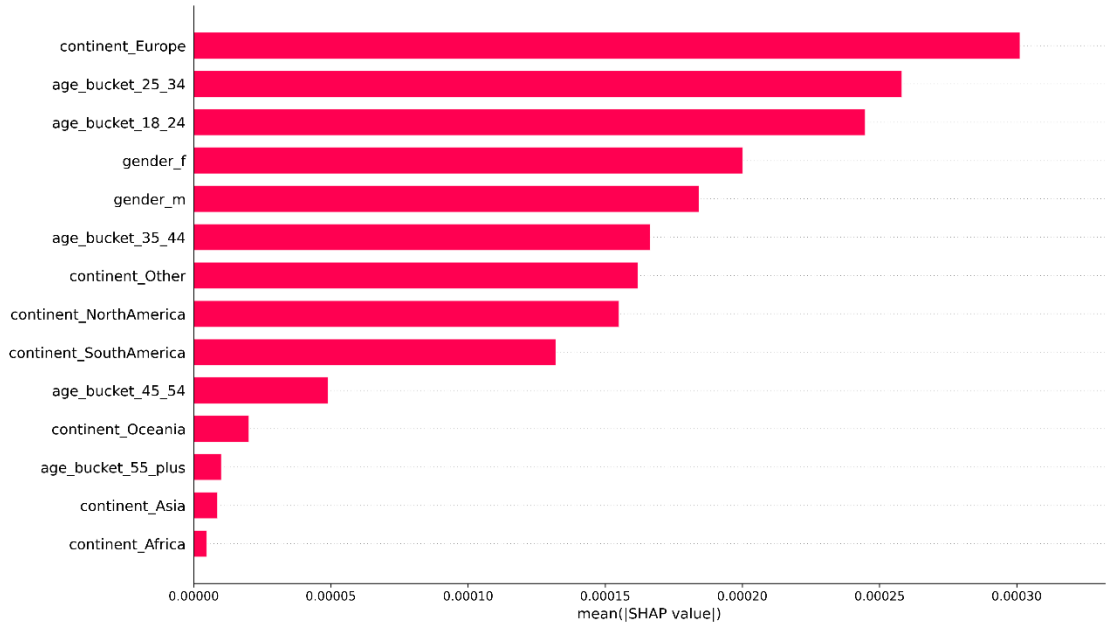


Figure 5.13 Bar plot over bucketed features and genders for absolute SHAP values

A noticeable observation is the continent of North America being ranked 8th, below the age group 35 to 44. With the US having the highest representative count among

countries in Fig. 4.3, while ages 35 to 44 have zero presence among the highest count user ages. One important factor that could justify this behavior, explained by Melchiorre et.al. [7], is the engagement level exhibited by different groups. For instance, while the US has the highest user count, it has a low level of male user engagement compared to Poland, which could explain why Europe appears higher on the list. A similar observation can be made of users older than 30 years. Their higher engagement than users aged 30 and below can explain why users aged 35 to 44 ended up higher on the list than expected. While the paper discusses LFM-2b-DemoBias dataset, LFM-360k is also expected to exhibit similar trends overall, given that it's a smaller sample from the same data faucet.

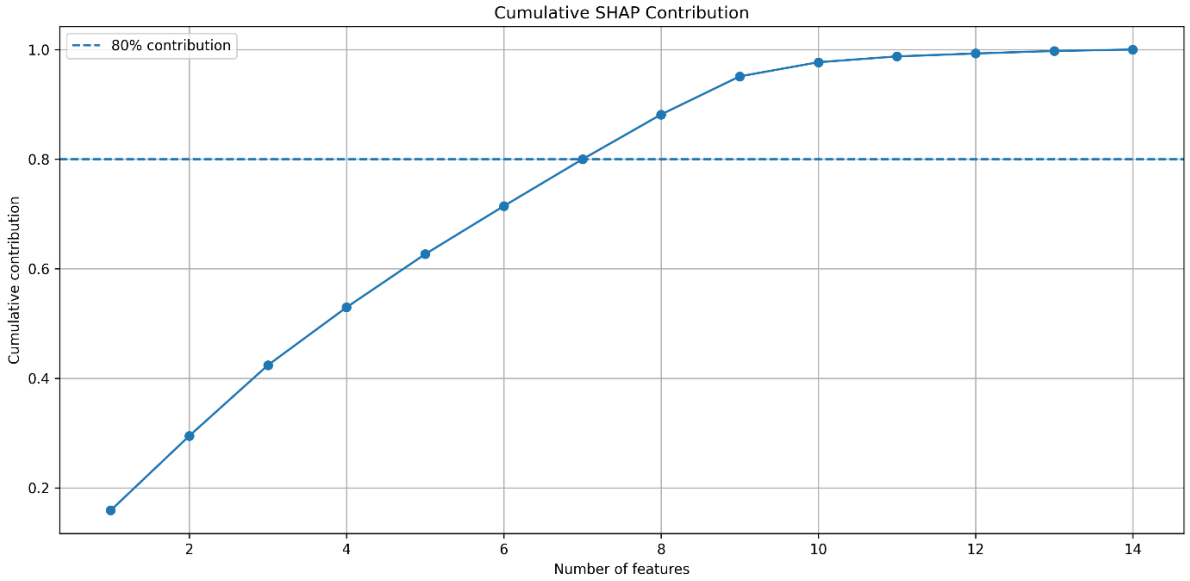


Figure 5.14 Cumulative SHAP graph across all features

As expected, bucketing of the features makes the recommender rely on more feature elements than normal. Fig. 5.14 shows the cumulative SHAP reaches an 80% cutoff using half of the bundled features. The performance may be improved by tuning the hyperparameters, as already discussed, to achieve higher efficiency.

However, due to the higher time consumption of the optimized model, implementation of the optimal hyperparameter setting in a real-time recommender scenario may be costly. Therefore, a tradeoff may be set to seek the local optima, while meeting the time constraints and a minimum performance quality, as the model has shown promising performance even in case of arbitrary hyperparameter values (Table 5.4).

Tuning of parameters in favor of real or pseudo users can be decisive in shaping the system's behavior. With the item-side features being elevated by implementing a pseudo-neighbor coefficient, we decided to divide the system's focus between feature characteristics, real user embeddings, and neighbors. The results show how this strategy led to subverting common expectations with the recommender's performance.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This study presented several novel and noteworthy recommendation models, and their capacity to reach fairness in the music recommendation scenario. LightFM as a model predominantly developed around overcoming the cold-start scenario is of particular interest. It performs exceptionally well, and is well-suited for a streaming scenario where huge datasets may be used, and high-quality recommendations are expected within a reasonable timeframe. Diffusion models were also investigated, as diffusion-based recommendation has the potential to reach high levels of utility in recommendation sphere. Scope of the study ranged from analyzing basic input models in case of EDiffuRec, to closely inspecting the system through profile reconstruction approach in case of CDiff4Rec.

Although EDiffuRec doesn't explicitly accept feature attributes as input, the algorithm's behavior shows how features link to top recommendation list in a similar manner to a feature-explicit model. It may highlight the diffusion network's robust performance, even in the case of a non-feature-based system. DiffNet is a Diffusion model which steps up from EDiffuRec by accepting user and item vectors, and establishing social influence between users. It outperforms EDiffuRec, and as evidenced by the study [30], also all the baselines under various ranking measures and parameters. Finally, CDiff4Rec provides an innovative solution to feature inclusion by inserting features as pseudo-users into the process. Modularity of the

system in terms of providing levers to tune the weight of different neighbor types makes it adjustable to any recommendation case.

6.2 Limitations and Open Challenges

Reaching fairness in a recommender system may be an ongoing challenge, with no definitive solution that can be applied to every single scenario. While in principle, setting fairness goals would mean reaching a fair tradeoff to satisfy the parties involved in a music recommendation scenario, in practice, definition of a role can inherently put one party at a disadvantage. Typically, the streaming services as providers benefit from this equation the most, by setting the framework, with the power to leverage visibility of item providers and shaping user preferences. Thus, it would be impossible to evaluate fairness when such non-fungible factors dominate the fairness metrics, by framing artists and users against the unmatched leverage power of streaming services.

Moreover, streaming services are effectively viewed as lucrative business conglomerates in their role as a data broker, linking a network of brands, labels, and listeners, which effectively incentives them to represent themselves as investors in data holding rather than music sale profiteers [6]. There's an ever-growing concern over companies compromising users' personal information, through leaks and data breaches. This framing can further fuel the customers' reluctance when sharing their personal data with streaming services as an untrustworthy entity. Therefore, the misalignment of data protection with fairness goals necessitates setting a tradeoff that may negatively impact fairness.

While a less detailed input design may appear counterintuitive to reaching a higher level of fairness, it serves as a persistent gap of information that is often overlooked in designing recommender systems. Obtaining quality recommendations requires

tracking the user's habits and activities through persistent data collection, as discussed in section 2.3. Gathering this type of sensitive data requires system-wide access to the user's personal device and constant tracking of their activities. This can be challenging, given the increasing scrutiny over user privacy, and hardware companies providing consumers with on-demand tools to limit access when sharing their own sensitive information with music services.

In particular, this limitation is imposed by EDiffuRec model, which is an overly simplistic diffusion algorithm. The model isn't particularly accommodating to the intricacies of a provided input. It is structured to set priorities in the preprocessing step, by ordering the reviews based on their recency (in the study's original approach [29]), or by play count (in the approach in section 4.2 used to feed last.fm dataset to the system). It doesn't offer an in-depth analysis by recognizing user features to any degree of sophistication. Thus, the results discussed here are purely indicative of the model's underlying methodology for creating a recommendation list, put into perspective in a music recommendation context. It may serve as an entry point into conditional diffusion systems and set initial expectations.

Lastly, current fairness study holds the capacity for further improvement. SHAP as the explanation apparatus is a fairly recently developed tool that is still to reach its full potential. As it stands, it can be argued that the underlying functionality and overall results of a SHAP study are a reliable source for understanding the system. However, previous studies in the area of music recommendation, as well as the incoherencies seen in the current study, have shown that feature rankings are prone to being slightly misattributed, and SHAP's feature sorting capability needs readjustment. To overcome this issue, there was an effort throughout the study to explain the SHAP results using different graphs and overall trends, rather than overly focusing on the final SHAP scores, individual graphs, or feature rankings. Any meaningful advancement in this area would require key modifications to improve

the SHAP's algorithm and architecture, which is outside of the coordinates that a recommendation researcher works in. Therefore, this gap remains as an area of focus that needs to be addressed and accounted for, with any SHAP analysis study.

6.3 Future Research Direction

In the current approach, the aggregation of SHAP scores and their mean values were used as the main tools in evaluating the model performance. However, this global score assessment comes with caveats, which as discussed, is the compromise of the importance of certain underrepresented categories. A new perspective into this study may replace the global SHAP averages with per-instance scores, and assess them individually against each other. It would be interesting to provide a basis for non-normalized scores to be investigated proportionately, as it can demonstrate the independent contribution of each feature in the recommendation process.

Bibliography

[1] Bahareh Rahmatikargar, Pooya Moradian Zadeh, Ziad Kobti. 2024. Two Decades of Recommender Systems: From Foundational Models to State-of-the-Art Advancements (2004-2024).

[2] Xu, Feiyu & Uszkoreit, Hans & Du, Yangzhou & Fan, Wei & Zhao, Dongyan & Zhu, Jun. (2019). Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges. 10.1007/978-3-030-32236-6_51.

[3] Dinnissen K and Bauer C (2022) Fairness in Music Recommender Systems: A Stakeholder-Centered Mini Review. *Front. Big Data* 5:913608. doi: 10.3389/fdata.2022.913608

[4] Avriel Epps-Darling, Romain Takeo Bouyer, Henriette Cramer. "ARTIST GENDER REPRESENTATION IN MUSIC STREAMING", 21st International Society for Music Information Retrieval Conference, Montréal, Canada, 2020.

[5] Andŕes Ferraro, Xavier Serra and Christine Bauer. "What is fair? Exploring the artists' perspective on the fairness of music streaming platform". 18th IFIP International Conference on Human-Computer Interaction (INTERACT 2021). Bari, Italy

[6] Georgina Born, Jeremy Morris, Fernando Diaz, and Ashton Anderson. "Artificial Intelligence, Music Recommendation, and the Curation of Culture", Schwartz Reisman Institute for Technology and Society White Paper (2021)

[7] Melchiorre, Alessandro B.; Rekabsaza, Navid; Parada-Cabaleiro, Emilia; Brandl, Stefan; Lesota, Oleg; Schedl, Markus: Investigating gender fairness of recommendation algorithms in the music domain. In: Information Processing & Management. Amsterdam : Elsevier, 2021

[8] Mehrotra, R., McInerney, J., Bouchard, H., Lalmas, M., and Diaz, F. (2018). "Towards a fair marketplace: counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems," in Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM

'18 (New York, NY: Association for Computing Machinery), 2243–2251. doi: 10.1145/3269206.3272027

[9] Mehrotra, R., Xue, N., and Lalmas, M. (2020). "Bandit based optimization of multiple objectives on a music streaming platform," in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '20 (New York, NY: Association for Computing Machinery), 3224–3233. doi: 10.1145/3394486.3403374

[10] Emanuele Bugliarello and Rishabh Mehrotra James Kirk Mounia Lalmas. 2022. Mostra: A Flexible Balancing Framework to Trade-off User, Artist and Platform Objectives for Music Sequencing. In Proceedings of the ACM Web Conference 2022 (WWW '22), April 25–29, 2022, Virtual Event, Lyon, France. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3485447.3512014

[11] Bauer, Christine & Ferraro, Andres. (2023). Strategies for Mitigating Artist Gender Bias in Music Recommendation: A Simulation Study. 10.5281/zenodo.8372477.

[12] Ferraro, Andres & Bogdanov, Dmitry & Serra, Xavier & Yoon, Jason. (2019). Artist and style exposure bias in collaborative filtering based music recommendations. 10.48550/arXiv.1911.04827.

[13] Shakespeare, Dougal & Porcaro, Lorenzo & Gómez, Emilia & Castillo, Carlos. (2020). Exploring Artist Gender Bias in Music Recommendation. 10.48550/arXiv.2009.01715.

[14] Andres Ferraro, Xavier Serra, and Christine Bauer. 2021. Break the Loop: Gender Imbalance in Music Recommenders. In Proceedings of the 2021 ACM SIGIR Conference on Human Information Interaction and Retrieval (CHIIR '21), March 14–19, 2021, Canberra, ACT, Australia. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3406522.3446033

[15] Afchar, Darius & Melchiorre, Alessandro & Schedl, Markus & Hennequin, Romain & Epure, Elena & Moussallam, Manuel. (2022). Explainability in music recommender systems. AI Magazine. 43. 190-208. 10.1002/aaai.12056.

[16] Jin Ha Lee, Liz Pritchard, Chris Hubbles. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). "Can we listen to it together?: factors influencing reception of music recommendations and post-recommendation behavior", 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 2019.

[17] Doshi-Velez, F., and B. Kim. 2017. “Towards a Rigorous Science of Interpretable Machine Learning.” arXiv preprint arXiv:1702.08608

[18] Ribeiro, M. T., S. Singh, and C. Guestrin. 2016. ““Why Should I Trust You?” Explaining the Predictions of Any Cassifier.” In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–44

[19] Andjelkovic, I., D. Parra, and J. O’Donovan. 2016. “Moodplay: Interactive Mood-based Music Discovery and Recommendation.” In Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, 275–9. New York: ACM.

[20] Sharma, A., and D. Cosley. 2013. “Do Social Explanations Work? Studying and Modeling the Effects of Social Explanations in Recommender Systems.” In Proceedings of the 22nd International Conference on World Wide Web, 1133–44. New York: ACM.

[21] Kouki, P., J. Schaffer, J. Pujara, J. O’Donovan, and L. Getoor. 2019. “Personalized Explanations for Hybrid Recommender Systems.” In Proceedings of the 24th International Conference on Intelligent User Interfaces, 379–90. New York: ACM.

[22] Kindermans, P.-J., S. Hooker, J. Adebayo, M. Alber, K. T. Schütt, S. Dähne, D. Erhan, and B. Kim. 2019. “The (un) Reliability of Saliency Methods.” In NIPS workshop on Interpreting, Explaining and Visualizing Deep Learning.

[23] Fel, T., and D. Vigouroux. 2020. “Representativity and Consistency Measures for Deep Neural Network Explanations.” arXiv preprint arXiv:2009.04521.

[24] Rudin, C. 2019. “Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead.” Nature Machine Intelligence 1(5): 206–15.

[25] Celma, Ò. (2010). *Music Recommendation and Discovery in the Long Tail*. Springer. https://doi.org/10.5281/zenodo.6090214

[26] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD ’19), August 4–8, 2019, Anchorage, AK, USA. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3292500.3330701

[27] Kula, Maciej. (2015). Metadata Embeddings for User and Item Cold-start Recommendations. 1448.

[28] Jianghao Lin, Jiaqi Liu, Jiachen Zhu, Yunjia Xi, Chengkai Liu, Yangtian Zhang, Yong Yu, and Weinan Zhang. 2018. A Survey on Diffusion Models for Recommender Systems. In Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX). ACM, New York, NY, USA, 33 pages.

[29] Lee, H.; Kim, J. EDiffuRec: An Enhanced Diffusion Model for Sequential Recommendation. Mathematics 2024, 12, 1795. https://doi.org/10.3390/math12121795

[30] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A Neural Influence Diffusion Model for Social Recommendation. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'19). Association for Computing Machinery, New York, NY, USA, 235–244. https://doi.org/10.1145/3331184.3331214

[31] Gyuseok Lee, Yaochen Zhu, Hwanjo Yu, Yao Zhou, and Jundong Li. 2025. Collaborative Diffusion Model for Recommender System. In Companion Proceedings of the ACM Web Conference 2025 (WWW Companion '25), April 28-May 2, 2025, Sydney, NSW, Australia. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3701716.3715516