# Enabling Targeted Music Exploration with Interactive Recommendations

Alec Nonnemaker

TU Delft

XITE

# Enabling Targeted Music Exploration with Interactive Recommendations

by

Alec Nonnemaker

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday December 13, 2024 at 14:00.


| | | |
|---|---|---|
| Student number: | 4953282 | |
| Project duration: | Feb 1, 2024 – December 13, 2024 | |
| Thesis committee: | Dr. Cynthia Liem, | TU Delft, supervisor |
| | Dr. Chirag Raman , | TU Delft |
| Supervisors: | Dr. Ralvi Isufaj, | XITE |
| | Dr. Zoltán Szlávik | XITE |


Cover: Photo by Priscilla Du Preez on Unsplash

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

XITE

# Preface

As is tradition, this preface should acknowledge all the effort put into the writing of this thesis, and highlight my struggles in the noble pursuit of knowledge. More importantly, I would like to skip straight to giving thanks.

First, I would like to thank the three people I am glad to call my supervisors, Dr. Cynthia Liem, Dr. Ralvi Isufaj, and Dr. Zoltán Szlávik. This project was made a lot more achievable because of your consistent guidance.

Cynthia, from our first email to our first meeting, to every other meeting since then your enthusiasm and positivity have lit up our conversations. Your guidance and support have given me confidence throughout the whole thesis. Despite your busy schedule, you would always find time for me when I needed it, for which I am very grateful.

Ralvi, thank you for the time and effort you put into countless meetings and for listening to my, often not yet well-supported, ideas. I could always count on your unwavering honesty, keeping me to high standards. This was always paired with even more unwavering support to give me the confidence I needed. I could not have asked for a better daily supervisor, thank you.

Zoltan, again thank you for all the time and effort you put into meetings and your always insightful comments and feedback. I appreciate that you always had my back, helping me with whatever I needed, and fighting for my future job. I look forward to working with you as my manager.

I want to give thanks to all my friends and family for their support throughout this journey.

To my housemates, Kian, Youri, Bob, Remco, and Isa, for creating a relaxing environment at home that I never needed to worry about.

To my friends back in Amsterdam, Mart, Julian, Sem, Freek, and Davin, thank you for all the support when I couldn't make it for another meetup and the always great times when I could make it.

To my gym friends, Olivia, Hiresh, Kimberley, Nathan, Eli, Nicky, and many more, for all the support and for making my countless hours in the gym a time when I could relax and forget about my thesis worries.

To my parents, Ingrid and Michael, my brother Liam, and my sister Emma, for their support patience, and care.

Finally, to my amazing girlfriend Sofie, thank you for your constant support and love, for always being there to take care of me when I forgot to take care of myself, encouraging me, and making these last few months infinitely better than I could imagine.

*Alec Nonnemaker*
*Delft, December 2024*

# Abstract

Recommender systems are widely used to help users navigate vast content catalogs, but they often limit users to suggestions that closely match their existing preferences, creating ”filter bubbles” that discourage exploration. We focus on solving this problem in the context of music recommendations, helping users discover and develop new musical tastes. We embed a knowledge graph containing expert-curated metadata, user interaction data, and audio similarity features, into a representation space where similar songs are mapped closely together. This enables the system to gradually guide users from their current preferences toward a new genre through personalized recommendations. Additionally, we apply a Bayesian active learning approach to iteratively update user preference models based on feedback, balancing exploration and exploitation to ensure user satisfaction while gathering information on the user's new preferences. We conducted a user study to evaluate the approach, demonstrating that a gradual, interactive approach outperforms directly introducing users to a new genre, increasing user engagement and their affinity toward the target genre. This research highlights the value of gradual, user-driven exploration in creating better music discovery experiences. Based on our findings, we provide recommendations for industry stakeholders and discuss opportunities for future research on targeted exploration in music recommendation.

# Contents

# 1

# Introduction

Recommender systems play a big part in our current culture of online content consumption, providing suggestions on movies, music, social media content, and more. The goal of these systems is to reduce information overload and help users quickly get matched with content tailored to their preferences [60]. Modern recommender systems excel at delivering highly personalized recommendations, utilizing implicit and explicit feedback from users' historical interactions to predict their preferences. While the improvement of these systems generally increases user satisfaction [41], there is concern that they may become overly personalized. This can result in "filter bubbles" [53], where users only receive suggestions closely aligned with their current preferences, leaving little room for exploration.

To address these concerns, researchers have proposed a shift in focus toward recommender systems that support "self-actualization." Such systems prioritize helping users develop, explore, and better understand their preferences [39]. In the context of music recommendation, supporting exploration can be a particularly meaningful goal. For example, a user who predominantly listens to dance music but wishes to connect with a friend or partner who prefers country music might want a recommender system that helps them with this exploration. A straightforward approach could involve suggesting representative or popular country songs [44]. However, a big jump directly into an unfamiliar genre can feel overwhelming or discouraging, especially when the target is significantly different from the user's current preferences. Alternatively, another strategy might involve guiding the user toward their goal gradually. The recommender system could introduce intermediate recommendations that bridge the gap between the user's current tastes and their exploration target.

Additionally, we note that exploration can be viewed as a unique cold-start scenario [64], where the user's current preferences are known, but their future, yet-to-be-discovered preferences remain unknown. In a scenario where the system is gradually guiding the user toward the target genre from the user's current preferences, there are several paths the system can take. Initially, we have no information on which paths will be most effective and satisfying for the user. Incorporating feedback through interactive recommendations could further improve the effectiveness of exploration, allowing the system to identify the most enjoyable paths of exploration for the user.

## 1.1. Industry context

The work presented in this thesis was conducted at XITE, a company that develops an interactive music video platform. XITE operates on linear and interactive television networks as well as offering an on-demand streaming service through its own TV app. Besides music videos, XITE's catalog contains over 300 themed playlists curated by music experts, covering various genres, moods, and eras. In the app, users can search for music, like music videos, create their own mix of songs, and access personalized playlists based on their viewing behavior and liked videos. Personalized recommendations of playlists or videos in 'For You' playlists are made possible by a music video graph they have developed containing a mix of factual, expert-curated, and user interaction data. This combination of data has the potential to facilitate unique recommendation scenarios beyond standard recommendations.

## 1.2. Thesis objective and research questions

In this thesis, we tackle the problem of targeted music exploration. This notion is further elaborated in our research question:

*How can we effectively guide a user towards developing a new taste based on their current preference profile?*

Our hypothesis for this question has two parts. First, we believe taking gradual steps, starting with current preferences and moving towards the new goal taste, is more effective than immediately providing the user with representative songs (**H1**). Providing users with recommendations from their preference profile and then slowly growing this profile toward the target can make the journey more approachable and engaging, increasing the likelihood of success. Second, we hypothesize that integrating user feedback and interaction can further improve the effectiveness of the approach for guiding users toward developing a new taste (**H2**). Ideally, the exploration method balances exploration and exploitation. Meaning it can effectively gather information about user preferences while delivering high-quality recommendations [78, 83]

To answer our research question and confirm our hypotheses, we identify two sub-research questions that must be addressed first. These sub-questions are:

**RQ1:** *How can we develop a multi-modal data representation of songs that clusters similar songs together, enabling the discovery and development of new musical tastes?*

To enable the system to take small incremental steps from a user's current preferences toward a goal we need a feature space where this is possible. This requires creating representations of songs where similar songs are mapped closely together.

**RQ2:** *How can user feedback and interaction be integrated into the exploration process to aid users in the discovery and development of new musical tastes?*

Logically following our second hypothesis, we need to find a way to incorporate user feedback into the targeted exploration process. Though some research has been done for the targeted music exploration scenario [72, 44], none of the work has tried to apply active learning and interactive recommendation techniques to increase the effectiveness of the exploration process.

## 1.3. Contributions

The work in this thesis has several contributions to the current literature:

1. Utilizing multiple data modalities we provide a method for creating song representations that allow for nuanced continuous item-item comparisons to facilitate algorithms for music exploration.
2. We propose a novel approach to targeted music exploration where the user takes gradual steps towards a target preference while the system incorporates their feedback using a Bayesian optimization procedure.
3. We conduct user experiments to evaluate the effectiveness of our active targeted music exploration approach and gain insight into its limitations and potential applications.

## 1.4. Thesis structure

We structure this thesis by dividing it into 8 chapters. In Chapter 2, we first introduce the background on recommender systems and the relevant approaches. Following, we outline the work that has been done in the various recommender systems domains that we touch upon in this thesis. Chapter 3 discusses our work done on developing data representations that can be utilized for music exploration. In Chapter 4 we describe the methods we used and adapted for targeted exploration. Our experimental setup is described in Chapter 5, and the results of those experiments are discussed in Chapter 6. Finally, we conclude the thesis with Chapter 7, where we summarize our findings, discuss the limitations of the work, and highlight potential future practical applications and research avenues.

# 2

# Related Work

The goal of this chapter is twofold. We first introduce relevant technical background on recommender systems which are the basis of this research. The second goal is to to review and analyze related work across the various domains of recommender systems that are explored throughout the thesis.

## 2.1. Background: Recommender Systems

Recommender systems are a class of information filtering systems designed to predict the utility of items for a user, providing personalized suggestions based on those predictions [59]. The recommendations generated by a recommender system are designed to assist users in making various decisions, such as choosing what items to buy, what music to listen to, or what news to read. We use the word utility here as it is commonly used for recommendation [2]. Another term that is frequently used instead of utility is preference [19]. We will use these terms interchangeably throughout the paper.

We formally define the recommendation task as follows: Let $\mathbf{U}$ represent the set of users and let $\mathbf{I}$ represent the set of all possible items that can be recommended. The system first learns representations for user $u$ and item $i$. A utility function, $f(u, i) = y_{u,i}$, is assumed to model the usefulness of item $i$ for user $u$. The system's goal is to predict the value of $y_{u,i}$ for various user-item pairs.

User preference can be recorded as explicit feedback on previously consumed items through numbered ratings such as 0-5 stars or a binary scale in the form of likes and dislikes. Another way is to record implicit feedback through item purchases or consumption of the item .e.g watching a video or listening to a song.

This historical feedback represents the known utility values for user-item pairs. The system uses this information to generalize and estimate a utility function $\hat{f}(u, i)$. Based on this estimated function, the system computes $\hat{y}_{u,i}$ for all possible user-item combinations and generates a ranked list of $K$ items with the highest predicted utility for recommendation.

### 2.1.1. Recommendation approaches

Broadly recommender systems can be divided into three different approaches. These are collaborative filtering (CF), content-based filtering (CBF), and a hybrid approach.

**Collaborative Filtering** (CF) is an approach to recommendation that bases its predictions and recommendations for a user on items previously rated by similar users [19]. The underlying assumption is that users with similar historical interactions (such as ratings, clicks, or purchases) are likely to prefer similar items. In its simplest form, it operates on a user-item interaction matrix to make predictions for a user on unobserved items (Figure 2.2).

We make a distinction between two different categories of collaborative filtering: neighborhood-based and model-based collaborative filtering [59, 3]. Neighborhood-based approaches identify relationships either between users (user-user relationships) or between items (item-item relationships) based on the similarity of their interactions. For example, user-based neighborhood filtering recommends items to a

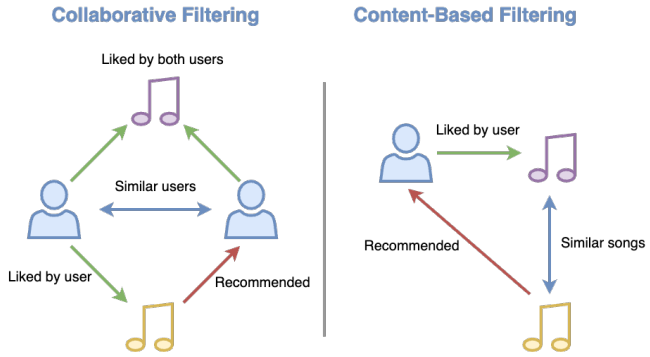



Figure 2.1

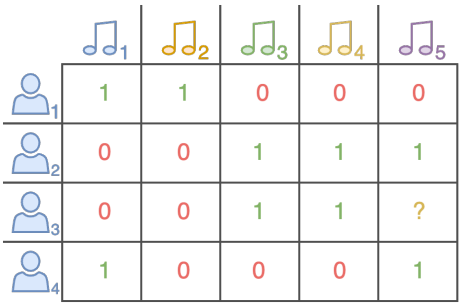



Figure 2.2

user by finding others with similar preferences, while item-based neighborhood filtering recommends items that are comparable to those the user has already interacted with.

In model-based collaborative filtering, predictive models are used to detect latent patterns within user-item interactions. A summarized model of the data is created up front, as with supervised or unsupervised machine learning methods. Therefore, the training is clearly separated from the prediction phase. These models are trained to capture complex relationships and offer more scalable solutions for large datasets. Examples of model-based methods include latent factor models, decision trees, Bayesian methods, and Neural Networks.

Collaborative filtering, in both its neighborhood and model-based forms, has proven effective in delivering relevant, often surprising recommendations tailored to users' preferences. However, CF algorithms tend to struggle in cold-start scenarios, where they lack sufficient interaction data for recommendations when presented with new users or new items. In addition, if a user has some niche unique preferences the system will not find enough genuinely similar users resulting in decreased recommendation accuracy.

**Content-Based Filtering** (CBF) recommends items to users by analyzing item attributes and comparing them to a user's historical interactions [46]. Unlike collaborative filtering, which relies on similarities between users or items, this method builds individual user profiles based on characteristics of previously interacted items. For example, in a music recommendation system, content-based filtering might suggest songs that share similar genres, artists, or moods with those a user has already enjoyed. This method works well in cold-start scenarios for new items by focusing on item features and user preferences. However, CBF still struggles with new users as the system needs sufficient information on the user's preference for content to give accurate recommendations [2]. Other drawbacks of CBF include, extracting content features for items is generally difficult and systems can often only recommend items closely related to known preferences, limiting opportunities for users to discover new or diverse content.

**Hybrid** approaches use a combination of collaborative signals and content features to leverage strengths and minimize weaknesses of the individual approaches [2, 9]. A hybrid approach combines user and item content attributes and historical user behavior data to utilize information from both types of data. For example, a hybrid approach might combine user-item interaction data with item feature data to improve recommendations for new or sparsely-rated items, addressing the cold-start problem more effectively. This combination allows systems to generate recommendations even when user interaction data is limited or when rich content attributes can improve personalization. Additionally, hybrid methods can reduce the limitations of CF's dependency on user similarity and CBF's narrow focus on known preferences, in that way offering more diverse recommendations personalized to users' unique tastes and evolving interests.

### 2.1.2. Knowledge Graphs in Recommendation

A knowledge graph (KG) is a structured network of information that models entities and the relations [25, 82]. These relationships are represented in the form of a graph, where nodes represent entities and edges capture the semantic connections between them. Formally, a KG is a directed graph $G = (V, E)$,

with entity nodes $V$ and relation edges $E$. Each edge is of the form $< h, r, t >$, indicating a relationship of $r$ from head entity $h$ to tail entity $t$. A KG-based recommender system typically has three main parts: the knowledge graph, a representation module, and a recommendation module. The KG stores the rich semantic information, that is turned into low-dimensional vectors by the representation model, after which the recommendation model calculates the user-item relations.

Typical challenges in recommender systems, such as sparse relation data between users and items and the cold-start problems [2], are difficult to fully address with just collaborative and content-based, as they rely on user interaction and complex item features. Knowledge graph-based systems can leverage rich semantic information allowing for a deeper understanding of both the content and context of the items being recommended, making them especially valuable for domains like music [6], where relationships between entities can be highly nuanced (e.g., artist collaborations, genre overlaps, subgenre correlation).

## 2.2. Music Recommendation

### 2.2.1. Music Similarity

Music similarity is a complex and highly subjective concept [80], posing significant challenges in both implementation and evaluation [80]. The debate around the ”semantic gap” between low-level music representations and high-level human understanding emphasizes the psychological nature of music perception [57, 38, 80]. To bridge this gap, researchers propose mid-level representations that combine low-level features with perceptually motivated knowledge meaning systems that aim to encode music similarity must, by definition, do so in a human-like way [80]. Examples are context-aware clustering of songs and audio features motivated by how humans perceive sound. The creation of these mid-level features requires a human-focused understanding of music, which can be found in music experts and curator teams, such as the ones employed by streaming services and radio stations.

**Music similarity evaluation:** The difficulties of defining music similarity make evaluating music similarity algorithms equally challenging. The similarity is context-dependent and based on a multitude of factors besides raw content. This often results in the absence of an objective ground truth [80]. Three main evaluation strategies have emerged [38]: using prelabeled data (e.g., genre [66]) as a proxy for similarity, human assessment of algorithm quality and analysis of user interaction data such as listening history [48] or playlists [11, 63]. Methods that rely on prelabeled data are often set up as classification or retrieval tasks and thus make use of traditional metrics such as precision and recall. The downside of this strategy is the dependence on the labeling of the data which, such as for the case of genres [66], is regularly not consistent throughout all people. Having humans assess the quality of music similarity algorithms is a potentially more accurate, though more costly, evaluation strategy. This method provides more meaningful evaluation results that align closely with human perceptions of similarity. However, relying on human judgment is both expensive and labor-intensive. Additionally, this strategy is still susceptible to biases due to the subjective nature of music perception. For instance, one person might find two songs similar because of their melodies or rhythms, while another might see them as different due to contrasting lyrical themes, such as war versus love. In an evaluation by analysis of user interaction, a retrieval task is constructed based on user listening history by splitting the collection into a training and test set [48]. The benefits of this evaluation method include its focus on the user experience and its use of real-world data. Additionally, it effectively avoids the need for explicitly labeling music or quantifying the degree of similarity between music pieces. However, in practice, users do not listen to one distinct type of music and will listen to different music depending on the context, such as their mood or time of day. To remedy this fact we can make use of another type of interaction data, user playlists. Playlists are often created to fit a theme or activity [15], making them more homogeneous in terms of music similarity compared to listening history. This allows tasks like the playlist completion task [11] to be used forthe evaluation of music embeddings [63].

## 2.3. KG embedding

The goal of Knowledge Graph Embeddings (KGE) is to simplify the processing of a knowledge graph while preserving its structural integrity and relative node relations. These embeddings map entities within a knowledge graph to vectors in an n-dimensional space for further use in downstream tasks. Vectors are often simpler and more efficient to work with than their graph structure counterparts. Dis-

tances in vector space can be determined using well-established distance metrics, and these vectors can also serve as feature vectors for subsequent machine-learning tasks. There are various different approaches for graph embedding. Below we will briefly discuss a few of them.

**Translation-based models** represent relationships between entities as translations in the embedding space. They use a distance-based scoring function and optimize the entity and relationship embeddings such that adding a relation vector to a head entity vector approximates the tail entity vector. One of the classic KG embedding algorithms is TransE [7] which represents entities and relations as vectors in the same space. For a triplet $(h, r, t)$, where $h$ and $t$ are entities and $r$ is the relation, TransE optimizes $||h + r - t||_2$. Despite its simplicity, TransE performs well on various benchmark datasets. However, it struggles with complex relations such as one-to-many, many-to-one, and many-to-many. Following TransE, numerous variants were introduced to improve on its shortcomings. TransH, TransR, and TransD have been proposed to overcome these problems by applying the relation transformation into different hyper-planes/subspaces [79, 45, 31]. Other models include ConvE [16], which applies 2D convolutional operations to the embedding vectors, and RotatE [71], which defines each relation as a rotation from the source entity to the target entity in the complex vector space. Translation-based knowledge graph embedding models, while effective, have several disadvantages. These models face difficulty in generalizing to different graph structures and are typically unable to handle edge weights, which in certain graphs convey a large amount of information.

**Graph neural network** (GNN) models are often applied to learn node embeddings by aggregating information from their neighbors. GCN [37] and its extension R-GCN [65] use graph convolution for efficient learning of node representations in large-scale graphs. GAT [74] together with its KG-specific adaptation KGAT [77] apply the concept of graph attention networks to KG embeddings, allowing them to handle more complex entity relation information and structural patterns. GNN models have shown great potential for knowledge graph embedding, however, they do suffer from limited interpretability as well as being resource intensive in terms of both computational power and memory.

**Random walk-based methods** focus on exploring the neighborhood structure in a graph in order to preserve proximity among nodes. These models ensure nodes that are close to each other in the graph get similar vector representations. One of the first of its kind is DeepWalk [54], which applies the ideas of the popular natural language processing method Word2vec to graphs instead of text. In DeepWalk, paths are created starting from a target node by uniformly choosing a neighbor of the current node as the next node for the path until a walk length $n$ is reached. For each node $m$ random walks are generated resulting in $N \cdot m$ paths of length $n$. These paths are then treated as sentences in the same way as Word2vec where using single hidden layer neural networks following the so-called skip-gram model, given a target ”word” the context, is predicted. The weights of the hidden layer are then used as vector representations of the graph nodes.

Node2vec [24] is an extension of DeepWalk where, instead of the decision for the next node to visit being completely random, a biased random walk is implemented. This is done using a weight parameter $\alpha$ that sets the probability of an edge to be traversed for balancing breath-first and depth-first graph search. This parameter in turn uses the parameters p and q that control the rate of exploration and how fast the walk leaves the neighborhood of the starting node. Parameter p controls the likelihood of revisiting the previous node in the walk. The higher the likelihood of jumping back, the more likely the random walk stays in the current neighborhood. Parameter q determines the likelihood of a depth-first search. A low value of q increases the likelihood of jumping to nodes that are further away from the previously visited node.

In metapath2vec [18] and HIN2vec [21], the random walks are guided by meta-paths which are established beforehand. While metapath-based approaches offer the advantage of incorporating domain-specific semantics through carefully designed paths, their complexity and sensitivity to path design can be significant drawbacks, especially when understanding the embedding results is of significant importance. Additionally, in the case of a graph with edge weights, it is not trivial to incorporate these weights. This is in contrast with Node2vec, where edge weights can simply modify the probability of transitions during random walks.

### 2.3.1. Multi modal KG embedding

Recently, work has been done to extend KG embedding methods by incorporating side information, such as text, video, or audio. [17] introduce W-KG2Vec, where the text-based similarity between entities is used to weigh certain metapaths in their metapath-based random-walk approach. In MKGAT [70], graph attention is first applied to aggregate information from an entity's multi-modal neighbors, which in their version of a multi-modal graph are also first-class entities. Following this, embeddings are computed in a traditional way using the transE model. In the field of music recommendation, MKGCN [14] was proposed to enhance music recommendation by additionally using the audio features of songs. MKGCN first fuses all types of multi-modal data of an entity by aggregating to obtain a multi-modal aggregated representation vector of the entity. Then a GCN aggregation layer models the high-order user and item representations by aggregating their neighbor representations layer by layer, starting from the outermost sampled neighbor. In [62], the authors opt for a simpler approach to incorporating acoustic features into knowledge graph-based recommendation to allow for the use of more explainable recommendation approaches. For the construction of the graph, edges between two different songs are created based on the similarity of their acoustic feature vector. Specifically, when the cosine similarity between $\boldsymbol{f}_i$ of song $m_i$ and $\boldsymbol{f}_j$ of song $m_j$ is greater than a threshold, a new edge is defined between their two nodes. Intuitively, the minimal changes to the graph allow you to use any KGE approach on top of it, including the one most appropriate for your goal.

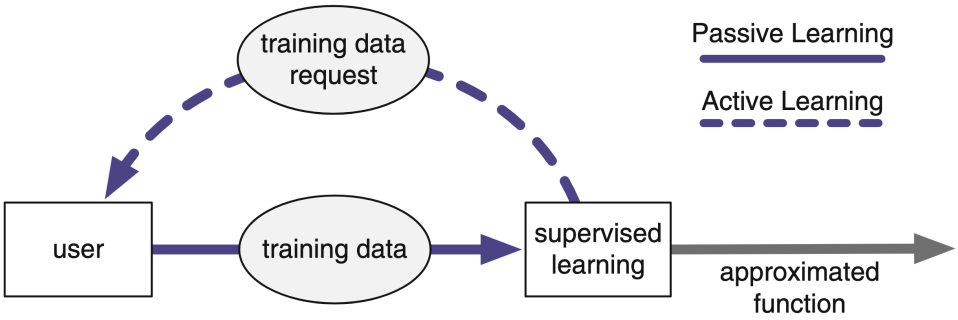



**Figure 2.3:** In active learning the system interactively/iteratively elicits training data from the user to refine its preference models [61].

## 2.4. Active Learning for Preference Elicitation

Efficient preference elicitation is essential in cold-start scenarios when the user model is not detailed enough to provide accurate personalized recommendations. Managing a lack of initial information about a new user is typically described as a cold-start problem. However, this problem is not unique to the first usage of a recommender system. Deliberately sending users to areas in the item space previously unknown to them may also encounter the challenge of insufficient information about their preferences. This cold-start problem is often tackled with Active Learning [61]. Active Learning (AL) is based on the concept that a machine learning algorithm can achieve higher accuracy with fewer labeled training examples if it is allowed to choose the training data from which it learns. In the context of recommender systems, this is achieved by allowing the system to influence the items a user is exposed to, enabling it to learn user preferences more efficiently (Figure 2.3).

Traditional AL recommendation strategies approach this problem by first splitting the process up into a preference elicitation (exploration) phase and a recommendation (exploitation) phase [30]. In the exploration phase, the system questions the users or recommends items to the user to maximize diversity [34], reduce the system's uncertainty [58], or reduce the model's error [23]. These methods have proven effective in reducing the number of interactions required to build an accurate model of user preferences. However, fully focusing on exploration for the initial recommendations could reduce the overall enjoyment of the user, even prompting the user to leave the service before their profile has been established [67]. An effective algorithm should balance exploration-exploitation, allowing the system to effectively gather information about user preferences while delivering high-quality recommendations. Bayesian approaches [75, 32] are particularly useful here, as they maintain a belief state over the utility of items that can be sampled and updated with a query selection strategy such as Thompson Sampling

[83], that naturally and dynamically balances exploration and exploitation of user preferences [78, 49].

Thompson Sampling's [10] simplicity and effectiveness make it a popular choice for applications such as multi-armed bandit frameworks [69] and personalized interactive recommendation systems [26]. In this strategy, the system maintains a probability distribution (posterior) over the parameters of each item's utility. To select an item for recommendation, it samples from these distributions to estimate the potential reward for each option and select the option with the highest sampled reward. This method naturally incorporates exploration, as less certain options with wider distributions are more likely to be sampled, while also exploiting known high-reward options as more data is gathered.

A limitation of Thompson Sampling is that when the set of possible actions is large or the number of rounds to elicit feedback is small it runs into sample efficiency problems. This means the algorithm cannot converge given the amount of data received. Several approaches have been proposed to address this limitation. In Collaborative Thompson Sampling [85] users are clustered into groups, and the feedback of all users in the same group is used to estimate the expected reward of an item. For a content-based approach similarity information can be used from items embedded into a representation space [12, 68, 84, 4, 81] to efficiently process feedback from sampled items.

## 2.5. Recommender Systems for Targeted Music Exploration

The role of recommender systems since their inception has been to help prevent information overload for users navigating the large amount of content available to them. While current systems effectively create personalized spaces for users, there is concern that these may be overly personalized, resulting in ”filter bubbles” [53]. This prompted researchers to think about the recommendation problem *”beyond accuracy”* [28] and consider evaluating recommender systems from the perspective of user experience [41, 40]. Systems should not just optimize for the highest utility items given the current data, but give the users control over the recommendation process [29] so that they can use these systems to support the development and exploration of their own unique tastes and preferences [39, 36].

### 2.5.1. Supporting Discovery and Exploration in Music Recommendation

In the music domain, supporting discovery has been identified as essential for improving user satisfaction and engagement [42, 22]. As such, several works have focused on ways to assist and encourage users to explore and develop new tastes.

One approach is to provide interfaces for users that visualize how their music tastes relate to the rest of the music space. In *Island of Music*, songs are visualized on a 2D map representing an artificial landscape [52]. Going further, *Music Tower Blocks* contains a 3D visualization in an interface where users can search, filter, and connect their personal streaming profiles to support manual exploration.

Another approach is to incorporate interactivity into the recommender systems letting users actively modify their own recommendations. In *TagFlip* [35], users can specify social tags that are associated with the next song. *TagFlip* was perceived to enable more control and transparency over recommendations, compared to the mobile Spotify interface. *TasteWeights* lets users get insight into and adjust the weights in a hybrid recommender system to get artist recommendations [8]. Combining visualization and interactive interfaces, *TastePaths* [55] helps users understand genre relations by presenting an overview of the genre landscape as a clustered graph of related artists. Evaluating between a personalized and non-personalized version they find that users prefer having their exploration ”anchored” by their personal music profile.

### 2.5.2. Targeted Music Exploration

Rather than giving users the freedom to discover without guidance, recent studies have looked at discovery in a new scenario where users set a goal to learn a new taste. In music recommendation, this can mean choosing a genre to explore.

In [44] users are presented with playlists that immediately introduce users to a new genre, but with various levels of personalization. Experimenting with representative, personalized, and mixed genre playlists they found that balancing personalization and genre representativeness could be important for effective new genre exploration. However, these effects were observed in a single-session study, whereas developing new preferences often takes time. Therefore, in a following work, the authors con-

duct a longitudinal study on users' exploration behavior and behavior change over time after using a music genre exploration tool for four sessions. In the study, users were randomly assigned to a more personalized or a more representative initial playlist for the first session. In sessions 2-4, participants used a trade-off slider for adjusting the recommendation personalization level, from the most representative to the most personalized, to adapt their playlist for the session. Experiment results show that the users perceived the system to be more helpful when their slider positions were set to more helpful in sessions 2 and 3. This suggests that giving the user access to a personalized trade-off slider for exploration allows them to explore a genre effectively. These studies hint at the potential for effective exploration of having recommendation playlists that become gradually less personalized and more representative but do not investigate this scenario. Additionally, recommendations always start within the target genre which could still be a significant jump if that genre is far away from current preferences. Finally, the generated playlists are only affected by the personalization-representativeness trade-off in the algorithm's content-based representation space, ignoring all other dimensions that could be explored to increase the effectiveness of recommendations for a user during exploration.

In other work [72], exploration of a new genre is guided by letting the user take gradual steps toward the target genre. The system identifies the shortest path from the user's current preferences to the target genre in a user similarity graph, with nodes corresponding to users and edges representing user-user similarity. This path consists of user nodes where the target genre is represented by a 'destination user' who is the best match for the selected target genre. For each user along the path, the top three artists matching best with the target genre will be selected and these are assembled as a sequence of recommended artists for exploration. Taking gradual steps benefits the users as they can reason how these recommendations relate to their tastes. However, it has some shortcomings limiting its effectiveness. The algorithm is mainly collaborative which in music recommendation has been shown to lack the depth of information needed for accurate music similarity. This can result in recommendations becoming noisy and hard to understand between the user's current preferences and the target. Furthermore, the path generated is only the shortest path from the user's current preferences towards the target, which does not necessarily mean that it is the most effective path to follow. Finally, the discovery process starts and ends with the generation of the list of recommended artists, not allowing the user any control in adapting the recommendations while they are following the path to their goal.

# 3

# Data Representation

In this chapter, we explore how multi-modal data representations can support targeted exploration algorithms in music recommendation systems. To achieve effective targeted discovery we guide the user toward their new music preferences by taking small, incremental steps in a feature space. This requires representations where similar songs are mapped closely together, allowing users to seamlessly transition from their current preferences to new discoveries.

In our approach we focus on integrating three key sources of information: expert-annotated content, user interaction data, and audio-based features. By combining these data modalities, we aim to create a richer, more nuanced representation of music that accurately reflects its inherent similarities. The ultimate goal is to ensure that users can be effectively guided toward their target preferences in a structured and intuitive manner.

## 3.1. Overview of Data and Modeling

### 3.1.1. Expert Based Collaborative Music Knowledge Graph

To organize the rich semantic content and relationship information that we intend to exploit, we utilize a Knowledge Graph (KG) constructed from a meta-data database. This database contains a combination of factual data, such as artist and decade, expert-curated data, and user interaction data. To manage these diverse relational structures accurately, distinct edge types are assigned unique weights contingent upon their relative significance. An overview of the graph structure can be found in Figure 3.1. In the following, we will outline the various kinds of data contained in the graph.

Genre and subgenre are two fundamental attributes that have been used to classify pieces of music based on common characteristics. These traits can include musical elements such as rhythm, tempo, instrumentation, and lyrical themes, as well as other factors such as cultural influence or intended audience. Songs can be categorized into genres through numerous methods, and there is no universally accepted definition for these genres, leading to extensive and highly subjective taxonomies. However, expert music curators are trained to consider a wide range of factors when developing and assigning songs to a taxonomy that is generally agreed upon by users. The subgenre nodes additionally include edges between them for subgenre correlation, capturing pairwise similarity between subgenres, as perceived by music experts, via an iterative manual process.

Next, we address curated playlists, which are collections of songs grouped based on shared attributes or a common theme, such as *90s Hip-Hop*, *Country Today*, or *Sing Along*. Unlike genres, which are typically more rigid in their classification, curated playlists offer greater flexibility in both creation and song assignment. New playlists can be generated at any time, and individual songs can belong to multiple playlists simultaneously. This overlap permits a more dynamic and versatile approach to organizing music collections, allowing for a broad range of dimensions for similarity.

After processing the expert-curated data, the graph is transformed into a Collaborative Knowledge Graph (CKG) with user interaction data, denoted by *like* and *play* edges connecting users and songs.

This allows us to exploit the advantages of CF-based recommendation, such as capturing measured collective preferences of users and identifying hidden latent factors driving user-song interactions.

The knowledge graph enables the discovery of latent relationships between songs via multi-hop connections. A notable feature of the graph is that it is essentially homogeneous in terms of the song nodes. Nearly all other types of nodes, with small exceptions, such as subgenres through subgenre correlation and hierarchical genre connection, are only connected to song nodes. In this way, these types of nodes can more so be seen as relational nodes, where, for example, two song nodes connected to the same genre node can be seen as a 'same genre' relation. Therefore, even though songs are not directly linked, the presence of a large number of short paths between two songs may indicate their semantic similarity and suggest that they should be grouped together. This allows relatively simple walk-based graph embedding models to accurately model node neighborhoods.

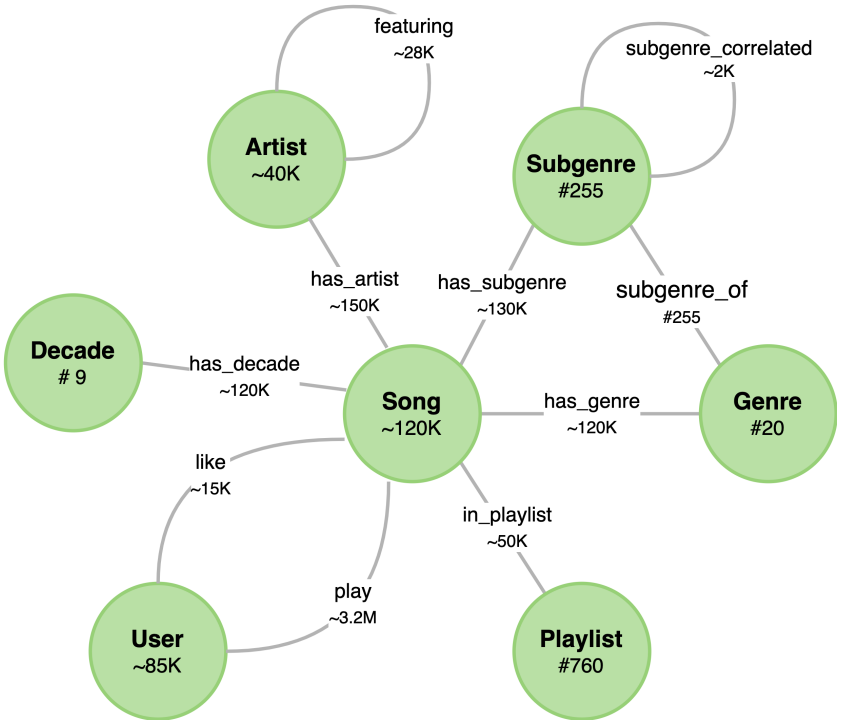
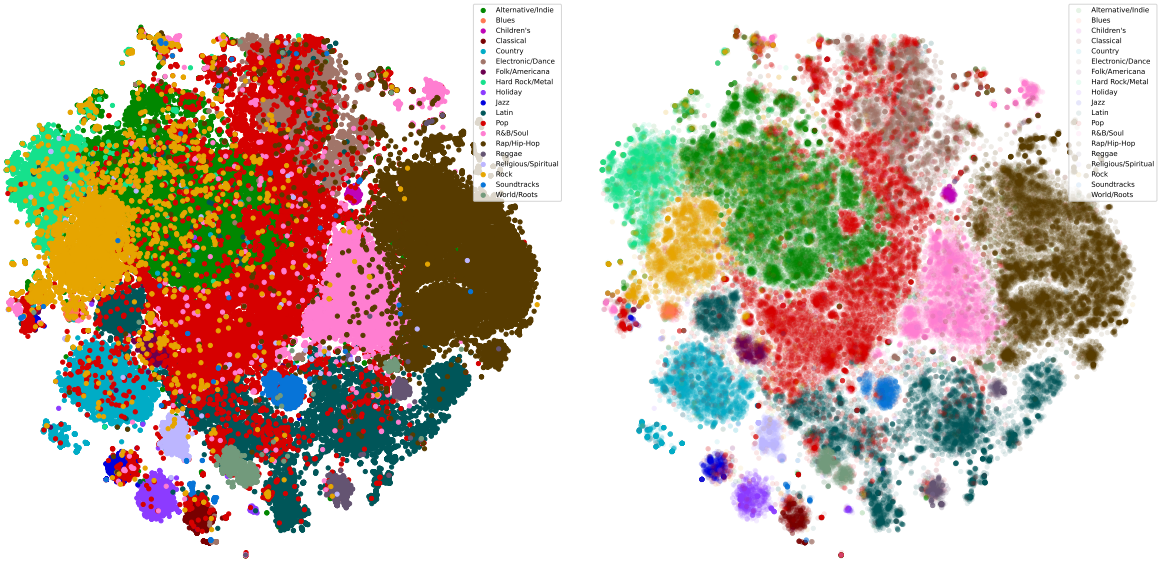



**Figure 3.1:** Graph schema for the full CKG with occurrence counts.

### 3.1.2. Enhancing Music KG through Audio Similarity Information

Up to this point, our CKG contains expert-generated semantic data and user-interaction data. Next, we explore the potential of a multi-modal approach by incorporating audio embedding information in our similarity computations. Low-level content representations, while on their own usually not sufficient for music understanding, can potentially aid our song representation with subtle patterns not picked up or recorded by the current combination of expert and user-interaction data.

We enhance the graph with song-song audio similarity edges, connecting nodes directly if one song is among the top 10 nearest neighbors of another based on audio representation, using the Faiss library for efficient similarity search. A direct edge between two song nodes allows for transitioning directly between similar songs without needing to pass through intermediary nodes like genre or artist. We regulate the influence of audio similarity by scaling edge weights proportionate to their cosine similarity and experiment with different scales. The MULE (Musicset Unsupervised Large Embedding) [47] model, which uses contrastive learning through the SimCLR objective on log-mel spectrograms, provides the audio embeddings. MULE, trained on the extensive Musicset dataset, has demonstrated state-of-the-art performance in music understanding tasks and can be directly applied to our data without additional fine-tuning.




(a) (b)

**Figure 3.2: (a)** A t-SNE visualization of the node2vec embeddings showing the separation of genres in the representations. **(b)** A reduced opacity ($\alpha = 0.1$) t-SNE visualization mostly emphasizes coordinates where a genre is densely located thereby filtering out overlapping outliers.

### 3.1.3. KG embedding

To enable the music exploration algorithm to effectively guide discovery and recommend songs, we embed the knowledge graph into a vector space where the relationships and similarity between songs are preserved. For this, we utilize Node2vec [24], an embedding technique that balances computational efficiency with the ability to capture nuanced graph relationships. Node2vec offers flexibility by allowing us to control how the algorithm explores the graph, ensuring it captures both local connections (like songs in the same genre) and broader patterns (like shared user preferences). Furthermore, Node2vec supports weighted edges, allowing us to integrate the varying importance of relationships within the graph, such as subgenre correlations or audio similarities. Its scalability and simplicity make it well-suited for large-scale graphs, ensuring computational efficiency while maintaining high-quality embeddings.

## 3.2. Evaluation of Data Representations Methods

### 3.2.1. Visualization

Visualizing embeddings is a valuable first step in evaluation as it provides an intuitive and immediate sense of how well the embeddings capture and preserve relationships among data points. As a fundamental requirement for our final discovery setting, it is crucial that groups of songs that may serve as a target, such as genres, are sufficiently separated in their representation. Unfortunately, our embeddings are high-dimensional vectors, therefore we cannot directly visualize them. For this, we need dimensionality reduction techniques to transform them into a 2D space.

To perform dimensionality reduction, we apply the t-Distributed Stochastic Neighbor Embedding (t-SNE) approach [73]. t-SNE is often used to visualize high-dimensional embeddings by reducing the number of dimensions while preserving the local structure and neighborhood relationships. This helps in revealing whether the underlying structure of the representations follows our fundamental requirement of sufficiently separating groups of songs, such as genres.

In Figure 3.2a we visualize our reduced-dimensionality embeddings in a 2D scatterplot, with genres labeled as colors. We can see a clear separation between genres, yet also some overlap. However, looking at the biggest overlaps, they generally are understandable. Pop music bleeds into multiple clusters, such as Country, Latin, and Electronic/Dance. Pop music is known to borrow elements from other styles [20] such as these named, which means that two relatively similar songs could be classified

differently into, for example, Pop and Country. Other overlaps include Rock into Hard Rock/Metal and Alternative/Indie, and Rap/Hip-Hop into R&B/Soul. In Figure 3.2b we reduce the opacity of the points to get more insight into the density in their clusters. If the genre color is still visible with the reduced opacity, the density of that genre at those coordinates is high. We can see here that the overlap between the genres is strongly reduced while the clusters remain visible. This would indicate that the overlap seen in the original t-SNE visualization are more likely to be outliers than a structural misrepresentation of the embedding model.

Overall this initial visual evaluation shows promising results for the effectiveness of the representation method in separating genres. However, this can be seen as an expected result as genre information was included in the training data as nodes in the KG. Additionally transforming the data and visualizing it does not necessarily result in an objective measure of embedding quality. For a more sound evaluation, we need a quantitative approach and a measure of similarity that is more easily separated from the data.

### 3.2.2. Playlist Completion

To quantitatively evaluate the quality of our KG song embeddings, we utilize the playlist completion task [63, 11]. The assumption for this task is that a playlist of songs is created by a user to fit a certain grouping of songs (e.g. genre, mood, task), making the songs in the playlist similar in at least one way. Experiments for this task can be done offline and involve recommending related songs given a number of songs from a playlist. The playlist completion task is as follows. For a playlist with $n$ songs, we randomly sample $n \cdot x\%$ of songs to be the seed and calculate a seed embedding by averaging over them. Next, we retrieve and rank the $n \cdot (100 - x)\%$ most similar songs to the seed embedding $z_s$ from all available song embeddings using the Faiss library [33]. Finally, this ranking is evaluated against the remaining non-seed songs from the playlist, using traditional ranking metrics such as Precision and Normalized Discounted Cumulative Gain (NDCG). We intend to fill the playlist up to its original size, meaning we use the top $n \cdot (100 - x)\%$ ranked songs as the playlist completion set returned by the embeddings.

Precision measures the proportion of correctly retrieved songs in the set returned by the embedding, focusing on the presence of relevant (non-seed) songs without considering their order. This makes it useful for simply evaluating the accuracy and ”cleanliness” of the embeddings in terms of matching the intended playlist content. In contrast, NDCG (Normalized Discounted Cumulative Gain) considers not just the presence but the rank order of retrieved songs, reflecting how well the most relevant songs are prioritized. Higher NDCG values indicate that relevant songs are not only retrieved but appear near the top.

We evaluate our embeddings on two datasets containing playlists. First, we use the curated playlist used to create the embedded CKG (Section 3.1.1). The knowledge of these music experts allows them to create playlists that feel coherent and cohesive to the common user. In creation of these playlists the music experts can directly see the meta-data of individual songs to aid in finding matching songs to add. Since the playlist creation and rest of the meta-data that are in the CKG are so closely linked, evaluating on a test set of these playlists gives us insight into how well the embedding models capture different, sometimes overlapping, clusters of songs that are semantically similar in some way. We take $20\%$ of the playlists in the dataset as a test set and remove them from the graph before training the embedding model.

Next, we evaluate on actual user created playlists from a Spotify dataset [56]. The dataset is created by crawling the playlists of Spotify users that share which songs they are listening to using #nowplaying on the popular social media platform X (formerly Twitter). The dataset consists of <user, track, artist, playlist>-quadruples with 15,345 unique users who listened to 1,878,457 unique tracks by 276,848 unique artists contained in 143,528 unique playlists. We match the track instances of this dataset with songs in the CKG data by exactly matching the song title and artist name which leaves us with 7,405 playlists containing 20,007 unique songs.

| | Graph | Precision | NDCG | Spotify Precision | Spotify NDCG |
|---|---|---|---|---|---|
| 1 | Full CKG | 0.4978 | 0.7488 | 0.0167 | 0.0963 |
| 2 | Full CKG + audio similarity (weights scaled to 0.5) | 0.5774 | 0.8631 | 0.0196 | 0.1101 |
| 3 | Full CKG + audio similarity (weight scaled to 0.25) | **0.6455** | **0.9096** | **0.0200** | **0.1154** |

**Table 3.1:** Average results of the Playlist Completion task with node2vec embeddings on full CKG compared with and without audio similarity edges, based on six runs.

## 3.3. Results

To determine the hyper-parameters for the Node2Vec implementation, we run a grid-search on the full CKG optimizing $d \in [8, 32, 128]$, $walk\_length \in [10, 20, 40]$, $context\_window \in [5, 10]$, and $p, q \in [0.5, 1, 2]$. We select $d = 128$, $walk\_length = 20$, $context\_window = 5$, $p = 0.5$, $q = 0.5$ for all graph versions and train for $200$ epochs. All models are run using the *fastnode2vec* [1] implementation configured to handle edge weights.

The results of the playlist completion evaluation task are presented in Table 3.1 and discussed below in terms of their implications for the project. We report Precision and NDCG to evaluate the quality of our model. Precision measures the proportion of songs returned by the embeddings that are in the playlist originally, meaning they are correctly retrieved. NDCG further evaluates the quality of the embeddings by valuing correctly retrieved songs that are placed higher in the rankings. Notably, NDCG is higher than Precision for all cases. This indicates that even in cases where the total mix of relevant songs in the playlist completion set is not as high, the relevant items are still ranked near the top of the set.

### 3.3.1. Evaluation on Curated Playlists

The first evaluation was performed on the curated playlist dataset. These playlists were created by experts and are highly structured, which makes them a good benchmark for testing how well the embeddings capture meaningful relationships between songs. Using the embeddings from the full CKG, we achieved a baseline Precision of 0.4978 and an NDCG of 0.7488. These scores show that the CKG embeddings are capable of identifying semantically similar songs based on the metadata relationships encoded in the graph.

When we added audio similarity edges to the graph, scaled with a weight of 0.5, the results improved noticeably. Precision increased to 0.5774, and NDCG rose to 0.8631, suggesting that including audio similarity helped the embeddings better capture relationships that are not apparent in the metadata alone. Reducing the weight of the audio edges to 0.25 led to further improvements, with Precision reaching 0.6091 and NDCG increasing to 0.8745. This indicates that combining metadata and audio features in a balanced way can enhance the quality of the embeddings, making them more effective for tasks like playlist completion.

### 3.3.2. Evaluation on Spotify Playlists

The second evaluation used the user-generated Spotify playlists. These playlists are less structured and often reflect individual user preferences, making them much harder to model compared to the curated playlists, especially given the structured nature of the data on which the embeddings were trained. For the embeddings generated from the full CKG, the baseline Precision was 0.0167, and NDCG was 0.0963. These scores are much lower than those for the curated playlists, which is expected given the noisy and diverse nature of user-generated playlists.

Adding audio similarity edges improved the results here as well. With audio edges scaled to 0.5, Precision increased to 0.0196, and NDCG improved to 0.1101. When the weight of audio edges was reduced to 0.25, the best results were achieved, with Precision rising to 0.0200 and NDCG to 0.1153. Although the overall performance is still low on this dataset, these improvements suggest that audio features can help fill in gaps where metadata alone fails to capture meaningful relationships.

### 3.3.3. Key Observations and Implications

The results show that combining user interactions and expert metadata from the CKG with audio-based similarity helps improve the quality of the embeddings in both datasets. The curated playlist dataset

benefited the most, as the embeddings already had a strong structure from the metadata, and the audio features provided additional fine-grained relationships. In the Spotify dataset, while the performance was lower overall, adding audio features still resulted in consistent improvements, which is promising given the noisy nature of the data.

Interestingly, scaling the weight of the audio edges to 0.25 provided the best performance across both datasets. This suggests that while audio features are valuable, they work best when combined with metadata in a balanced way. Overweighting audio similarity may make the embeddings less focused on the broader semantic relationships encoded in the graph.

In this chapter, we have demonstrated the ability of our knowledge graph embedding approach to effectively cluster songs by combining metadata and audio features, creating a rich and nuanced representation space. This enables smooth and meaningful transitions between user preferences and target genres, forming the backbone for our exploration algorithms. Our evaluation results confirm that this approach supports effective, user-driven exploration.

# 4

# Targeted Exploration

This chapter focuses on the methods and techniques developed to enable users to gradually transition from their current musical tastes to a target genre, offering a personalized and engaging discovery experience. Our objective here is to effectively define transitions within the representation space that connect user preferences to the target genre. To achieve this, we leverage the song representations discussed in the previous chapter. Several challenges must be addressed to facilitate effective exploration in this representation space, including the sparsity of data, and diversity of user preferences and target genres.

We will first address the challenges of navigating high-dimensional spaces using graph-based approaches (Figure 4.1). Next, we will introduce an active learning approach to guiding users toward their exploration targets (Figure 4.3). Together, these methods aim to enable a smooth and adaptive discovery process that incrementally guides users towards their target preferences while incorporating their feedback along the way.

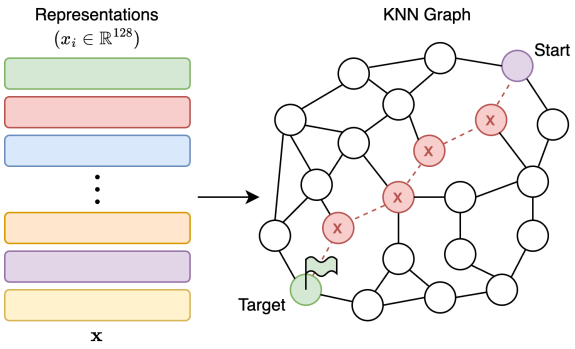



**Figure 4.1:** We turn 128-dimensional song representations into a KNN graph where we can set appropriate start and target nodes. This allows us to move between them through nearest-neighbor edges.

## 4.1. Moving through the feature space

The feature space which contains all song representations should allow us to take the small steps needed to effectively guide the user to the target preference. Unfortunately, since the feature space is high-dimensional we run into the Curse of Dimensionality. This refers to the exponential increase in data sparsity and computational complexity as the number of features (dimensions) grows. Especially this data sparsity becomes a problem when trying to take steps through the feature space. With the data being so sparse a very small amount of songs would align themselves sufficiently between the start and end locations such that they get sampled when taking steps between these locations. A common solution to the Curse of Dimensionality problem is to do dimensionality reduction [76]. However,

experimenting with various ways of doing dimensionality reduction we found that reducing dimensionality led to a significant degradation of representations' performance on the playlist completion task we could not accept. As an alternative to traditional dimensionality reduction techniques, we chose to create a K-Nearest Neighbor(KNN)-graph where steps can be taken by following edges between the song nodes. This approach parallels our method of evaluating representations through the playlist completion task. Similarly to the evaluation task, we employ the Faiss library [33] to retrieve nearest neighbors for a query, in this case a song. This way, we can increase confidence that the performance observed during the representation evaluation task translates effectively to our discovery setting. Specifically, we construct the KNN graph with each song node connected to its 10 closest neighbors in the representation space. The choice of using 10 neighbors (K=10) balances sufficient connectivity and avoiding overly dense graph structures, which could obscure meaningful relationships between songs.

### 4.1.1. Defining Start and Target

In our final representation space, we need to define both the starting position and the target position for the recommendation algorithm to gradually move between them. To this end, we need to characterize both the user and the target by some point or collection of points in the space. User preferences are often not uniform, with individuals often listening to and liking several different styles of songs that can be dissimilar from each other. For example, a user might enjoy music from both the Country and the Rap/Hip-Hop genre. Similarly, targets preferences such as genres are often comprised of several distinct sub-styles/genres, for instance, the Rap/Hip-Hop genre contains the sub-genres "Southern Hip-Hop" and "Pop Rap". We want to make sure that these subgroups are well represented in regards to the starting and ending locations because any of these user-target subgroup combinations could be the optimal one for maximizing recommendation effectiveness and user satisfaction.

We represent these subgroups by identifying central nodes within their subgraphs, enabling the creation of paths towards these groups. To ensure diversity, we select these central nodes such that each is a sufficient distance away from previously chosen nodes. This approach helps capture the centers of distinct clusters within the subgraph. For user preferences, these central nodes serve as starting points, while for the target genres, they act as destinations. To identify central nodes, we use the PageRank algorithm [51]. This process generates a starting set of user nodes $S_{start} = \{s_1, ..., s_m\}$ of size $M$, and a target set of nodes $S_{target} = \{t_1, ..., t_n\}$ of size $N$.

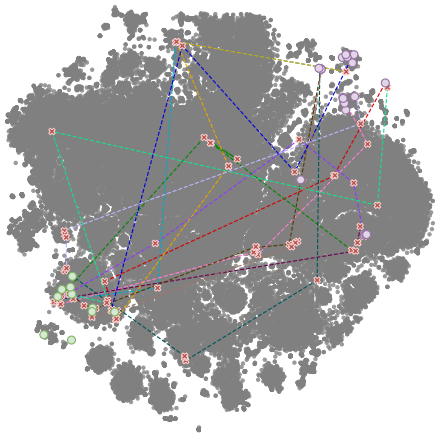


**Figure 4.2:** Visualizing a subset of paths between start nodes ⊙ in the top right and target nodes in the bottom left with t-SNE [73].

### 4.1.2. Generating Paths

To give the user the possibility of moving through the graph using any of the combinations of user-target subgroup combinations we generate paths $P$ with

$$P = \{p(s,t) \mid s \in S_{start}, t \in S_{target}\}$$

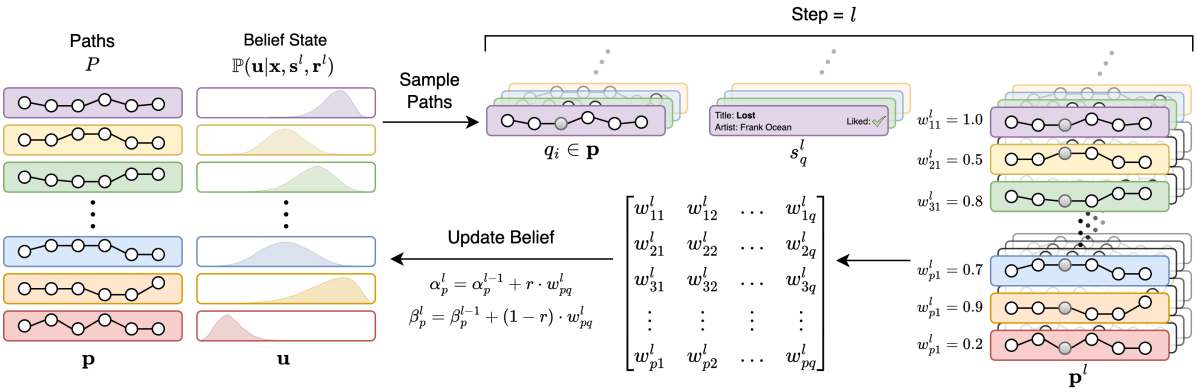



**Figure 4.3:** An overview of the the active learning of path utilities algorithm for targeted exploration. We maintain a belief over every path in the form of a posterior distribution. These posteriors are sampled to get a set of paths to recommend and songs at the current step are presented to the user. Following user feedback we update all beliefs with $w$ based on their similarity to the sampled paths at the current step.

where $p(s,t) = [s, x_1, x_2, ..., x_l, t]$ is a path standardized for length $L$ between $s$ and $t$ in the graph. These paths are created by finding the shortest path with Breadth-First Search and then standardized. If the shortest path is shorter than $L$ nodes are inserted that connect two consecutive nodes in the path. In the case that the path is longer than $L$, we keep searching for and removing node $x_i$ where $x_{i-1}$ and $x_{i+1}$ have the highest cosine similarity between their embedding. Both of these actions are repeated until the path is of length $L$.

We now have paths $p_i \in P$ where every subsequent song $s \in S_p$ in a path is closer to the target than the previous songs on the path, based on their representation $x_s \in \mathbf{x}$. This means following the path you get gradually closer to the target that has been set for discovery. In Figure 4.2 we use t-SNE [73] to visualize an example subset of paths in a 2-dimensional representation space. T-SNE tries to preserve local relationships between points by keeping close neighbors together while reducing dimensions. This focus on neighbors parallels our KNN graph, which means we can use its visualization as a way to get an intuition for how the paths run through the graph.

## 4.2. Active learning for targeted exploration

As we gradually take steps between the current preferences and the target, the user is likely to encounter unfamiliar areas of music. Since we know little about the user's enjoyment of these areas it is unclear which paths will result in the most enjoyment for the user while discovering these unexplored areas. To address this uncertainty, we generate a large number of possible paths and dynamically adjust which ones to follow based on user feedback. This process creates an exploration-exploitation trade-off: we need to explore a diverse range of paths to discover new preferences, while also focusing on paths the user has already shown to enjoy. To balance this trade-off, we employ a Bayesian Optimization approach in a novel application of path-following algorithms. At each step, the algorithm maintains a Bayesian belief state that estimates the utility of a given path for the user.

### 4.2.1. Path Utility Beliefs

#### Prior Beliefs

Before any recommendations are made we set a prior belief $\mathbb{P}(\mathbf{u})$ over all paths $p$ on their utility to the user. We assume the prior for each utility $u_p$ is a Beta distribution

$$\mathbb{P}(u_p) = Beta(\alpha_p^0, \beta_p^0). \tag{4.1}$$

Initially we have no information on the utility of these paths so we initialize them with a uniform Beta prior of $Beta(1,1)$. Beta distributions lie in the domain $[0,1]$, which makes them suited for the context of recommendation where feedback of the user often comes in the form of a like or dislike of an item. For paths of songs we can interpret a utility value of $u_p = 1$ as the user enjoying the whole path, while

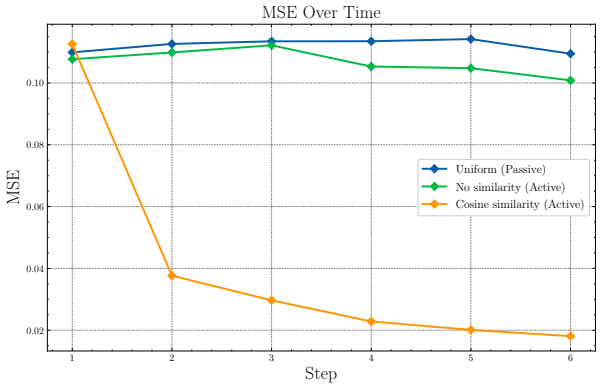



**Figure 4.4:** Comparison of efficient TS method using cosine similarity with traditional TS and passive approach with no belief updating on a simulation exploration setting.

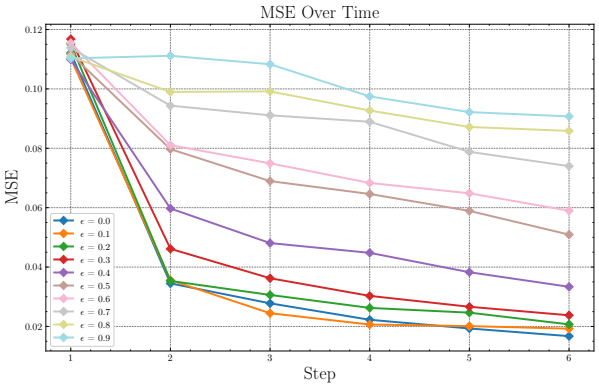



**Figure 4.5:** Results of experimenting with threshold $\epsilon$ for updating remaining paths.

values $u_p = [0, 1)$ represent some various strength of non-complete enjoyment starting from complete dislike.

### Sampling Based on Belief

We update the utility belief by incorporating observed responses $\mathbf{r}^l$ to the songs $\mathbf{s}^l$ available at step $l$ across all paths. This requires modeling the likelihood of these responses, expressed as $\mathbb{P}(\mathbf{r}^l|\mathbf{x}, \mathbf{u}, \mathbf{s}^l)$. In our recommendation setting, the response we can observe from the user is binary with $r^l \in (0, 1\}$, where 1 means the user liked the song and 0 means the user disliked the song. We can now simply model the likelihood as

$$\mathbb{P}(r_p^l|x_s, u_p, s_p^l) = Bernoulli(u_p), \tag{4.2}$$

where $x_s$ is the representation of song $s_p^l$, sampled at step $l$.

Due to the conjugacy rule, when the prior is a Beta distribution (4.1 and the likelihood is Bernoulli (4.2, the posterior distribution remains a Beta distribution with updated parameters based on the observed successes and failures. This means our posterior looks like

$$\mathbb{P}(u_p|x_s, \mathbf{s}_p^l, \mathbf{r}_p^l) = Beta(\alpha_p^l, \beta_p^l), \tag{4.3}$$

where $\mathbf{r}_p^l$ are historic rewards from songs $\mathbf{s}_p^l$ up until step $l$. To use the posterior distribution for sampling paths and updating our belief of their utility we adapt the popular Thompson Sampling (TS) strategy. With TS, we explore more when beliefs have a higher uncertainty and exploit more as the system becomes more confident. TS takes a sample of each paths utility $u_p$ from the posterior. In the typical case, TS selects the item with the highest sampled utility. For our case we select the top 20 paths with the highest sampled utilities and recommend these to the user as independent songs.

### Efficient Belief Updating

Using traditional TS, observed feedback for a path $p$ would update the posterior distribution for that path. At step $l$, if a like (reward $r = 1$) or dislike (reward $r = 0$) is observed for song $s_p^l$ of path $p$ the posterior distribution of $u_p$ is updated to be

$$Beta(\alpha_p^l + r, \beta_p^l + (1 - r)) \tag{4.4}$$

As we have limited steps we want to improve the sample efficiency of TS. We do this by not only updating the observed path put also updating all other paths based on their similarity to the observed path. For this we introduce the weight parameter $w_{pq}^l$, calculated by the cosine similarity between songs

$s_p^l$ and $s_q^l$, which are songs of paths $p$ and $q$ at step $l$. This way the posterior distribution update of $u_p$ becomes

$$Beta(\alpha_p^l + r \cdot w_{pq}^l, \beta_p^l + (1 - r) \cdot w_{pq}^l) \tag{4.5}$$

where $q$ is a sampled path for which we received user feedback on song $s_q^l$.

This efficient belief updating should lead to a faster convergence to the actual path utilities, compared to the traditional way of TS. To demonstrate this we simulate a targeted exploration setting using user historical interaction data. In this example the user almost exclusively enjoys *Rap/Hip-Hop* and *R&B/Soul* music, and has set the *Country* genre as their target for exploration.

We generate the set of standardize paths P for the user to move through. As a ground truth for every path we set a true success rate between 0 and 1. For the simulation we pick one path to be the perfect path for the user to move through with a true success rate of 1. All other paths success rate are then set based on the average cosine similarity of their songs at every step in the path to the songs of the same step in the perfect path. This way highly similar paths will also have a close to 1 true success rate.

In the simulation we sample the paths exactly like the real case by sampling from their posterior distribution and selecting the top 20 sampled utilities. Then to simulate user feedback we sample a binomial distribution using the paths' true success rate as the probability of success. We evaluate by using the sampled path utilities and calculating their Mean Squared Error (MSE) with the ground truth success rates. As can be seen in Figure 4.4, efficiently updating the belief by updating remaining paths based on cosine similarity achieves a large improvement in speed of MSE reduction compared to the traditional TS approach without similarity. As a baseline we also show the case where all path posterior distributions stay uniform throughout the steps meaning we do not take user feedback into account.

Additionally, we experimented with controlling the threshold of cosine similarity to the sampled path for when to update a remaining path, using the hyperparameter $\epsilon$ as the threshold number. For example, a threshold of $0.5$ means only paths for which the current step songs similarity has a cosine similarity with the sampled path current step song of higher than $0.5$ will be updated using Equation 4.5. Results for this experiment can be seen in Figure 4.5. We can see as the threshold goes down the reduction in MSE becomes greater and steeper. A threshold of $0$, meaning all remaining paths get updated, delivered the best result, which meant we decided on keeping the threshold at $0$ for the final algorithm.

By addressing the challenges of sparsity, and user preference and target diversity, we developed a robust methodology for path generation. These steps form the foundation for enabling personalized, user-driven exploration in the representation space. The next sections will discuss the evaluation of this discovery approach and the implications for exploration music recommendation systems.

# 5

# Experimental Setup

We conducted an online experiment to evaluate how effectively our approach helps guide users toward developing a new taste based on their current preference profile. Specifically, we investigate whether taking gradual steps toward the target and integrating user feedback during the exploration process increases the effectiveness of the recommendation process.

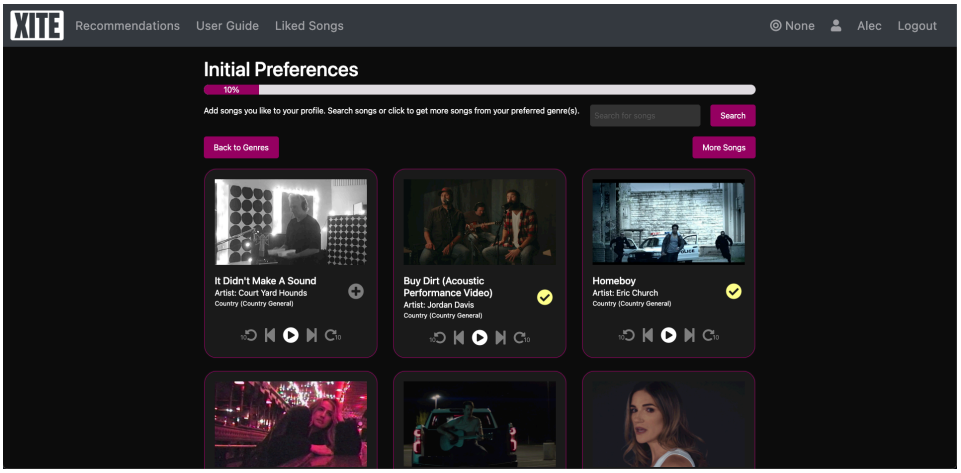



**Figure 5.1:** A cold-start user can search for and flip through their preferred genre of songs to fill their initial preference profile.

## 5.1. Experiment Scenario & Platform

To carry out the experiment, we created an experiment platform through a website[1] where we can set up our recommendation scenario for evaluation. The website is built using the popular Python-based web framework *Django* and uses a *PostgreSQL* database. Experiment subjects can create an account on the website under which all their discovery progress is stored.

Our recommendation scenario goes as follows. We start off with a completely cold-start user, for which we have no information on their music interests and preferences. To establish a simple user model, we prompt the user to specify 20 songs which they enjoy to create the initial user preferences the personalized recommendations can be based off (Figure A.12). Next, the user picks a target genre that they would like to explore. Completion of this initial phase sends them straight into the exploration phase in which the user will go through 6 exploration steps, each step containing 20 recommended songs in random order (Figure A.11). Songs in a step are played in random order to control for any order effects.

[1] https://expandyoursound.xite.com/

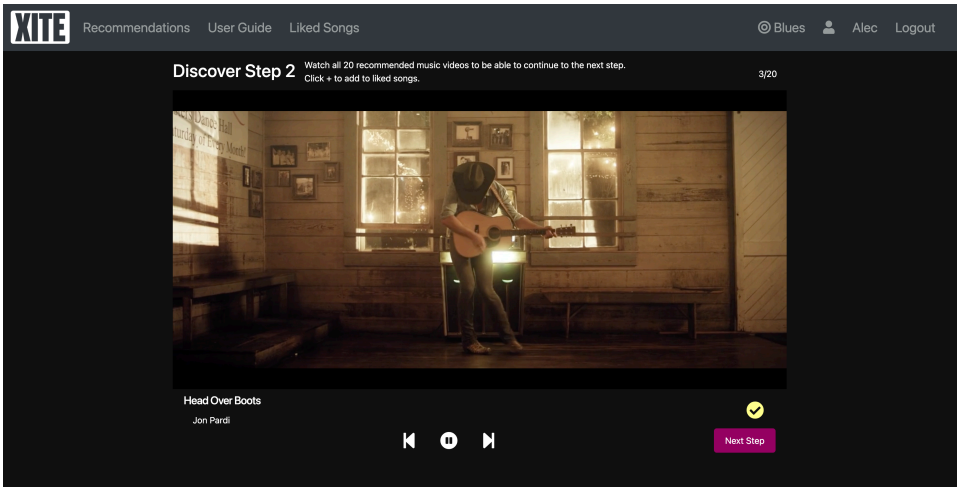



**Figure 5.2:** In every discovery step the user receives 20 recommendations one-by-one in random order. Users can add a song to the set of liked songs in their user profile and can move to the next step after going through all 20 songs once.

## 5.2. Participants

Around half of the participants were recruited from XITE, with most of them coming from the music curation team. The rest were recruited via convenience sampling. They were invited to the online study by group or personal messages and sent the website link. Upon visiting the home page of the website they are presented with the basic procedures of the study and an informed consent statement. This includes the explanation: ”This experience is designed to help you explore new music based on your current preferences, guiding you toward a genre or style you’d like to enjoy. Each user will follow a unique path, making the journey both personalized and enjoyable.”

## 5.3. Evaluation

Evaluating our system beyond accuracy is essential to gaining insight into the effectiveness of our methods. To this end, we follow a user-centric evaluation framework [40] that will allow us to answer our research questions. This framework allows us to examine how users’ interactions with the system (INT), perceived subjective system aspects (SSA), and user experience (EXP) relate to different conditions (Objective System Aspects: OSA) to which we subject the users.

In our case, we evaluate two OSAs that relate to our hypotheses (Ch. 1), namely, (1) taking gradual steps starting from current user preferences is more effective than one large step and (2) incorporating feedback improves effectiveness of the approach. Adopting a between-subjects study design, we aim to evaluate our approach by splitting subjects into three groups, covering our OSAs:

- **Big Step (BS):** From the start of the experiment we immediately jump to the target genre and start recommending items from the target. This group is meant as a baseline for evaluating the effect of taking small gradual steps towards the target.
- **Small Steps Passive (SSP):** These users start of by getting recommendations close to their current preferences and gradually get recommendations that get closer and closer to the target genre by randomly following any of the generated paths through the KNN-graph. Since in this group user feedback is not incorporated for picking which paths to follow, it serves as an intermediate baseline for evaluating the effect of integrating user feedback in the recommendation process.
- **Small Steps Active (SSA):** Similar to SSP, recommendations for these users gradually move from current preferences to the target genre. However, for this group, we update our belief on the enjoyment of a path for a user while taking step and receiving feedback. We then use this belief to guide the sampling of paths for the user.

The next step for evaluation is to select our outcome measures from observed behaviors (INT) and user survey feedback (EXP). As a first observed behavior, we track the number of songs liked throughout

| EXP/SSA | Question item |
|---|---|
| **Perceived helpfulness** | This approach supports me in getting to know the new genre. |
| | This approach motivates me to more delve into the new genre. |
| | This approach is useful in exploring a new genre. |
| **Affinity Toward Target** | I enjoy the music from [target taste]. |
| | My enjoyment of music from [target taste] has increased since the start of the experiment. |
| | My enjoyment of music from [target taste] has increased since the previous step. |
| **Quality of Direction** | I can notice the recommendations going in the direction of the target. |
| | The recommended songs seem to be in between my preferences and the target. |
| **Personalization** | I feel like the recommended songs take my preferences into account. |
| | I find the songs from the playlist appealing. |
| | I would listen to the playlist again. |
| **Control** | I found it easy to modify the recommendations in the recommender. |
| | The recommender allows only limited control to modify the recommendations. |
| | I feel in control of modifying the recommendations. |
| **Understandability** | I understand how the recommended songs relate to my musical taste. |
| | It is easy to grasp why I received these recommended songs. |
| | The recommendation process is clear to me. |

**Table 5.1:** Individual survey questions for each subjective construct.

the different steps. This gives us an objective measure of the enjoyment of a user during exploration both as the total amount of songs liked and the progression of number of songs liked at each step. The second observed behavior we look at is songs liked from the target genre. The total number of target genre songs liked gives us an indication of the user's affinity to the target genre and the progression of this number shows us how this affinity changes throughout the discovery process.

To supplement these objective measures, we survey the user's subjective experience in questionnaires throughout the experiment. Our EXP variables are perceived *helpfulness* and perceived *affinity towards target*. These EXP variables combined with the INT variables we see as representing effectiveness. A targeted discovery process being effective means the user's affinity towards the target is shifted positively and the user has enjoyed listening to the music presented to them during the discovery process.

We have established the OSAs for our subjects to experience and have INT and EXP outcomes that align with our defined measures of user effectiveness. This would be sufficient to answer our hypotheses for the research questions. However, to get more insights into why the OSAs result in INT and EXP outcomes we also include subjective system aspects (SSA) in the experiment questionnaires. These SSAs serve both as a dependent variable (in the hypothesized effect of OSA → SSA) and an independent variable (in the hypothesized effect of SSA → EXP). These kind of variables are often called mediating variables. The SSAs we measure are (1) perceived *control*, (2) perceived *understandability*, (3) perceived quality of *direction*, (4) perceived *personalization*. These are all factors that can differ based on the OSAs and can influence the user's perceived helpfulness and affinity towards target. We hypothesize that taking small gradual steps starting from current preferences will increase the user's perception of personalization and quality of direction. Further, we hypothesize that incorporating feedback increases the user's perception of control over the recommendations and their understanding of the recommendation process. Finally, we hypothesize that all measured SSAs have a positive effect on the INT and EXP measures.

All EXP and SSA factors are measured by multiple question items on a 7-point Likert scale ranging from *Strongly Disagree* to *Strongly Agree* (Table 5.1). For all factors except helpfulness we repeat these questions after every step to measure how they progress throughout the steps. Questions for control, understandability, personalization and helpfulness are adapted from [43], and new question items are constructed for perceived quality of direction and affinity towards target.

# 6

# Results and Discussion

## 6.1. Results

The online experiment ran from October to November 2024 and accrued 21 valid responses. We only included results for users who finished all 6 steps. In total 49 people signed up for the platform. However, 19 people did not start the experiment, 5 people stopped after creating their initial user preferences, and 4 people did not come back after going through the first step. The average age of participants was 28.1 years (std. 6.43), with 11 females and 10 males. During sign-up participants were randomly assigned to each of the user groups, BS (N=6), SSP (N=8), SSA (N=7).

The number of valid responses combined with the complexity of the required model to encompass all survey and interaction variables means we cannot achieve a valid fit for a structural equation model as in [40]. We focus on exploring and discussing the observed patterns and relationships, leveraging available data to identify areas of potential significance where possible. We follow the approach by [5] to evaluate hypothesized mediation effects of SSAs between the OSAs and the INT/EXP outcomes, and bootstrap confidence intervals to test for significance [27]. Especially the dispersion of data points over multiple recommendation steps complicates the model. To remedy this, for statistical evaluation, we individually aggregate every measure over the time steps in a way that makes sense for that measure. These aggregate measurements can be found in Table 6.2. Further, we examine the measurements across time steps in search of potentially interesting insight and present them here. For all aggregated survey factors (EXP & SSA) convergent validity holds as the average variance extracted between question items in each factor is larger than 0.50. An overview of all measurements can be found in Appendix A.

| **OSA** | **INT** | | | | | |
|---|---|---|---|---|---|---|
| **User Group** | **Total Likes** | | **Target Likes** | | **Target Likes/Song** | |
| | Mean | Std. | Mean | Std. | Mean | Std. |
| BS | 14.33 | 10.65 | 14.33 | 10.65 | 0.12 | 0.09 |
| SSP + SSA | 44.80 | 9.35 | 11.67 | 6.33 | 0.34 | 0.16 |
| SSP | 42.25 | 6.80 | 9.63 | 4.93 | 0.28 | 0.22 |
| SSA | 47.71 | 11.47 | 14.00 | 7.30 | 0.40 | 0.19 |

**Table 6.1:** Overview of aggregated like data with total likes, total liked songs from target genre, ratio between total recommended songs from target genre and total liked songs from target genre.

### 6.1.1. Liked songs

For the observed behaviors, number of liked songs and number of liked songs from the target genre, we first look at the total number of liked songs and liked target songs (Table 6.1). When comparing total liked target songs between, we look at the ratio between target songs liked and received, as the BS group naturally receives more songs from the target genre. Separating our OSAs from the user groups, we look at the BS group compared to the combined group of SSP and SSA for the effect of gradual steps and at the SSP group compared to the SSA group for the effect of incorporating feedback on total

liked songs. We use Poisson regression to analyze the total number of likes because this method is well-suited for count data, where the response variable consists of non-negative whole numbers [13]. For the ratio of target likes to the number of target songs, we apply regular linear regression [50]. This method is appropriate because the ratio is a continuous variable, and linear regression helps us test for significant differences across conditions.

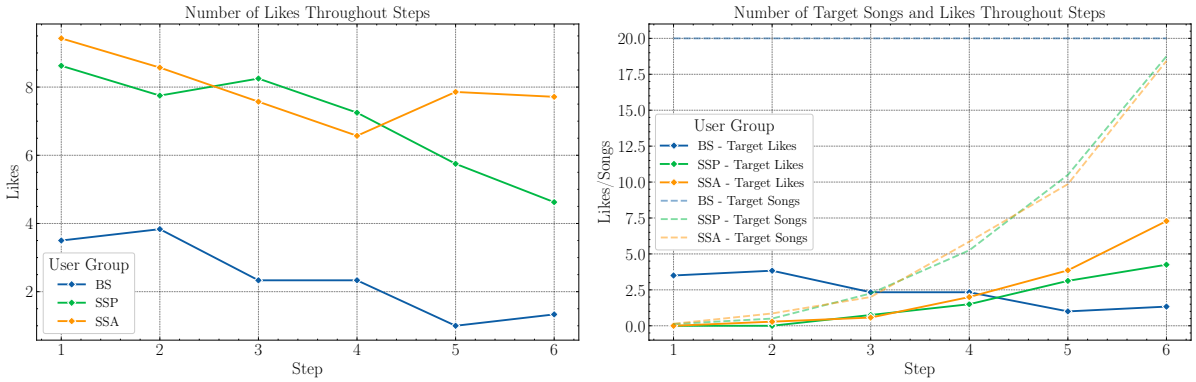



**Figure 6.1:** Average like progress throughout the steps for different user groups.

### Total Likes

Between BS and SSP + SSA we observe a significant difference between their expected total likes ($\beta$[1] $= 1.14; se = 0.11; p < .001$). This shows that having the user take gradual steps starting from their current preferences towards the target will mean that they encounter more songs that are enjoyable to them during the discovery process. Comparing the effects of incorporating feedback between the SSP and SSA groups shows a slight increase in total likes for the SSA group ($\beta = 0.12; se = 0.08; p = .11$). However, this difference between SSP and SSA was not statistically significant. This means looking at total likes we cannot conclude that incorporating feedback significantly improves the user's enjoyment during the discovery process.

To get more insight into the number of songs liked for each user group, we will look at their progress throughout the steps in Figure 6.1 (left). We can now see that both SSP and SSA start with a high like count when song recommendations are still close to their initial preferences. For both groups this like count drops in the following steps. In the last two steps we can see the difference between SSP and SSA become apparent. The like count for SSA seems to recover, while the like count for SSP keeps falling. This could indicate that the incorporated feedback of the system for the SSA group allowed them to find the subgroup of the genre that they enjoy as opposed to the SSP group that still get recommendations spread through all possible paths to the genre.

For BS, the like count starts relatively low and keeps slowly reducing. This further reduction could be explained by the users growing tired of the volume of target genre recommendations that the SSP and SSA groups did not get. However, the number of likes from their first exposure to the target genre is still lower than the number of likes for the SSP and SSA groups in step five and six where they most likely first got fully exposed to their target genre.

### Liked Target Songs

When looking at the ratio of liked to received target songs, we observe a significant uplift between the SSP + SSA group and the BS group ($\beta = 0.22; se = 0.08; p = .02$). This means that taking gradual steps results in users enjoying a larger proportion of the target songs they receive. Between SSP and SSA, this increase in proportion of target songs liked, similar to total likes, is smaller and nonsignificant ($\beta = 0.12; se = 0.1; p = .25$). Again, the sample size limits our ability to make any strong conclusions on the effect of incorporating feedback, but a detectable effect is present.

Figure 6.1 (right) shows the progression of both received target songs and liked target songs. Both SSP and SSA get recommended a similar amount of songs from their targets throughout the steps.

[1] Note that $\beta$ in the total likes regressions is the coefficient for Poisson regression.

The number of liked target songs also starts off growing similarly. However, in the last two steps, the SSA group starts liking more target songs, with the last step having a difference of around 3.5 likes. Similar to total likes, this shows users from the SSA group eventually more often find the subgroup of their target that they enjoy.

| OSA | EXP | | | | SSA | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| User Group | Helpfulness | | Final Affinity | | Avg. Direction Quality | | Avg. Personalization | | Avg. Control | | Avg. Understandability | |
| | Mean | Std. | Mean | Std. | Mean | Std. | Mean | Std. | Mean | Std. | Mean | Std. |
| BS | 3.39 | 1.14 | 1.94 | 0.57 | 2.58 | 0.62 | 2.38 | 0.59 | 2.15 | 0.31 | 2.59 | 0.45 |
| SSP + SSA | 5.51 | 1.32 | 4.87 | 1.30 | 4.70 | 1.19 | 4.40 | 1.02 | 4.36 | 0.99 | 4.95 | 1.22 |
| SSP | 5.00 | 1.55 | 4.58 | 1.32 | 4.31 | 1.36 | 4.08 | 1.23 | 3.92 | 1.14 | 4.47 | 1.41 |
| SSA | 6.10 | 0.71 | 5.19 | 1.29 | 5.14 | 0.84 | 4.76 | 0.61 | 4.87 | 0.42 | 5.51 | 0.70 |

**Table 6.2:** Aggregated measures from questionnaires for different user groups.

### 6.1.2. Perceived quality of direction and personalization (SSA)

We examine the SSA measures, perceived quality of direction and perceived personalization, hypothesized to be influenced by taking gradual steps towards target and to influence all INT and EXP measures (Table 6.2). As expected, a comparison between the BS group and the SSP+SSA group shows us a large effect ($d = 1.98; p < .001$) on the user's average perceived quality of direction. This means the user directly notices the songs moving from their current preferences towards the target. Surprisingly, between the SSP and SSA groups, there is a moderate effect detectable, even though not statistically significant ($d = 0.72; p = 0.18$). Observing the measure throughout the steps can give us more insight to why this effect exists (Figure 6.2). The perception of the quality of direction is relatively close between the groups for the first few steps but their relative difference grows towards the final steps. This suggests as the users from SSA group get further along the discovery and more feedback is integrated to receive recommendations the user is more likely to enjoy, their perception of moving between their preferences increases.

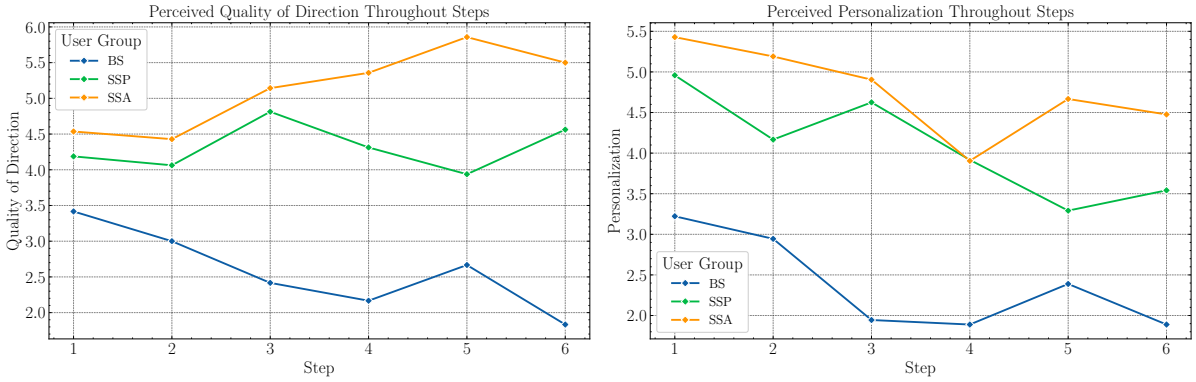



**Figure 6.2:** Average perceived quality of direction and personalization throughout the steps for different user groups.

Looking at the effects of the OSAs on average perceived personalization we observe a strong significant effect when taking gradual steps towards the target ($\beta = 2.02; se = 0.45; p < .001$) and a moderate insignificant effect when incorporating feedback ($\beta = 0.68; se = 0.51; p = .21$). These results confirm that users actually perceived the recommendations that start at their current preferences as more personal. This perception holds even when moving further away from initial preferences (Figure 6.2). We observe a moderate effect when incorporating feedback due to an increase in perceived personalization in the final steps for the SSA group after decreasing in the first few steps. This means when more feedback is integrated the user again perceives the recommendations as being more close to their preferences.

### 6.1.3. Perceived control and understandability (SSA)

We explore perceived control and perceived understandability, hypothesized to be influenced by incorporating feedback and to influence all INT and EXP measures (Table 6.2). The effect of gradual

steps is average perceived control ($\beta = 0.99; se = 0.23; p < .001$). A similar significant is observed for average perceived understandability ($\beta = 2.35; se = 0.52; p < .001$). These are not effects we initially hypothesized; however these findings align with an intuitive explanation: users may feel more control and better understand how the recommendations relate to their musical when the system begins with preferences close to their own. Going from the SSP group to the SSA group where feedback is incorporated we observe an effect on control ($\beta = 0.45; se = 0.24; p = .09$) and a strong effect on understandability ($\beta = 1.04; se = 0.59; p = .1$). Results for both suggest that with a larger sample size, a statistically significant effect might be detected. This would confirm our hypothesis that incorporating feedback increases the user's perception of control and understandability of the system.

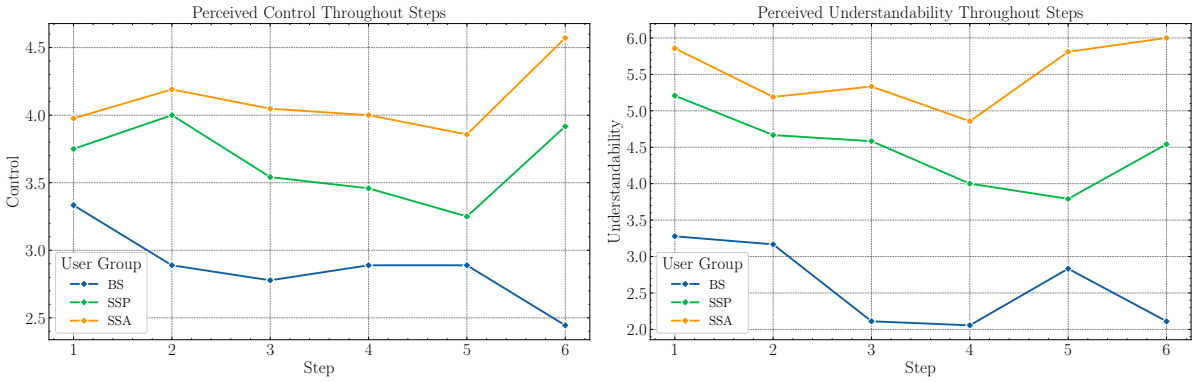



**Figure 6.3:** Average perceived control and understandability throughout the steps for different user groups.

For the progression of both perceived control and perceived understandability, the SSP and SSA groups follow similar patterns. Both start relatively high for the first two steps, then decrease in the intermediate steps, increasing again for the final steps. When the user's current preferences and the target are significantly far apart, the intermediate steps often consist of songs that neither align closely with the user's current preferences nor belong to the target genre. This mismatch could explain the observed negative trend in control and understandability during these steps.

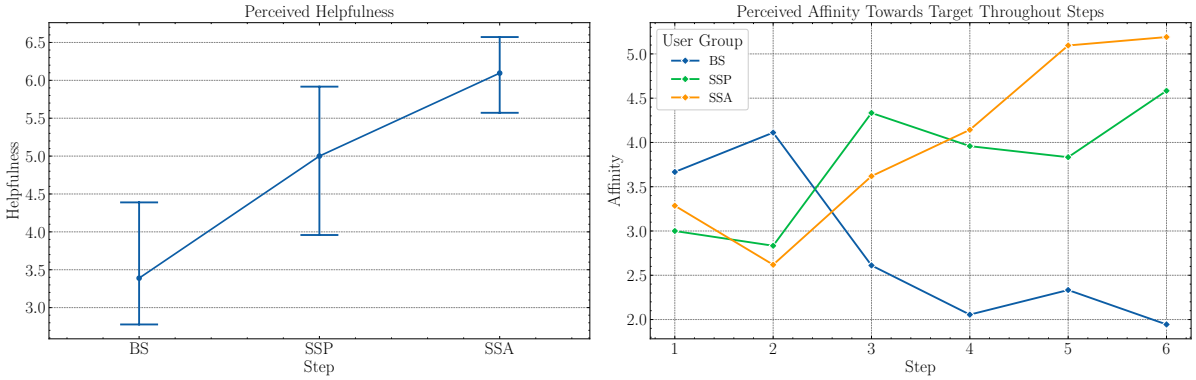



**Figure 6.4:** Perceived helpfulness (error bars 95% confidence interval) and average perceived affinity towards target throughout the steps for different user groups.

### 6.1.4. Perceived helpfulness and affinity towards target (EXP)

We inspect the effects of the different OSAs, as well as any significant mediating effects from the SSAs, on the subject outcome measures we set to represent the effectiveness of our system, allowing us to answer our main research question and second sub-research question. Table 6.2 shows the mean and standard deviation of the aggregated measures from questionnaires for the different conditions. As seen in Figure 6.4 (left), there is a clear difference in perceived helpfulness between groups, with error bars indicating 95% confidence intervals. This visual aligns with the statistical evaluation: there is a strong direct effect between user groups BS and SSP+SSA for perceived helpfulness ($\beta = 2.12; se = 0.62; p = .003$) showing that taking gradual steps from the current preferences to-

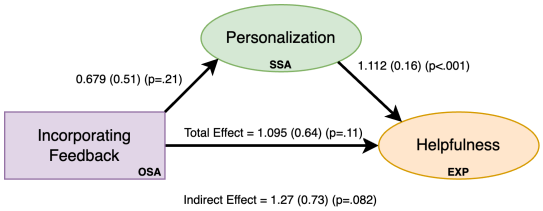



**Figure 6.5:** Mediated effect of incorporating feedback on helpfulness through personalization. Individual coefficients, standard errors and p-values shown.

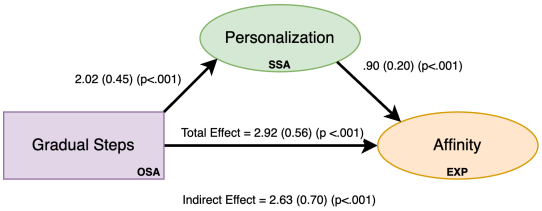



**Figure 6.6:** Mediated effect of taking gradual steps on affinity through personalization. Individual coefficients, standard errors and p-values shown.

wards the target is perceived as more helpful for exploration of the target. Additionally, we measure a borderline statistically significant indirect effect between taking gradual steps and helpfulness through perceived personalization ($\beta = 1.76; se = 0.96; p = .067$). This lack of significance may come from the limited sample size, which reduces the power to detect mediation effects. From the SSP to SSA group there is a direct effect on helpfulness, but it is not statistically significant ($\beta = 1.095; se = 0.64; p = .11$). A stronger (indirect) effect is found through personalization ($\beta = 1.27; se = 0.73; p = .082$) (Figure 6.5) and a slightly weaker effect is found through understandability ($\beta = 0.92; se = 0.65; p = .16$) which shows, despite not being statistically significant, that potentially the increased perception of personalization and understandability of the recommenders due to the incorporation of feedback increases the user's perception of the helpfulness of the system.

For perceived affinity towards the target, we decide to aggregate by selecting the value from the final discovery step (Table 6.2). This represents the user's affinity at the end of the discovery process, which is what we are most interested in. Similar to helpfulness, between user groups BS and SSA+SSP there is a strong direct effect (($\beta = 2.92; se = 0.56; p < .001$).This effect is also found indirectly though perceived personalization ($\beta = 2.63; se = 0.70; p < .001$) indicating that taking gradual steps results in significantly more affinity towards the target, which is partly explained by an increase in the perception of personalization of the recommendations (Figure 6.6). The difference in affinity to target between the SSA and SSP groups is noticeable ($\beta = 0.6; se = 0.68; p = .38$); however, not statistically significant. The relative strength of effect still suggests a potentially meaningful impact of incorporating feedback.

Figure 6.4 (right) shows the progression of affinity towards target over the discovery steps. Interestingly, the affinity towards target starts off higher for the BS group as they receive target specific song recommendation immediately and even increases for step two. However, affinity starts decreasing rapidly in the following steps. This negative change is potentially the result of saturation from receiving exclusively target songs at each step, decreasing the users enjoyment of the process, and in turn, their affinity toward what they are trying to explore. The progression of affinity for the SSP and SSA groups follow similar patterns, starting off relatively low and slowly increasing. Affinity towards target for the SSA group increases marginally faster and therefore ends up higher after the final step.

## 6.2. Discussion

Analyzing the results of this user experiment reveals several key insights into the effectiveness of gradual steps and feedback incorporation in guiding users towards new musical preferences. These findings contribute to a deeper understanding of how recommendation systems can enable targeted exploration while ensuring user satisfaction.

The strongest increases in perceived helpfulness and affinity were observed when users took gradual steps from their current preferences toward the target genre. This aligns with our hypothesis (**H1**) that gradually introducing users to the target is more effective than directly exposing users to representative songs from the target. The results also revealed that gradual steps significantly increased both the total number of liked songs and the proportion of target songs liked. These findings highlight the importance of incremental exploration in improving user engagement and aligning recommendations with user preferences.

Incorporating feedback into the recommendation process resulted in positive but non-significant effects on several measures, including total likes, perceived personalization, and perceived understandability.

Notably, the SSA group showed a recovery in total likes during later steps, even when far away from initial preferences. This, along with the increased proportion of target songs liked in later steps, suggests that feedback helped users to discover subgenres they enjoyed from their target genre. These observations align with our second hypothesis (**H2**) that integrating user feedback and interaction could further improve the effectiveness of the approach for guiding users toward developing a new taste. Enabling users to fine-tune recommendations with their feedback helps the system discover and leverage their hidden preferences, increasing both engagement and satisfaction. Although the current experiment lacked sufficient statistical power to confirm these effects, the results highlight the potential of feedback integration for personalizing and optimizing the discovery process.

Gradual steps also significantly improved perceived control and understandability. These measures are critical for user engagement, as they ensure users feel their preferences influence the system and can understand the reasoning behind recommendations. Interestingly, both control and understandability decreased during intermediate steps, likely due to recommendations falling outside both user preferences and the target genre. However, similar to total likes they recovered in later steps, indicating the importance of making users understand and feel control over where their recommendations come from. This suggests that for future systems incorporating explanations or insight into where users are in the discovery process could increase user engagement and satisfaction, especially in situations where the recommendations are not intuitive.

Although the results were promising, our experiment has some limitations. The low sample size restricted our ability to fit a structural equation model (SEM) accounting for all factors simultaneously. Instead, we evaluated individual effects in isolation, which could overlook interactions between variables. Additionally, some hypothesized effects, such as the impact of feedback on personalization and control, lacked statistical significance due to limited power. These constraints should be carefully considered when interpreting the findings.

The small sample size resulted from both resource constraints and experiment design choices. Convenience sampling limited the number of participants and their diversity, while the large time commitment necessary to complete the experiment led to high drop-out rates. Participants were required to select 20 initial preference songs and complete 6 rounds of 20 recommendations, which likely contributed to the high drop-out rate. Additionally, users had to perform the experiment on a separate website, which may have introduced friction and reduced engagement. Future studies could address these issues by simplifying the process, such as reducing the number of steps or initial song selections, and integrating the experiment into platforms users already engage with, such as music streaming services. This approach could improve ease of use and likely eliminate the need for an initial preference elicitation process, making participation more intuitive and natural for users.

Another limitation of the experiments is that users had complete freedom to determine how long they spent on each step and the time they waited between steps. This could have contributed to user fatigue for the control group if they went through the consecutive steps shortly after each other, as they received similar content in all steps.

# 7 Conclusion

## 7.1. Summary

In this thesis, we aimed to enable targeted music exploration with interactive recommendations. Addressing the main research question, “How can we effectively guide a user towards developing a new taste based on their current preference profile?”, we propose an approach that gradually guides the user from their current preferences to a target genre.

Facilitating our gradual exploration approach, we combined high-level expert-annotated content, low-level audio-based features, and user interaction data to create a representation space where similar songs are mapped closely together. In our approach to constructing the representation space, we embed a music knowledge graph enhanced with audio similarity information. We performed an initial visual evaluation, confirming that our method qualitatively aligns with our expectations. Following, through quantitative evaluation on a playlist completion task, we demonstrated the effectiveness of our approach for separating songs in a way that aligns with human understanding of music in context.

We evaluate our approach in extensive user experiments, showcasing the effectiveness of guiding users incrementally from their current preferences toward a target genre. Our findings show that gradual exploration improved engagement and satisfaction compared to directly introducing users to songs from the target genre. Incorporating user feedback through a Bayesian approach to active learning further enhanced the effectiveness of the recommendation process, allowing the system to gather information on the user’s preferences while ensuring the recommendations remain enjoyable.

## 7.2. Industry Recommendations

Based on the work done in this thesis we can make some recommendations for industry players who wish to provide their users with alternative ways to approach listening to music. Despite the limited sample size of experiments, the benefits of gradual exploration in improving satisfaction and engagement are clear. While the impact of incorporating feedback remains inconclusive, it is reasonable to expect including additional user feedback beyond historical interactions should not negatively affect recommendations.

Industry applications would address several limitations of this work. Embedding a version of our methods into existing music streaming platforms could allow for seamless user interaction and greater accessibility. This would likely improve the sample size of any experiments and ensure that the evaluation of the system aligns more closely with how users are likely to interact with the algorithm in real-world scenarios.

We envision two approaches to the industry application of this recommendation scenario. One approach would be to create a separate section on their platform dedicated to targeted exploration, where the user can set a target genre, access and listen to the recommendations playlist of their current step, and look back at their journey through previously recommended playlists. Another approach would be to seamlessly integrate targeted exploration into the platform by providing the user with a targeted

exploration playlist alongside other weekly updating recommendation playlists such as 'Your Weekly' and 'Discover Weekly'.

Implementing targeted exploration in industry does however involve trade-offs between resource investment, user engagement, and system complexity. Gradual exploration requires computationally efficient graph representations but enhances user engagement by guiding users incrementally. Feedback integration could further improve user satisfaction and engagement but demands real-time processing.

## 7.3. Future Work

Despite its contributions, this research has limitations that could be addressed in future work on targeted exploration.

**Experiment design:** The small sample size of user experiments in this work restricted the generalizability of findings to broader, real-world applications. Future studies with a larger pool of subjects would make it possible to provide more conclusive statements on the effect of targeted exploration approaches on user satisfaction and engagement. Furthermore, investigating the long-term impacts of gradual exploration on user satisfaction and taste development would provide valuable insights into the sustainability of this approach.

**Evaluation of AL algorithm:** We evaluated the effectiveness of the active learning algorithm in a simulation, demonstrating its potential effectiveness. However, when evaluating on real users this algorithm was only compared to the absence of interactive recommendations. Future research could focus on implementing several active learning algorithms in live systems to evaluate their effectiveness with actual users. This could help determine whether alternative approaches to incorporating feedback might yield improved results or uncover new insights into user behavior during exploration.

**Improvements on methods:** Integrating richer prior information from similar users could further accelerate the belief update process and improve recommendation efficiency. Additionally, providing users with explanations or insights into the recommendation process during exploration may help address potential moments of confusion or disengagement, especially during intermediate steps where recommendations are far from current preferences and the intended target.

**Applicability to other target types:** In this work we focused on the exploration of target genres. However, we believe that the methods outlined have the potential to apply to several target types, such as subgenres and artists. As the size and diversity of subgenre and artist groups differ largely from most genres this belief would have to be validated in future studies.

# References


[1] Louis Abraham. *fastnode2vec*. 2020. DOI: 10.5281/zenodo.3902632. URL: https://github.com/louisabraham/fastnode2vec.

[2] G. Adomavicius and A. Tuzhilin. "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions". In: *IEEE Transactions on Knowledge and Data Engineering* 17.6 (June 2005). Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 734–749. ISSN: 1558-2191. DOI: 10.1109/TKDE.2005.99. URL: https://ieeexplore.ieee.org/abstract/document/1423975 (visited on 11/13/2024).

[3] Charu C Aggarwal et al. *Recommender systems*. Vol. 1. Springer, 2016.

[4] David Austin et al. "Bayesian Optimization with LLM-Based Acquisition Functions for Natural Language Preference Elicitation". In: *Proceedings of the 18th ACM Conference on Recommender Systems*. RecSys '24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 74–83. ISBN: 9798400705052. DOI: 10.1145/3640457.3688142. URL: https://dl.acm.org/doi/10.1145/3640457.3688142 (visited on 10/29/2024).

[5] Reuben M Baron and David A Kenny. "The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations." In: *Journal of personality and social psychology* 51.6 (1986), p. 1173.

[6] Niels Bertram, Jürgen Dunkel, and Ramón Hermoso. "I am all EARS: Using open data and knowledge graph embeddings for music recommendations". In: *Expert Systems with Applications* 229 (Nov. 1, 2023), p. 120347. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2023.120347. URL: https://www.sciencedirect.com/science/article/pii/S0957417423008497 (visited on 05/28/2024).

[7] Antoine Bordes et al. "Translating Embeddings for Modeling Multi-relational Data". In: *Advances in Neural Information Processing Systems*. Vol. 26. Curran Associates, Inc., 2013. URL: https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html (visited on 07/09/2024).

[8] Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. "TasteWeights: a visual interactive hybrid recommender system". In: *Proceedings of the sixth ACM conference on Recommender systems*. RecSys '12. New York, NY, USA: Association for Computing Machinery, Sept. 9, 2012, pp. 35–42. ISBN: 978-1-4503-1270-7. DOI: 10.1145/2365952.2365964. URL: https://dl.acm.org/doi/10.1145/2365952.2365964 (visited on 12/02/2024).

[9] Robin Burke. "Hybrid web recommender systems". In: *The adaptive web: methods and strategies of web personalization* (2007), pp. 377–408.

[10] Olivier Chapelle and Lihong Li. "An Empirical Evaluation of Thompson Sampling". In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011. URL: https://proceedings.neurips.cc/paper/2011/hash/e53a0a2978c28872a4505bdb51db06dc-Abstract.html (visited on 12/01/2024).

[11] Ching-Wei Chen et al. "Recsys challenge 2018: Automatic music playlist continuation". In: *Proceedings of the 12th ACM Conference on Recommender Systems*. 2018, pp. 527–528.

[12] Minmin Chen et al. "Off-Policy Actor-critic for Recommender Systems". In: *Proceedings of the 16th ACM Conference on Recommender Systems*. RecSys '22. New York, NY, USA: Association for Computing Machinery, Sept. 2022, pp. 338–349. ISBN: 978-1-4503-9278-5. DOI: 10.1145/3523227.3546758. URL: https://dl.acm.org/doi/10.1145/3523227.3546758 (visited on 11/06/2023).

[13] Stefany Coxe, Stephen G West, and Leona S Aiken. "The analysis of count data: A gentle introduction to Poisson regression and its alternatives". In: *Journal of personality assessment* 91.2 (2009), pp. 121–136.

[14] Xiaohui Cui et al. “MKGCN: Multi-Modal Knowledge Graph Convolutional Network for Music Recommender Systems”. In: *Electronics* 12.12 (Jan. 2023). Number: 12 Publisher: Multidisciplinary Digital Publishing Institute, p. 2688. ISSN: 2079-9292. DOI: 10.3390/electronics12122688. URL: https://www.mdpi.com/2079-9292/12/12/2688 (visited on 03/20/2024).

[15] Sally Jo Cunningham, David Bainbridge, and Annette Falconer. “” More of an art than a science”: Supporting the creation of playlists and mixes”. In: (2006).

[16] Tim Dettmers et al. “Convolutional 2d knowledge graph embeddings”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[17] Phuc Do and Phu Pham. “W-KG2Vec: a weighted text-enhanced meta-path-based knowledge graph embedding for similarity search”. In: *Neural Computing and Applications* 33.23 (Dec. 1, 2021), pp. 16533–16555. ISSN: 1433-3058. DOI: 10.1007/s00521-021-06252-8. URL: https://doi.org/10.1007/s00521-021-06252-8 (visited on 04/16/2024).

[18] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. “metapath2vec: Scalable Representation Learning for Heterogeneous Networks”. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’17. New York, NY, USA: Association for Computing Machinery, Aug. 4, 2017, pp. 135–144. ISBN: 978-1-4503-4887-4. DOI: 10.1145/3097983.3098036. URL: https://dl.acm.org/doi/10.1145/3097983.3098036 (visited on 04/22/2024).

[19] Michael D. Ekstrand, John T. Riedl, and Joseph A. Konstan. “Collaborative Filtering Recommender Systems”. In: *Foundations and Trends® in Human–Computer Interaction* 4.2 (May 4, 2011). Publisher: Now Publishers, Inc., pp. 81–173. ISSN: 1551-3955, 1551-3963. DOI: 10.1561/1100000009. URL: https://www.nowpublishers.com/article/Details/HCI-009 (visited on 11/11/2024).

[20] Simon Frith, Will Straw, and John Street. *The Cambridge companion to pop and rock*. Cambridge University Press, 2001.

[21] Tao-yang Fu, Wang-Chien Lee, and Zhen Lei. “Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning”. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, pp. 1797–1806.

[22] Jean Garcia-Gathright et al. “Understanding and Evaluating User Satisfaction with Music Discovery”. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR ’18. New York, NY, USA: Association for Computing Machinery, June 27, 2018, pp. 55–64. ISBN: 978-1-4503-5657-2. DOI: 10.1145/3209978.3210049. URL: https://dl.acm.org/doi/10.1145/3209978.3210049 (visited on 12/02/2024).

[23] Nadav Golbandi, Yehuda Koren, and Ronny Lempel. “On bootstrapping recommender systems”. In: *Proceedings of the 19th ACM international conference on Information and knowledge management*. CIKM ’10. New York, NY, USA: Association for Computing Machinery, 2010, pp. 1805–1808. ISBN: 978-1-4503-0099-5. DOI: 10.1145/1871437.1871734. URL: https://dl.acm.org/doi/10.1145/1871437.1871734 (visited on 12/01/2024).

[24] Aditya Grover and Jure Leskovec. *node2vec: Scalable Feature Learning for Networks*. July 3, 2016. DOI: 10.48550/arXiv.1607.00653. arXiv: 1607.00653[cs,stat]. URL: http://arxiv.org/abs/1607.00653 (visited on 01/30/2024).

[25] Qingyu Guo et al. “A Survey on Knowledge Graph-Based Recommender Systems”. In: *IEEE Transactions on Knowledge and Data Engineering* 34.8 (Aug. 2022). Conference Name: IEEE Transactions on Knowledge and Data Engineering, pp. 3549–3568. ISSN: 1558-2191. DOI: 10.1109/TKDE.2020.3028705. URL: https://ieeexplore-ieee-org.tudelft.idm.oclc.org/document/9216015 (visited on 07/09/2024).

[26] Negar Hariri, Bamshad Mobasher, and Robin Burke. “Adapting to User Preference Changes in Interactive Recommendation”. In: ().

[27] Jason S Haukoos and Roger J Lewis. “Advanced statistics: bootstrapping confidence intervals for statistics with “difficult” distributions”. In: *Academic emergency medicine* 12.4 (2005), pp. 360–365.

[28] Chen He, Denis Parra, and Katrien Verbert. “Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities”. In: *Expert Systems with Applications* 56 (Sept. 1, 2016), pp. 9–27. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2016.02.013. URL: https://www.sciencedirect.com/science/article/pii/S0957417416300367 (visited on 12/01/2024).

[29] Dietmar Jannach, Sidra Naveed, and Michael Jugovac. “User Control in Recommender Systems: Overview and Interaction Challenges”. In: *E-Commerce and Web Technologies*. Ed. by Derek Bridge and Heiner Stuckenschmidt. Cham: Springer International Publishing, 2017, pp. 21–33. ISBN: 978-3-319-53676-7. DOI: 10.1007/978-3-319-53676-7_2.

[30] Dietmar Jannach et al. “A Survey on Conversational Recommender Systems”. In: *ACM Computing Surveys* 54.5 (June 30, 2022), pp. 1–36. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3453154. arXiv: 2004.00646[cs]. URL: http://arxiv.org/abs/2004.00646 (visited on 02/05/2024).

[31] Guoliang Ji et al. “Knowledge graph embedding via dynamic mapping matrix”. In: *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*. 2015, pp. 687–696.

[32] Rong Jin and Luo Si. *A Bayesian Approach toward Active Learning for Collaborative Filtering*. July 11, 2012. DOI: 10.48550/arXiv.1207.4146. arXiv: 1207.4146. URL: http://arxiv.org/abs/1207.4146 (visited on 12/01/2024).

[33] Jeff Johnson, Matthijs Douze, and Hervé Jégou. “Billion-scale similarity search with GPUs”. In: *IEEE Transactions on Big Data* 7.3 (2019), pp. 535–547.

[34] Serdar Kadıoğlu, Bernard Kleynhans, and Xin Wang. “Integrating optimized item selection with active learning for continuous exploration in recommender systems”. In: *Annals of Mathematics and Artificial Intelligence* (Apr. 5, 2024). ISSN: 1573-7470. DOI: 10.1007/s10472-024-09941-x. URL: https://doi.org/10.1007/s10472-024-09941-x (visited on 04/15/2024).

[35] Mohsen Kamalzadeh et al. “TagFlip: Active Mobile Music Discovery with Social Tags”. In: *Proceedings of the 21st International Conference on Intelligent User Interfaces*. IUI ’16. New York, NY, USA: Association for Computing Machinery, Mar. 7, 2016, pp. 19–30. ISBN: 978-1-4503-4137-0. DOI: 10.1145/2856767.2856780. URL: https://dl.acm.org/doi/10.1145/2856767.2856780 (visited on 02/29/2024).

[36] Nedim Karakayali, Burc Kostem, and Idil Galip. “Recommendation Systems as Technologies of the Self: Algorithmic Control and the Formation of Music Taste”. In: *Theory, Culture & Society* 35.2 (Mar. 1, 2018). Publisher: SAGE Publications Ltd, pp. 3–24. ISSN: 0263-2764. DOI: 10.1177/0263276417722391. URL: https://doi.org/10.1177/0263276417722391 (visited on 03/25/2024).

[37] Thomas N Kipf and Max Welling. “Semi-supervised classification with graph convolutional networks”. In: *arXiv preprint arXiv:1609.02907* (2016).

[38] Peter Knees et al. “Introduction to music similarity and retrieval”. In: *Music Similarity and Retrieval: An Introduction to Audio-and Web-based Strategies* (2016), pp. 1–30.

[39] Bart P. Knijnenburg, Saadhika Sivakumar, and Daricia Wilkinson. “Recommender Systems for Self-Actualization”. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys ’16. New York, NY, USA: Association for Computing Machinery, Sept. 7, 2016, pp. 11–14. ISBN: 978-1-4503-4035-9. DOI: 10.1145/2959100.2959189. URL: https://dl.acm.org/doi/10.1145/2959100.2959189 (visited on 04/12/2024).

[40] Bart P. Knijnenburg and Martijn C. Willemsen. “Evaluating Recommender Systems with User Experiments”. In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA: Springer US, 2015, pp. 309–352. ISBN: 978-1-4899-7636-9 978-1-4899-7637-6. DOI: 10.1007/978-1-4899-7637-6_9. URL: https://link.springer.com/10.1007/978-1-4899-7637-6_9 (visited on 05/16/2024).

[41] Bart P. Knijnenburg et al. “Explaining the user experience of recommender systems”. In: *User Modeling and User-Adapted Interaction* 22.4 (Oct. 1, 2012), pp. 441–504. ISSN: 1573-1391. DOI: 10.1007/s11257-011-9118-4. URL: https://doi.org/10.1007/s11257-011-9118-4 (visited on 11/17/2024).

[42] Pan Li et al. “PURS: Personalized Unexpected Recommender System for Improving User Satisfaction”. In: *Fourteenth ACM Conference on Recommender Systems*. RecSys ’20: Fourteenth ACM Conference on Recommender Systems. Virtual Event Brazil: ACM, Sept. 22, 2020, pp. 279–288. ISBN: 978-1-4503-7583-2. DOI: 10.1145/3383313.3412238. URL: https://dl.acm.org/doi/10.1145/3383313.3412238 (visited on 11/06/2023).

[43] Yu Liang. “Supporting Personalized Music Exploration through a Genre Exploration Recommender”. ISBN: 9789038659015. PhD thesis. Eindhoven: Eindhoven University of Technology, Nov. 27, 2023.

[44] Yu Liang and Martijn C. Willemsen. “Personalized Recommendations for Music Genre Exploration”. In: *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*. UMAP ’19. New York, NY, USA: Association for Computing Machinery, June 7, 2019, pp. 276–284. ISBN: 978-1-4503-6021-0. DOI: 10.1145/3320435.3320455. URL: https://dl.acm.org/doi/10.1145/3320435.3320455 (visited on 02/15/2024).

[45] Yankai Lin et al. “Learning entity and relation embeddings for knowledge graph completion”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 29. 1. 2015.

[46] Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. “Content-based Recommender Systems: State of the Art and Trends”. In: *Recommender Systems Handbook*. Ed. by Francesco Ricci et al. Boston, MA: Springer US, 2011, pp. 73–105. ISBN: 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3_3. URL: https://doi.org/10.1007/978-0-387-85820-3_3 (visited on 11/13/2024).

[47] Matthew C. McCallum et al. *Supervised and Unsupervised Learning of Audio Representations for Music Understanding*. Oct. 7, 2022. arXiv: 2210.03799[cs,eess]. URL: http://arxiv.org/abs/2210.03799 (visited on 04/11/2024).

[48] Brian McFee et al. “The million song dataset challenge”. In: *Proceedings of the 21st International Conference on World Wide Web*. 2012, pp. 909–916.

[49] James McInerney et al. “Explore, exploit, and explain: personalizing explainable recommendations with bandits”. In: *Proceedings of the 12th ACM Conference on Recommender Systems*. RecSys ’18. New York, NY, USA: Association for Computing Machinery, Sept. 27, 2018, pp. 31–39. ISBN: 978-1-4503-5901-6. DOI: 10.1145/3240323.3240354. URL: https://dl.acm.org/doi/10.1145/3240323.3240354 (visited on 06/20/2024).

[50] Douglas C Montgomery, Elizabeth A Peck, and G Geoffrey Vining. *Introduction to linear regression analysis*. John Wiley & Sons, 2021.

[51] Lawrence Page. *The PageRank citation ranking: Bringing order to the web*. Tech. rep. Technical Report, 1999.

[52] Elias Pampalk. *Islands of music: Analysis, organization, and visualization of music archives*. na, 2001.

[53] Eli Pariser. *The filter bubble: What the Internet is hiding from you*. penguin UK, 2011.

[54] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. “DeepWalk: Online Learning of Social Representations”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. Aug. 24, 2014, pp. 701–710. DOI: 10.1145/2623330.2623732. arXiv: 1403.6652[cs]. URL: http://arxiv.org/abs/1403.6652 (visited on 02/05/2024).

[55] Savvas Petridis et al. “TastePaths: Enabling Deeper Exploration and Understanding of Personal Preferences in Recommender Systems”. In: *27th International Conference on Intelligent User Interfaces*. IUI ’22. New York, NY, USA: Association for Computing Machinery, Mar. 22, 2022, pp. 120–133. ISBN: 978-1-4503-9144-3. DOI: 10.1145/3490099.3511156. URL: https://dl.acm.org/doi/10.1145/3490099.3511156 (visited on 02/22/2024).

[56] Martin Pichl, Eva Zangerle, and Günther Specht. “Towards a Context-Aware Music Recommendation Approach: What is Hidden in the Playlist Name?” In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 2015 IEEE International Conference on Data Mining Workshop (ICDMW). ISSN: 2375-9259. Nov. 2015, pp. 1360–1365. DOI: 10.1109/ICDMW.2015.145. URL: https://ieeexplore.ieee.org/abstract/document/7395827?casa_token=lcVJAMzBWl8AAAAA:nwsUjzLIgI_6IUTsu5JbsWKuHRRiPLyuzuHCjhZeoSb8PNOuuMlv4un2UM4HEKsgzBD1KILB6Fg (visited on 05/15/2024).

[57] Dulce B Ponceleon and Malcolm Slaney. "„Multimedia Information Retrieval."". In: *Modern Information Retrieval–The Concepts and Technology behind Search* (2011), pp. 587–639.

[58] Al Mamunur Rashid et al. "Getting to know you: learning new user preferences in recommender systems". In: *Proceedings of the 7th international conference on Intelligent user interfaces*. IUI '02. New York, NY, USA: Association for Computing Machinery, Jan. 13, 2002, pp. 127–134. ISBN: 978-1-58113-459-9. DOI: 10.1145/502716.502737. URL: https://dl.acm.org/doi/10.1145/502716.502737 (visited on 12/01/2024).

[59] Francesco Ricci, Lior Rokach, and Bracha Shapira, eds. *Recommender Systems Handbook*. New York, NY: Springer US, 2022. ISBN: 978-1-07-162196-7 978-1-07-162197-4. DOI: 10.1007/978-1-0716-2197-4. URL: https://link.springer.com/10.1007/978-1-0716-2197-4 (visited on 10/12/2023).

[60] Francesco Ricci et al., eds. *Recommender Systems Handbook*. Boston, MA: Springer US, 2011. ISBN: 978-0-387-85819-7 978-0-387-85820-3. DOI: 10.1007/978-0-387-85820-3. URL: https://link.springer.com/10.1007/978-0-387-85820-3 (visited on 11/11/2024).

[61] Neil Rubens et al. "Active Learning in Recommender Systems". In: *Recommender Systems Handbook*. Ed. by Francesco Ricci, Lior Rokach, and Bracha Shapira. Boston, MA: Springer US, 2015, pp. 809–846. ISBN: 978-1-4899-7637-6. DOI: 10.1007/978-1-4899-7637-6_24. URL: https://doi.org/10.1007/978-1-4899-7637-6_24 (visited on 11/29/2024).

[62] Keigo Sakurai et al. "[Paper] Deep Reinforcement Learning-based Music Recommendation with Knowledge Graph Using Acoustic Features". In: *ITE Transactions on Media Technology and Applications* 10.1 (2022), pp. 8–17. DOI: 10.3169/mta.10.8.

[63] Antonia Saravanou et al. "MULTI-TASK LEARNING OF GRAPH-BASED INDUCTIVE REPRESENTATIONS OF MUSIC CONTENT". In: (2021).

[64] Andrew I. Schein et al. "Methods and metrics for cold-start recommendations". In: *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*. SIGIR '02. New York, NY, USA: Association for Computing Machinery, Aug. 11, 2002, pp. 253–260. ISBN: 978-1-58113-561-9. DOI: 10.1145/564376.564421. URL: https://dl.acm.org/doi/10.1145/564376.564421 (visited on 12/01/2024).

[65] Michael Schlichtkrull et al. "Modeling relational data with graph convolutional networks". In: *The semantic web: 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*. Springer. 2018, pp. 593–607.

[66] Klaus Seyerlehner, Gerhard Widmer, and Peter Knees. "A comparison of human, automatic and collaborative music genre classification and user centric evaluation of genre classification systems". In: *Adaptive Multimedia Retrieval. Context, Exploration, and Fusion: 8th International Workshop, AMR 2010, Linz, Austria, August 17-18, 2010, Revised Selected Papers 8*. Springer. 2011, pp. 118–131.

[67] Nicollas Silva et al. "User Cold-start Problem in Multi-armed Bandits: When the First Recommendations Guide the User's Experience". In: *ACM Trans. Recomm. Syst.* 1.1 (Jan. 27, 2023), 2:1–2:24. DOI: 10.1145/3554819. URL: https://dl.acm.org/doi/10.1145/3554819 (visited on 12/01/2024).

[68] Aleksandrs Slivkins. "Contextual Bandits with Similarity Information". In: *Proceedings of the 24th Annual Conference on Learning Theory*. Proceedings of the 24th Annual Conference on Learning Theory. ISSN: 1938-7228. JMLR Workshop and Conference Proceedings, Dec. 21, 2011, pp. 679–702. URL: https://proceedings.mlr.press/v19/slivkins11a.html (visited on 12/01/2024).

[69] Aleksandrs Slivkins et al. "Introduction to multi-armed bandits". In: *Foundations and Trends® in Machine Learning* 12.1-2 (2019), pp. 1–286.

[70] Rui Sun et al. "Multi-modal Knowledge Graphs for Recommender Systems". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. CIKM '20. New York, NY, USA: Association for Computing Machinery, Oct. 19, 2020, pp. 1405–1414. ISBN: 978-1-4503-6859-9. DOI: 10.1145/3340531.3411947. URL: https://dl.acm.org/doi/10.1145/3340531.3411947 (visited on 03/20/2024).

[71] Zhiqing Sun et al. “Rotate: Knowledge graph embedding by relational rotation in complex space”. In: *arXiv preprint arXiv:1902.10197* (2019).

[72] Maria Taramigkou et al. “Escape the bubble: guided exploration of music preferences for serendipity and novelty”. In: *Proceedings of the 7th ACM conference on Recommender systems*. RecSys ’13. New York, NY, USA: Association for Computing Machinery, Oct. 12, 2013, pp. 335–338. ISBN: 978-1-4503-2409-0. DOI: 10.1145/2507157.2507223. URL: https://dl.acm.org/doi/10.1145/2507157.2507223 (visited on 02/22/2024).

[73] Laurens Van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE.” In: *Journal of machine learning research* 9.11 (2008).

[74] Petar Veličković et al. “Graph attention networks”. In: *arXiv preprint arXiv:1710.10903* (2017).

[75] Ivan Vendrov et al. “Gradient-Based Optimization for Bayesian Preference Elicitation”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.6 (Apr. 3, 2020). Number: 06, pp. 10292–10301. ISSN: 2374-3468. DOI: 10.1609/aaai.v34i06.6592. URL: https://ojs.aaai.org/index.php/AAAI/article/view/6592 (visited on 10/29/2024).

[76] Michel Verleysen and Damien François. “The curse of dimensionality in data mining and time series prediction”. In: *International work-conference on artificial neural networks*. Springer. 2005, pp. 758–770.

[77] Xiang Wang et al. “KGAT: Knowledge Graph Attention Network for Recommendation”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. July 25, 2019, pp. 950–958. DOI: 10.1145/3292500.3330989. arXiv: 1905.07854[cs, stat]. URL: http://arxiv.org/abs/1905.07854 (visited on 03/12/2024).

[78] Xinxi Wang et al. “Exploration in Interactive Personalized Music Recommendation: A Reinforcement Learning Approach”. In: *ACM Trans. Multimedia Comput. Commun. Appl.* 11.1 (Sept. 4, 2014), 7:1–7:22. ISSN: 1551-6857. DOI: 10.1145/2623372. URL: https://dl.acm.org/doi/10.1145/2623372 (visited on 12/01/2024).

[79] Zhen Wang et al. “Knowledge graph embedding by translating on hyperplanes”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 28. 1. 2014.

[80] Geraint A Wiggins. “Semantic gap?? Schemantic schmap!! Methodological considerations in the scientific study of music”. In: *2009 11th IEEE International Symposium on Multimedia*. IEEE. 2009, pp. 477–482.

[81] Hojin Yang et al. “Bayesian Preference Elicitation with Keyphrase-Item Coembeddings for Interactive Recommendation”. In: *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. UMAP ’21. New York, NY, USA: Association for Computing Machinery, June 21, 2021, pp. 55–64. ISBN: 978-1-4503-8366-0. DOI: 10.1145/3450613.3456814. URL: https://dl.acm.org/doi/10.1145/3450613.3456814 (visited on 10/29/2024).

[82] Jin-Cheng Zhang et al. “A review of recommender systems based on knowledge graph embedding”. In: *Expert Systems with Applications* 250 (Sept. 15, 2024), p. 123876. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2024.123876. URL: https://www.sciencedirect.com/science/article/pii/S0957417424007425 (visited on 12/05/2024).

[83] Xiaoxue Zhao, Weinan Zhang, and Jun Wang. “Interactive collaborative filtering”. In: *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. CIKM ’13. New York, NY, USA: Association for Computing Machinery, 2013, pp. 1411–1420. ISBN: 978-1-4503-2263-8. DOI: 10.1145/2505515.2505690. URL: https://dl.acm.org/doi/10.1145/2505515.2505690 (visited on 12/01/2024).

[84] Sijin Zhou et al. “Interactive Recommender System via Knowledge Graph-enhanced Reinforcement Learning”. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’20. New York, NY, USA: Association for Computing Machinery, July 25, 2020, pp. 179–188. ISBN: 978-1-4503-8016-4. DOI: 10.1145/3397271.3401174. URL: https://dl.acm.org/doi/10.1145/3397271.3401174 (visited on 12/01/2024).

[85] Zhenyu Zhu, Liusheng Huang, and Hongli Xu. “Collaborative Thompson Sampling”. In: *Mobile Networks and Applications* 25.4 (Aug. 1, 2020), pp. 1351–1363. ISSN: 1572-8153. DOI: 10.1007/s11036-019-01453-x. URL: https://doi.org/10.1007/s11036-019-01453-x (visited on 10/03/2024).

# A

# Experiment Supplementary

## A.1. Individual Survey Question Responses

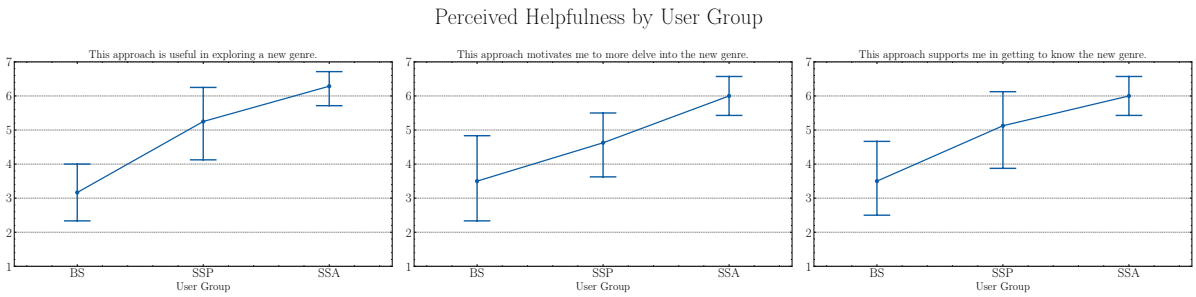



**Figure A.1:** Average individual question responses for helpfulness per group.

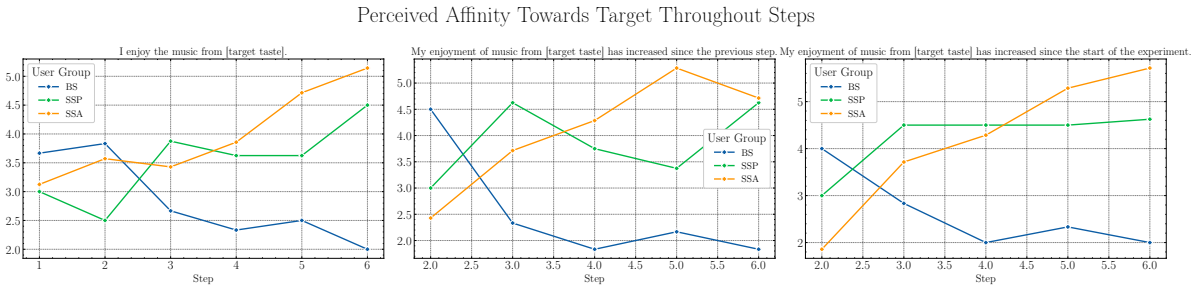



**Figure A.2:** Average individual question responses from factor affinity towards target throughout the steps for different user groups.

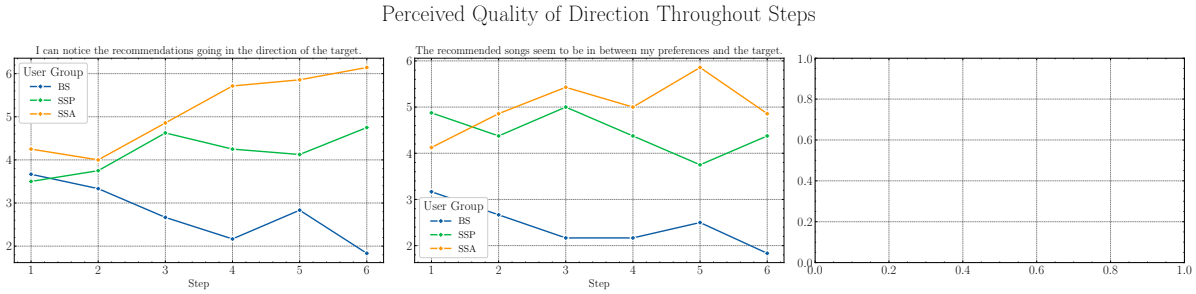



**Figure A.3:** Average individual question responses from factor quality of direction throughout the steps for different user groups.

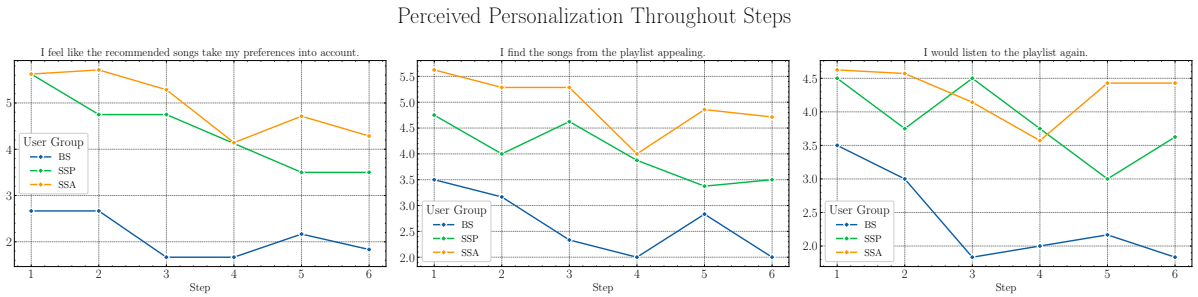



**Figure A.4:** Average individual question responses from factor personalization throughout the steps for different user groups.

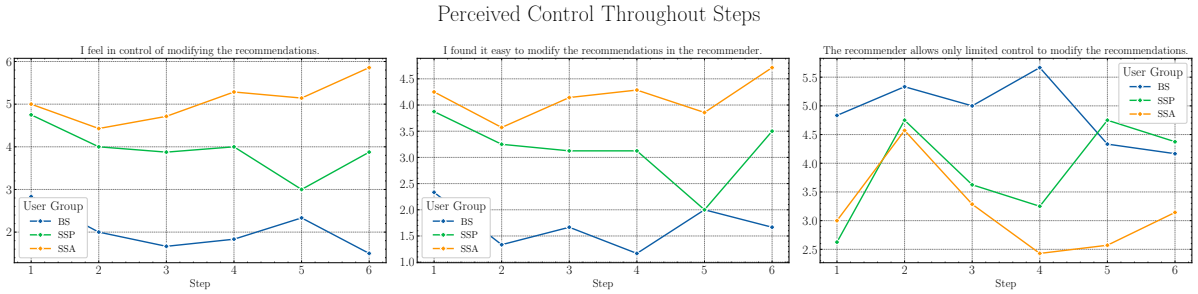



**Figure A.5:** Average individual question responses from factor control throughout the steps for different user groups.

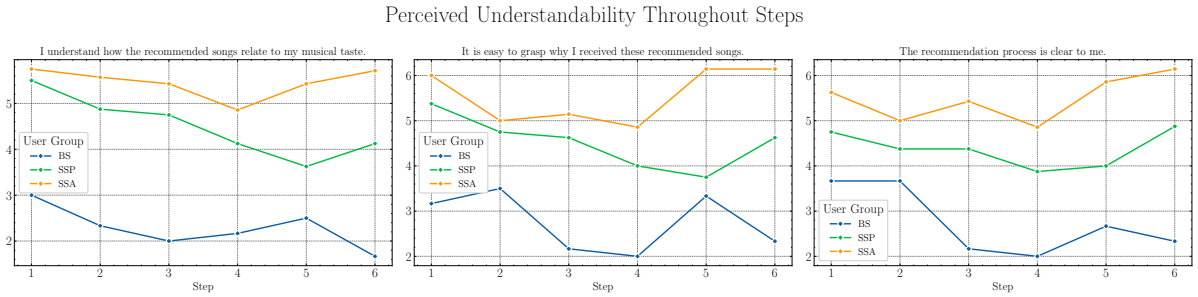



**Figure A.6:** Average individual question responses from factor understandability throughout the steps for different user groups.

# A.2. Experiment Platform Steps

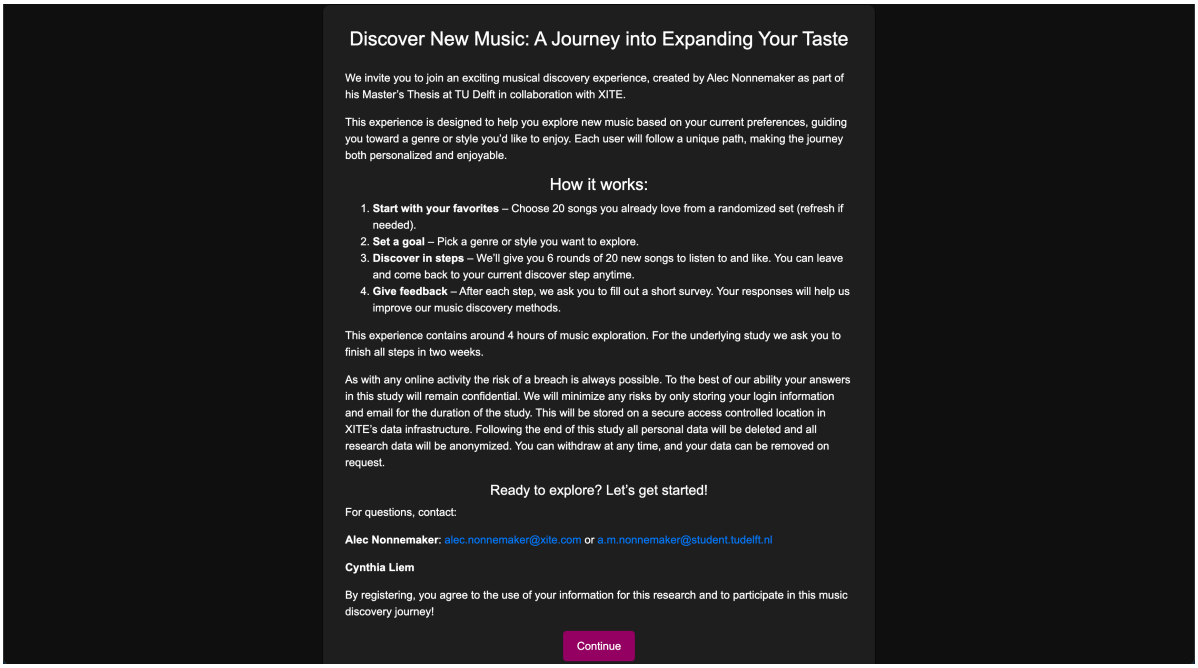



**Figure A.7:** The opening screen combining a welcome message and explanations with informed consent.

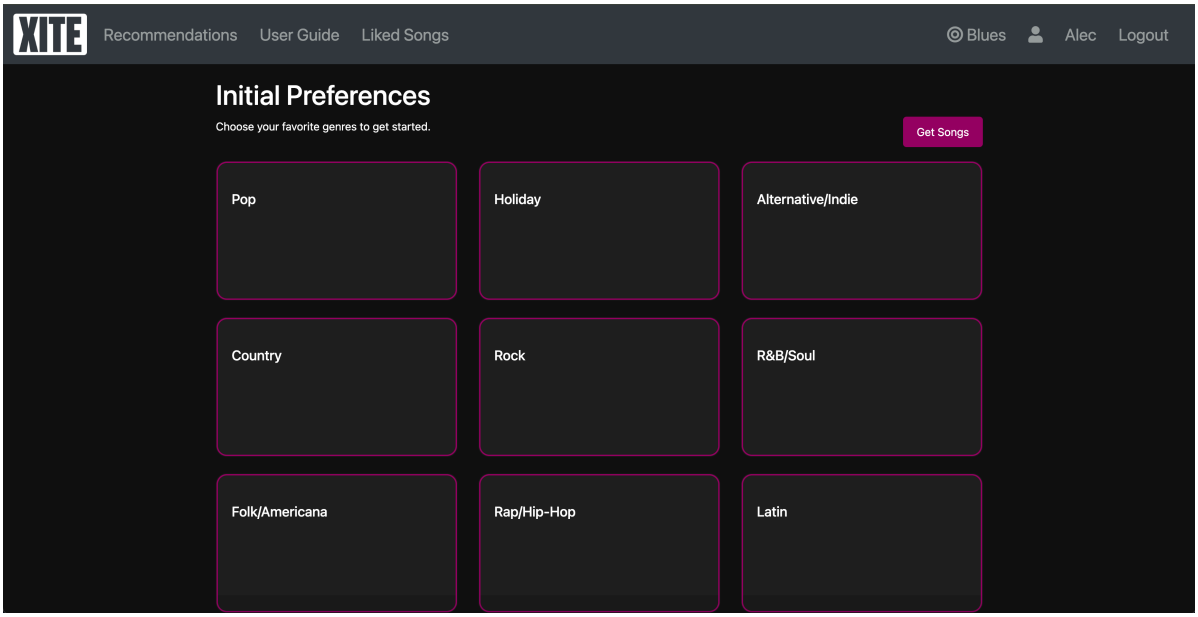



**Figure A.8:** The user is asked to select their preferred genre of songs.

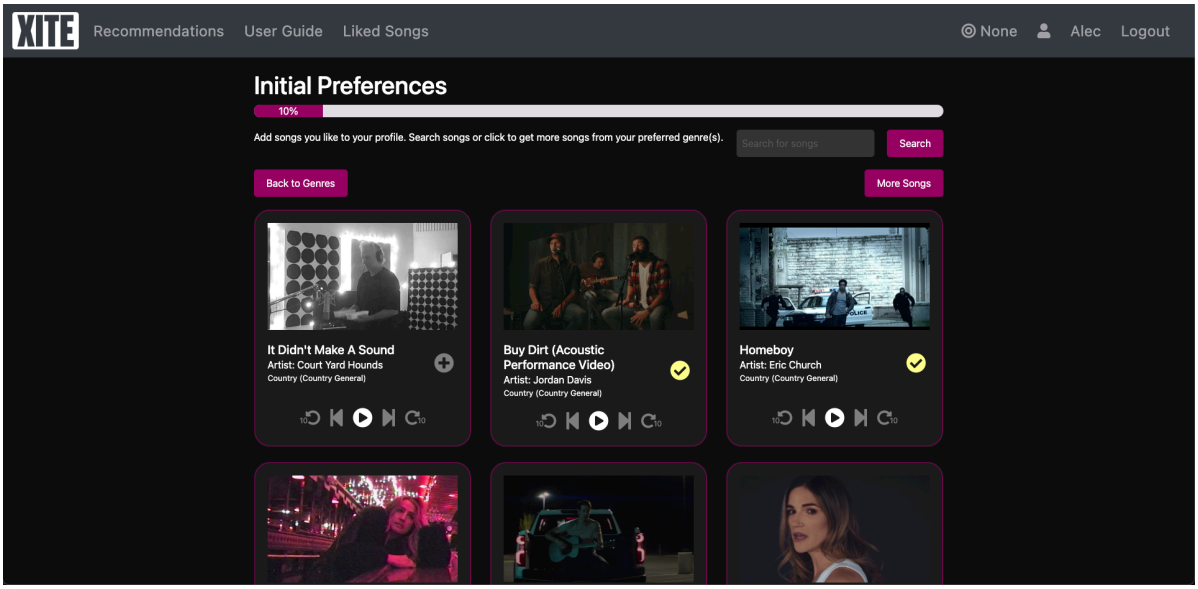



**Figure A.9:** A cold-start user can search for and flip through their preferred genre of songs to fill their initial preference profile.

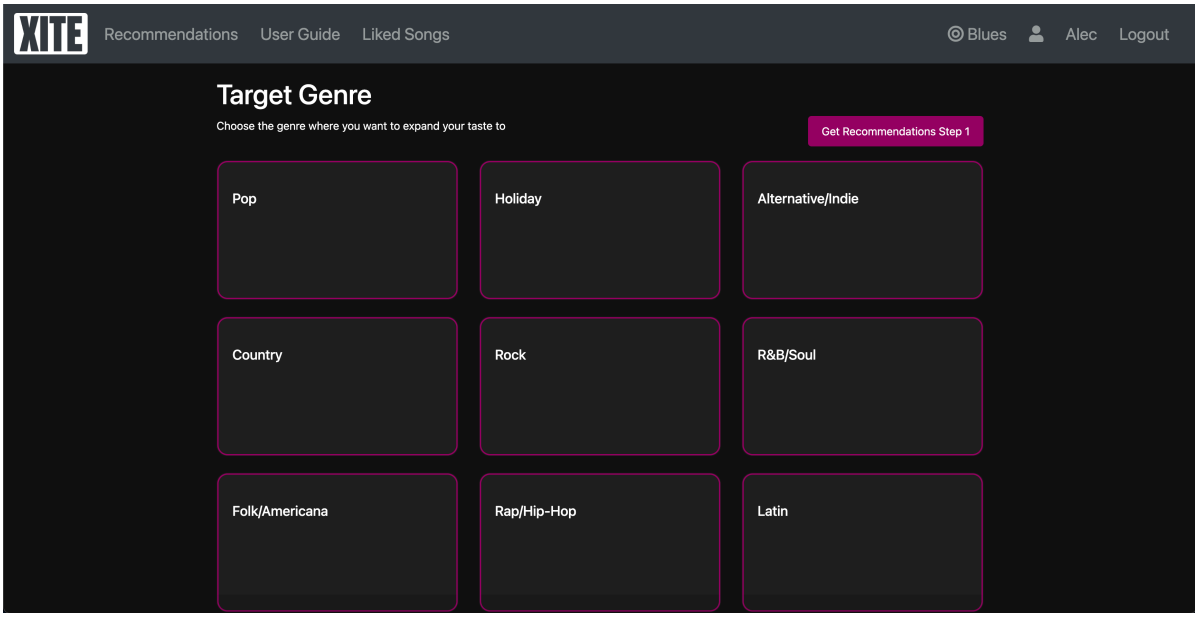



**Figure A.10:** The user is asked to set a target genre for exploration.

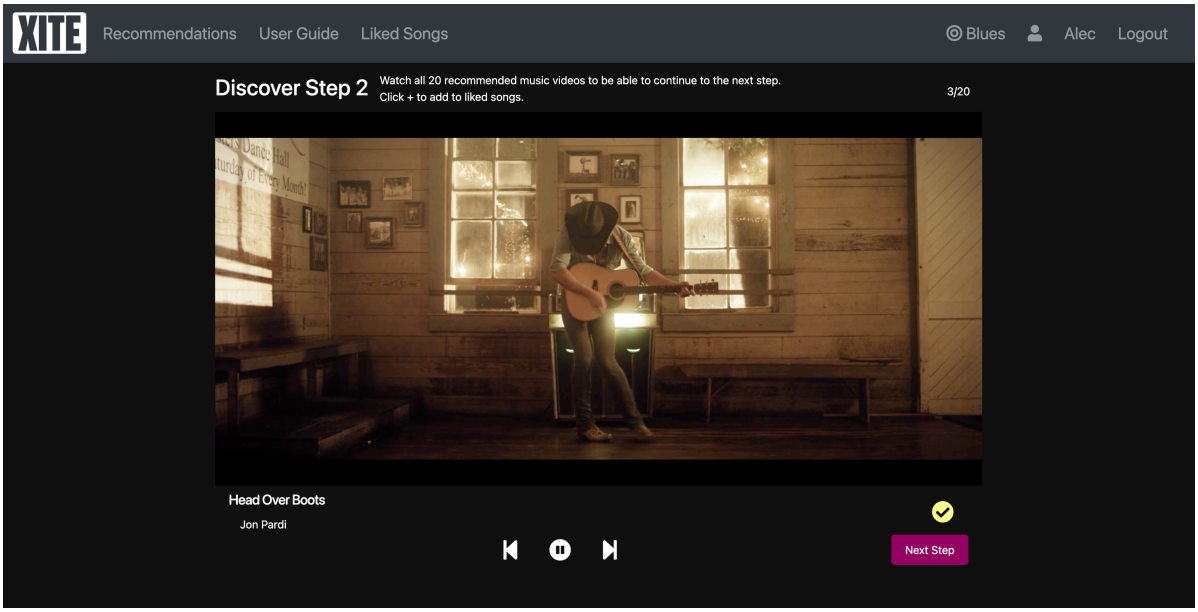



**Figure A.11:** In every discovery step the user receives 20 recommendations one-by-one in random order. Users can add a song to the set of liked songs in their user profile and can move to the next step after going through all 20 songs once.

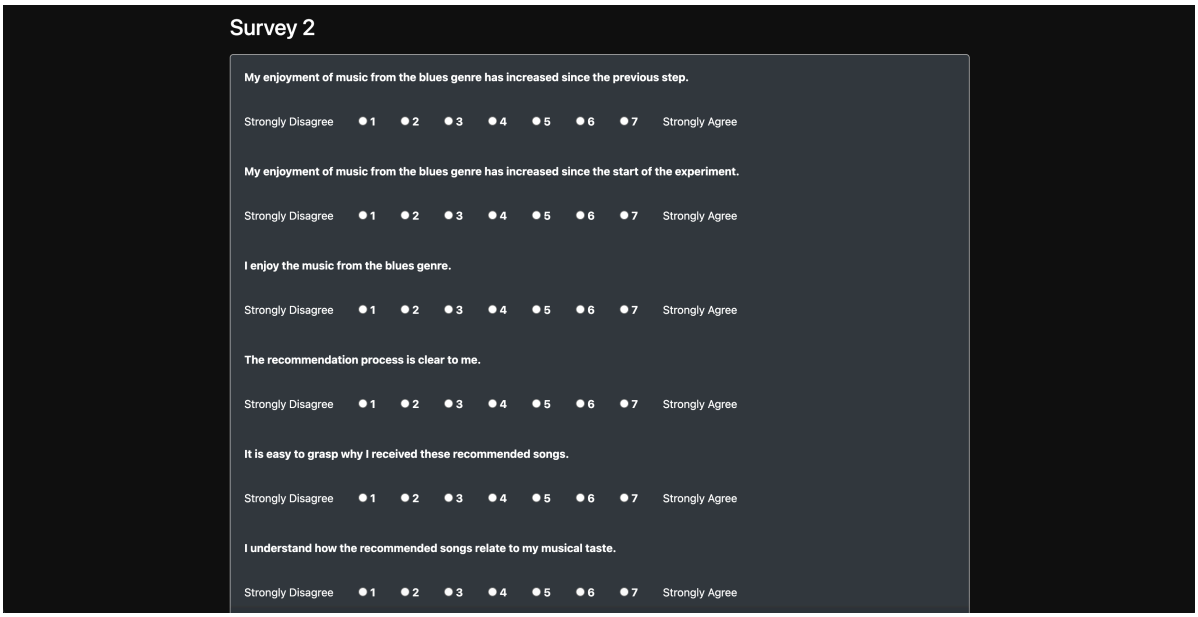



**Figure A.12:** After every step the user is asked to fill in a survey on their experience.