People know how diverse their music recommendations should be; why don't we?

by

Kyle Robinson

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2021

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

The entirety of this thesis consists of work authored by Kyle Robinson, with supervision and editing provided by Dr. Dan Brown at the University of Waterloo. Major contributions to this thesis have been adapted and extended from the following conference publication:

Kyle Robinson, Dan Brown, and Markus Schedl. 2020. **User Insights on Diversity in Music Recommendation Lists**. In *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR 2020)*, 446-453 ISBN: 978-0-9813537-0-8

For this publication, Kyle Robinson was the sole author and developer, primary supervision and editing was provided by Dr. Dan Brown at the University of Waterloo, and additional guidance was provided by Markus Schedl at Johannes Kepler University Linz.

Additionally, some content for this thesis (specifically chapters 4-6) has been adapted from as of yet unpublished works.

Abstract

While many researchers have proposed various ways of quantifying recommendation list diversity, these approaches have had little input from users on their own perceptions and preferences in seeking diversity. Through a set of user studies we provide a better understanding of how users view the concept of diversity in music recommendations, and how intra-list diversity can be adapted to better represent their diversity preference. Our results show that users have a clear idea of what music recommendation diversity means to them, accuracy metrics do not model overall list satisfaction, and filtering recommendations on genre before list diversification can positively impact list satisfaction. More importantly, our results highlight the need to base music recommendation metrics on insights from real people.

Acknowledgements

The past two years have lead me through the highest and lowest points of my life thus far, and there is absolutely no way I could have completed the work presented here alone.

Thank you Mom, Dad, Brandon, Jesse, and Jamie for always being there despite my lackluster performance at keeping in touch. You provided the foundation needed to push through, and an absolutely unwavering amount support regardless of your own personal thoughts and opinions. A similar thank you to my, now official, unofficial family: Bailey, Chris, Colin, Dallas, Dave, Kenny, Mason, and Quin.

This work would also have been impossible without the passive and active support of so many I met at Waterloo. Thank you to all those in HCI lab who motivated the evolution of my work and adopted me, especially Johann, and to the UW Badminton Club team. An extra thank you to Jay, Jessy, Ellie, and Mike for helping maintain my sanity during the dreaded spring and summer of 2020, and just being absolutely fantastic friends.

Most of all I would like to thank Dan for his endless support and accommodation of my ideas, ever-changing research interests, and personal circumstances. Your support through one of the most gratifying and humbling periods of my life will not be soon forgotten.

Table of Contents

List of Tables ix

List of Figures x

1 Introduction 1

1.1 Outline . . . 3

2 Background and Related Work 4

2.1 Diversity in Recommender Systems . . . 4

2.2 Optimising for Diversity . . . 5

3 Study 1 - User Insights on Diversity in Music Recommendation Lists 6

3.1 Overview . . . 6

3.2 Methodology . . . 6

3.2.1 Interactive Recommendation Lists . . . 7

3.2.2 User Study . . . 9

3.3 Results . . . 11

3.3.1 Music Discovery . . . 11

3.3.2 Recommendations . . . 12

3.3.3 Interactive Diversity . . . 12

3.3.4 User Perceptions of Diversity . . . 13

3.4 Discussion . . . 16
3.4.1 RQ1: How do users feel about diversity in personalized music recommendation lists? . . . 17
3.4.2 RQ2: How might users optimise their own level of diversity in personalized recommendation lists? . . . 18
3.4.3 Summary . . . 19

4 Study 2 - Expanding Diversity Scope Through Personalized Filtering 20
4.1 Overview . . . 20
4.2 Methodology . . . 20
4.2.1 Recommendation Overview . . . 21
4.2.2 Item Features . . . 25
4.2.3 Music Recommendation Lists . . . 25
4.2.4 User Study . . . 27
4.3 Results . . . 30
4.3.1 Data and Demographics . . . 30
4.3.2 Pre-Interaction Survey . . . 31
4.3.3 Recommendation List Evaluation . . . 36
4.4 Discussion . . . 45
4.4.1 Satisfaction . . . 45
4.4.2 Diversity . . . 46
4.4.3 Genre and Filtering . . . 46
4.4.4 Diversity and Personalization . . . 47
4.4.5 Pushing Diversity Further . . . 47
4.4.6 Summary . . . 48

5 Large-scale Experimental Design for Music Recommender Systems: Challenges and Possible Solutions **49**

5.1 Running a Live Recommendation Study 49

5.1.1 General Architecture 50

5.1.2 Users and Data 50

5.1.3 Computation 51

5.1.4 Summary 52

6 Conclusion **54**

References **56**

List of Tables

3.1 Positives and negatives of more and less diversity in recommendation lists expressed by participants. 17

4.1 Breakdown of implicit data collected from Last.fm and filtered to remove tracks with 10 plays or less. 21

4.2 Final results of both recommendation algorithms. Note that the test results reflect models trained using the combined training and validation data sets. 25

4.3 P-values from applying Dunn's (LQ0), and Nemenyi (LQ1-LQ7) tests to analyse Likert response distributions. Significant differences ($p \leq 0.001$) were first identified using Kruskal-Wallis (LQ0), and Friedman (LQ1-LQ7) tests. Note that p-values close to 0.05 (i.e., 0.046) are not necessarily relevant due to the large number of entries. 41

List of Figures

3.1 The mid-motion user interface state directly after moving the diversity slider seen at the top. The top 7/100 songs as ranked by Equation (3.2) are displayed as 30 second song previews. As the slider is moved the songs shift from the old order to the new order over a period of 2 seconds. Songs which leave the top 7 move off the bottom and new songs appear from the bottom highlighted in green for 5 seconds. The circular *Well Known* buttons on the left remove songs from the list entirely. 10

3.2 Responses to Likert questions completed after every recommendation list, split between static lists and lists which were selected using the diversity slider. The final question was customized for each individual using their own definition of diversity obtained during the pre-interaction interview (see Section 3.3.4). 13

3.3 All user selections for diversity using the slider found in Figure 3.1. The legend corresponds to Likert responses to: "The list of recommended music was diverse." 14

4.1 The UI for music preview ratings. Each song is displayed as a 30 second preview from Spotify 28

4.2 Time taken to complete recommendation lists per user (a) and per list type (b). Users whose completion time was below 35s (black line) were removed from analysis. 31

4.3 Distribution of LEs per user in our base recommendation data set (50,437 users), compared to MTurk participants (92 users) before and after removing tracks not found in our base data set. 32

4.4 Processed and counted responses to the pre-interaction open-response questions: "What platform(s) do you primarily use to find new music?", and "What platform(s) do you primarily use to listen to music?" 33

4.5 Likert responses to pre-interaction survey questions on music diversity preference. 34

4.6 Responses to pre-interaction survey questions with binary responses. These questions were asked in order to find correlations between user selections and recommendation satisfaction. 35

4.7 Each bar shows the total count of tracks which participants identified as known in some form out of ≈ 920 total recommendations served. Colours show counts per-participant. 37

4.8 Likert responses to LQ3 between participants with different LE counts. Only the list types displayed showed any statistical significance in a prior Kruskal-Wallis test. We did not complete post hoc tests due to a small between-subject sample size and unbalanced groups. 40

4.9 Likert responses to recommendation list questions pertaining to satisfaction and recommendation quality. Black bars signify statistically different response distributions (thin: $p <= 0.05$, bold: $p \leq 0.001$). 42

4.10 Likert responses to questions on recommendation list diversity. Black bars signify statistically different response distributions ($p \leq 0.001$). 43

4.11 Likert responses to additional questions about recommendation list diversity. Black bars signify statistically different response distributions (thin: $p <= 0.05$, bold: $p \leq 0.001$). 44

Chapter 1

Introduction

As music consumption has moved from physical media to digital collections to streaming, people have changed the way they discover new music. As with other forms of consumption which have made the shift to digital media and marketplaces such as movies, television, and consumer products, data on music listening habits is more prevalent than ever. Accordingly, systems which use this data to market or recommend new content to users have become ubiquitous. These *music recommender systems* aim to provide satisfying music recommendations to users across a wide variety of contexts [38].

One common way of recommending music is to create a ranked list where the items are formed by the top-n recommendations, as produced by the used recommendation algorithm, sorted by recommendation relevance. To judge the quality of recommendations, various forms of accuracy metrics have been proposed. Typically borrowed from the field of *information retrieval*, these accuracy measurements aim to quantify how well a recommendation (or set of recommendations) aligns with a user's known preferences, or in some cases how satisfied a user will be with those recommendations [17].

In addition to accuracy, various other metrics have been proposed [17, 20]. These aptly named *beyond-accuracy* metrics include novelty, coverage, serendipity, and diversity [17]. Novelty relates to items which are unknown to the user, coverage relates to the proportion of items that can be recommended (item coverage) or to the proportion of users for which at least one recommendation can be made (user coverage), serendipity relates to the unexpectedness of a recommendation, and diversity relates to the dissimilarity of recommended items[17]. We focus our attention to diversity as it is well researched, and easily understood for music [41, 17, 20].

While diversity in *music recommender systems* is well researched, we are unaware of any

research which specifically explores use-provided perceptions of diversity in this domain. Additionally, most implementations of diversity treat the metric as a static variable between or within users. We argue that desire for diversity in recommendations may instead be situationaly dependent and use insights gleaned from users themselves to build and measure a simple top-n diversification approach. We present the results of two different user studies.

The first is an exploratory study in which 17 participants experienced an interactive recommendation system before and after completing surveys on their ideas of diversity in music recommendations. These participants were also interviewed and their responses were coded for common themes. In this first study, we found that users presented a range of definitions for diversity, linked ideal diversity levels to their mood, and distinguished between what we call inner and outer diversity. When asked to optimise their own level of diversity using our system, participant selections differed greatly within and between subjects.

The second is a larger and more targeted study in which 92 online participants were asked to evaluate three differently diversified personal recommendation lists each of two different collaborative filtering recommendation algorithms. Participants were also asked questions about their preference for novel music and diversity as they relate to concepts discovered in the first study. Among the many results we identified that accuracy and individual song ratings differed from overall list satisfaction, and our implementation of *inner diversity* filtering resulted in higher levels of list satisfaction despite no significant decrease in perceived diversity. We also found that participants were less satisfied with the recommendations from the neural network model's recommendations despite its superior performance in offline testing accuracy. Finally, we found that none of our diversification methods resulted in too diverse recommendations, suggesting that were not able to match all users diversity preferences.

The idea that accuracy metrics and offline experiments do not well represent real-world recommendation performance is not new [29]. We hope the work presented here serves as a reminder that recommender systems rely heavily on *human-computer interaction*, and user studies should be a vital part of their evaluation. If beyond-accuracy metrics are to help solve the disconnect between offline accuracy and online evaluations then they themselves should have a stronger basis in real people.

1.1 Outline

This thesis is organised as follows:

- Chapter 2 provides a brief overview and explanation of recommender system diversity.
- Chapter 3 describes the first study, which qualitatively explores user ideas and perceptions of diversity through an interactive system
- Chapter 4 describes the second study, which uses ideas gleaned from the first study to quantitatively analyse differently generated diverse recommendations.
- Chapter 5 presents a series of challenges to running live music recommendation studies and corresponding solutions.
- Chapter 6 summarizes the contents and results of this thesis and suggests future extensions.

Chapter 2

Background and Related Work

2.1 Diversity in Recommender Systems

Alongside novelty, coverage, and serendipity, diversity has long been identified as an important metric in providing satisfying automated recommendations to users across varying domains [20]. Diversity in this context traces back to information retrieval tasks, where it was used to resolve ambiguity in search queries [7]. Within recommender systems, diversity prevents over-personalization of recommendations to users, thereby increasing user satisfaction with recommendations [22, 20]. Research on diversity in recommender systems is extensive, and numerous different definitions have been proposed [22]. More generally, recommender system diversity has been described as the opposite of *similarity* [17, 1]. Among the most commonly researched and implemented definitions of diversity in *music recommender systems* is intra-list diversity (ILD) which measures the average pairwise dissimilarity of items using some chosen similarity metric; typically calculated using content features [1, 49]. Numerous other definitions for diversity have been proposed. These definitions range from modifications of ILD [2, 26], to completely novel approaches which do not rely on pairwise dissimilarity [44, 45, 15].

One highly cited alternative diversity metric proposed by Vargas *et al.* uses the distributions of genres in a users listening history and recommendations to satisfy three *properties* of diversity: genre coverage, redundancy, and size-awareness. The first two properties they borrow directly from *information retrieval* literature, and the last they contrive. Their method achieves the best performance in offline tests, but it is not evaluated with any users. Unsurprisingly, their method also outperforms others on the same *properties* they

define earlier. We are unaware of any existing research which grounds these *properties* as good evaluators of recommendation diversity.

2.2 Optimising for Diversity

Research on selecting optimal levels of diversity for recommender systems is extensive. In their original paper defining diversity as the opposite of similarity, Bradley and Smyth show that traditional recommender system outputs are not diverse, and diversity, in one metric, can be increased with minimal negative impact on accuracy [1]. Ziegler *et al.* further showed that user satisfaction with recommendation lists relies on more than accuracy by computing precision, recall, and satisfaction curves in a large user study [49]. Studies following this theme of incorporating existing diversity metrics with minimal negative impact on accuracy and/or satisfaction are plentiful [33, 48]. Whereas these works applied a global level of diversity to recommendations, recent work has focused on selecting levels of diversity on a per-user basis through user modeling [26, 11, 28, 14]. Interactive systems that allow users to explore recommendations through diversity have been explored outside of the music domain, but these systems aim to abstract diversity into a multi-dimensional user interface rather than allow for user selection of existing diversity metrics and algorithms [43, 47, 36].

Differences in user perceived diversity levels have been identified across varying recommendation algorithms for movies [12], and varying levels of intra-list diversification (ILD) for music [46]. Finally, user listening habits on diversity have been extracted from social networks [13], and playlists [32].

We are not aware of any research which explores user provided perceptions of diversity in personalized music recommendations, or allows them to directly modify existing diversity metrics on the fly. We begin to fill in this gap by providing knowledge on how well ILD aligns with user perceptions of diversity.

Chapter 3

Study 1 - User Insights on Diversity in Music Recommendation Lists

3.1 Overview

Study 1 is a largely exploratory experiment on how users themselves define diverse music recommendations. We aim to answer two primary research questions:

- RQ1: How do users feel about diversity in personalized music recommendation lists?
- RQ2: How might users optimise their own level of diversity in personalized recommendation lists?

A small set of users were asked a series of questions on diversity before and after interacting with a system showing personalized recommendations. The system allowed them to modify the diversity level of their recommendations on the fly, and they were asked to select the optimal level. Participants were also interviewed on their ideas of diversity, and these interviews were coded into a series of more general observations. The results of this study were presented as a full paper at the 21st International Society for Music Information Retrieval Conference (ISMIR 2020) [34].

3.2 Methodology

To control all aspects of recommendation and diversity inclusion, and to minimise restricting participants' consumption method, we implemented a collaborative filter recommender.

We used Last.fm as a source of raw listening data, and presented song previews in the form of standardised 30 second track previews from Spotify.

3.2.1 Interactive Recommendation Lists

3.2.1.1 Data

To get a background sample of listening behaviour, we collected a total of 341,764,569 unique listening events (LEs) from 51,669 unique users whose region was set to North America using the Last.fm API. Users were found by crawling the Last.fm social graph using the *user.getFriends* endpoint. We had a limit of 10,000 LEs accepted per user, and only accepted LEs between January 12, 2019 and when we collected them in February 2020. The median number of LEs per user is 7744, 25^{th} percentile is 3502, and 75th^{th} percentile is 9842.

We used a simple key consisting of artist and track name tuples in order to identify individual tracks. The final user-track-interaction matrix, used to generate recommendations (see Section 3.2.1.2), contains 141,205,668 non-zero entries (play counts) across 12,300,857 unique artist-track tuples, resulting in a $51,669 \times 12,300,857$-sparse matrix. This system does not account for potentially inaccurate metadata obtained from Last.fm, but does account for the same track across different releases. Entries in this matrix are integers which correspond to the number of unique times a user (row) played the track (column). An anonymized version of this data is available upon request.

3.2.1.2 Collaborative Filtering & Diversity

For generating recommendations we used an Alternating Least Squares (ALS) matrix factorization algorithm which is designed specifically for implicit feedback data sets [19, 14]. This algorithm generates a latent matrix factorization consisting of a smaller user-matrix, and item-matrix. Each user and item then has a corresponding vector, and the product of any user-item vectors represents the relevance of the user to the item. On initial training, all users and items are used to generate the latent factorization. After training, vectors for unseen users can be calculated using their listening history and the latent item factorizations. This method is known as the *fold-in* method [35].

In practice, ALS results in one vector for each user consisting of a non-negative real number (recommendation relevance) for each track in the database; higher numbers are considered more relevant recommendations. The ALS collaborative filter recommender

was implemented using the *Implicit* python library [16], and was trained using the dataset described in Section 3.2.1.1. Hyper-parameters were optimised using 5-fold cross-validation and Mean Average Precision for top-10 recommendations (MAP@10) over 60 iterations of randomized search resulting in 160 factors, 28 iterations, a scaling factor of $\alpha = 774$, and regularization term of $\lambda = 1$.

The trained collaborative filter recommender was used to generate top-400 track recommendation lists for a single Last.fm username (see Section 3.2.2). To facilitate multiple recommendation lists per-user we split this list evenly into four smaller lists of 100 tracks each. We iteratively divided tracks between lists by oscillating from the first to fourth list and adding tracks by ascending rank resulting in four lists with identical track rank sums. Each track within each of the four lists was then re-assigned a rank from 1-100 with one being the most relevant. In order to measure diversity we used the latent vectors generated for each track during matrix factorization as descriptors. Similar to previous work [46, 14], we calculated a form of ILD (d_i) by summing the Euclidean distance of one track's descriptors (v_i) from all other descriptors (v_j) in each top-100 list.

$$d_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} ||v_i - v_j|| \tag{3.1}$$

This calculation differs from previous work in that diversity is only calculated once and not as part of a greedy diversification algorithm. Higher values of d_i correspond to more diverse tracks in relation to others in the list. Tracks are assigned additional ranks from 1-100 where rank one is the most diverse. We are left with four unique top-100 recommendation lists for a given user where each track is assigned a rank for relevance (R_i), and diversity(D_i).

The final ranking (F_i) is calculated as a trade off between relevance and diversity controlled by a convex combination of both ranks, with a diversity parameter β.

$$F_i = (1 - \beta) * R_i + \beta * D_i \tag{3.2}$$

The user interface, shown in Figure 3.1, displays the top-7 tracks of each top-100 recommendation list based on F_i in the form of 30 second previews using Spotify Play Button widgets.[1] We chose to use top-7 recommendation lists to ensure user study session times under 70 minutes (see Section 3.2.2). An interactive slider that controls the value of

[1] https://developer.spotify.com/documentation/widgets/generate/play-button/

β is situated above the song previews. The left of this slider corresponds with $\beta = 0$ and the right side corresponds with $\beta = 1$ with a step size of 0.001. A *Well Known* button appears to the left of each song preview allowing users to remove songs which are not new to them.

Due to differences in the music collection available on Spotify and our own music database, as well as to avoid false-positives in retrieving song previews, we omitted all songs which did not match exact artist and song string queries to Spotify. This typically resulted in final recommendation lists of 95-100 tracks each.

3.2.2 User Study

Participants were recruited on the University of Waterloo campus through internal email lists and posters in February and March of 2020. After completing a digital information consent form participants were asked to complete a brief survey. As part of this survey they were asked to provide their Last.fm usernames, or alternatively were provided instructions on how to set up a Last.fm account and record their listening events to it. We required that participants had a minimum of 5 hours of LEs recorded before continuing to the interactive portion.

The interactive portion of the study involved a pre-interaction interview, two conditions of 4 trials each using four unique recommendation lists, and a post-interaction interview. Interviews were semi-structured. Pre-interaction interview questions focused on the importance of music discovery to the participant, how the participant finds new music, and what a diverse list of personalized recommendations means to them. Post-interaction interview questions focused on the perceived effect of the slider on recommendations, the static or variable nature of their selections across trials, and positives and negatives of diversity in music recommendations.

Trials 1-4 consisted of static top-7/100 recommendation lists each corresponding with one evenly split quarter of their top-400 recommendations as ranked only by recommender output (relevance) (see Section 3.2.1.2). The user-interface was similar to Figure 3.1 but without the slider. Participants were asked to listen to each preview, remove well known tracks, mark if they were familiar with the artist, and rate the recommendation on a four-point Likert scale of 'Strongly Dislike', 'Dislike', 'Like', or 'Strongly Like'. Only once every track was rated could the participant move to the next trial.

Trials 5-8 consisted of the same ranked lists as trials 1-4 (minus tracks marked as well-known) with the addition of the interactive slider to re-rank the larger hidden list based on the participants' selected level of diversity (see Section 3.2.1.2). The user-interface can

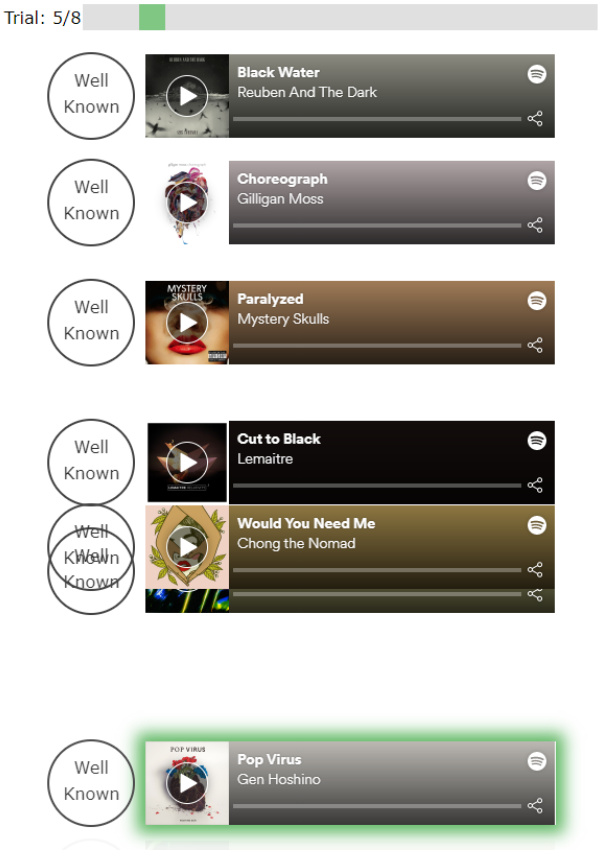


Figure 3.1: The mid-motion user interface state directly after moving the diversity slider seen at the top. The top 7/100 songs as ranked by Equation (3.2) are displayed as 30 second song previews. As the slider is moved the songs shift from the old order to the new order over a period of 2 seconds. Songs which leave the top 7 move off the bottom and new songs appear from the bottom highlighted in green for 5 seconds. The circular *Well Known* buttons on the left remove songs from the list entirely.

be seen in Figure 3.1. Participants were not told what the slider did and were instructed to find the position on the slider that resulted in the most satisfying recommendation list as a whole while removing tracks that were well known to them. Once the participant locked in this position they were again asked to mark if they were familiar with each song's artist, and rate each individual recommendation on the same four-point Likert scale before moving to the next trial.

Between each trial participants completed a survey with questions on their satisfaction with the final recommendation list, the level of diversity in the recommendation list, and how well the recommendation list portrayed the definition of diversity they provided in their pre-interview survey. Participants were paid $10 CAD upon completing the interactive portion of the study.

Pre- and post-interaction interviews were transcribed, and comments were then sorted into three categories: interaction, music discovery, and diversity. As in other qualitative music consumption studies we extracted individual ideas as statements from transcriptions and proceeded to build connections and groupings through affinity diagramming [6, 30]. Main ideas were highlighted and categorized into groupings of similar themes, and finally counts of each theme were collected. We specifically focused on responses regarding diversity.

3.3 Results

We recruited 18 participants, and removed one participant for marking all recommendations as *Well Known*, leaving 17 total participants. The median participant age was 23; the oldest was 29 and the youngest 19. Each user session took 50-70 minutes inclusive of interviews. Some sessions were completed face to face, and others involved the users connecting remotely to the interactive system.

3.3.1 Music Discovery

When asked how they discovered new music, 9 said they used Spotify, 9 used YouTube, 5 used movies and/or television, 4 relied on friends, 3 used radio, and 4 used some other online service such as Amazon or Soundcloud. The importance and frequency of finding new music varied significantly from user to user, and no clear patterns were observed. Some users noted that the primary reason they use music services such as Spotify is to enable

easier music discovery. When asked how important finding new music is to them, one user reported previously spending 5 hours per week looking for new music, but added:

> While it's still very important to me, I basically don't do it very often on my own anymore; I rely on Spotify to do almost all of it for me.

3.3.2 Recommendations

None of the participants had an existing Last.fm account, and the length of time during which users recorded their listening histories to Last.fm varied from one to three weeks. The median percentage of user LEs which existed in our CF database was 95%, with a maximum of 100% and a minimum of 65%. Median LE counts per-user used for recommendation generation were 256, with a max of 1156 and minimum of 86. All users marked and removed fewer than 100 tracks as well known across all trials, with the exception of one user who marked and removed 208.

When asked to rate individual recommendations on a 4-point Likert scale (Strongly Dislike, Dislike, Like, Strongly Like) 72.69% of songs were rated as 'Like' or 'Strongly Like' after locking in the diversity slider, and 74.79% in static lists. In addition to rating individual songs, participants were asked if they were satisfied with the list of recommended music for every trial. On a 5-point Likert scale (Strongly Disagree, Disagree, Undecided, Agree, Strongly Agree) 75% of diversified recommendation lists resulted in a positive response, with 50% for static recommendation lists.

3.3.3 Interactive Diversity

In addition to the task of selecting an optimal position for the diversity slider, participants were asked a series of questions on how diverse they felt each recommendation list was. Responses to these questions can be seen in Figure 3.2. In order to visualise how participant responses on diversity align with their diversity selections, Figure 3.3 shows all 17 user's diversity selections coded with their Likert response on diversity. User selections varied greatly between their own recommendation lists and between other users'. Likert responses for perceived diversity did not fall in line with levels of β.

As a part of the post-interaction interview participants were asked to identify what they thought the slider was changing within their recommendation lists. Of the 17 participants, 5 identified it to increase diversity directly, 3 identified some change in genres, and 4 had

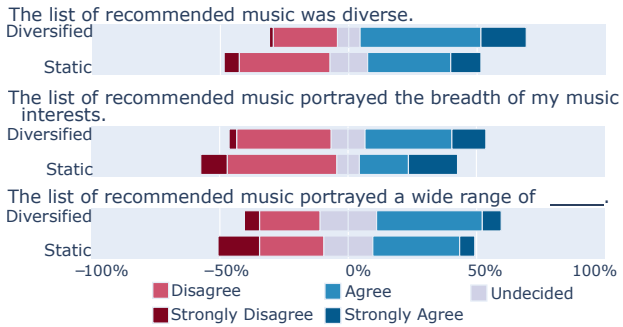


Figure 3.2: Responses to Likert questions completed after every recommendation list, split between static lists and lists which were selected using the diversity slider. The final question was customized for each individual using their own definition of diversity obtained during the pre-interaction interview (see Section 3.3.4).

no explanation. The remaining participants identified the slider to change the perceived gender of vocalists, increase 'newness', increase distaste, increase quality, and decrease quality. In one case where a participant identified the slider to effect genre they stated:

> I noticed initially that the first side of the slider was giving me a bunch of songs from different genres. The more I was sliding it the more it was giving me the songs... from the genre which I like.

In another case where a participant was unable to identify the effect of the slider and was asked what they would like the slider to do they answered:

> The way I imagined it was... less diverse on the one side and more and the other side. That's something I could definitely use.

When asked about their experience using the system some users expressed difficulty in remembering which locations of the slider they preferred most, and frustration over which songs remained on the list and which were moved off. In total, 10 users preferred interacting with the static list, and 7 preferred using the interactive slider.

3.3.4 User Perceptions of Diversity

During the pre-interaction interview participants were asked what they would mean if they were looking for diverse recommendations. In addition to their open ended responses, they

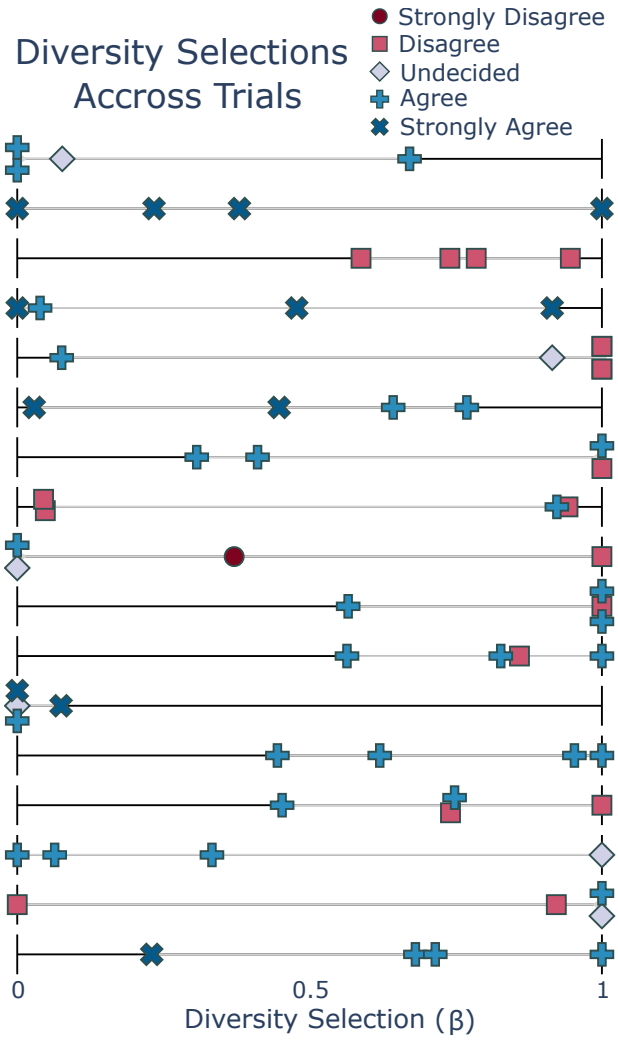


Figure 3.3: All user selections for diversity using the slider found in Figure 3.1. The legend corresponds to Likert responses to: "The list of recommended music was diverse."

were challenged to come up with a single word or idea that could be used in place of diversity. Of the responses to this question, 13 answered a difference in **genres**, 2 answered **cultural differences**, and the remaining participants responded with **originality**, **variety**, and differences in **artists**.[2] These definitions were used to complete the third question in Figure 3.2.

Coding of participants' comments on diversity in their own personalized music recommendations resulted in two primary themes which we labeled **diversity meaning**, and **listener mood**. Comments which we classified under **diversity meaning** are deeply intertwined with personal definitions of diversity, and can be more specifically categorized into what we identify as **inner and outer diversity**; that is music within the bounds of existing preferences, and music outside of these bounds. In answering the interview question on the meaning of diverse recommendations, 8 participants made reference to a preference for this idea of inner or outer diversity. Participant comments expressing a preference for inner diversity include:

> Diverse in the–within the boundaries of the things that I like.

> I like a playlist which recommends me songs on the genre I like... the important thing is to get diversified music in my genre only... to stay in the same genre but diversity in artists.

> A diverse music recommendation I think should still be within the category of music that I usually listen to, but it should be different artists or different albums that I haven't listened to so far.

Comments expressing a preference for outer diversity include:

> [Diverse recommendations are] something new, something exciting. Something that I'm not used to, like I've never heard before.

> [Diverse recommendations] would be music from other genres that maybe I haven't listened to very much, but still somewhat akin to the ones that I have listened to.

[2] One participant was unable to choose between genre and culture.

Mood was the second most common theme referenced, and was explicitly mentioned by 7 participants. Only 2 participants mentioned context. Participants referenced mood as a primary factor in how much diversity they want in their music recommendations at any given time. Notable comments on mood include:

> ...depending on my mood–whether I'm looking for more of the same things that I already like–I could set that slider to show me less diverse music–if I'm in the mood.

> [I] like a piece of music right now because of the mood that I am in, but I might not like it while I'm listening to a very different kind of music. So diversity is good but I think in a weird way the recommender system should know when to recommend it.

> Sometimes you're in the mood of listening to one specific–like you don't want [a] diverse playlist. You just want to listen to sad songs. You just want a playlist that has a sad song. You don't want diversity

> If you're in a melancholic mood and then you don't have a very diverse playlist of melancholic music then you'd be happy about your music because that's your mood.

Participants also provided their thoughts on the positives and negatives of diversity in personalized recommendations, and a summary of these thoughts can be found in Table 3.1. Participants generally felt that while diversity could enable music discovery, it also increased the risk of disliking some recommendations.

3.4 Discussion

In this study we provided a primary analysis of user perceptions on diversity in personalized music recommendations. We also provided users an opportunity to directly optimise a diversity metric which until now had been algorithmically optimised for them. Although our results do not hold statistical power due to the small sample size, our semi-structured interviews facilitated valuable insights and answers to our posed research questions. These insights add to the growing number of other qualitative works in Music Information Retrieval research [30, 6, 23, 24, 25].

	More Diversity	Less Diversity
Possitive	Music Discovery ($N = 11$) Preference Discovery ($N = 4$) Interesting ($N = 4$)	Likely to Like ($N = 2$)
Negative	Likely to Dislike ($N = 8$) Dissatisfaction/Annoyance ($N = 2$)	Restrictive ($N = 4$) Repetitive ($N = 2$) High Risk/Reward ($N = 2$) Unremarkable ($N = 2$)

Table 3.1: Positives and negatives of more and less diversity in recommendation lists expressed by participants.

3.4.1 RQ1: How do users feel about diversity in personalized music recommendation lists?

Despite a large variability in user's feelings towards diversity in music recommendations, their ideas on its positives and negatives (Table 3.1) mostly align with the metric's purpose of reducing over-personalization. Beyond this, however, users attached more complex ideas such as personal preference discovery and interestingness to more diversity. Ideas such as this may in part explain the higher levels of satisfaction reported by users given more diverse recommendations.

The prevalence of mood in participants descriptions of diversity is especially notable when compared to the lack of references to their context; that is their location and current activity. As more focus is directed towards context-aware recommender systems [39], careful attention should be paid to not assume that ideal diversity levels can be determined by context alone. Diversity optimisation may also serve as an ideal jumping off point for mood-based recommendation [9, 42]. In designing systems which incorporate diversity, it is also important to note that preferred diversity levels may not remain static on an individual user basis.

Although most participants described diversity as a difference in genres, genre was not the exclusive answer. To some participants, a recommendation list which spans genre may not be considered diverse unless those genres span a range of cultures, and to other users a recommendation list which spans artists in just one genre may be considered diverse.

The occurrence of inner and outer diversity–that is, diversity within the bounds of existing preference, and outside of those bounds–was an unexpectedly binary result, and

neither of these ideas are well defined by existing beyond-accuracy metrics. Inner diversity is not well described as novelty, nor is outer diversity well described by serendipity. The idea of inner diversity does however align with idea of user genre coverage [45]. We further explore the ideas of inner and outer diversity preference by showing three differently diversified lists to a larger pool of participants in Study 2 (Chapter 4).

In their foundational paper on diversity in *information retrieval*, Clarke et al. use a query for 'jaguar' as an example to show the usefulness of diversity; a diverse response might include the cars, the cats, and the classic Fender guitar [7]. In the case of music recommendations, all diverse responses may be simultaneously correct to one user, and incorrect to another.

3.4.2 RQ2: How might users optimise their own level of diversity in personalized recommendation lists?

The interactive system we implemented (Figure 3.1) represents a first attempt in allowing users to optimise diversity metrics in line with how they are optimised in existing studies. As such, all variables other than the level of diversity (Equation (3.2)) were fixed. We note that in allowing users to remove well-known songs the system represents a specific use for diversity in discovering novel music.

Diversity selections accross the interactive trials varied widely within and between users. Ideally in Figure 3.1, users' Likert ratings would be distributed with positive responses on the right ($0.5 \leq \beta \leq 1$), and negative responses on the left ($0 \leq \beta \leq 0.5$). While results do not follow this distribution, the responses in Figure 3.2 show that users generally found the slider system to enable more diversity. We hypothesise a combination of three reasons for these results. First, the Likert survey provided no frame of reference for diversity and participants used their own idiosyncratic definitions. Second, the users' responses were heavily impacted by music previewed before locking in a diversity value. Third, the diversity metric did not match users' models of diversity.

We also note that while our selection of CF recommender and diversity metric have a basis in previous work, there are countless combinations of them which may be used to comprise of a system such as ours. Recently, neural recommendation approaches have become much more common in the recommender systems literature despite a lack of user evaluations [39, 8].

The design of our follow-up study found in Chapter 4 was informed by users diversity selections in order to control for the three hypothesised sources of variation. We also include

a modern neural recommendation approach to evaluate and compare its performance on real users.

3.4.3 Summary

This study provides a much needed connection between quantitative diversity metrics and user perceptions of diversity in music recommendation lists. We identified two primary themes on user selections for diversity: listener mood, and diversity meaning. Many users expressed a clear distinction between diversity within the bounds of their existing preferences, and diversity outside of these preference. Participants used *genre* most commonly to describe music recommendation diversity, although this was not universal. We explore these ideas further and with more statistically significant sample size through a follow-up study in Chapter 4.

Chapter 4

Study 2 - Expanding Diversity Scope Through Personalized Filtering

4.1 Overview

In Study 2 we aim to use the some of the primary qualitative results gleaned from Study 1 on a larger and more statistically significant sample size. Specifically, we build an online application to have participants evaluate three different conditions of recommendation lists generated by two different collaborative filtering algorithms. The first condition is the algorithm's native output, the second is inspired by the concept of *outer diversity*, and the third is inspired by *inner diversity*, where both of these terms arose from our study in Chapter 3. We evaluate the quality and diversity of the recommendations through statistical analysis of questions on a 5-point Likert scale.

4.2 Methodology

To control all aspects of recommendation and diversity inclusion, and to minimise restricting participants' consumption method, we again trained our own recommendation models. We used Last.fm as a source of raw listening data, and presented song previews in the form of standardised 30 second track previews from Spotify.

Dataset	Interaction Count (User-Track)	User Count	Track Count
Training	127,267,022	50,437 for all	2,817,819 for all
Validation	22,460,349		
Test	26,423,939		
Total	176,151,310		

Table 4.1: Breakdown of implicit data collected from Last.fm and filtered to remove tracks with 10 plays or less.

4.2.1 Recommendation Overview

4.2.1.1 Data

We extended the Last.fm data set used in Study 1 by retroactively topping up each users listening history. For each username we collected up to 10,000 new LEs starting from July 2020 and working back to their latest LE in our existing data set. Users who did not have any new LEs during this period were removed from the data set. Tracks were once again identified using unique artist and track name tuples. The total un-processed data set consists of 520,134,112 unique LEs, and 15,804,356 unique artist-track tuples recorded by 50,440 users over a period of roughly 2 years during 2019 and 2020. More details on the collection method can be found in Section 3.2.1.1.

In order to remove some noise in the data set, we removed tracks which were listened to 10 or less times. This filtering resulted in a drastic reduction in unique artist-track tuples to 2,817,819 (82.2% decrease), while only modestly reducing the number of LEs to 488,528,514 (6% decrease) and the number of users to 50,437 ($< 0.01\%$ decrease). These decreases can be explained by a combination of the 'long tail' as well as the noisy and unstructured nature of Last.fm listening events [4].

In the filtered data set, the median number of LEs per user is 9,857, the 25^{th} percentile is 4,663, and the 75th^{th} percentile is 14,277. The user-track-interaction matrix, used to generate recommendations, contains 176,151,310 non-zero entries (play counts) across 2,817,819 unique tracks, resulting in a $50,437 \times 2,817,819$-sparse matrix. Entries in this matrix are integers which correspond to the number of unique times a user (row) played the track (column). An anonymized version of this data is available upon request.

Data was split into training, validation, and test sets using weak generalization, where user-item interactions are sampled at random from the entire dataset to form the subsets.

This differs from the strong generalization used by Liang *et al.* which samples entire users resulting in each user occuring in only one subset [27]. We chose to use weak generalization because matrix factorization can not efficiently deal with a large number of unseen users. A description of the data splits can be seen in Table 4.1.

4.2.1.2 Algorithms

For generating music recommendations we chose two collaborative filtering recommendation algorithms designed for implicit feedback data sets: Alternating Least Squares matrix factorization (ALS), and Variational Autoencoders for Collaborative filtering (MultVAE) [27, 19]. We provide a brief overview of how these algorithms work in practice, and refer readers to the original papers for detailed descriptions and mathematical processes.

ALS is based on classical matrix factorization, and has been used frequently in recommendation research and production with positive results [16, 19, 14]. The algorithm generates recommendations by factorizing a large sparse matrix of user and item play-counts, then uses these latent factorizations to compute a low-rank matrix approximation. The factorizations themselves consist a vector for each user and item. The product of any user vector and item vector represents the relevance of that item to that user. On initial training, all users-item play-counts are used to generate the latent factorization. After training, vectors for unseen users can be calculated using their listening history and the latent item factorizations. This method is known as the *fold-in* method [35].

To generate a top-n recommendation list for some user, we find the dot product of the user's vector with all track vectors, sort the result for each track in decreasing order, and select the top-n items.

In addition to generating recommendations, the column factorizations can be seen as latent features representing each item; in this case each song. These latent representations are especially useful for public research use as generating content-based features for up-to-date tracks is nearly impossible due access and copyright restrictions.

MultVAE is a modern neural network approach based on a Variational Autoencoder architecture. It is the only neural network approach identified by Dacrema *et al.* to outperform basic top-n recommendation algorithms using various measurements of accuracy on commonly used benchmark data sets [8]. MultVAE takes a dense input vector with length equal to the total number of recommendable items (x). This vector is passed through an encoder (g_ϕ) to a lower dimensional latent representation (z), and then through a decoder (f_θ) which has an inverse architecture to the encoder. The general architecture of MultVAE

is:

$$x \longrightarrow g_\phi \longrightarrow z \longrightarrow f_\theta \longrightarrow x'$$

The authors suggest that g_ϕ and f_θ consist of 0 or 1 densely connected perceptron layers with a dimensionality of 600, and the dimensionality of z to be 200. The dimensionality of x and x' is equal to the number of items in the database. Given the vector of play counts x, x' is a vector of expected play counts which we can sort in decreasing order and select top-n items from.

4.2.1.3 Hyperparameter Optimization and Training

In order to measure the general performance of the models for hyperparameter optimization and baseline comparison we adopt binary Normalized Discounted Cumulative Gain (NDCG) [17]. Discounted Cumulative Gain (DCG) is based on recall and defined as:

$$DCG = \sum_{i=1}^{k} \frac{rel_i}{\log(i+1)}$$

where rel is a binary value representing whether the recommendation at rank i appears in the unseen portion of the users listening history. The denominator then *discounts* the relevance based on how far from rank 1 it appears. NDCG for one user is then defined as:

$$NDCG = \frac{DCG}{DCG'}$$

where DCG' is the ideal DCG: rel_i is always equal to 1. Total NDCG@k is the average value across all users for some defined list length k.

Hyperparameters for ALS matrix factorization are:

- **Factors:** Dimensionality (i.e., rank) of the latent matrix factorizations (per user and item)
- **Regularization (λ):** Scaler term which controls the level of model regularization to prevent over-fitting
- **Iterations:** Number of ALS optimization steps to make
- **Rate of Increase (α):** Scaler value which all non-zero values of sparse data matrix are multiplied by

We optimized these hyperparameters using randomized search over 60 iterations. The best performance on validation data (NDCG@100=0.217) was achieved using 224 factors, $\lambda = 1$, $\alpha = 1$, after 98 iterations.

For MultVAE , we consider only the parameters and ranges referenced by the original authors [27]:

- **Architecture:** Either 0 or 1 fully connected hidden layers in the encoder and decoder with a dimensionality of 600.
- **Dropout:** Probability of ignoring nodes during each pass of training.
- **Annealing Cap:** The maximum value for the β term which is multiplied with the KL-divergence loss to reduce its weight in the two-objective VAE optimization process. The authors specifically suggest that of $0 \leq \beta \leq 1$ [27].
- **Annealing Steps:** The value of β will be increased each optimization step so as to reach the Annealing Cap after this many steps.
- **Learning Rate:** The size of each gradient descent optimization step, typically either .001 or 0.0001, but any value between 0 and 1.
- **Batch Size:** The number of unique users to include in one single optimization step.
- **Epochs:** The number of times all training data is used for training.

Our implementation of MultVAE was based on the original author's Tensorflow 1 implementation, and a PyTorch implementation by James Le[1] . Due to the large number of unique tracks in our data set, full cross-validation of MultVAE architectures was not computationally feasible. We instead trained a number of different models and architectures concurrently. The best performance on validation data (NDCG@100=0.223) was achieved using 0 hidden encoder/decoder layers, annealing cap of 1, 10000 annealing steps, learning rate of 0.001, and batch size of 500 over 250 epochs. We implemented early stopping based on NDCG@100, but it was not triggered. The final dimensionality of the model was:

$$[2,817,819] \longrightarrow [200] \longrightarrow [2,817,819]$$

[1]The original authors code can be found at https://github.com/dawenl/vae_cf. Permission to use James code was obtained through email correspondence. This code can be found at https://github.com/khanhnamle1994/MetaRec

Model	Validation NDCG@100	Test NDCG@100
ALS	0.217	0.325
MultVAE	0.223	0.349

Table 4.2: Final results of both recommendation algorithms. Note that the test results reflect models trained using the combined training and validation data sets.

Both models were retrained on the combined training and validation data, and evaluated on the unseen test data. The final evaluation results can be seen in Table 4.2.

The process of generated recommendations for new users is similar for each algorithm. For ALS, a new latent user vector is generated using their listening history and the existing latent item vectors. We multiply this new user vector with all item vectors to generate item relevance. For MultVAE we feed the user's listening history through the trained network and obtain a new vector containing each item's relevance.

4.2.2 Item Features

Diversity was calculated similarly to Section 3.2.1.2 using the latent item features generated from ALS matrix factorization. Before calculating diversity, we explored the effect of various simple normalization and standardization techniques on the data. To view their effects we performed PCA on the transformed data to reduce its dimensionality and plot the data. We found that item feature vector magnitude was highly related to popularity, which was calculated as the number of times a track was listened to in the entire dataset. In order to lessen the effect of popularity on the latent features, we chose to normalize the latent features. Each track's feature vector was l2-normalized to unit-length.

4.2.3 Music Recommendation Lists

We used three different techniques to generate top-10 music recommendation lists for each recommendation algorithm, resulting in a total of 6 different top-10 recommendation lists per user. Recommendation lists generated using ALS are prefixed with *als*, and recommendation lists generated using MultVAE are prefixed with *vae*.

4.2.3.1 Control (*als*, *vae*)

Control recommendation lists consist of the raw ranked output from each recommendation algorithm after removing tracks which appeared in the user's listening history. This is the method most commonly used to evaluate new and novel recommendation approaches, and reflects the metrics reported in Table 4.2.

4.2.3.2 Maximally Diverse (*als_max_div*, *vae_max_div*)

We generated these recommendation lists using the greedy ILD diversification method described by Ziegler *et al.* using $\beta = 1$ [49]. Our method differs slightly in that the first item selected is the most diverse rather than the most relevant. This greedy diversification algorithm starts with the maximally diverse track from some larger recommendation list. In order to provide a relatively diverse pool of tracks we chose to start with the top-1000 recommendations from each algorithm. The algorithm proceeds to add the track which is maximally distant from the already selected tracks until the list is of the desired length. This method ensures that the tracks are not only maximally distant from all other tracks, but that the final recommendation list traverses multiple extremes in the item feature space. The beta value controls a convex combination between track relevance (algorithm output), and track diversity at each point of selection. In trading relevance for diversity, the beta value also controls a trade off of model accuracy (i.e., NDCG@k) and list diversity. Using $\beta = 1$ means that we do not consider the relevance ranking within the top-1000 recommendations when generating diverse recommendation lists.

4.2.3.3 Filtered Diverse (*als_filt_div*, *vae_filt_div*)

The filtered diverse lists follow the same greedy diversification process as the maximally diverse lists, but the top-1000 recommendations are first put through a filter based on the user's existing listening history. This list serves as a first attempt to better align recommendations with user preference for *inner diversity* identified in Chapter 3. We considered two methods for filtering recommendations to be within users' existing preference: feature clustering, and genre filtering.

Feature clustering was inspired by the idea that we may be able to remove tracks which are too distant in the track feature space from a user's existing listening events. We theorised that if a user's listening history could be clustered into n groups, then we could select a threshold around each groups centroid and filter recommendations which fall outside. Due to poor initial clustering results, we chose to move on to genre filtering.

Genre filtering involves retrieving a list of genres representing individual listening events and recommendations, and removing recommendations in genres which do not appear in the user's existing listening history.

Both Spotify and LastFM provide a form of genre tags for music. LastFM tracks each contain tags which are crowd-sourced from users, whereas most Spotify artists contain a set of genre tags. We used Spotify artist genre tags, because LastFM tags often relate to irrelevant qualities (i.e., the 'seen live' tag). We defined a track's genres as the genres of that track's artist retrieved from Spotify.

For each user, we first created a hash table of all genres which appear in the user's listening history and their frequencies. Next, we find the most diverse track among the top-1000 recommendations (the first track in the Maximally Diverse list) and its genres. The user's genre hash table is searched for this track's genres, and we save the lowest frequency found (or 0 if none) to be the user's genre threshold. For all other top-1000 recommendations, we complete the same search and remove from the list any tracks with a single genre either not in the user's hash table, or with a frequency below the found threshold.

Once the top-1000 recommendations have been filtered, the greedy diversification method is run as for the Maximally Diverse list. The threshold ensures that the Filtered Diverse list is seeded with a different maximally diverse track, and therefore differs from the Maximally Diverse list.

4.2.4 User Study

Our interactive user study consisted of a pre-interaction survey on personal music consumption, discovery, and preference, followed by 6 personalized top-10 music recommendation lists as described in Section 4.2.3. The recommendation lists included a 5-point Likert evaluation for each track, as well as questions on the recommendation list as a whole using the same 5-point Likert scale. The music recommendations were displayed as 30 second song previews obtained using Spotify Play Button widgets.[2] The UI for music recommendation lists can be seen in Figure 4.1. The study was hosted as an online web-app which collected participant LastFM data and generated recommendations in the background while participants completed the surveys.

[2] https://developer.spotify.com/documentation/widgets/generate/play-button/

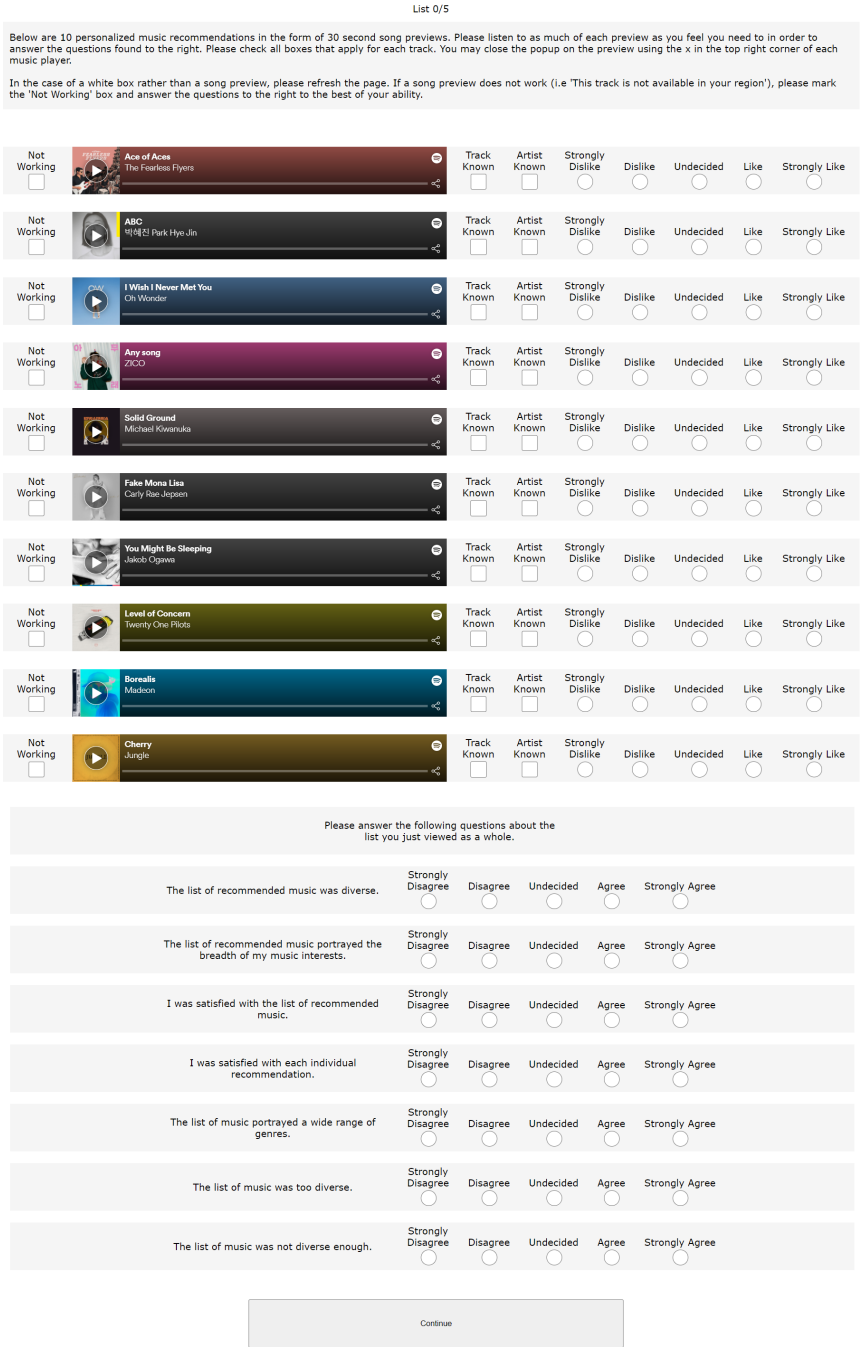


Figure 4.1: The UI for music preview ratings. Each song is displayed as a 30 second preview from Spotify

4.2.4.1 Participant Groups

We split our study into two seperate streams which we call the MTurk and non-MTurk groups. We made the decision to split the study for two reasons. First, we wanted to complete interviews with some participants to help us interpret quantitative results, and such interviews are not feasible using crowdsourcing platforms. Second, we wanted to pilot the study with a small subset of participants to ensure usability and interpretability.

LastFM usernames were never stored so as to maintain participants' anonymity, and participants from both groups were assigned random participant ID numbers. Although the MTurk and non-Mturk groups interacted with the same system, we kept each data set isolated.

In addition to the recruitment groups, all participants were assigned to one of six order groups from A to F. These groups were created using a balanced latin square to account for potential effects of the order participants were shown recommendation types.

4.2.4.2 Non-MTurk

The non-MTurk participants who were recruited through email lists and social media. After obtaining informed consent, we asked these participants for their LastFM username. If participants did not have an existing LastFM account, we provided instructions on how to set one up and begin recording their music listening events to it (known in LastFM as scrobbling). We required that non-Mturk participants listen to at least 5-hours of music per week, and had at least one weeks worth of listening history recorded by the time their recommendations were generated. Once they had met the LE requirements, this set of participants scheduled a virtual face-to-face session with the researchers. The sessions consisted of a video call with one of the researchers during which they would complete the interactive study using a web browser on their own computer. Video and audio were muted while the participants completed the survey, but they were invited to ask questions and clarifications at any time. After they completed reviewing their last recommendation list, we resumed the video call and asked them a series of semi-structured interview questions on their interpretation of, and preference for, diversity in music recommendations. Non-Mturk participants were compensated $10CAD for their participation, and sessions lasted between 40-70 minutes.

There were no major changes made to the application or generation methods between Non-MTurk and MTurk participation.

4.2.4.3 MTurk

The MTurk group were recruited through Amazon Mechanical Turk. The Mechanical Turk terms of service prohibit asking workers (participants) to register for a service, or log into an existing service. We therefore required workers to have a LastFM account in order to participate, and specified such in the HIT description, the HIT layout, and as a question on the consent form. After obtaining informed consent, we had workers provide their LastFM username which we used to obtain their public listening history. We also required that the LastFM account had at least 50 LEs recorded in the last 6 months. To verify ownership without requiring a login, workers were given 3 attempts to name one artist they had listened to in the previous 6 months. The MTurk group completed the same interactive study as the non-Mturk group, but without the interview or live video call. MTurk participants were compensated $4USD for their participation.

4.3 Results

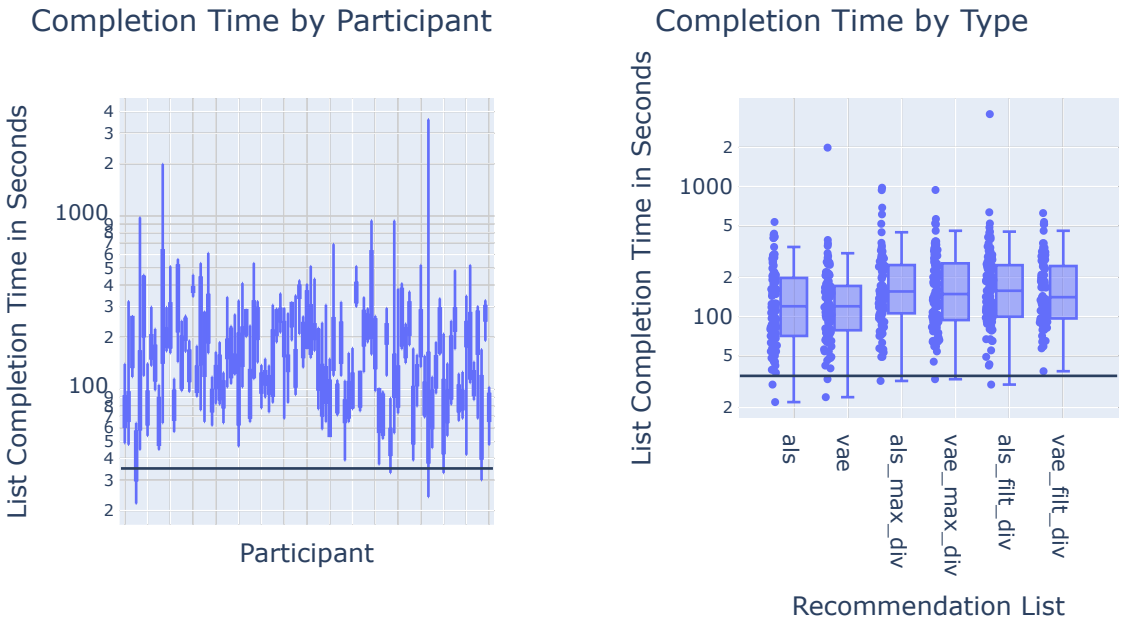
4.3.1 Data and Demographics

We recruited 9 non-MTurk participants, and 97 MTurk participants. Only MTurk participant data was used for analysis, as the non-MTurk participants are not samples from the same population.

In order to remove low-quality responses, participants who completed any single recommendation list survey (as shown in fig. 4.1) in less than 35 seconds was removed. The distribution of completion times per participant before removing low quality responses can be seen in Figure 4.2a, and completion times per list can be seen in Figure 4.2b. In total, 5 participants were removed for completing lists too quickly, resulting in a final participant count of 92. After removing these participants the median completion times for *vae* and *als* were still lower than the diversified lists. The proceeding results include only these 92 participants.

The median participant age was 29; the youngest was 19 and the oldest 62. When asked about their gender, 50 identified as male (54%), 40 identified as female (43%), and 2 identified as non-binary (2%).

The use of collaborative filtering algorithms for recommendation means that only tracks in our base training data set can be used to generate recommendations and be recommended. The median count of LEs per participant was 857 before removing tracks not in



(a) (b)

Figure 4.2: Time taken to complete recommendation lists per user (a) and per list type (b). Users whose completion time was below 35s (black line) were removed from analysis.

the base data set, and 627 after. This is compared to a median value of 3110 for users in the base data set (fig. 4.3). The distribution of the base data set is skewed by the two collection periods over 2 years each capped at 10,000 events per user, while the participant data only covers a listening period of up to 6 months with no cap.

4.3.2 Pre-Interaction Survey

In addition to questions on demographics, the pre-interaction survey asked questions focused on music consumption, discovery, and music recommendation preferences.

Results to open-answer questions on what platforms participants use to discover, and consume music were broken down into individual platforms/methods, counted, and their relative percentages are shown in Figure 4.4. Spotify, YouTube, or LastFM was included by more than 60% of participants as a method for both discovery and consumption. Par-

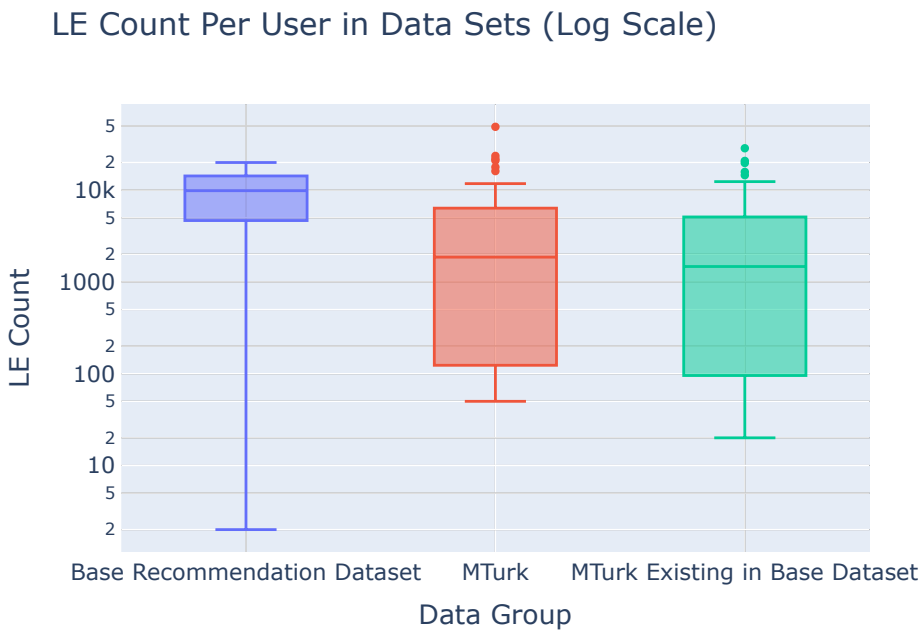


Figure 4.3: Distribution of LEs per user in our base recommendation data set (50,437 users), compared to MTurk participants (92 users) before and after removing tracks not found in our base data set.

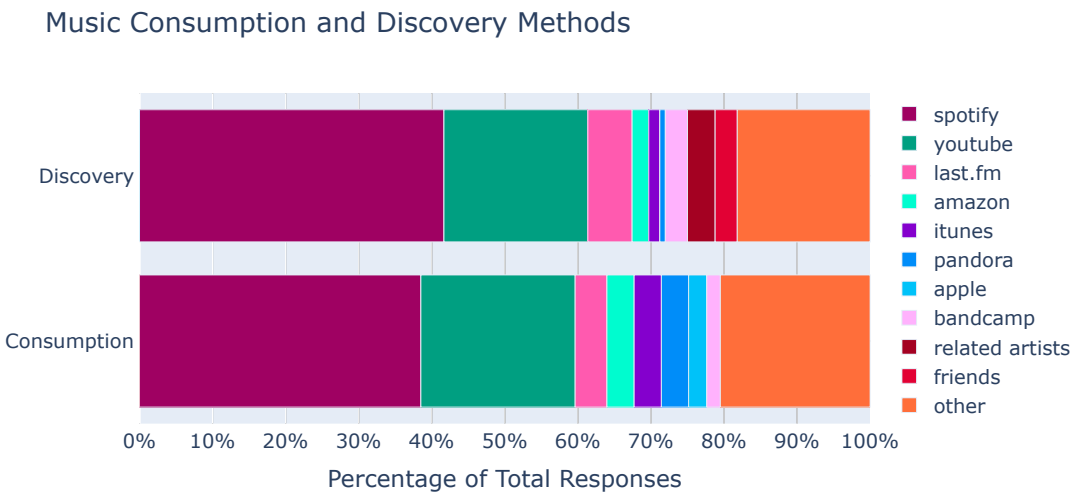


Figure 4.4: Processed and counted responses to the pre-interaction open-response questions: "What platform(s) do you primarily use to find new music?", and "What platform(s) do you primarily use to listen to music?"

ticipants generally used the same platforms for music discovery and consumption.

Responses to Likert questions on music diversity preference are seen in Figure 4.5. A large majority of participants agreed that their preference depended on who makes the recommendations, the quality of the recommendations, what they are doing, and their mood, while also indicating their preference was static. Almost 50% of participants disagreed or strongly disagreed that their location was important to how they felt at a given time about music diversity.

Questions which focused on music recommendation preferences used a binary yes/no scale with an additional undecided option to remove noisy responses. We asked these questions to examine correlations between participant responses and participant satisfaction with each recommendation list type. They were labeled CQ1 through CQ8. The questions and responses are shown in Figure 4.6. Similar to the Likert questions on music discovery above, the vast majority of participants selected "yes" for all CQ, with the notable exception of CQ7, for which 32% indicated they were unsure, and 15% responded "no". Correlations between CQ7 and list satisfaction are explored in Section 4.3.3.3.

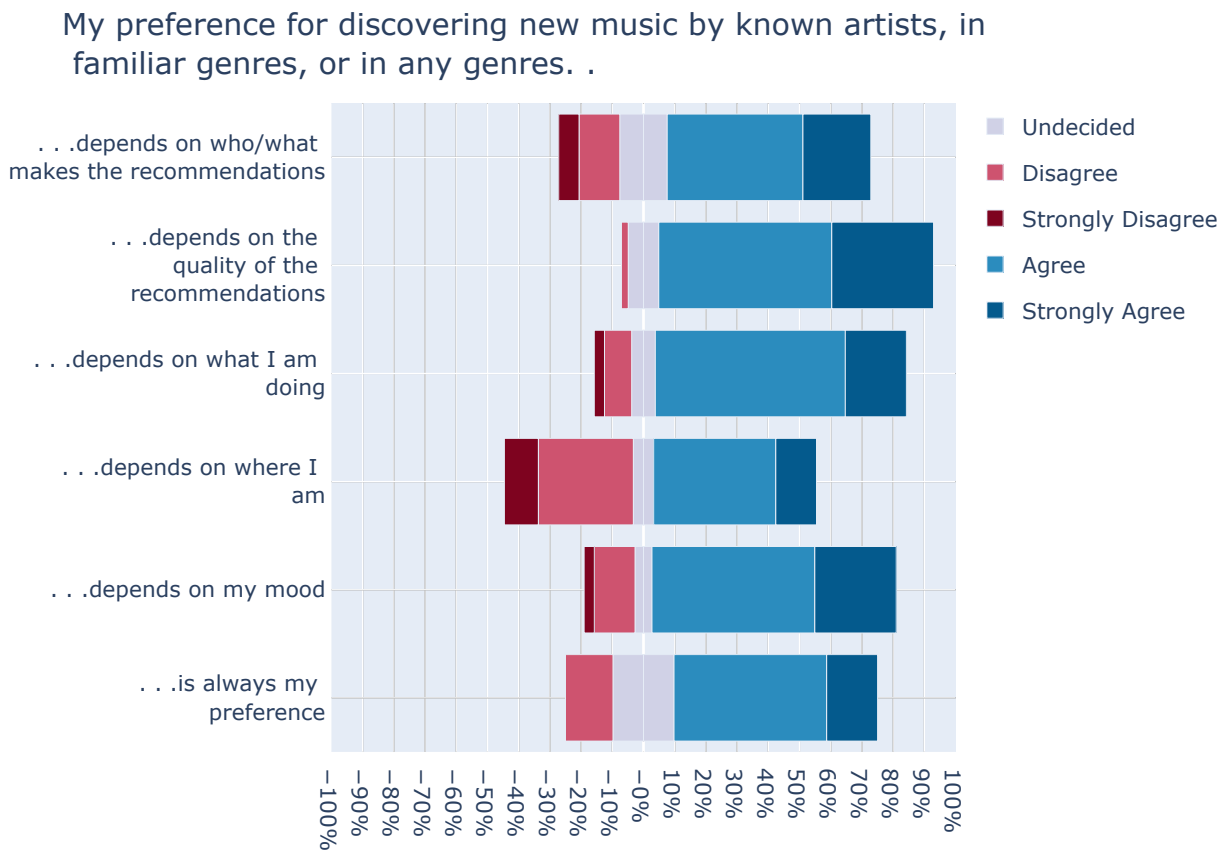


Figure 4.5: Likert responses to pre-interaction survey questions on music diversity preference.

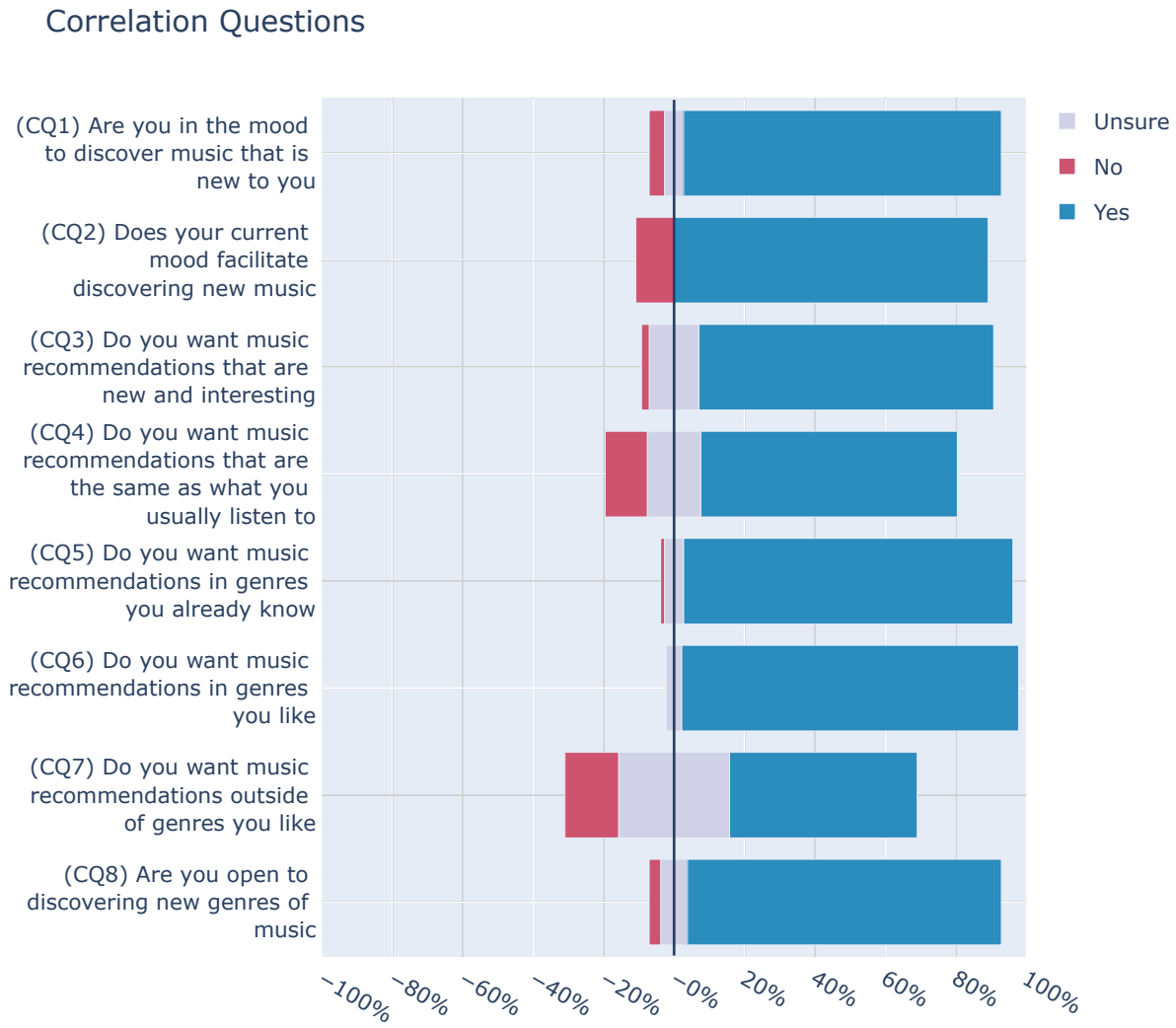


Figure 4.6: Responses to pre-interaction survey questions with binary responses. These questions were asked in order to find correlations between user selections and recommendation satisfaction.

4.3.3 Recommendation List Evaluation

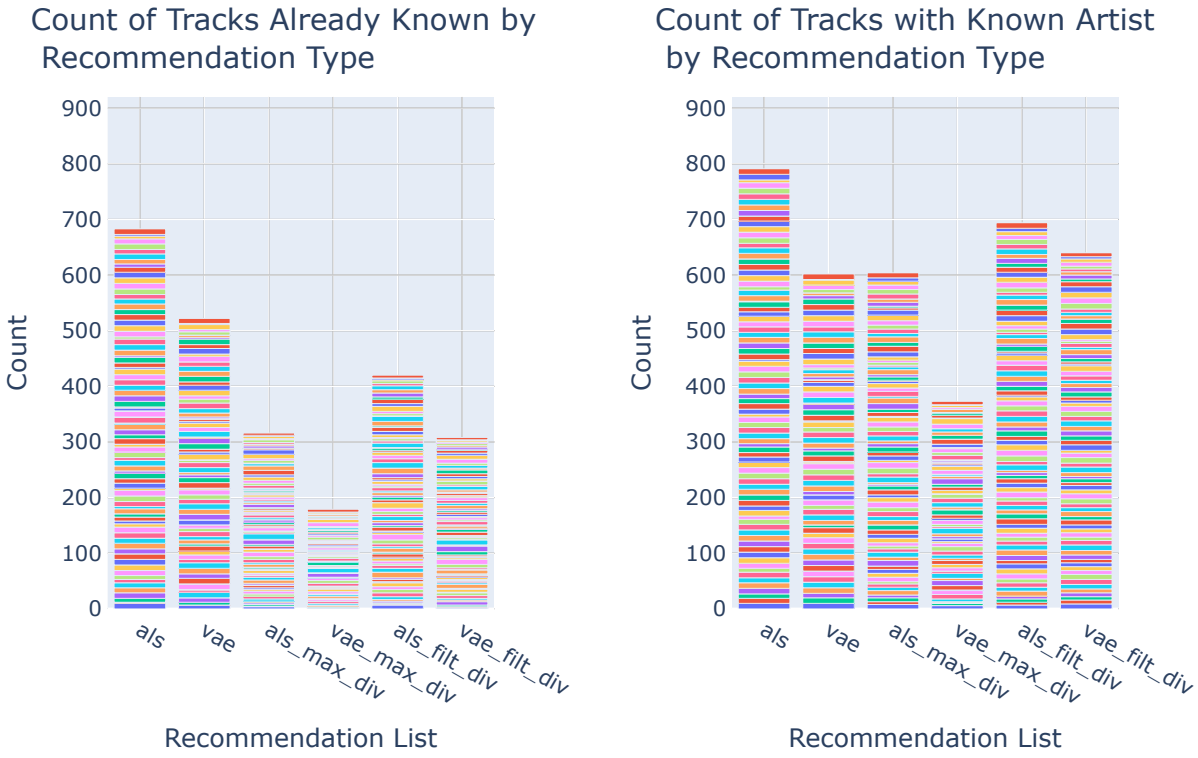
Alongside Likert questions on individual recommendations and on the recommendation lists as a whole, each song preview contained checkboxes to indicate a track error, to mark the artist as known, and to mark the track as known. Less than 1% of track previews were reported as having an error. Counts for known tracks and artists are shown in Figure 4.7a, and Figure 4.7b respectively. The control lists contained the highest counts of known tracks, with the maximally diverse lists containing the fewest, and the filtered lists containing slightly more than the maximally diverse. In all cases, comparable *als* lists contained more known tracks than *vae* lists. This pattern of *als* lists containing more known entities than *vae* lists also holds true for artists. The *vae_max_div* list also shows significantly fewer known artists than any of the other lists.

As with the pre-interview correlation questions, we assigned labels to the recommendation list evaluation questions based on the order in which they were presented to participants (fig. 4.1). The questions were:

- **(LQ0)** Individual Likert responses for each track in the 10-track recommendation list.[3]
- **(LQ1)** The list of recommended music was diverse.
- **(LQ2)** The list of recommended music portrayed the breadth of my music interests.
- **(LQ3)** I was satisfied with the list of recommended music.
- **(LQ4)** I was satisfied with each individual recommendation.
- **(LQ5)** The list of music portrayed a wide range of genres.
- **(LQ6)** The list of music was too diverse.
- **(LQ7)** The list of music was not diverse enough.

We preformed a series of Kruskal-Wallis tests on the Likert distributions of order groups A to F for all LQ, and found a statistically significant difference in distributions for LQ0 ($H = 18.761, p = 0.002$) and LQ4 ($H = 18.089, p = 0.003$). This strongly suggests that the order recommendations were presented had an impact on track satisfaction responses, though post hoc tests did not show any clear pattern in these variances. The remaining analysis is completed on the combined order groups to control for this variance.

[3]Due to tracks removed for errors and some duplicate track ID's returned by the Spotify API not all lists contain 10 valid ratings.



(a) (b)

Figure 4.7: Each bar shows the total count of tracks which participants identified as known in some form out of ≈ 920 total recommendations served. Colours show counts per-participant.

4.3.3.1 List Comparisons

In order to better compare results, we split the LQ into two different groups for evaluation: satisfaction, and diversity. We preformed a Friedman test on the distributions of responses between each list for all LQ and found that at least one list type's distribution differed significantly for each LQ ($p < 0.001$ for all)[4]. A post-hoc Nemenyi test is performed to identify which list's distributions differ from each other. The raw results of the Nemenyi test can be seen in Table 4.3, and they are also visualised in Figures 4.9, 4.10 and 4.11, which also show the raw frequencies of responses to the questions.

There are at least four results of note in the responses to rating questions (LQ0) and satisfaction questions (LQ3, LQ4) found in Figure 4.9. Statistical significance is found more readily among LQ0 because there are 10 samples per list, and the use of Dunn's tests instead of Nemenyi tests. First, the control *als* recommendations were consistently rated more positively than the *vae* recommendations (LQ0: $p \leq 0.001$, LQ4: $p \leq 0.001$). Second, *als_max_div* recommendations were consistently rated higher than *vae_max_div* (LQ0: $p \leq 0.001$, LQ3: $p = 0.007$, LQ4: $p \leq 0.001$). Third, *als_filt_div* and *vae_filt_div* were rated similarly or better than their un-diversified counterparts in list satisfaction (LQ3, LQ4,) despite receiving less positive individual track ratings (LQ0). Fourth, the filtered lists also performed similar to or better than their maximally diverse counterparts in all cases.

We also examined the distributions of responses to LQ0 and LQ3 by list type using a Kruskal Wallis test, and found significant differences among the control lists ($p \leq 0.001$), which highlights a clear distinction between track ratings and overall list satisfaction, especially for control lists.

The diversity results shown in Figures 4.10 and 4.11 display at least two clear results of note. First, all diversified lists were recognised to be significantly more diverse, and to portray a wider range of genres, than their non-diversifed controls ($LQ1, LQ5, LQ7 : p \leq 0.001$). Second, no significant differences in perceived diversity or genre range were found between filtered diverse and maximally diverse lists. Participants did not find any list to be overly diverse, though they did feel more strongly that the control lists were not overly diverse as compared to most diversified lists ($LQ6 : p \leq 0.05$). A crucial finding is that the *als_filt_div* and *vae_filt_div* lists were most consistently found to best portray the breadth of participants' music interests (LQ2).

An additional Kruskal-Wallis test was performed on the distributions of responses to

[4]We preform Kruskal-Wallis and Dunn's (with Bonferroni adjustment) tests for LQ0 instead of Friedman and Nemenyi due to the unbalanced data.

LQ1 and LQ5 for all list types. This test also showed no statistically significant difference in distribution for diversified list types, supporting the idea that one way users perceive diversity is genre range.

4.3.3.2 LE Quantity

To check for effects of user LE count on recommendation satisfaction, we arbitrarily sorted participants into low ($LEcount \leq 100, N = 22$), medium ($100 < LEcount \leq 1000, N = 14$), and high ($LEcount > 1000, N = 55$) groups. We completed another Kruskal Wallis test on the distribution of LQ3 and found statistically significant differences in *als* ($p \leq 0.001$), *vae* ($p = 0.028$), *vae_max_div* ($p \leq 0.001$), and *vae_filt_div* ($p \leq 0.001$). We chose not to complete a post hoc test due to the small between-subject sample size and unbalanced group sizes. Instead, we show the raw results from significant likert responses in Figure 4.8. All vae list types showed a significant difference in list satisfaction by LE count, however the *vae_filt_div* list had much more positive responses even with low LE counts. The als results show an interesting inverse case, where the highest satisfaction was found in the low LE group.

4.3.3.3 Correlation Questions

To analyse correlations between the questions seen in Figure 4.6 and responses to overall list satisfaction seen in LQ3 of Figure 4.9, we first removed neutral responses from both and then converted the Likert responses to binary based on sentiment resulting in two distributions of binary responses for each list for every CQ. A pearson correlation analysis showed only two statistically significant results. CQ3 correlated positively with *als_max_div* ($r = 0.330, p = 0.002$), and CQ7 correlated negatively with *vae* ($r = -0.317, p = 0.005$). Without stronger correlations across more recommendation lists, and less skewed question responses, these results hold little meaning.

4.3.3.4 Summary of Statistically Significant Results

Through analysis of participant responses to Likert questions we found statistically significant differences among recommendation algorithms and list generation approaches. In general, recommendations from the ALS model were more satisfying than those from the VAE model, and filtered lists performed similar to or better than control, and maximally diversified lists from the same model. List satisfaction and individual track ratings also differed significantly.

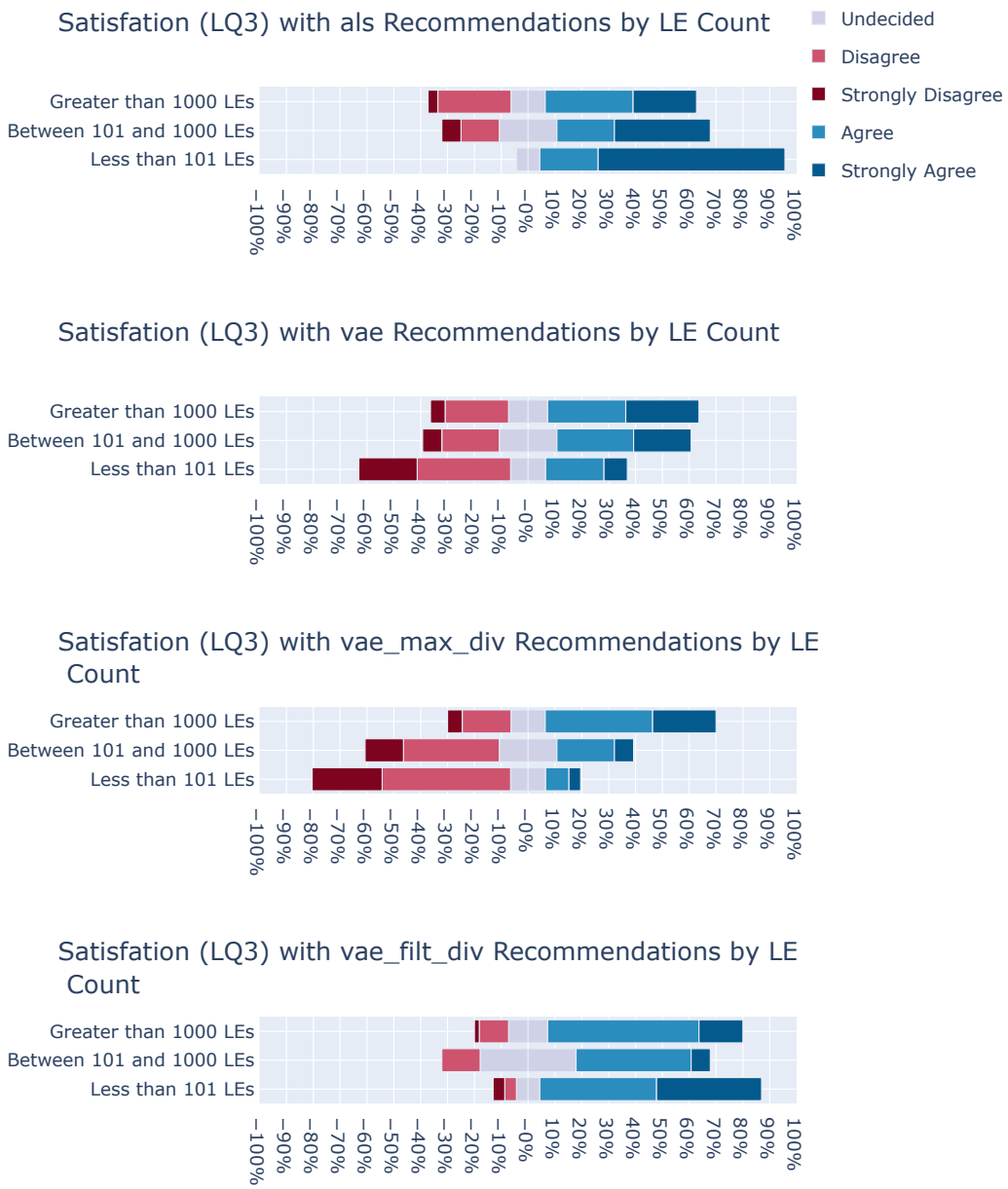


Figure 4.8: Likert responses to LQ3 between participants with different LE counts. Only the list types displayed showed any statistical significance in a prior Kruskal-Wallis test. We did not complete post hoc tests due to a small between-subject sample size and unbalanced groups.

		Recommendation Lists				
Question	**Rec. List**	**als**	**vae**	**als_max_div**	**vae_max_div**	**als_filt_div**
LQ0	**vae**	0.001				
	als_max_div	0.001	0.001			
	vae_max_div	0.001	0.432	1.000		
	als_filt_div	0.001	0.001	1.000	0.020	
	vae_filt_div	0.001	0.001	0.001	0.001	0.001
LQ1	**vae**	0.244				
	als_max_div	0.001	0.001			
	vae_max_div	0.001	0.001	0.900		
	als_filt_div	0.001	0.001	0.900	0.900	
	vae_filt_div	0.001	0.001	0.900	0.872	0.900
LQ2	**vae**	0.900				
	als_max_div	0.041	0.002			
	vae_max_div	0.900	0.900	0.020		
	als_filt_div	0.124	0.008	0.900	0.068	
	vae_filt_div	0.024	0.001	0.900	0.011	0.900
LQ3	**vae**	0.262				
	als_max_div	0.900	0.058			
	vae_max_div	0.055	0.900	0.007		
	als_filt_div	0.900	0.235	0.900	0.046	
	vae_filt_div	0.826	0.011	0.900	0.001	0.860
LQ4	**vae**	0.005				
	als_max_div	0.208	0.747			
	vae_max_div	0.001	0.092	0.001		
	als_filt_div	0.118	0.894	0.900	0.003	
	vae_filt_div	0.900	0.020	0.447	0.001	0.293
LQ5	**vae**	0.9				
	als_max_div	0.001	0.001			
	vae_max_div	0.001	0.001	0.900		
	als_filt_div	0.001	0.001	0.900	0.900	
	vae_filt_div	0.001	0.001	0.900	0.900	0.900
LQ6	**vae**	0.900				
	als_max_div	0.012	0.129			
	vae_max_div	0.002	0.023	0.900		
	als_filt_div	0.002	0.031	0.900	0.900	
	vae_filt_div	0.004	0.044	0.900	0.900	0.900
LQ7	**vae**	0.900				
	als_max_div	0.001	0.001			
	vae_max_div	0.001	0.001	0.900		
	als_filt_div	0.001	0.001	0.900	0.900	
	vae_filt_div	0.001	0.001	0.900	0.900	0.900

Table 4.3: P-values from applying Dunn's (LQ0), and Nemenyi (LQ1-LQ7) tests to analyse Likert response distributions. Significant differences ($p \leq 0.001$) were first identified using Kruskal-Wallis (LQ0), and Friedman (LQ1-LQ7) tests. Note that p-values close to 0.05 (i.e., 0.046) are not necessarily relevant due to the large number of entries.

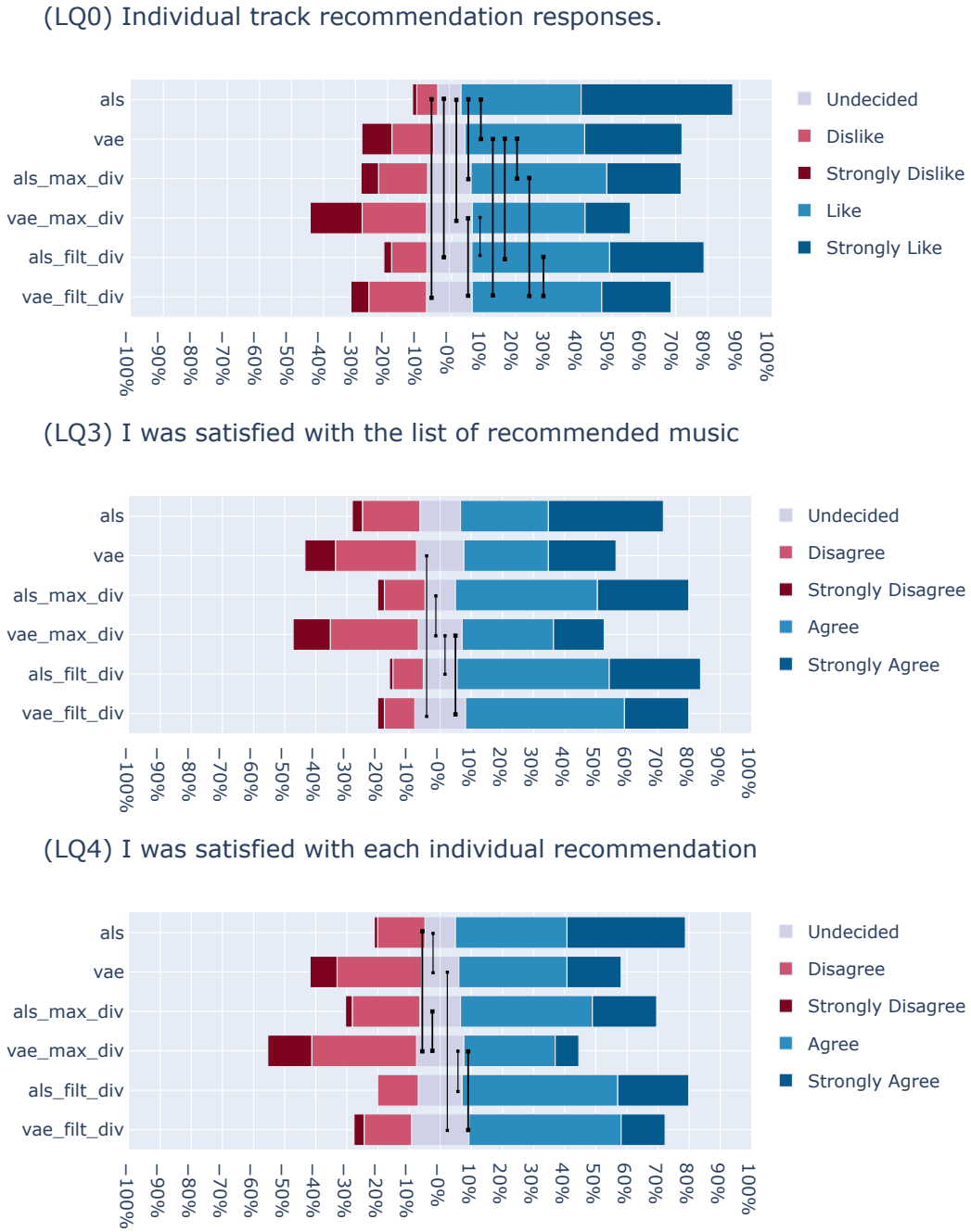


Figure 4.9: Likert responses to recommendation list questions pertaining to satisfaction and recommendation quality. Black bars signify statistically different response distributions (thin: $p <= 0.05$, bold: $p \leq 0.001$).

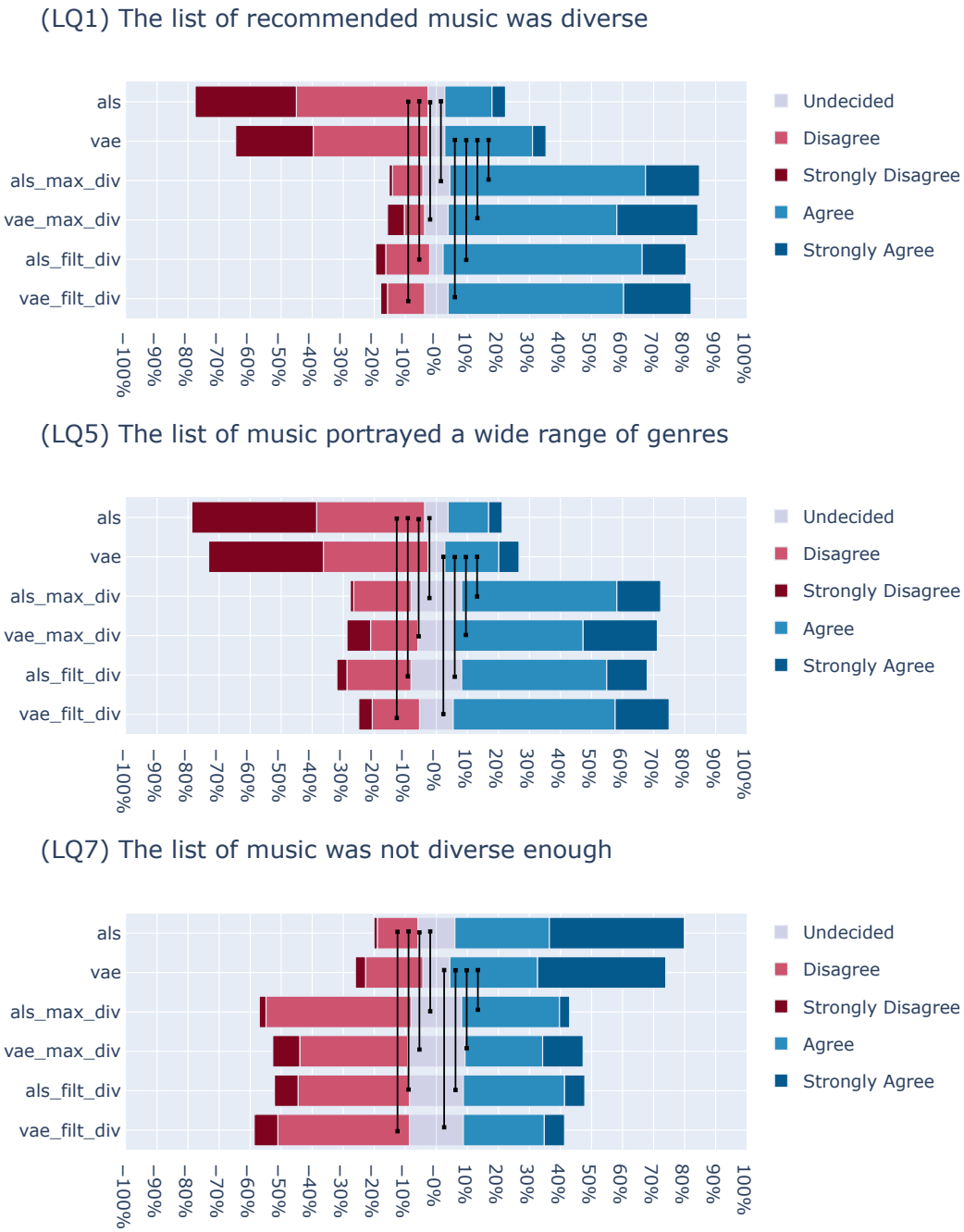


Figure 4.10: Likert responses to questions on recommendation list diversity. Black bars signify statistically different response distributions ($p \leq 0.001$).

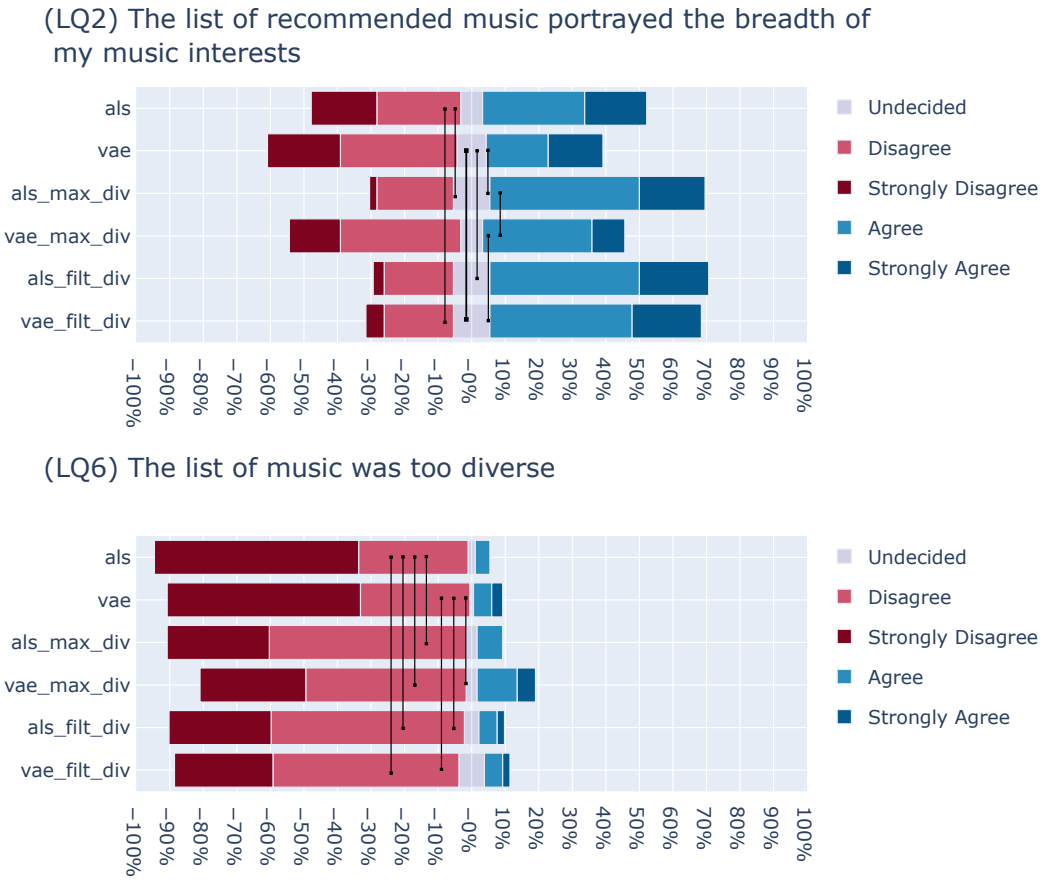


Figure 4.11: Likert responses to additional questions about recommendation list diversity. Black bars signify statistically different response distributions (thin: $p <= 0.05$, bold: $p \leq 0.001$).

In analysing diversity responses we found that all diverse recommendation lists were recognised as such, and filtered lists were found to be just as diverse as maximally diversified ones. Filtered lists also most consistently conveyed the breadth of participants interests. Additionally, participant responses on genre range mirrored their evaluations of diversity, and no list types were found to be too diverse.

In the next section, we explore what our results suggest about how to build good music recommender systems.

4.4 Discussion

4.4.1 Satisfaction

In recommender systems research, the quality of a recommendation list is often inferred from some accuracy measures computed on known data sets [17]. The results presented here show that accuracy certainly does not tell the whole story. Despite the modern neural network-based recommendation method (MultVAE) outperforming the more classical Matrix Factorization approach (ALS) on the base data set using NDCG@100 (Table 4.2), participant responses on individual recommendations showed markedly higher satisfaction from the ALS model with an ostensibly lower test accuracy. This gap in satisfaction only grows larger for the maximally diverse recommendations generated from top-1000 lists.

In their 2019 ACM RecSys Conference best paper, Dacrema *et al.* highlight the fact that the majority of modern neural recommendation methods are either not reproducible (with reasonable effort) or are outperformed on accuracy measures by simple heuristics [8]. We posit that even if a novel recommendation algorithm outperforms existing algorithms in test accuracy, this has little implication on the quality of its recommendations to real users. Further complicating recommendation evaluation is the statistically significant difference in how participants rated tracks vs. their satisfaction with the list as a whole. We are unaware of any accuracy metric which can measure overall recommendation list satisfaction, as such a concept depends on what qualities each user considers most important. This is not only difficult to measure, but it is a moving target affected by mood and is probably different for different recommendation tasks (among other factors). Finding a good playlist is different than picking out a movie to watch.

4.4.2 Diversity

In Chapter 3 we identified that mood and context were important factors in determining how much diversity a user wants, and this is further supported by the results in Figure 4.5 which show that users identify mood and context (among other factors) as important. The difference in responses between 'what I am doing' vs. 'where I am' emphasise the difference between context and location. One can imagine extrapolating that a user is working out if they are detected at a gym, but the location alone is not as significant as the action.

Previous work on optimising diversity levels in recommendation lists has also depended heavily on accuracy measurements [22, 20, 17]. Our results suggest that the difference between control and diversified lists is not well portrayed in individual recommendation ratings. Although maximally diverse lists did result in lower individual track ratings (LQ0), there was no detectable impact on overall list satisfaction (LQ3). This is despite strong statistical evidence that both filtered and maximally diversified lists were identified as significantly more diverse. It is especially important to keep in mind that the maximally diverse lists were created using a beta value of 1 from all top-1000 participant recommendations. This means that either the additional diversity of the lists made up for a decrease in the quality of each recommendation, or that the top-1000 recommendations are all of a relatively high quality. It is likely a combination of both factors.

4.4.3 Genre and Filtering

The nearly identical responses to questions about list diversity and the range of genres further solidify the relationship between the two concepts as expressed by users in Chapter 3 and existing research [45]. When users are asked to evaluate the diversity of a music recommendation list, genre is clearly one of the primary factors they consider. Although genre can be a difficult idea to study quantitatively, its importance from the human perspective of diversity should not be ignored.

Overall, the filtered recommendation lists performed as well or better than the maximally diversified lists for satisfaction while also portraying similar levels of diversity. When maximally diverse recommendations were good (as for *als_max_div*) the filtering had no statistically significant impact on list satisfaction or diversity. Alternatively, when maximally diverse recommendations were poor (as for *vae_max_div*) the filtering had a sizeable positive impact on satisfaction without impacting perceived diversity.

The filtered and maximally diverse lists can be viewed as simple implementations of a system for inner and outer diversity. Unfortunately, when asked about their preference

for diversity nearly all participants responded positively to all questions, and therefore no significant correlations were identified. Future studies may benefit from asking contrasting questions rather than yes/no questions to better model the finite top-n recommendation space.

4.4.4 Diversity and Personalization

Diversity in recommender systems is a heavily researched domain, and numerous different methods of defining and quantifying diversity have been proposed [22, 20, 17]. The ILD method we chose is arguably the simplest. Despite its simplicity, our results add to the existing evidence that increasing ILD is perceived by users as increasing diversity; this time in the domain of music [2, 14]. In fact, the significant negative impact of this diversification method was only observed in the MultVAE recommendations, and was removed through the process of genre filtering. Whereas a more complex genre diversification approach such as that proposed by Vargas *et al.* loses the ability to effectively differentiate between feature-distant items with the same genre tags [45], considering genre (or a content-based equivalent) as a pre-filtering step maintains this granularity.

We extend this one step further by noting that the filtered recommendations were generated with a beta value of 1. We have demonstrated that filtering can result in positive satisfaction even with maximal ILD, suggesting that any and all values of beta will present viable recommendation lists for each user. This may simplify the task of selecting an optimal level of diversification based on mood and context.

4.4.5 Pushing Diversity Further

Despite our attempts to do so, we were unable to generate recommendation lists which reached outside the bounds of our participant's diversity preferences. The fact that maximally diverse recommendations were seen not seen as too diverse signals that it is very hard to generate overly diverse recommendations using either model. In an ideal collaborative filtering system, diversity preference would be implicitly considered; a user's top-n recommendations would mirror those of users with a similarly diverse listening history. Aside from problems with popularity bias [5], this does not account for the idea of *outer diversity* where some subset of users have indicated that they would like recommendations which differ from their existing listening preference (Section 3.4). This problem is exacerbated further by the fact that hyper-parameter optimization of recommendation models make use of accuracy measurements such as NDCG@k (Section 4.2.1.3) which incentivize

only recommendations in training users' hidden listening history; most existing recommender systems, because they so strongly focus on accuracy, are unlikely to make risky recommendations.

In order to generate **truly** diverse music recommendations that match user preference, we first need to understand the extents of their preferences for diversity. The idea of recommending surprising items is typically associated with the related beyond-accuracy metric of serendipity [3], which broadly aims to measure the unexpected or surprising nature of recommendations. It is easy to equate user preference for *outer diversity* to a preference for serendipity, but this does not explain why even maximally diverse recommendations were not found to be too diverse. We feel it is important to try and reach the point of over diversification within the context of collaborative filtering before exploring the impact of serendipitous recommendations. We hope that by extending beyond the top-1000 most relevant recommendations we can find more diverse recommendations, and once again compare them against recommendations filtered for existing user preference.

If new collaborative filtering algorithms can not generate adequate levels of diversity at all, then are they really working towards generating better music recommendations for users?

4.4.6 Summary

The results we presented in this study build on those identified in our prior study (Chapter 3) and extend even further to highlight the large disconnect between offline and online accuracy and diversity evaluations of music recommender systems. Through a large within-subjects study, we evaluated two collaborative filtering algorithms and found that the offline accuracy–and even the user provided track ratings–were not good indicators of overall list satisfaction. In fact, the modern neural network-based approach which achieved higher testing accuracy performed more poorly when judged by real people.

Diversity continues to be an important topic of discussion in recommender systems. Our genre filter-based diversification approach was successful at enabling satisfying and diverse recommendations within users existing preferences despite using arguably the simplest definition of diversity. We found success in modeling diversity based on user ideas of the term, and then asking them to evaluate it. In doing so, we brought to light the limited diversity contained within collaborative filtering recommendation algorithms.

Chapter 5

Large-scale Experimental Design for Music Recommender Systems: Challenges and Possible Solutions

5.1 Running a Live Recommendation Study

The results and discussion from Chapter 4 add to the existing evidence that there are clearly benefits to evaluating music recommender systems with real users [21]. A key finding is that, accuracy evaluations are not necessarily good indicators of individual track ratings, and overall list satisfaction is not a function of individual track ratings alone. A decade and a half before this manuscript, McNee *et al.* informally argued that there must be more emphasis put on user-centric recommender system evaluation [29], yet just one year ago Dacrema *et al.* highlighted a disturbing lack of attention to evaluation even in strictly offline analyses [8].

User studies and online analysis require significantly more resources and time than strictly offline analysis. In the hope of assisting researchers completing live evaluations of their methods, we present some of the struggles faced in developing the music recommendation applications used in Chapters 3 and 4, and their resolution. For further reading on live evaluation of recommender systems we refer readers to the relevant chapters of the Recommender Systems Handbook [17, 21].

5.1.1 General Architecture

The goals of our system were twofold: to generate current recommendations for previously unknown participants using the same models described in research, and to generate these recommendations on demand. The final system consisted of an online application which, after obtaining consent, collected participants' listening histories from the LastFM API, fed their data through each recommendation algorithm to obtain top-n lists, obtained music previews and metadata from the Spotify API, and displayed previews alongside questionnaires. We implemented the system using a Flask backend which served static HTML and Javascript[1]. The study was hosted on a single AWS EC2 server instance using Elastic Beanstalk.

5.1.2 Users and Data

5.1.2.1 Up-to-date Training Data

A first challenge is that for collaborative filtering recommendation (especially for music) training data must be collected as close to the study as possible in order ensure that participant data is known by the model. Real data is also difficult to locate and obtain.

In the domain of music, LastFM continues to provide a useful API for implicit data collection, though collection is not always a straightforward process. For this reason, we contacted authors of prior research utilizing data sets which fit our needs. Although their data was not up-to-date, we were able to develop our own collection method after corresponding. This data was topped-up before each subsequent study.

5.1.2.2 Live Participant Data

An especially interesting challenge was that we needed to be able to collect data from participants as they connected to the system. This data also needed to be as current as possible.

Our solution to this problem was to use a media-tracking application. For music, LastFM worked well. For smaller pools of participants, we helped them register an account and monitored it over a collection period of a few weeks. For larger pools of participants

[1] An un-maintained repository of our application can be found at https://github.com/Stack-Attack/music_rec_div_study

we specified in recruitment and consent materials that they must have an existing account containing some minimum number of LEs. Amazon Mechanical Turk specifically does not allow researchers to ask participants to log into any accounts, and so the platform must allow data to be collected without user authentication. Some participants appear to have created new accounts just to complete the study without being asked to.

5.1.2.3 Showing Music Recommendations

To evaluate a recommendation, participants need to be able to listen to it!

Music previews can typically be accessed without having to authenticate with a music service. We used the Spotify API to obtain 30 second previews with album art in the form of HTML iframes embedded in the page. The relevant track previews were retrieved by searching for tracks using their exact song and artist names as well as the region a participant was connecting form. We discarded the small portion of tracks that did not return any results; this is likely unavoidable.

5.1.3 Computation

5.1.3.1 Model Training

An obvious concern is that writing the necessary code to implement models efficiently is time-consuming and error-prone. Additionally, training models on huge data sets seems infeasible due to size and dimensionality.

Using open-source libraries can save time and help alleviate the risk of errors impacting results, though they may mis-implement key algorithmic features, or be difficult to extend. Comparing multiple implementations online can help, but one must ensure to follow any licenses and reference the source.

In order to reduce the size of data, we filtered out irrelevant items and users. By filtering out tracks with 10 or fewer LEs we reduced the number of unique tracks by 82% while only decreasing LE count by 6%. Even after filtering, our MultVAE model was too large to fit on our GPU, and so we trained multiple model variations concurrently using CPU's in order to make up for lost time. Some models may simply be infeasible without access to High Performance Computing (HPC) resources.

5.1.3.2 Complex Architecture and Resource Requirements

The size of trained models is too large to fit in memory, especially if a new model is loaded for each server connection.

The size of trained models can be **very** large even after removing unnecessary data (i.e., neural network optimizer information). Our trained MultVAE model was over 6.5GB in size. Luckily, online cloud computing platforms often offer specific instances with large amounts of dedicated memory at the expense of processing power. These instances are a great fit for running user-studies which will inherently have a low number of concurrent users. Simple hosting services such as Heroku will be infeasible due to the memory requirements [18]. AWS Elastic Beanstalk provided very cost effective solutions by selecting a memory optimized EC2 instance [40].

Even with the low number of concurrent users, there will still be some asynchronous computation required. The ideal, yet complex, solution to this problem is to decouple the longer tasks (recommendation and data collection) from the main server using a separate worker process or even server. Developing this architecture can be time-consuming, expensive, and unnecessary for such small temporary applications. We found success by limiting the server to one Python process, and running data collection and recommendation on separate threads using the built in *concurrent.futures* library[2]. As data collection was input/output bound it did not block the server from handling requests, and as most recommendation tasks utilized multiple cores these tasks were handled relatively quickly. This kind of architecture would certainly not work for large-scale applications, but was ideal for our user-study due to its simplicity and efficient use of only one compute node.

5.1.4 Summary

The benefits of evaluating music recommender systems on real users are as intuitive as they are founded in empirical evaluation [21]. In implementing the studies found in Chapters 3 and 4, we developed a music recommendation system which could generate and present recommendations to new users within a single un-moderated interactive session. To assist and encourage future researchers in developing similar systems, we provided a series of challenges and solutions to problems we encountered. Among the problems we addressed were training data collection, live user data collection, and obtaining music previews. We also discussed our technical implementation; specifically dealing with issues of memory

[2]In practice, we used the Flask-Executor python library to manage our futures: https://pypi.org/project/Flask-Executor/

management and availability. In general, we hope that researchers embrace collaboration with others to better base our analysis of recommender systems in users themselves.

Chapter 6

Conclusion

The work we present here strengthens the important connection between quantitative diversity metrics and user perceptions of diversity in music recommendation lists.

Through analysis of semi-structured interviews with 17 participants we identified two primary themes on user selections for diversity: listener mood, and different meanings for the term "diversity". More specifically, many users expressed a clear distinction between diversity within the bounds of their existing preferences, and diversity outside of these preferences. This *inner* and *outer diversity* was often expressed as a binary preference; some users did not want eider diversity, while that was exactly the goal of others. Additionally, we found that when given the ability to select their own level of diversity in recommendation lists, user selections varied widely within and between subjects.

In a larger qualitative analysis using 92 online participants, we showed that offline music recommendation accuracy metrics do not always align with real user sentiments. Our implementation of a simple genre filter inspired by the idea of inner diversity also showed that pre-processing methods can ensure satisfying and accurate recommendation lists even when heavily diversified.

Evaluating music recommender systems through user studies can be an arduous task, but it is a necessary step in order to better under how we can tailor music recommendation diversity to match actual user preferences. To help future researchers we also presented a series of challenges and the corresponding solutions we arrived at through many months of troubleshooting and development.

Much future work is required in order to understand and evaluate music recommendation diversity on a more granular level, and further inform *music recommender systems*

from the user's perspective. Our initial attempt at building a simple interactive system of diversity was unsuccessful, but this should not stop future research which enables more direct user interactions. The success of our simple genre filter and diversification method demonstrate that the answer to better music recommendation diversity does not necessarily lie in defining newer and more complex diversity definitions, although there are no doubt many more precise and tailored filtering methods to be discovered and evaluated. We believe it is important that these future methods find a basis in user perceptions and evaluations. Additionally, there is room to explore the connection between listener mood, and their preference for inner and outer diversity through a long term music recommendation study.

If beyond-accuracy metrics are to help solve the disconnect between offline accuracy and online evaluations, then they themselves should have a stronger basis in real people. Diversity in music recommendations should have at least as solid a foundation in user perception as in *information retrieval.*

References

[1] Keith Bradley and Barry Smyth. Improving recommendation diversity. In *Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science, Maynooth, Ireland*, pages 85–94. Citeseer, 2001.

[2] Sylvain Castagnos, Armelle Brun, and Anne Boyer. When Diversity Is Needed... But Not Expected! In *International Conference on Advances in Information Mining and Management*, pages 44–50, 2013.

[3] Pablo Castells, Neil J Hurley, and Saul Vargas. Novelty and diversity in recommender systems. In *Recommender Systems Handbook, Second Edition*, pages 881–918. Springer, 2015.

[4] Òscar Celma. The Long Tail in Recommender Systems. In *Music Recommendation and Discovery*, pages 87–107. Springer, 2010.

[5] Keke Chen, Patrick PK Chan, Fei Zhang, and Qiaoqiao Li. Shilling attack based on item popularity and rated item correlation against collaborative filtering. *International Journal of Machine Learning and Cybernetics*, 10(7):1833–1845, 2019.

[6] Li Chen, Wen Wu, and Liang He. How Personality Influences Users' Needs for Recommendation Diversity? In *Conference on Human Factors in Computing Systems - Proceedings*, volume 2013-April, pages 829–834, apr 2013.

[7] Charles L.A. Clarke, Maheedhar Kolla, Gordon V. Cormack, Olga Vechtomova, Azin Ashkan, Stefan Büttcher, and Ian MacKinnon. Novelty and diversity in information retrieval evaluation. In *ACM SIGIR 2008*, pages 659–666, 2008.

[8] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 101–109, 2019.

[9] Shuiguang Deng, Dongjing Wang, Xitong Li, and Guandong Xu. Exploring User Emotion in Microblogs for Music Recommendation. *Expert Systems with Applications*, 42(23):9284–9293, 2015.

[10] Tommaso Di Noia, Vito Claudio Ostuni, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. An analysis of users' propensity toward diversity in recommendations. In *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, pages 285–288, 2014.

[11] Tommaso Di Noia, Jessica Rosati, Paolo Tomeo, and Eugenio Di Sciascio. Adaptive multi-attribute diversity for recommender systems. *Information Sciences*, 382-383:234–253, 2017.

[12] Michael D. Ekstrand, F. Maxwell Harper, Martijn C. Willemsen, and Joseph A. Konstan. User perception of differences in recommender algorithms. In *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, pages 161–168, oct 2014.

[13] Katayoun Farrahi, Markus Schedl, Andreu Vall, David Hauger, and Marko Tkalčič. Impact of listening behavior on music recommendation. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 483–488, 2014.

[14] Bruce Ferwerda, Mark Graus, Andrew Vall, Marko Tkalčič, and Markus Schedl. The Influence of Users' Personality Traits on Satisfaction and Attractiveness of Diversified Recommendation Lists. In *Proceedings of the 4th Workshop on Emotions and Personality in Personalized Services (EMPIRE 2016)*, pages 43–47, 2016.

[15] Daniel M. Fleder and Kartik Hosanagar. Recommender systems and their impact on sales diversity. In *EC'07 - Proceedings of the Eighth Annual Conference on Electronic Commerce*, pages 192–199, 2007.

[16] Ben Frederickson. Fast Python Collaborative Filtering for Implicit Datasets. https://github.com/benfred/implicit, 2019.

[17] Asela Gunawardana and Guy Shani. Evaluating recommender systems. In *Recommender Systems Handbook, Second Edition*, pages 265–308. Springer, 2015.

[18] Heroku. Dyno Types. https://devcenter.heroku.com/articles/dyno-types, 2021.

[19] Yifan Hu, Chris Volinsky, and Yehuda Koren. Collaborative filtering for implicit feedback datasets. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 263–272, 2008.

[20] Marius Kaminskas and Derek Bridge. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 7(1):1–42, 2016.

[21] B. P. Knijnenburg and M. C. Willemsen. Evaluating recommender systems with user experiments. In *Recommender Systems Handbook, Second Edition*, pages 309–352. Springer, 2015.

[22] Matevž Kunaver and Tomaž Požrl. Diversity in recommender systems – A survey. *Knowledge-Based Systems*, 123:154–162, 2017.

[23] Jin Ha Lee. How similar is too similar?: Exploring users perceptions of similarity in playlist evaluation. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 109–114, 2011.

[24] Jin Ha Lee and Rachel Price. Understanding users of commercial music services through personas: Design implications. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, pages 476–482, 2015.

[25] Jin Ha Lee, Liz Pritchard, and Chris Hubbles. Can we listen to it together?: Factors influencing reception of music recommendations and post-recommendation behavior. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 663–669, nov 2019.

[26] Amaury L'Huillier, Sylvain Castagnos, and Anne Boyer. Understanding usages by modeling diversity over time. In *CEUR Workshop Proceedings*, volume 1181, pages 81–86, 2014.

[27] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 World Wide Web Conference*, pages 689–698, 2018.

[28] Feng Lu and Nava Tintarev. A diversity adjusting strategy with personality for music recommendation. In *CEUR Workshop Proceedings*, volume 2225, pages 7–14, 2018.

[29] Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 1097–1101, 2006.

[30] So Yeon Park, Audrey Laplante, Jin Ha Lee, and Blair Kaneshiro. Tunes together: Perception and experience of collaborative playlists. In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 723–730, 2019.

[31] Lorenzo Porcaro and Carlos Castillo. Music Recommendation Diversity : A Tentative Framework and Preliminary Results. In *Proceedings of the 1st Workshop on Designing Human-Centric Music Information Research Systems*, pages 11–15, 2019.

[32] Lorenzo Porcaro and Emilia Gomez. 20 Years of Playlists : a Statistical Analysis on Popularity and Diversity. *Proceedings of the 20th International Society for Music Information Retrieval Conference*, pages 4–11, 2019.

[33] Marco Tulio Ribeiro, Anisio Lacerda, Adriano Veloso, and Nivio Ziviani. Pareto-efficient hybridization for multi-objective recommender systems. In *RecSys'12 - Proceedings of the 6th ACM Conference on Recommender Systems*, pages 19–26, 2012.

[34] Kyle Robinson, Daniel G. Brown, and Markus Schedl. User insights on diversity in music recommendation lists. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, pages 446–453, 2020.

[35] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth international conference on computer and information science*, volume 1, pages 27–8. Citeseer, 2002.

[36] J Ben Schafer, Joseph A Konstan, and John Riedl. Meta-recommendation systems: User-controlled integration of diverse recommendations. In *International Conference on Information and Knowledge Management, Proceedings*, pages 43–51, 2002.

[37] Markus Schedl. Deep learning in music recommendation systems. *Frontiers in Applied Mathematics and Statistics*, 5:44, 2019.

[38] Markus Schedl, Peter Knees, Brian McFee, Dmitry Bogdanov, and Marius Kaminskas. Music recommender systems. In *Recommender Systems Handbook, Second Edition*, pages 453–492. Springer, 2015.

[39] Markus Schedl, Hamed Zamani, Ching Wei Chen, Yashar Deldjoo, and Mehdi Elahi. Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval*, 7(2):95–116, jun 2018.

[40] Amazon Web Services. EC2 Instance Types. https://aws.amazon.com/ec2/instance-types/, 2021.

[41] Malcolm Slaney and William White. Measuring playlist diversity for recommendation systems. In *Proceedings of the ACM International Multimedia Conference and Exhibition*, pages 77–82, 2006.

[42] Marko Tkalčič, Andrej Košir, and Jurij Tasič. Affective recommender systems: The role of emotions in recommender systems. In *CEUR Workshop Proceedings*, volume 811, pages 9–13, 2011.

[43] Chun Hua Tsai and Peter Brusilovsky. Leveraging interfaces to improve recommendation diversity. *UMAP 2017 - Adjunct Publication of the 25th Conference on User Modeling, Adaptation and Personalization*, pages 65–70, 2017.

[44] Saúl Vargas. New Approaches to Diversity and Novelty in Recommender Systems. In *FDIA'11: Proceedings of the Fourth BCS-IRSG conference on Future Directions in Information Access*, pages 8–13, 2011.

[45] Saúl Vargas, Linas Baltrunas, Alexandros Karatzoglou, and Pablo Castells. Coverage, redundancy and size-awareness in genre diversity for recommender systems. In *RecSys 2014 - Proceedings of the 8th ACM Conference on Recommender Systems*, pages 209–216, 2014.

[46] Martijn C. Willemsen, Bart P Knijnenburg, Mark P Graus, Linda C.M. Velter-Bremmers, and Kai Fu Eindhoven. Using latent features diversification to reduce choice difficulty in recommendation lists. In *CEUR Workshop Proceedings*, volume 811, pages 14–20, 2011.

[47] David Wong, Siamak Faridani, Ephrat Bitton, Bjoern Hartmann, and Ken Goldberg. The Diversity Donut: Enabling participant control over the diversity of recommended responses. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 1471–1476, 2011.

[48] Yuan Cao Zhang, Diarmuid Ó Séaghdha, Daniele Quercia, and Tamas Jambor. Auralist: Introducing serendipity into music recommendation. In *WSDM 2012 - Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 13–22, 2012.

[49] Cai-Nicolas Ziegler, Sean M. McNee, Joseph A. Konstan, and Georg Lausen. Improving recommendation lists through topic diversification. In *Proceedings of the 14th international conference on World Wide Web - WWW '05*, page 22, 2005.