KU LEUVEN

FACULTY OF
ENGINEERING SCIENCE

A Playlist-based Group Music Recommendation System

Alexander Joossens

Thesis submitted for the degree of
Master of Science in Engineering:
Computer Science, option Artificial
Intelligence

Supervisor:
Prof. dr. K. Verbert

Assessors:
Dr. Maria-Mihaela Trusca
Mingxiao Li

Assistant-supervisor:
Ivania Nadine Donoso Guzmán

Academic year 2022 – 2023

Preface

This thesis is conducted at the Department of Computer Science at the KU Leuven. It has been an extensive journey, involving a lot of research, brainstorming, programming, and numerous late nights of work. Overall, it has been a highly enriching and captivating learning experience.
I would like to thank my thesis-mentor Ivania Nadine Donoso Guzmán for her guidance, time, patience, and valuable advice.
I also want to thank my supervisor, Prof. dr. Katrien Verbert, for giving me the opportunity to choose my own thesis topic, and for her helpful feedback.
This final work will also mark the ultimate milestone towards my graduation. That is why I want to express my heartfelt gratitude to my parents who gave me the opportunity to achieve this diploma.
Finally, I am thankful for all 80 participants of the user study for their time and effort.

Alexander Joossens

Contents

Abstract

Recommender systems have become an important part of our everyday lives, impacting our choices and preferences. A major challenge encountered in numerous recommender systems is their perceived black box nature, leaving users unaware of the underlying basis of their recommendations. Furthermore, another concept that plays an important role on the user's perceptions is the use of contextual filtering.
This work focuses on different filtering algorithms (pre- and post-filtering) and different visual explanation types (content- and user-based explanations) in a group music recommender system. A web application was built that takes these selected playlists of the users as input and generates a new recommended playlist as output. It does this using API calls from the Spotify Web API. Users can actively influence the system by making their own playlist selection, steering the recommendations.
A between-subjects user study with 80 participants was conducted to investigate the influence of these two filtering techniques and visual explanation types on the users' perceptions. Pairs of participants tested the web application under four use case combinations. Afterwards, they completed a post-test questionnaire, including questions related to perceived fairness, accuracy, explanation, behavioural intentions, and ease of use. The data analysis identified the optimal question combinations, assessed main and interaction effects, and determined specific group differences.
The results show that a combination of post-filtering (providing only new recommendations) and content-based explanations (like matching song characteristics with the user's taste) can have a positive influence on the *perceived explanation* and the *behavioural intentions*.
Ultimately, the participants responded positively to the application. Therefore, this study has the potential to enhance current commercial music platforms, such as Spotify's *Blend* feature.

Samenvatting

Aanbevelingssystemen spelen een belangrijke rol in ons dagelijks leven en hebben een impact op onze keuzes en voorkeuren. Een grote uitdaging bij veel aanbevelingssystemen is dat ze vaak overkomen als een 'zwarte doos'. Hierdoor zijn de gebruikers zich niet bewust van de onderliggende basis van hun aanbevelingen. Een ander concept dat ook een belangrijke invloed heeft op de percepties van de gebruiker is het gebruik van contextueel filteren.

Dit werk richt zich op verschillende filteralgoritmen (pre- en post-filtering) en verschillende soorten visuele uitleg (inhoudsgebaseerde en gebruikersgebaseerde uitleg) in een muziekaanbevelingssysteem voor groepen. Een webtoepassing werd gebouwd die deze geselecteerde afspeellijsten als invoer neemt, en een nieuwe afspeellijst als uitvoer genereert. Het doet dit m.b.v. 'API calls' van de Spotify Web API. Gebruikers kunnen actief invloed hebben op het systeem door hun eigen selectie van afspeellijsten te maken en zo de aanbevelingen mee te sturen.

Via een between-subjects onderzoek met 80 deelnemers werd de invloed van deze twee filtertechnieken en typen visuele uitleg op de percepties van de gebruiker onderzocht. Deelnemersparen testten de webtoepassing onder vier combinaties van gebruiksscenario's. Daarna vulden ze een 'post-test' vragenlijst in die vragen bevatte omtrent de ervaren eerlijkheid, nauwkeurigheid, uitleg, gedragsintenties en gebruiksgemak. De data-analyse toonde de optimale combinaties van vragen aan, beoordeelde hoofd- en interactie-effecten, en bepaalde specifieke groepsverschillen.

De resultaten tonen aan dat een combinatie van post-filteren (met het geven van enkel nieuwe aanbevelingen) en inhoudsgebaseerde uitleg (zoals overeenkomende eigenschappen van liedjes met de smaak van de gebruiker) een positieve invloed kan hebben op de *ervaren eerlijkheid* en de *gedragsintenties*.

Ten slotte werd de webtoepassing positief beoordeeld door de deelnemers. Daarom kan dit onderzoek bestaande commerciële muziektoepassingen helpen verbeteren, zoals de toepassing Spotify *Blend*.

List of Figures and Tables

List of Figures

List of Tables

List of Abbreviations

CFA	Confirmatory Factor Analysis
ANOVA	Analysis of Variance
HSD	Honestly Significant Difference
API	Application Programming Interface
REST	Representational State Transfer
ResQue	Recommender systems' Quality of user experience
TLX	Task Load Index
RMSEA	Root Mean Square Error of Approximation
TLI	Tucker-Lewis Index
CFI	Comparative Fit Index
AVE	Average Variance Extracted

Chapter 1

Introduction

1.1 Motivation

Many aspects of our lives are influenced by recommender systems: the posts we see on Instagram, the routes we take to go places, the videos we watch, and the advertisements we see. Often these recommender systems know very well what we want, even without us explicitly having to give them our preferences. This is because the recommender systems consist of advanced models that combine various techniques to identify patterns, trends, and correlations within the data. By combining data from many users, products, and services, the model will accurately predict what the user wants to see.

However, one of the major problems with all of these popular recommender systems is that they appear to be black boxes. This means that users do not know what their recommendations are based on and where they come from. The predictions seem to be 'magically' generated by a secret model whose inner workings are unknown. According to He et al. [33], this 'creepy' behaviour of recommendation systems can lead to a lack of trust for the users, who want to know where the recommendations come from. Until now, most of the research in recommender systems has an emphasis on the algorithms behind them. Little research has been done yet on the interface and explainability, which are crucial in improving user experience [6]. Therefore, this thesis researches the explanations that are shown to the users, next to the recommended items. Different explanation styles may have different influences on various usage aspects.

Group recommender systems provide suggestions to a group of multiple users. Using data and characteristics from multiple users, a set of recommended items is produced. One critical aspect of group recommender systems is the users' perception of fairness and accuracy. These concepts can be defined and measured in many ways, as precisely explained in section 2.4. In a nutshell, the goal is to provide relevant recommendations to each of the users and to give them all an equal amount of influence on the outcome of the system.

One aspect that can have a major impact on the perceived fairness is the use of pre- and post-filtering. In pre-filtering, the input data (e.g. from multiple users in a group) is adapted or weighted in some way, after which the data is used in the recommender algorithm. As an example, the individual data set of each user can be accumulated into one single data set. This data will then be used as input for the recommender system. In post-filtering, this change in the data happens after the execution of the recommender algorithm. For example, when a recommender system provides a list of recommended items, the list can be filtered in such a way that the suggested items align with the interests of all users as evenly as possible.

A popular application of group recommender systems is the music we listen to. In June 2021, Spotify released the beta version of *Blend*[1]. To use this feature, the user presses the 'Blend' button to select any of their friends, and the recommender system will automatically create a new playlist based on both users. The playlist will keep itself up-to-date by checking what the individual users listen to daily and changing the mutual playlist accordingly. However, the problem with this feature is that it works as a black box: the inner workings of the recommender algorithm have not been made public, and users do not know what the recommendations are based on. The only indication they get in the resulting playlist is an icon that visualises which of the two users has already 'liked' the song in their library. Although this is a minimal explanation, it may not be enough. After doing an initial questionnaire 4.1 about what people think of Spotify Blend, the results spoke for themselves: 50% of the participants tried Spotify Blend before, but only 30% of them liked it. They mentioned a clear need for explanation, and more influence on the recommendations.

For these reasons, the explanations of recommender systems are becoming more and more important nowadays. Not only because the EU legislation decided in 2016 that users should get explanations, but also because the users themselves are more satisfied with the suggestion when they receive an explanation. This was confirmed by Gkika and Lekakos [30], who investigated the black box nature of recommender systems and concluded that providing a clear explanation can change the consumer's attitude and increase their acceptance of the recommendation, despite their initial resistance or hesitation.

1.2 Problem statement

To help unveil this black box nature of recommender systems, this thesis focuses on building and evaluating a new group music recommender system based on the user's playlists in Spotify. The goal is to give users visual explanations about the generated playlist. Also, users influence the outcome of the recommender system by first making a selection of their playlists. This way, they can steer the recommendations and actively influence the results.

To summarise these problem statements, we formulated two research questions:

- How do pre- and post-filtering in a group recommendation system influence the perceived accuracy, fairness, explanation and behavioural intentions?

[1] Blend is a feature in Spotify that automatically generates a playlist based on the music taste of you and a friend. See release note.

- How do different visual explanations of group recommendations affect the perceived fairness, accuracy, explanation, behavioural intentions, and ease of use?

1.3 Methodology

The application focuses on groups of two persons who want to generate a mutual playlist to listen to. To produce the new playlist, two group recommender algorithms were created using pre- and post-filtering. Furthermore, two types of explanatory visualisations were built to ensure the users understand why these songs are recommended to them.
The evaluation of the application was performed with a between-subject user study. A total of 80 participants were recruited to test the web application. They were randomly assigned to one of the four use case combinations (pre- or post-filtering with content- or user-based explanation). The goal was to understand the influence of the different algorithms and visualisations on the perceived accuracy, fairness, explanation, behavioural intentions and ease of use for the user. This was done by using a questionnaire that is based on the ResQue model [62].

1.4 Overview

First of all, this research is motivated by an extensive literature study in chapter 2. It starts with a general informative study about the three types of recommender systems. After this follows a more specific discussion about group recommender systems, explanations, fairness in group recommender systems, and music recommender systems.
The literature study gives a motivation for the research questions. These are described in detail in chapter 3.
After this follows a description of the application design in chapter 4. Starting from an initial questionnaire and multiple think-aloud studies, the application flow and visualisations were decided. Also, the inner workings of the pre- and post-filtering algorithms are discussed thoroughly.
Subsequently, in chapter 5 follows an explanation of the application implementation. This consists of the backend, frontend, and Spotify Web API.
Chapter 6 describes the user study that has been done for this application. The results from this study are described in chapter 7. It also provides an explanation of the conducted data analysis.
Finally, the discussion of the results of the research follows in chapter 8, after which a conclusion is stated in chapter 9. Suggestions for further research are also included in this chapter.

Chapter 2

Literature Study

This section gives an overview of the relevant work that has already been done in the field of recommender systems. It provides information about research in five different domains: recommender systems in general, group recommendation systems, explanations, fairness and music recommender systems.

2.1 Recommender systems

2.1.1 Goal

Recommender systems provide suggestions to users and exist in many different forms. Recommendations can have several goals: increase sales, sell more diverse items, increase user satisfaction, increase user fidelity or better understand what the user wants [68]. Because of these many goals, the applications of recommender systems are also very diverse. Itmazi and Gea [37] highlight the importance of recommender systems in the scope of eLearning, Covington et al. [19] add a Deep Neural Network to a recommender system to generate YouTube recommendations, Bui et al. [15] use a recommender system for traffic problems, and Pérez-Marcos and Batista [63] create a recommender system to suggest music to users of Spotify. They all share the goal of providing relevant suggestions for users based on input data. In many applications, there is an enormous number of items to choose from for the user. The recommender system can then filter all of these items in the best possible way, only showing the most relevant items to the user.

2.1.2 Methods

Many recommender algorithms and models exist, each having advantages and disadvantages. Patel et al. [58] explain that there are three ways of recommending. These are collaborative filtering, content-based filtering, and a combination of both, called hybrid recommender systems.
In collaborative filtering, items are suggested to users based on what other users have consumed [58]. The idea behind these methods is that users with similar interests in the past will also have similar interests in the future. This method's advantage is that it does

not need knowledge about the recommended items. The model only uses the information of the feedback that the user provides. However, a disadvantage of this approach is the cold start problem, which emerges when a new user without past records enters the system [1, 40, 47]. Another disadvantage is that a large amount of data is required to provide the correct recommendations. However, not every item is rated by every user. This problem is referred to as sparsity [40, 47].
In content-based filtering, items are suggested to users based on the characteristics of the items themselves [1, 58]. The idea is that users who have bought or used many items with similar characteristics in the past, will also do this in the future. This type of suggestion can make use of a large number of characteristics and features of the items. The advantage of this approach is that the model does not need multiple users, because the recommender system can provide provide suggestions based only on the characteristics of these items. Therefore, the cold start problem does not occur in content-based filtering. However, a downside of this approach is that if the profile of the item does not contain enough information, the recommendation will not be precise [25].
Finally, there are also hybrid recommender systems. They combine the previous two approaches into a single model. Adomavicius and Tuzhilin [3] propose four ways to do this:

- '*Implementing collaborative and content-based methods separately and combining their predictions*'
- '*Incorporating some content-based characteristics into a collaborative approach*'
- '*Incorporating some collaborative characteristics into a content-based approach*'
- '*Constructing a general unifying model that incorporates both content-based and collaborative characteristics*'

Hybrid recommending is the most effective [57], as it combines the advantages of the previous two approaches and is, therefore, the most popular method. However, this technique also combines the limitations of both models, such as the earlier discussed cold-start problem and data sparsity.

2.1.3 Challenges

Recommender systems come with many different challenges. A first big challenge of collaborative-filtering recommender systems is handling sparsity [29]. Generally, the majority of the users do not rate most of the items, and consequently, the rating matrix becomes very sparse. The typical values of sparsity can become close to 100% in most real-world recommender systems [65]. An analysis of the Yahoo! Music database by Dror et al. [22] shows that there is a sparsity of 99.96%. The Netflix database on the other hand, has a sparsity of 'only' 98.82%. These big sparsity values mean that the chance of finding a set of users with similar ratings declines. This is a big disadvantage of the collaborative filtering approach.

Another challenge, caused by a consequence of data sparsity, is the cold-start problem (shown in figure 2.1). This happens when the production of predictions for a new user having very few ratings is not possible because of insufficient data to profile them. In these situations, it is very hard for the recommender system to make a good recommendation since it does not know anything about the new group or the new items. Khusro et al. [42] propose three solutions:

- Asking the users at the beginning to rate some items
- Asking the users explicitly to state their taste
- Suggesting items to the users based on the collected demographic information

An example of the first solution in the music domain would be to let the user rate some songs. With this information, the algorithm will learn what genres the user likes and can recommend other songs from this genre to the user. For the second solution, the algorithm could do the same thing, but instead of asking the user to rate some songs, it can ask the user to state their taste in music (e.g.: '*What is your favourite artist?*'). Finally, the algorithm can also use demographic information to generate more accurate suggestions already from the start. For example, the recommender system can use the target group's nationality to recommend songs currently popular in the hit lists of that country.

Another solution to tackle the cold start problem is given by Halder et al. [31]. They propose a movie swarm mining concept, using two pruning rules and frequent item mining. However, the proposed method has a shortcoming of finding clusters of users when they enjoy a diverse set of movies. A number of other methods have been proposed to deal with sparsity and cold start problems [55].

A third challenge consists of training and evaluating a recommender system, for which a data set is needed. The data set should be correctly chosen to have all the necessary information needed for the proper implementation of the recommender system [45]. Luckily, a lot of data sets have been put together: *MovieLens*[1] is one of the largest movie data sets with over 25 million movie ratings. *Social Network Influencer*[2] offers a data set for social media data, including users with their volume of interactions, number of followers, etc. *Million Song Dataset*[3] and *Free Music Archive*[4] are very large music data sets with thousands of artists, albums and songs. As these data sets are extremely large, sparsity and scalability problems get larger as well. To cope with this large amount of data, Aiolli [5] focusses on memory-based collaborative filtering approaches since they are able to deal with large data sets in an efficient and effective way. With this technique, they won the *Million Song Dataset Challenge* [50].

Finally, another complex challenge is knowing what the purpose of the users is; Do they want to get suggestions that they already know about (and like), or do they want

[1] https://grouplens.org/datasets/movielens/25m/
[2] https://www.kaggle.com/c/predict-who-is-more-influential-in-a-social-network
[3] http://millionsongdataset.com/
[4] https://github.com/mdeff/fma

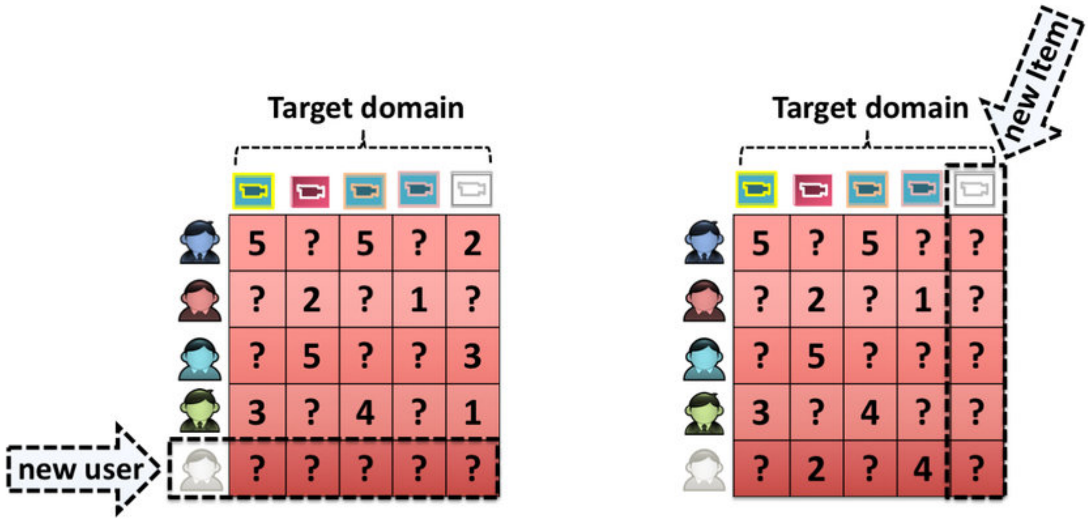


Figure 2.1: Cold Start Problem in CF [23]

to discover new items? This is called the exploration/exploitation trade-off. Both exploration and exploitation are needed to achieve the user's goal of wanting the best possible suggestion, but they cannot be carried out at the same time [66]. According to Barraza-Urbina [9], balancing the exploitation and exploration dilemma is key to the long-term success of a recommender system, leading to benefits such as:

- '*Increase novelty/serendipity, thereby facing the filter bubble*'
- '*Improve catalogue coverage and increase overall item visibility*'
- '*Deal with both user and item cold-start problems*'
- '*Respond to dynamic user preferences or evolving user intent*'
- '*React to contextual changes*'
- '*Dynamically adapt to changing business goals*'

2.2 Group recommender systems

2.2.1 Motivation

A lot of progress has been made on individual recommender systems. The challenge of suggesting relevant items to a user has been tackled in many different ways and has brought some very effective solutions [34, 39, 76]. Despite this success, often, these individual recommender systems are not sufficient. There are many situations where a group would like a joint recommendation. For example, when a group of people want to

watch a movie together or when a family wants to buy flight tickets for a holiday. Other examples include interactive television, advertising in cinemas or booking a hotel with friends.
The domain 'group recommender systems' is still a less explored research topic. In RecSys 2021, there were no papers about group recommender systems and in RecSys 2022 only two papers. Recommending to groups of users is also more complicated than recommending to individuals [48]. Instead of only taking into account the tastes and data of a single user and using a model to generate recommendations on one source of input, group recommender systems have to combine the tastes and data from multiple persons to get a common list of suggested items.
However, some studies have already used group recommender systems. Ashu and Debasish [7], Villavicencio et al. [71] and Pessemier et al. [59] use it for movie recommendations, Wang et al. [73] and Fang et al. [24] use it for trusted social networks, Barile et al. [8] use it for smart city applications and Nozari and Koohi [53] use group recommender systems to compute the leader's impact on the members' preferences in a group. Also, the growing number of group activities available on the web has increased research on group recommender systems [53].

2.2.2 Approaches

Group recommender systems focus on recommendations for a group of persons. After gaining an understanding of how to create recommendations for individual users, the next step is to explore the process of merging these individual user models into a unified group recommender model. To do this combination, Masthoff [48] suggests an overview of aggregation strategies (see table 2.1).
Perhaps the most simple one from these aggregation strategies is the average. This strategy will give equal weight to all of the individual outcomes and return the average. Besides this, the multiplicative strategy favours similar individual outcomes. For example, when two individuals rate a song 7/10, the multiplication (0.49) is higher than if one of the individuals would rate the song 6/10 and the other 8/10 (which results in a score of 0.48). This characteristic can be useful for combining the tastes of both users.
As the application will have to generate a list of songs, the sequence order will also greatly impact user satisfaction, perceived fairness, and accuracy. The results could be shown in descending order based on the individual song scores. However, Masthoff [48] states that there are some issues to take into account. These consist of a good narrative flow, mood consistency, and a strong ending. Choosing the optimal strategy is very domain specific and should also be further investigated in the music domain.
Apart from the aggregation strategy, also filtering can be applied in group recommender systems. First we explain this concept for individual recommender systems: Manipulation of input data can have a big influence on the performance of a recommender system. This manipulation can be done using contextual filtering and has been studied by Adomavicius et al. [4]. There are three categories: Contextual pre-filtering, contextual post-filtering and contextual modelling. The concepts are visualised in figure 2.2. The idea of contextual pre-filtering is to only select the relevant set of input data to use in the recommender

Strategy	How it works	Example
Plurality Voting	Uses 'first past the post': repetitively, the item with the most votes is chosen.	A is chosen first, as it has the highest rating for the majority of the group, followed by E (which has the highest rating for the majority when excluding A).
Average	Averages individual ratings	B's group rating is 6, namely (4+9+5)/3.
Multiplicative	Multiplies individual ratings	B's group rating is 180, namely 4*9*5
Borda Count	Counts points from items' rankings in the individuals' preference lists, with bottom item getting 0 points, next one up getting one point, etc	A's group rating is 17, namely 0 (last for Jane) + 9 (first for Mary) + 8 (shared top 3 for Peter)
Copeland Rule	Counts how often an item beats other items (using majority vote) minus how often it looses	F's group rating is 5, as F beats 7 items (B,C,D,G,H,I,J) and looses from 2 (A,E).
Approval Voting	Counts the individuals with ratings for the item above a approval threshold (e.g. 6)	B's group rating is 1 and F's is 3.
Least Misery	Takes the minimum of individual ratings	B's group rating is 4, namely the smallest of 4,9,5.
Most Pleasure	Takes the maximum of individual ratings	B's group rating is 9, namely the largest of 4,9,5.
Average without Misery	Averages individual ratings, after excluding items with individual ratings below a certain threshold (say 4).	J's group rating is 7.3 (the average of 8,8,6), while A is excluded because Jane hates it.
Fairness	Items are ranked as if individuals are choosing them in turn.	Item E may be chosen first (highest for Peter), followed by F (highest for Jane) and A (highest for Mary).
Most respected person	Uses the rating of the most respected individual.	If Jane is the most respected person, then A's group rating is 1. If Mary is most respected, then it is 10.

Table 2.1: Overview of Aggregation Strategies [48]

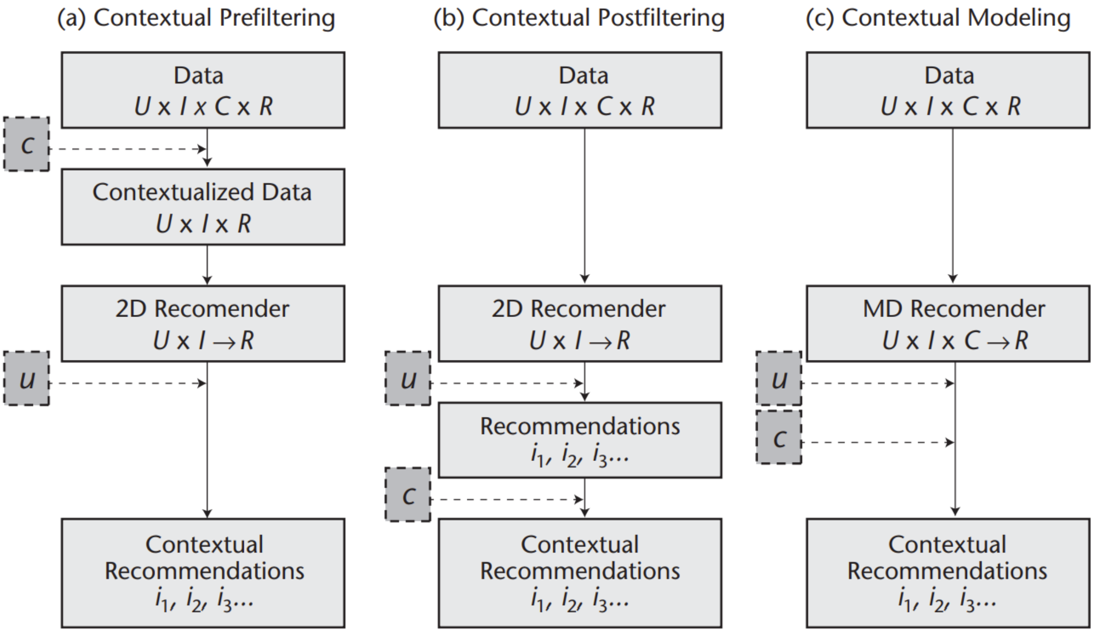


FIGURE 2.2: Pre/post filtering and contextual modelling [4]

system. Post-filtering is similar but modifies the output data. In contextual modelling, contextual information is used directly in the modelling technique. Panniello et al. [56] compare pre- and post-filtering in an e-commerce application and concludes that the post-filtering approach achieved a better performance in their collaborative contextual recommender system.

What has not yet been investigated is how these types of filtering can be applied in group recommender systems. For example, the *combination of different individual user inputs* can be seen as a pre-filtering for the group recommender system. Before the actual recommendation happens, the input data is already pre-filtered based on some individual user aspects. Another possibility is to see the *combination of different individual user inputs* as a post-filtering for the group recommender system. What happens in this case, is that each of the users' input data is used for individual recommender systems. The outcomes of these recommender systems is then aggregated in a post-filtering step (e.g. based on some mutual user characteristics). This is exactly what will be investigated in this research. A more detailed explanation of the pre- and post-filtering concepts follows in section 3.3.

2.2.3 Challenges

There are still many challenges in the domain of group recommender systems. Different types of groups lead to different ways in which the preferences can be modelled [13]. Boratto [12] states that there are three main types of groups:

- Established groups: These are groups of people who share long-term mutual interests, like a group of fans of a director. (e.g. [60], [43])
- Occasional groups: These are groups of people with the same specific aim, like watching a movie together. (e.g. Mccarthy and Anagnost [49] use people working out in a gym together)
- Random groups: These groups consist of people with no common interests or connections (e.g. the recommendation of background music in a supermarket, or Carolis and Pizzutilo [17] use random groups of people that are in a public space at a specific time)

Furthermore, the way of collecting preferences from the users to the group recommender system is still a challenge. Just like in individual recommender systems, the model can obtain the preferences implicitly or let the users give them explicitly. However, in group recommender systems users can be given access to view and copy the preferences of other users. This can have a big influence on the system [38]. Gartrell et al. [28] allow individuals and groups to express their preferences. It is also possible to let the user express negative ratings [18, 49]. This way, for example, it is possible to avoid watching a movie that is disliked by someone from the group.
Another challenge in group recommender systems is the arrival of all group members at a final decision. For individual recommender systems, the final decision for multiple recommended options (for example which movie to watch) is just a choice for a single person. Within a group setting however, negotiation may be necessary, posing a particular challenge when communication among members is limited. Jameson [38] describes three possible solutions to this problem.

- *The system simply translates the highest rated solution into action without requesting the consent of any users*
- *It is assumed that one group member is responsible for making the final decision*
- *It is assumed that group members will arrive at the final decision through straightforward face-to-face discussion*

It is however still a challenge to determine which of these solutions is the best option in a specific context.

2.3 Explanations

2.3.1 Overview of explanation types

Many papers agree that providing good explanations to the users of a recommender system is critical. According to Tintarev and Masthoff [67], explanations in recommender systems can lead to seven possible advantages:

- '*Transparency: Explaining how the system works*'
- '*Scrutability: Allow users to tell the system it is wrong*'
- '*Trust: Increase users' confidence in the system*'
- '*Effectiveness: Help users make good decisions*'
- '*Persuasiveness: Convince users to try or buy*'
- '*Efficiency: Help users make decisions faster*'
- '*Satisfaction: Increase the ease of usability or enjoyment*'

Explaining recommendations can be done in many ways. A good overview is given by Kouki et al. [44] (shown in table 2.2). They did a crowd-sourced user study that evaluated the effects of explanation style, number, and format on user preferences for explanations. They concluded that '*People prefer item-centric but not user-centric or socio-centric explanations*', '*People prefer to see at most three to four explanation styles*', and '*Textual explanations are ideal*'.

Explanation Style	**We recommend *U2* because:**
(I) User-based	User *Aren* with whom you share similar tastes in artists, listens to U2.
(II) Item-based	(a) People who listen to your profile item *AC/DC* also listen to U2. (b) Last.fm's data indicates that U2 is similar to *Coldplay* that is in your profile.
(III) Content	(a) U2 has similar tags as *Beatles* that is in your profile. (b) U2 is tagged with *rock* that is in your profile.
(IV) Social	Your friend *Cindy* likes U2.
(V) Item popularity	U2 is very popular in the last.fm database with 3.5 million listeners and 94 million playcounts.

TABLE 2.2: Explanation Styles [44]

In textual explanation styles, it has been found by Carenini and Moore [16] that in general, short arguments are better and more understandable than long arguments with a lot of detail. Also Schaffer et al. [64] conclude that detailed, full explanations have a negative impact on user confidence and enjoyment. However, when we look at high-risk products, users would prefer more detailed textual explanations, while for low-risk products, users tend to prefer short textual explanations [61].
Many companies use explanations for their recommendations in their software. A very common approach is offering other items at the checkout page of an online shopping basket. Amazon does this even already when you are just viewing a product (shown in figure 2.3). The products that are suggested are explained in a user-based textual way, saying: '*Customer who viewed this item also viewed X*'

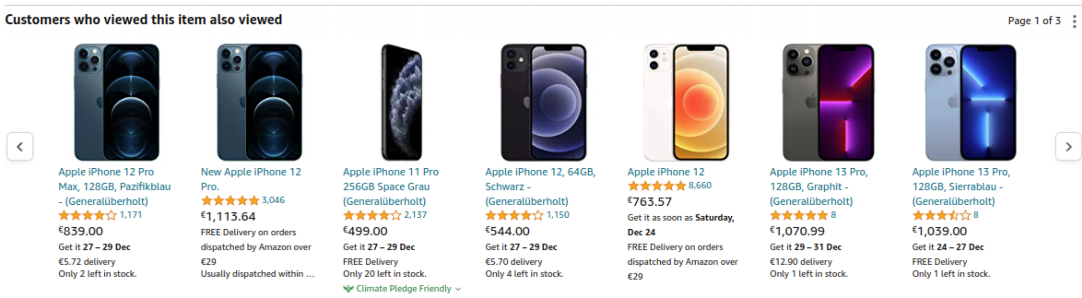


Figure 2.3: User-based textual explanation for Amazon suggestions

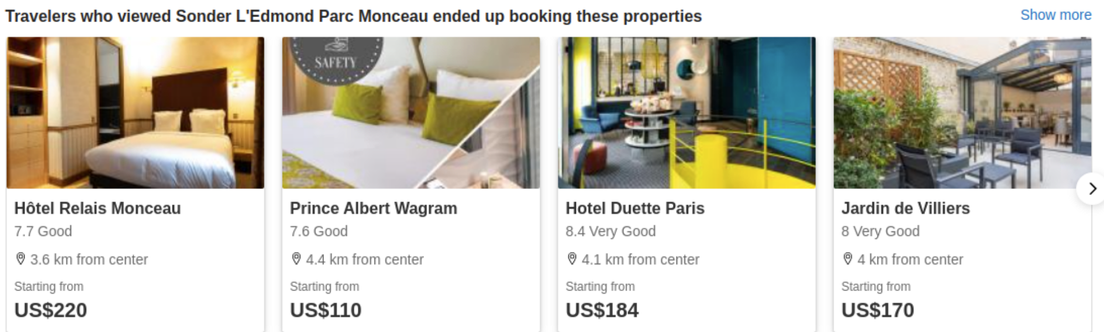


Figure 2.4: User-based textual explanation for Booking.com suggestions

Another example was found on the website of Booking.com. When you do a search and click on any hotel to see the availability, at the bottom of the page, other suggestions are given. A recommender system offers other hotels to the user, based on the actual hotels that were booked by the travellers who have also visited this page (shown in figure 2.4). Again, the provided suggestions are given together with a user-based textual explanation saying: '*Travelers who viewed Sonder L'edmond Parc Monceau ended up booking these properties*'.

2.3.2 Visualisations

One of the first studies to ever investigate how visualisation techniques can improve explanations, was done by Herlocker et al. [34]. They used the MovieLens web-based movie recommender to provide recommendations for movies and videos. In their study, 21 presentation styles were compared. They found that it can be efficient in explanations for automatic collaborative filtering systems to use rating histograms, e.g. with the text 'This movie has 4 stars because of the ratings of other similar users'. Furthermore, they showed that simple graphical explanations are more accepted than complex graphical presentations.

Tsai and Brusilovsky [70] proposed *Relevance Tuner+*, which provides a controllable interface for the user to fuse social recommendations from multiple sources. Four types

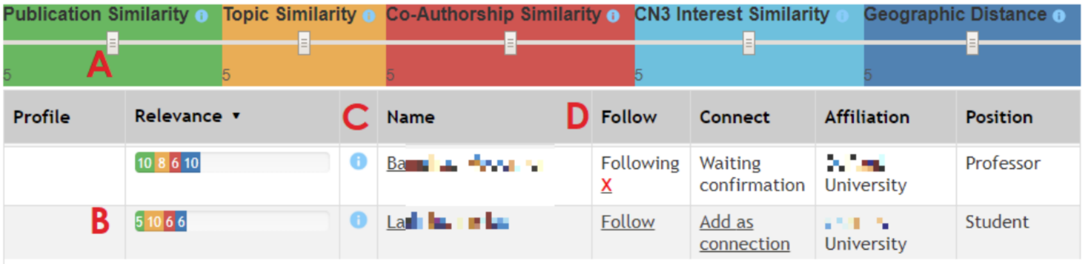


FIGURE 2.5: Design of *Relevance Tuner+*, containing four types of visualisations[70]

of visualisation of the recommendation are investigated (shown in figure 2.5), using (A) relevance sliders; (B) stackable score bar; (C) explanation icon; (D) user profiles. They found that an interactive interface helps to improve the user experience and initiate user-driven exploration. They also stated that the explanations were not used as heavily as expected. Furthermore, the most popular explanations were the ones that were the most understandable, persuasive, and enjoyable. Furthermore, they conclude that the possibly overwhelming amount of information caused the users to decrease the perception of controllability.

A final interesting research on visualisations in recommender systems has been done by Millecamp et al. [51]. They compared two different techniques, using sliders and a radar chart, to allow the users to manipulate five musical attributes used to produce recommendations. Figure 2.6 shows these two visualisations. They concluded that '*the radar chart helped the participants discover a significantly higher number of new songs compared to the sliders*'. Furthermore, they state that the most important music attributes to take into account when implementing a visual control technique are *energy, acousticness, danceability, instrumentalness, tempo and valence.*

2.3.3 Visualisations for group recommender systems

Visually explaining the results of group recommender systems is more complex than for individual recommender systems. Since they have to account for more users, the visual presentations can quickly become cluttered and messy. This is confirmed by Htun et al. [35], who state that scalability is an important aspect of visualisations. They also find that a lot depends on the number of users of the application. A bar chart, for example, may become chaotic when many users are involved. In their design phase, they also stated that visualisations should not be too complex because users do not want to put much effort into understanding what they mean. This means that the demand for strong cognitive skills should be limited as much as possible. For example, graphs with many edges or charts with a large number of bars should be avoided.

A hierarchy visualisation method for group recommender systems is proposed by Wang et al. [72] to provide visual and intuitive explanations. A hierarchy graph and a pie chart were used to illustrate visualisations for movie recommendations based on the MovieLens data set. Figure 2.7 shows the visualisation. Users are allowed to explore the group

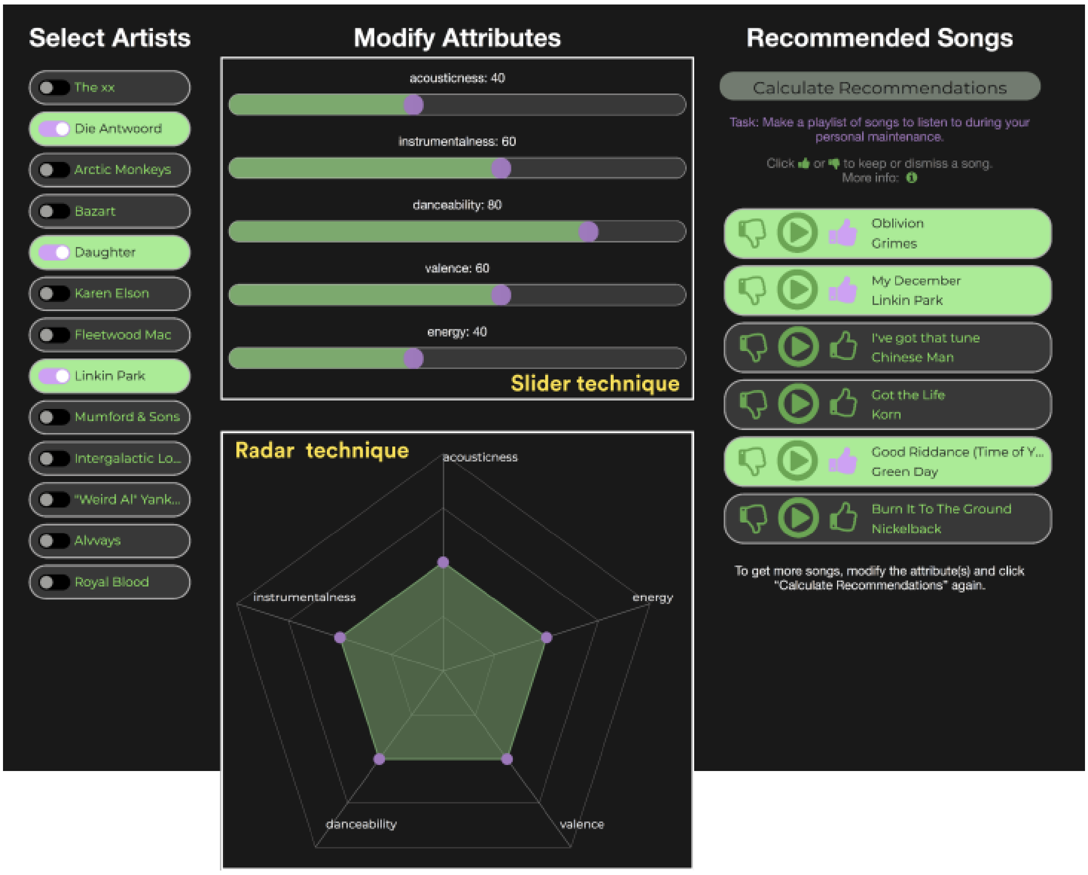


Figure 2.6: Selected artists are used with sliders or radar techniques to control the outputted recommended songs [51]

recommender process. The method organises and presents information using multilevel nodes and edges, which explain the recommender procedures. According to the authors, '*The group members can understand their influence and relative relationships with other members by mapping members by the pie chart for every node*'.

To visualise the current status of a group decision process, Felfernig et al. [26] use a *spider diagram* (shown in figure 2.8). These diagrams can be applied to visualize the preferences of nearest neighbours. In this case, the spider diagram represents the rating of five items (the coloured symbols) by five users (named 'gp'). The further the item is placed from the centre of the web, the higher the user rated the item. From this, we can see that item t_4 scores the best in total and is recommended to the group. This is a very easy way to visually explain why an item is recommended to a group of users.

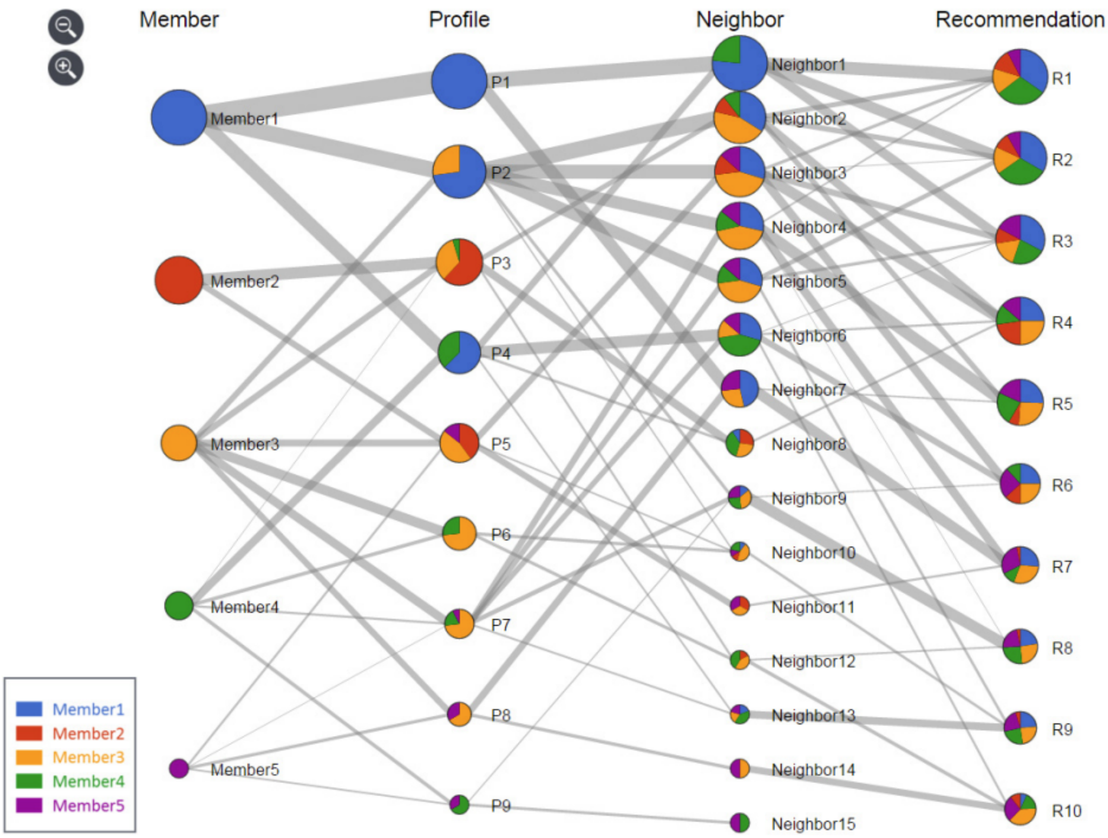


FIGURE 2.7: Graph with pie charts showing the recommendation compositions per member [72]

2.4 Fairness in group recommender systems

There are many aspects that decide a good (group) recommender system. Some examples include *transparency, recommendation quality, user control, perceived ease of use, perceived usefulness, enjoyment, decision confidence, perceived effort, trust in recommendation, satisfaction* and *satisfaction* [62]. Another important aspect is fairness [35], which we will focus on in this section.

Definition

There exist many definitions of fairness, ranging from statistical bias, group fairness, individual fairness, to process fairness, and others resulting in 21 different definitions [38].
Wang et al. [74] describe two facets of fairness, namely *Individual Fairness* and *Group Fairness*. The former is described as '*Individual fairness believes that outcomes should be fair at the individual level*' and the latter as '*Group fairness holds that outcomes should be fair among different groups*'. A more detailed description of *individual fairness* is given

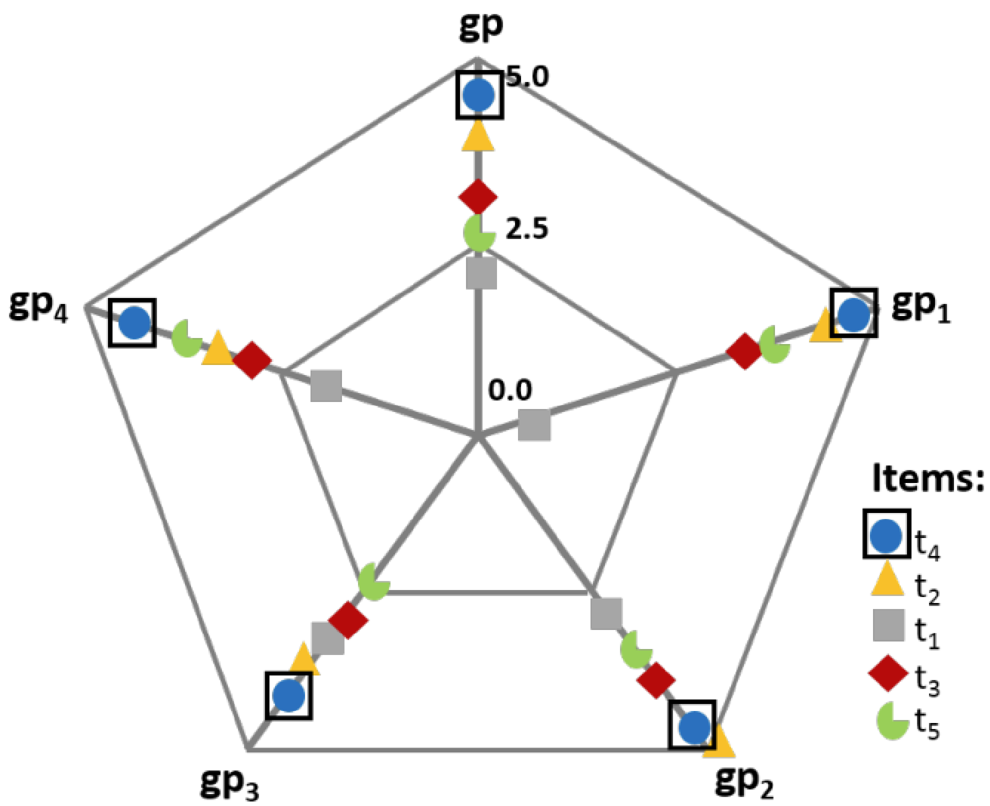


Figure 2.8: Spider diagram showing the rated items of five neighbours[26]

by Biega et al. [11]: '*The aim at treating each individual fairly by requiring that subjects who are similar to each other receive similar decision outcomes*'.

Htun et al. [35] state that '*To satisfy all users, their preferences should be taken into account equally. Especially if a set of items is recommended to a group, the system should try to balance the user utilities inside the group. Fairness in group recommender systems ensures that even the songs of users with music taste diverging from the group are included in the recommended playlist*'. They made a group music recommender system including two versions of a song list. One list contains all the selected songs based on the *time they were added*, and the second list ranked the songs based on *their dissimilarity to the individual profiles*. In their user study, they used the ResQue model [62] to test the perceived fairness. The results of the questionnaire show that the second version (based on dissimilarity-based ranking) scores higher in fairness.

This perceived fairness, however, does not always correspond with the objective measure of fairness. To measure this, Htun et al. [35] use '*the number of songs included in the final playlist over the number of selected/liked songs for each individual*'. Another way to objectively measure fairness (in the form of 'accuracy') is to check whether the

recommended items' characteristics match all users equally. Kaya et al. [41] define a top-N (list of recommendations) as 'fair' if *the relevance of the items to the group members is 'balanced' across the group members for each prefix of the top-N.*
Another study about fairness in group recommendations was done by Xiao et al. [75]. Fairness is quantified in their model by measuring the proximity of individual users' utility balance when group recommendations are provided. Moreover, they define the fairness-aware group recommendation problem as optimising user utility and fairness in the group recommendation. They prove that the problem is NP-Hard, which means that the problem is not solvable in polynomial time but can be verified in polynomial time. This demonstrates the complexity of improving fairness in group recommender systems.

2.4.1 Improving fairness

Improving fairness in group recommender systems does not always but may conflict with recommendations performance [74]. They state, 'Existing methods often improve the fairness in recommendation with a loss in recommendation performance, and many papers have revealed such a trade-off between fairness and performance.'
Since fairness can be defined in a number of ways, it can also be improved in a number of ways. Htun et al. [35] focus on two aspects of fairness: The influence of the personality of the users on the perception of fairness and the influence of the song ranking algorithm on the perception of fairness of the usability of the system. They investigated this by making an application where users can select their favourite songs, and based on this, a group playlist is recommended. The application involves two ranking algorithms: One which bases the song ranking on the moment in time that the song was added by the user and one which applies content-based filtering and aggregated predictions. Using three questionnaires (pre-test, fairness, and post-test), a user study was conducted. They conclude that the latter algorithm achieves much better fairness results, while the former algorithm is better at predicting the popularity of the suggested songs.
Based on three different explanation types (based on preference aggregation strategies, decision history, and future decision plans), Ngoc et al. [52] investigate which explanation is the most effective to increase the fairness perception of group members. They found that the explanations that considered the preferences of all or a majority of group members achieved the highest perceived fairness. Furthermore, they found a positive correlation between users' satisfaction levels with group recommendations and the perceived fairness levels of explanations. They say that '*In these explanations, higher perceived fairness levels of explanations associate with higher satisfaction levels of users with regard to group recommendations*'. From this, we can conclude that perceived fairness is one way to improve user satisfaction.
Despite this clear importance of fairness in recommender systems, not a lot of work has been done yet to better it. Dinnissen and Bauer [21] state that the large majority of works analyse the current situation of fairness in music recommender systems, whereas only a few works propose approaches to improve it. Therefore, this research focuses on finding the best design aspects to improve the perceived fairness in group recommender systems. As explained in detail in chapter 3, two different recommender algorithms and

two different visualisation types will be investigated on what influence they have on the perceived fairness of the users.

2.5 Music recommender systems

2.5.1 Motivation

Providing suggestions in the music domain is a popular application of recommender systems. The largest music streaming platforms (Spotify, Apple Music, YouTube Music, Amazon Music, etc.) all use some form of recommender system. Often times, users want to discover new music, artists or genres. A recommendation system can then provide new songs to the user, who can add them to their playlist or library if they like them. Also, new albums, playlists, and even artists can be suggested to the user. This way, users are able to discover new music, and they do not have to listen to their own registered songs all the time.

2.5.2 Examples of music recommender systems

Adiyansjah et al. [2] use the characteristics of the previous songs that the user has listened to in order to predict and offer newly suggested songs. They developed a music recommender system that gives recommendations based on the similarities of features of the audio signal. These similarities were found by using convolutional recurrent neural networks for feature extraction. The study concludes that users favour suggestions that take music genres into account, compared to suggestions that are only based on similar audio features. The interface of the application is shown in fig. 2.9.
Sometimes, it may be easier for the user to say which songs they do not like instead of which songs they do like. Chao et al. [18] use this principle for recommending music in a shared environment. As long as there is no negative feedback from anyone in the group, the system keeps playing suggested songs. When a user expresses negative feedback about a song, the system adapts and learns from it. fig. 2.10. shows the interface of the adaptive radio.
Another music recommender system is *Genius*[5]. It is based on a massive amount of user data (thanks to its former collaboration with iTunes) and uses collaborative filtering. Barrington et al. [10] compare Genius with two canonical music recommender systems: one based solely on the similarity of artists and another based solely on the similarity of acoustic content. After a study with 185 participants, they conclude that Genius provides better recommendations in general. However, when the names of the songs and artists were shown to the users, the method using artist similarities scored very well. This finding is especially present in playlist evaluation, indicating that recommender systems must be designed with applications in mind.
Pandora (shown in figure 2.11) offers an online radio station where a user builds up '*stations*' based on musical interests [14]. Users can specify whether they like each song

[5] `https://genius.com/`

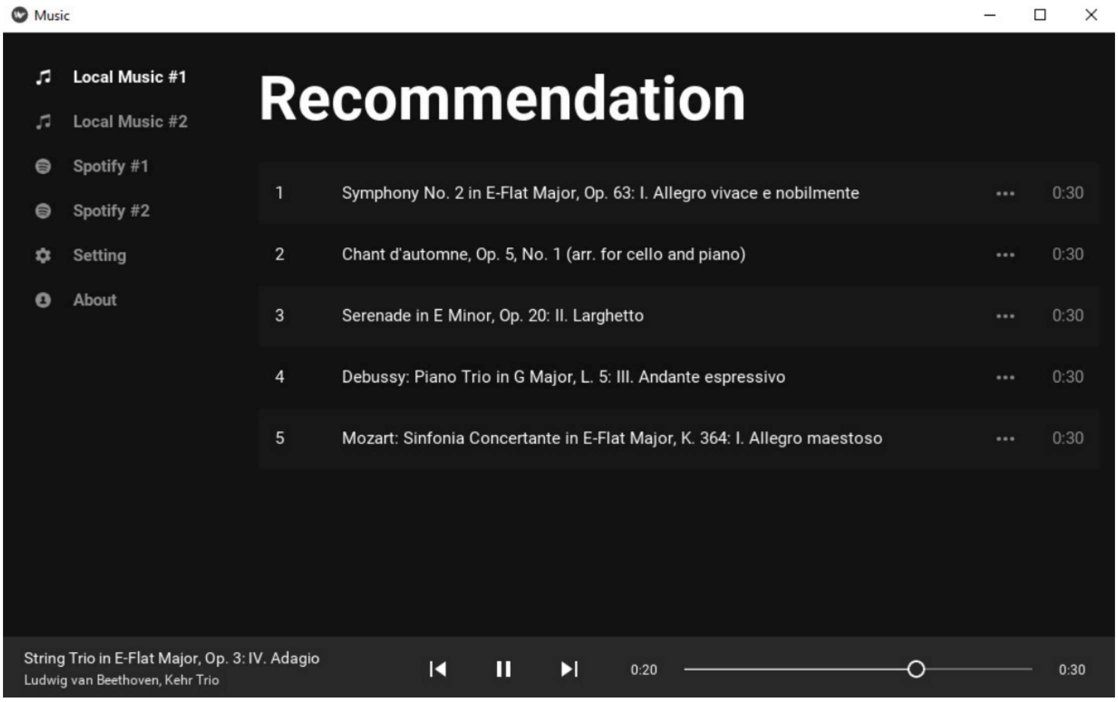


FIGURE 2.9: Music recommender interface [2]

and artist. Based on this input, Pandora can learn and play recommended songs for the users. To this day, Pandora is a popular music streaming platform with 6.3 million active subscribers.

Another advanced music recommender system is TasteWeights [14]. The main focus of this system lies in providing a very detailed visual interface that is also user-adaptive. The interface is shown in fig. 2.12. The interactive interface works as follows: on the left-hand side, the user's music library is shown. Each song is connected to the middle layer, which shows 'context'. This context is divided into three sections: Wikipedia, showing objective characteristics of the songs (like genre and prizes); Facebook, showing a list of the user's friends that have also registered the song; and Twitter, providing information from experts. The *John's music* layer, and the *Context* layer are user-adaptive. This means that the user can use sliders to adapt the weights of the parameters. These two layers are connected to the *Recommendations* layer, which provides recommended songs based on the weighted parameters from the *Context* layer. From this advanced interactive interface, the study concludes three things:

- *User satisfaction can be improved by explaining the recommendation process through a user interface.*
- *Recommendation accuracy and user experience can be improved by interaction at recommendation time.*

Figure 2.10: Interface of Adaptive radio [18]

- *Hybrid methods which merge multiple social APIs can give better recommendations than traditional collaborative filtering.*

Finally, Liang and Willemsen [46] explore the influence of nudging in a music recommender system. They do this by making a user-adaptive trade-off slider. Using this slider, users could modify the suggested songs from the most personalised songs of a music genre to the most representative songs of a genre (songs representing the mainstream tastes of a certain genre). They changed the default slider position in each of the three user groups. These positions were *the most personalised*, *the most representative*, or *the middle* position. They conclude that defaulting the slider to a certain position affects the user's usage of the slider. The interface of the genre-exploring music recommender is shown in fig. 2.13.

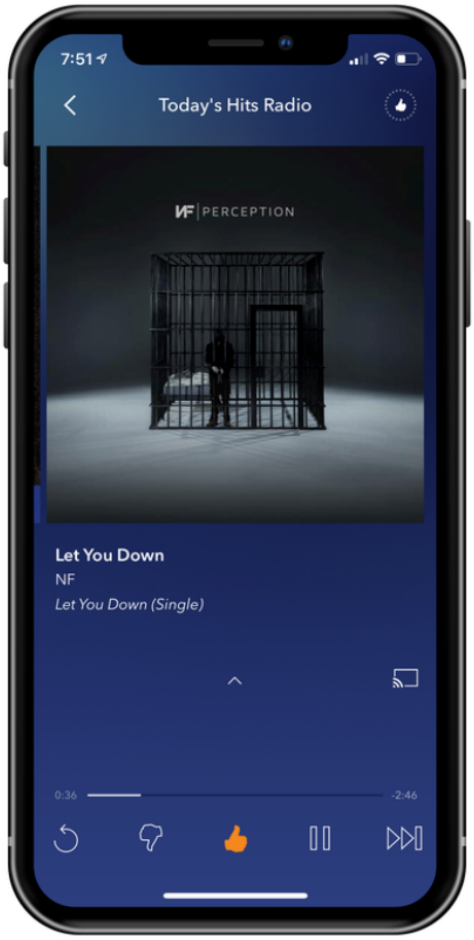


FIGURE 2.11: Interface of Pandora (source: areiner.com)

2.5.3 Spotify

The world's most popular music streaming platform, Spotify, has 350 million users and 150 million subscribers. Like many other streaming platforms, it also offers music recommendations to users. One simple example is at the bottom of every playlist. fig. 2.14 shows the bottom of a Spotify playlist, where multiple songs are suggested with the possibility to quick-add to the playlist. No information or explanation is given to the user about what these songs are based on.

Another feature of Spotify that uses a group recommender system, is the *Blend* function (shown in figure 2.15). As described in 1.1, the feature recommends a new playlist to a group of users. To understand where these recommendations come from, Spotify shows a single 'About recommendations' page. However, they do not explain a lot of aspects about the inner workings about the recommender algorithm. They only mention a couple of influencing factors: '*What you are listening to and when, the listening habits of people*

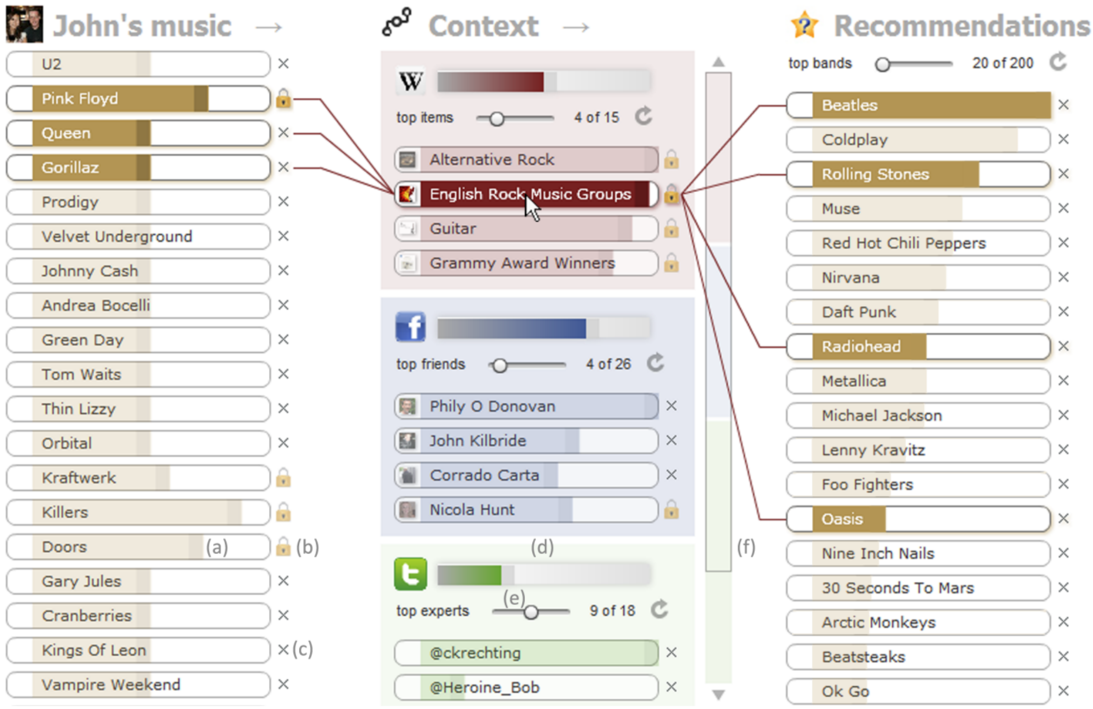


Figure 2.12: Interactive interface of TasteWeights involving the *John's music* layer, the *Context* layer and the *Recommendations* layer [14]

who have similar taste in music and podcasts (collaborative filtering), and the expertise of our music and podcast specialists'. Besides this, they also mention that commercial considerations might impact the suggestions. Furthermore, the only visual explanation that is given are two small profile pictures next to each song, indicating which of both users this song recommendation is based on. Therefore, the Blend feature leaks a lot of explainability and user control.

In a nutshell, the Blend function generates a playlist based on the general music tastes of two users. Furthermore, it is also possible to make a *Blend* with an artist. This way, it is possible to generate a playlist based on your music taste, combined with the one from the artist. When three one-on-one Blends are made between each of three friends, it is also possible to get a *Friends Mix*. This is essentially the same as a Blend playlist, but just with more than two persons.

However, the users still do not have much control over the recommended playlist. For example, users cannot add or remove songs from this playlist. Also, they cannot base the mutual playlist only on a part of their music library (e.g. a playlist), instead of all of their listening activity. Moreover, the users do not have the choice to add more influence from a specific genre, artist, or song characteristic on the recommended playlist. In conclusion, the Blend feature does not offer a lot of user control. Nevertheless, providing control to the users can lead to many advantages, as explained in 2.3. Therefore, this research

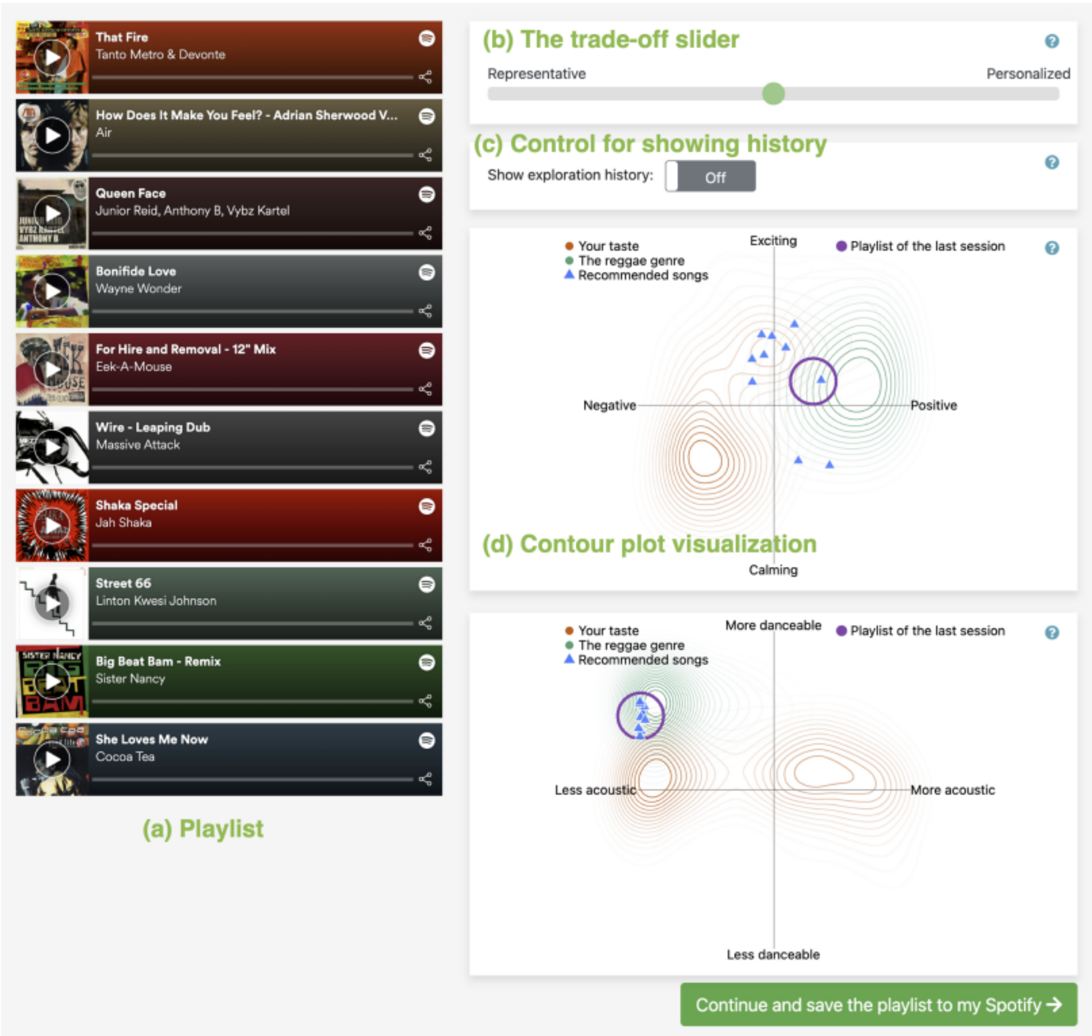


FIGURE 2.13: Interface of genre-exploring recommender with a slider to influence the recommended songs [46]

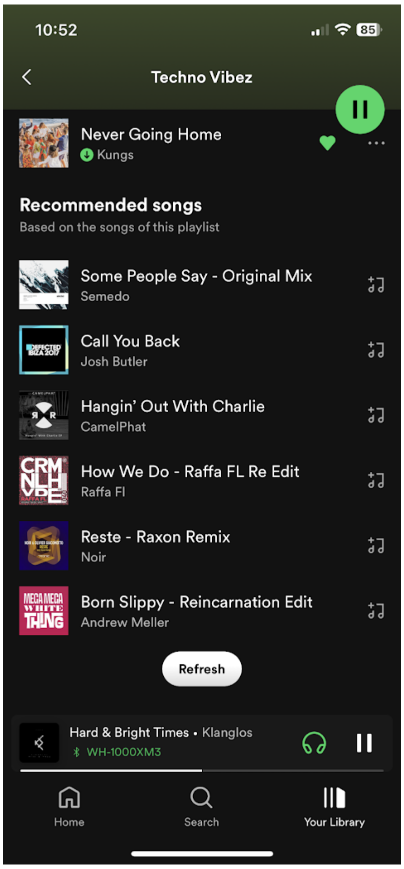


Figure 2.14: Recommendations below a playlist

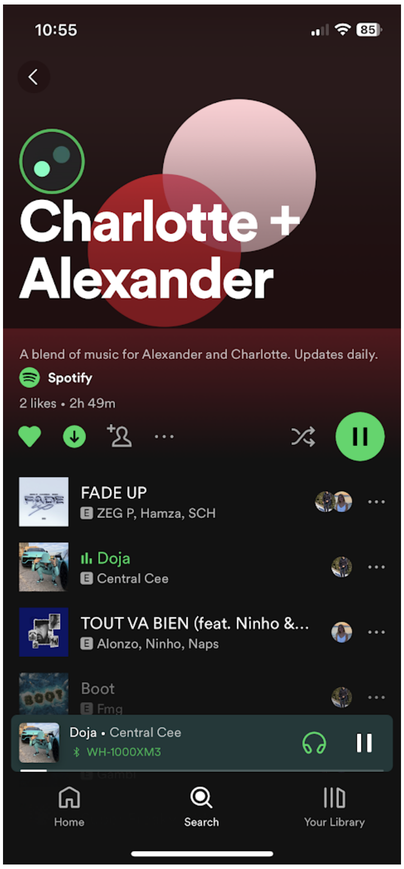


Figure 2.15: Spotify Blend playlist

will focus on a group recommender system where the users have more control over the recommendations.

2.6 Summary

This chapter has given an overview of the relevant work that has already been done for this research. The basic techniques that recommender systems use lay a solid foundation for this study. Furthermore, the focus on group recommender systems has helped to illustrate the challenges of providing suggestions to multiple users. Going deeper into explanations has provided a lot of insight into existing explanation styles. Especially visualising explanations for group recommender systems is a highly relevant topic for the playlist-based group recommendation system of this research. Besides this, fairness was one of the most important aspects of this literature overview, focusing on its definition and ways to improve it. Finally, investigating other music recommender systems has provided insight and inspiration for building a new recommender system using the Spotify Web API.

This research focuses on two aspects in the group recommender system domain. The first aspect is pre- and post-filtering. As described in 2.2.2, these types of filtering are used to modify the input data or output data of the recommender system respectively. The second aspect that will be focused on is the explanation type of the outcome of a group music recommender system. Especially the content-based and user-based explanation styles will be investigated. The influence of changing these two aspects (pre- and post-filtering and user-based and content-based explanation types) will be researched with respect to the perceived accuracy, fairness, explanation, behavioural intentions and ease of use in the recommender system.

Chapter 3

Problem Statement

3.1 Motivation

As mentioned in the literature study 2.2.2, pre- and post-filtering can have a big influence on the performance of recommender systems. However, the effect of pre- and post-filtering in group recommender systems has, to the best of our knowledge, not been investigated yet. Nevertheless, when a group of users uses a group recommender system, it is often the case that pre- and post-filtering are being used. For example, online mobile advertising or flight bookings for families often make use of group recommender system where pre- and post-filtering are applied [20]. Pre-filtering is applied when only a selection of the data of both users is used for the recommender algorithm. The individuals' data is combined to generate a mutual recommendation for the group of users. Furthermore, post-filtering can be used by generating a group recommendation and modifying the final recommendations based on the individual data of each of the users or by using a specific context to produce contextualised recommendations on top of what a traditional recommender system would suggest [20]. An example of this is showing to a group of users only a selection of a final advertisement recommendation list that includes the mutual interests of both users. This raises the interesting question of the influence of these filtering methods on not only the performance of the group recommender system but also on perceived fairness and user satisfaction.

Additionally, the use of explanations in recommender systems has been shown to have a significant influence on perceived transparency, trust, efficiency and satisfaction [67]. A relevant research question would thus not be if these explanations have an influence on the perceived accuracy and fairness (this has already been proven by many papers mentioned in 2.3), but what type of explanation has the biggest influence on the perceived accuracy and fairness (the precise meaning of these measure quantities is explained in section 3.2). As explained by Kouki et al. [44], the five big explanation styles consist of user-based, item-based, content, social and item popularity explanation styles. This brings up the captivating query of what the influence would be of these explanation styles on the perceived accuracy, fairness, explanation, user satisfaction, behavioural intentions and ease of use in a group recommender system.

This motivates two questions regarding the perceived accuracy, fairness, explanation, user satisfaction, behavioural intentions and ease of use in group recommender systems:

- How do pre- and post-filtering in a group recommendation system influence the perceived accuracy, fairness, explanation and behavioural intentions?
- How do different visual explanations of group recommendations affect the perceived fairness, accuracy, explanation, behavioural intentions, and ease of use?

3.2 Evaluation factors

In this section, each of the concepts of perceived accuracy, fairness, explanation, behavioural intentions and ease of use in a group recommender system are explained in more detail.
Perceived accuracy: The recommendations delivered by a group recommender system can, just like in an individual recommender system, be judged on the user's perceived accuracy. Felfernig et al. [27] define accuracy in a group recommender system as three metrics: '(1) classification metrics that evaluate to which extent a recommender is able to determine items of relevance (interest) for the user, (2) error metrics that evaluate how well a recommender predicts ratings, and (3) ranking metrics that help to evaluate how well a recommender predicts the importance ranking of items'. Metrics (2) and (3) are more objective definitions of accuracy, but they might not always overlap with the perceived accuracy for the user. Therefore, we rather use definition (1) in this research. This way, it is the user who decides whether the recommended items are relevant, and thus whether the system is perceived as accurate.
Perceived fairness: Unlike perceived accuracy, this is a totally different concept: It does not necessarily take into account how good each of the final recommendations is, but rather on what scale each of the recommendations is based on each of the user's interests. Felfernig et al. [27] define fairness in group recommender system as 'the extent of imbalance between group member specific utilities.'. Htun et al. [35] explain that when the recommendations are clearly and evenly based on each of the different users, the perceived fairness will be higher than when the recommendations are only based on one user of the group of users.
Behavioural intentions: The meaning of this concept speaks for itself: It indicates whether the user would take action after using the application. The behavioural intentions towards a system are associated with the system's ability to influence users' decisions to use the system and engage in purchasing or using the recommended outcomes. One criterion of this concept includes the intention to introduce this system to friends [62].
Perceived explanation: According to Tintarev and Masthoff [69], a popular definition of explain is 'to justify'. Therefore, they explain that '*an explanation can be an item description that helps the user to understand the qualities of the item well enough to decide whether it is relevant to them or not*'.
Ease of use: Finally, Ozok et al. [54], state that interface issues such as page layout and navigation are the most important factors relating to the overall ease of use and

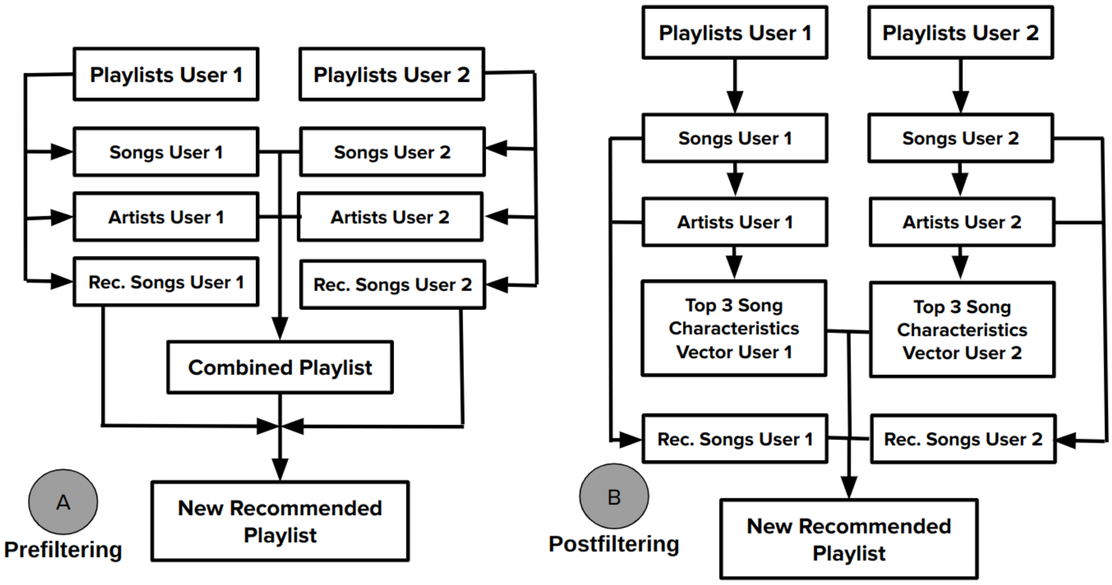


FIGURE 3.1: Pre- and post-filtering in a playlist-based context

perceived usefulness of a system. Since the application flow will be identical for both types of algorithms, the ease of use aspect is only investigated for research question 2, concerning the visualisation types.

3.3 Research question 1

How do pre- and post-filtering in a group recommendation system influence the perceived accuracy, fairness, explanation and behavioural intentions?
As stated in 2.2.2, contextual pre-filtering changes the input data of the recommender system by modifying the input data before the recommendation happens. Contextual post-filtering modifies the output data after the recommendation has already happened. This thesis investigates the influence of these two types of filtering in the context of a playlist-based group recommender system.

Applying these two filtering strategies to a playlist-based music recommendation is possible in two ways, as shown in figure 3.1. The sketch A illustrates pre-filtering. The playlists of both users are first combined, from which a playlist is formed based on both users' interests. This playlist is input for the recommendation algorithm, which outputs a final recommended playlist for both users. The sketch B illustrates post-filtering. First, the playlists of both users are used individually to generate recommendations for both users. Afterwards, these two recommendations are combined into one new recommended playlist based on both the users' interests.

By implementing both filtering methods, it is possible to investigate whether they affect the perceived accuracy, fairness, behavioural intentions, explanation and ease of use in a user study. The exact course of this user study is described in detail in section 6.

3.3.1 Hypothesis

The hypothesis for the first research question is: *The pre-filtering algorithm will score better on perceived fairness and accuracy but lower on behavioural intentions and perceived explanation.* This could be the case because in pre-filtering, both of the users first combine their selected playlists, resulting in a list of songs and artists that they both know for sure. This could lead to a feeling of both of the users that the playlist certainly keeps both of their interests into account. Therefore the perceived fairness and accuracy may be higher. However, the second way of recommending (using post-filtering) first generates a new recommendation for each of the individual playlists of the users, and later joins them into one list. Since the joining of these lists can be done in a transparent way (described in 3.3), the perceived explanation and behavioural intentions could be better.

3.4 Research question 2

How do different visual explanations of a group recommendations affect the perceived fairness, accuracy, explanation, behavioural intentions and ease of use?

As the literature study 2.3 has already shown, more explanations can positively affect the perceived fairness, accuracy, user satisfaction, and trust in recommender systems. A more interesting study would be to compare two different ways of visual explanations rather than comparing more or less explanations. This is something that will be investigated specifically for a group recommender system. Since Kouki et al. [44] concluded that people prefer to see at most three to four explanation styles and that textual explanations are ideal, these design aspects were taken into account. From the five explanation styles that the paper described (shown in table 2.2), two of them were used as inspiration for the visualisations. More precisely, the user-based and the content-based explanation styles will be investigated about what influence they have on the perceived fairness, accuracy, explanation, behavioural intentions and ease of use.

3.4.1 Visualisations

In this section, the two visualisations will be discussed in more detail. Two sketches of the visual explanation types are shown in figure 3.2.

The first visualisation makes use of a user-based explanation. This means that the explanations are based on the influence of the users on the recommendations. In the context of a playlist group recommender application, the group of users are provided with an explanation about the final playlist based on data and characteristics of the music interests of each of the individual users. In sketch A, the match percentages with the selected playlist(s) of both users are shown. Therefore the users get information about who this recommendation is (mostly) based on.

This explanation type could affect the opinion of the users in many ways, for example on their perceived fairness of the system. This is because they can use the user-based

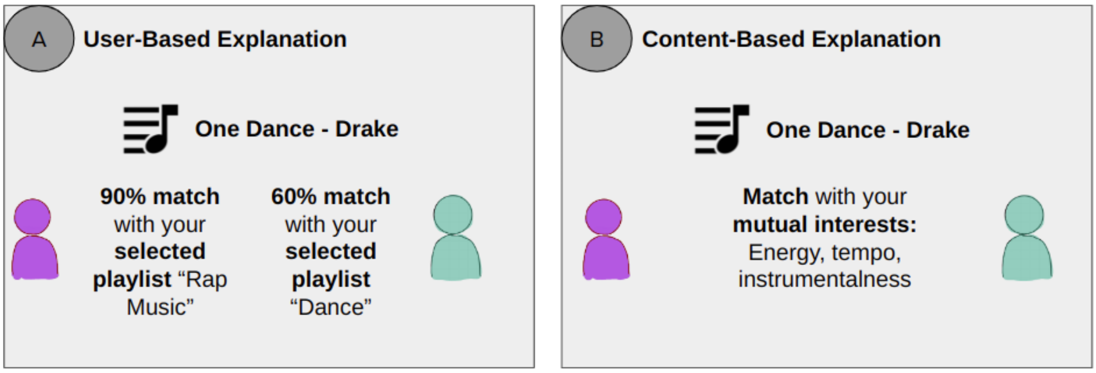


FIGURE 3.2: (A) User-based and (B) Content-based explanation styles

information to judge if the recommended song is based on the interests of each of the users.
The second visualisation uses a content-based explanation style, which means the explanations are based on the influence of the characteristics of the recommended items on the recommendations. For this study, the visual explanations for the final playlist provide the users with information about the musical characteristics of the specific recommended song. Sketch B shows three characteristics of the recommended song to the users. Therefore the users get information about the aspects of the recommended item itself and the explanation is content-based.
Because the users will get more information about the attributes and facets of the recommended items, this content-based explanation could affect the perceived accuracy, fairness and other aspects of the users. As the users get to see a direct match indication of the certain song characteristics with their own musical interests, it may for example influence their perceived accuracy. This is exactly what the second research question will investigate.
A more clear and more detailed design of the explanation styles will further be discussed in 4.2.

3.4.2 Hypothesis

The hypothesis for the second research question is: *The visualisation with the user-based explanation style will score higher on the perceived fairness and explanation, while the visualisation with the content-based explanation style will score higher on perceived accuracy and behavioural intentions.* The reason that the user-based explanation could have an advantage w.r.t. the perceived fairness and explanation, is because users could use the explanation to make a direct link with each of the users' interests. Therefore they can immediately judge that the group recommender system provides a good and fair explanation because it is visually shown in the explanation that each of the users' interests was taken into account. The content-based explanation, however, could score higher on the perceived accuracy and behavioural intentions. This could be the case because the users will see direct information about the recommended items. Therefore they can judge immediately if the recommendations are accurate, taking into account their own personal interests. When the recommendations are indeed relevant to their personal music taste, individuals may be inclined to use the recommender again in the future.

Chapter 4

Application Design

To conduct the research, a web application was built. This includes the implementation of two different recommendation algorithms (using pre- and post-filtering) and two different visualisations (using user-based and content-based explanation styles). Furthermore, the web application is designed to be easy to use, intuitive and overall satisfy the users in its usage.

4.1 Initial Questionnaire

To design a user-friendly web application, the desires of the users were researched beforehand. As mentioned in 2.5.3, Spotify has its own Blend feature that generates a new playlist for two users. As this is one of the most famous group music recommender systems, an initial questionnaire was created to obtain a clearer depiction of the thoughts of the users. Through this approach, it was possible to become acquainted with the weaknesses of the Spotify Blend functionality. Improving these weaknesses could then be the strength of the new web application.

The initial questionnaire and the results are shown in appendix C. From the answers of 31 participants, it was concluded that 50% of the participants had already used the Spotify Blend function, and only 30% of them liked it. There was a clear need for influence on the recommended songs and more explanation, as one participant requested: '*Possibility to see on which of the users' playlists the song in the recommended playlist is based on*'.

It was decided that the input of the music recommender system would consist of a number of selected playlists of each user. This way, the users were given the control to make their own playlist selection and thus had some influence on the input of the recommender system. Providing a way of control to the users of an recommender system has been proven to be effective by Tsai and Brusilovsky [70].

The initial questionnaire provided a clear image of the features that the web application should provide. There should be enough overall explanation about the recommendations and a certain degree of control over the recommendations.

4.2 Think-aloud studies

4.2.1 Iteration 1: Application flow

To decide the exact application flow and the goal of each individual web page, some initial design layouts were made. During a first think-aloud study, these different versions and drafts were shown to potential participants. Every time, the participant had to use the application completely autonomously, without any external help. At the same time, the participant had to say aloud their thoughts on the use of the application. This way, the think-aloud study was executed without external influence. The design of the first prototype is shown in figure 4.1.

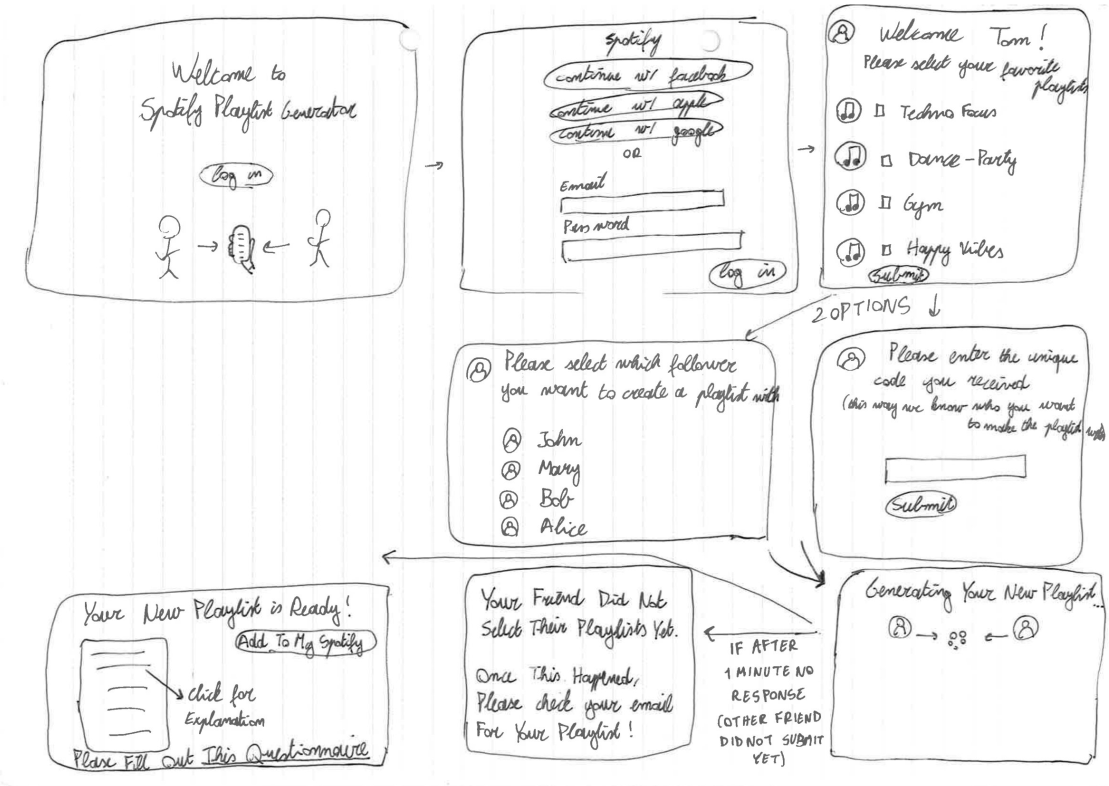


Figure 4.1: Think-aloud study 1: Application flow

The homepage is a very simple page, displaying an image and the title of the application with a button to connect to Spotify. The authorisation page is the standard page of Spotify for Developers that is used for authorising third-party applications (as described in detail in section 5.3). It asks the the users' permission to obtain their Spotify data, by letting them fill out their Spotify username and password.
After this, their playlists will be retrieved and shown on the playlist selection page. This page consists of a welcome to the user (with the user's name and profile picture) and a list of their Spotify playlists.

After the user made their playlist selection, two options of verification pages with two different strategies were considered for creating a connection between the two users. The first option gave the first user a list of his/her Spotify followers, saying: '*Please select which follower you want to create a playlist with*'. The user could then select who to make a new playlist with. The second option provided an input field with the text: '*Please enter the unique code you received (this way, we know who you want to create a playlist with)*'. Here they would have to fill out the unique code they received for the research (described in section 6).
Then, two things can happen. Since the web application is designed for groups of two users, they both have to go through the playlist selection process. The first option is when the first user entered the code. He/she will not get to see the final playlist yet, because the playlist selection from both users is needed to create the final recommended playlist. For this reason, the first user will be redirected to the '*Waiting for your friend*' page. This page informs the user that he/she successfully completed all the steps and that the web application is waiting for the second user to complete the steps as well. The second option happens after the second user went through the web application pages as well (and the recommender algorithm has all the data to generate the new playlist). He/she will then immediately see the final playlist page. Also, the first user who is still on the waiting page, will receive an email with a link to the final playlist page.
Now that both of the users can see the final playlist page, it would be helpful for them to listen to the final playlist as well. For this reason, a preview button is shown next to each of the recommended songs in the final playlist. Using this button, the users can listen to a 30-second preview of each of the recommended songs.
When the users are ready inspecting the final playlist and the explanatory visualisations, they can click on the '*Add to my Spotify Account*' button. When this button is clicked, the final recommended playlist will be added to their personal Spotify account, such that they can listen to it even after using the web application.

Results

From the think-aloud study, many things were learned. First, the participants would often click on the image instead of on the '*log in*' text on the first welcome page. This was improved by making the button larger and more notable. Secondly, the authorisation part was found to be very clear. One participant complained about having to find his Spotify password. This problem was, however, hard to work around. The playlist selection part was also clear to all users. After this, most users preferred the '*Please select which follower you want to create a playlist with*'-style. Nevertheless, in the implementation phase of the application, it was found that it is not possible to retrieve the list of followers using the API. For this reason, the second option of using unique codes was used.

Final version and extra features

The final version of the homepage is shown in fig. 4.2. After this, an information page was added to give the user more information about the goal of the research. The information

Figure 4.2: Homepage

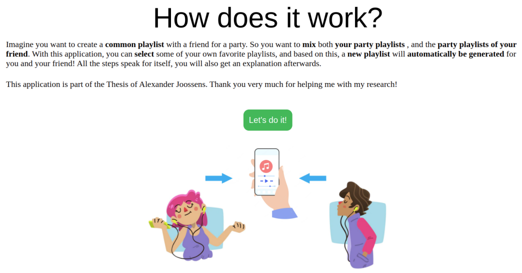


Figure 4.3: Info page

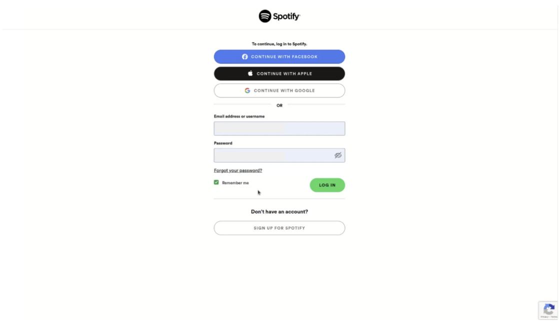


Figure 4.4: Login page

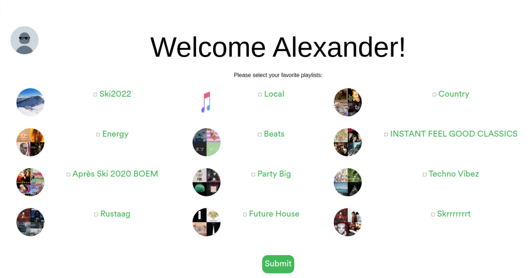


Figure 4.5: Playlist selection page

page is shown in fig. 4.3. After this, the user gets to see the login page, the playlist selection page, the code page and the waiting page, shown in figures 4.4, 4.5, 4.7 and 4.8. On the playlist selection page, it could be possible that the user does not exactly remember which songs are present in each of their playlists. For this reason, it was designed that the user can click on the name of any of their playlists, and then be redirected to the playlist page. The playlist page contains a list of all the songs for a specific playlist, with their artists next to them, a preview button is placed next to each song such that the user can listen to it and recall the song. Since not every song on Spotify supports a preview link through the API, some songs do not have the preview button. A big green '*Back*' button is shown on the top of the playlist page to return to the playlist selection page. The playlist page is displayed in figure 4.6.
Also the waiting page was redesigned. A new idea of providing the waiting user with a redirect button to the final playlist page seemed more convenient. This way, the first user does not need to go check their email. Instead, he/she can just keep waiting on the waiting page. Once the second user has finished the process as well, the first user is able to click on a button (saying '*Click here when your friend is ready*'), and gets redirected to the final playlist page.

4.2.2 Iteration 2: Final playlist page

The final playlist page should show the new recommended playlist to the group of users. Furthermore, it should contain a user-based and content-based explanation about the

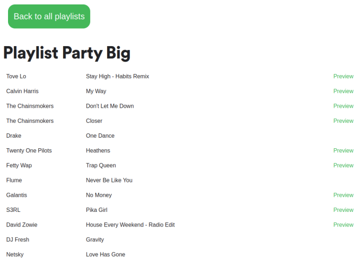


FIGURE 4.6: Playlist-page

FIGURE 4.7: Code page

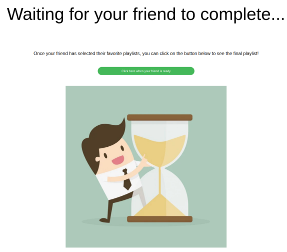


FIGURE 4.8: Waiting page

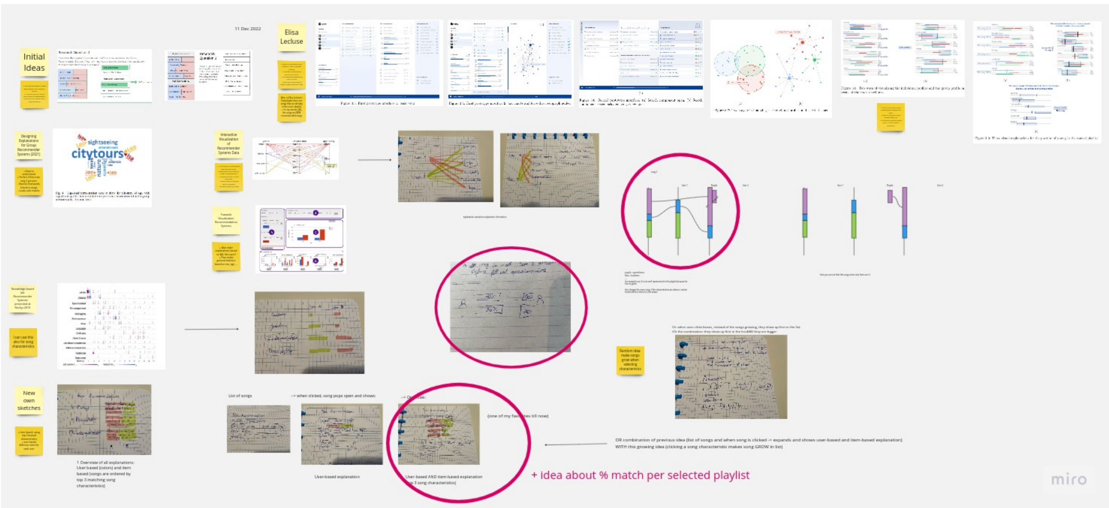


FIGURE 4.9: Initial visual explanation sketches

recommended song list. This is what the second research question focuses on. As shown in figure 4.9, many initial sketches were made for the visual explanation. From all of these sketches, two were selected and worked out in more detail. A second think-aloud study was then done for each of the two sketches. For each sketch, three slightly different variations were designed. The three variations (A, B and C) of each of the two selected sketches are shown in figure 4.10 and figure 4.11.

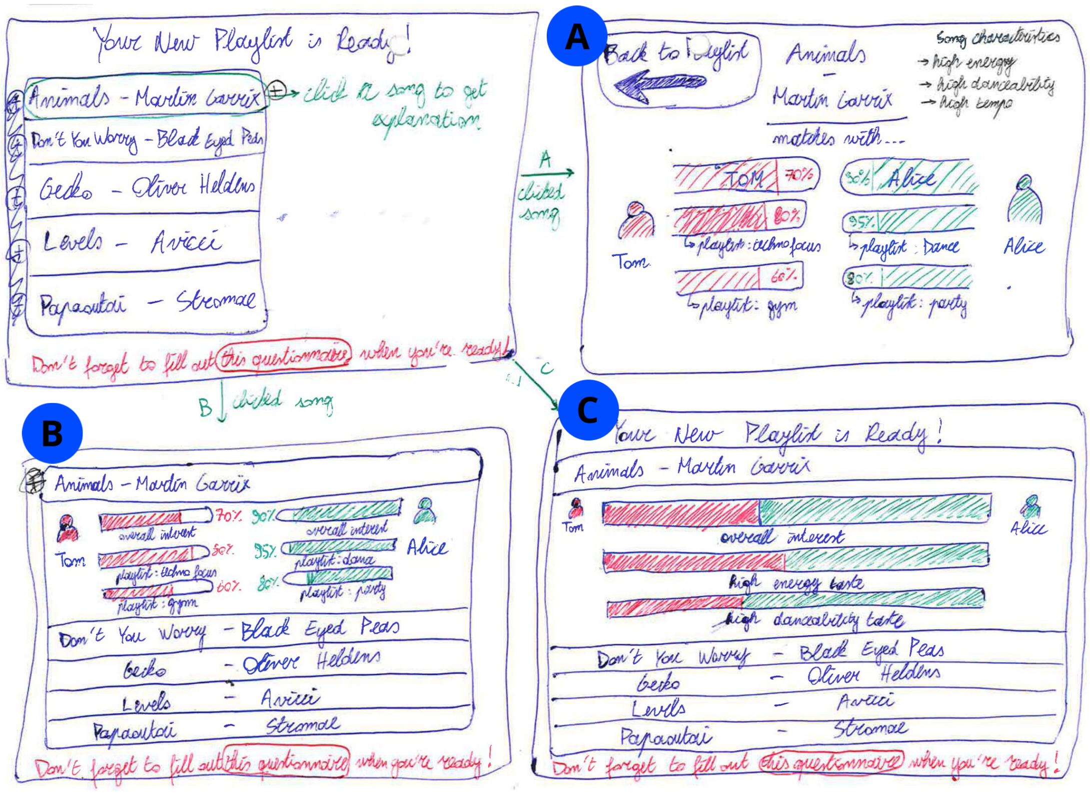


FIGURE 4.10: Think-aloud study 2: Three variations of sketch 1

The three variations (A, B and C) of the first sketch (fig. 4.10) provide the users with an overall match percentage that indicates the match of the song with their music taste. In variation A and C, the users also get information about the top three song characteristics that match both of their profiles. It is, therefore, a mix of a content-based and a user-based explanation since the information is about the characteristics of the recommended items themselves but also about the relation between the two users. In variation B, these song characteristics are not shown, but instead, there is more information given about the match percentage of the song with each of their selected playlists. It is, therefore, a purely user-based explanation. No information about the particular recommended item is given.

Results

Most of the users preferred accordion-style designs. This way, they did not have to click a '*Back to playlist*' button all the time to go back to the playlist page. Variation B was the favourite of most users since they liked the fact that you could see the individual playlist match percentages for each of the songs. With these percentages, they could clearly see on which of the playlists of both users the new song recommendation was based.
Opposed to this, the three variations (A, B and C) of the second sketch (figure 4.11)

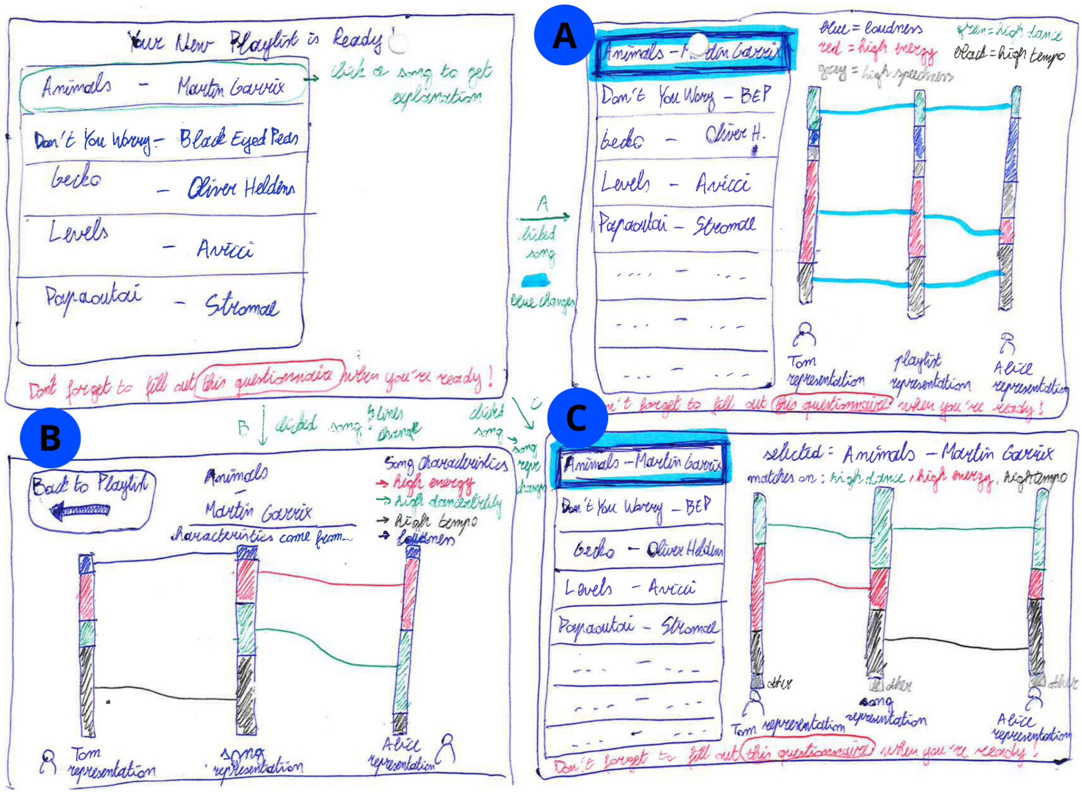


FIGURE 4.11: Think-aloud study 2: Three variations of sketch 2

were not well appreciated by the participants at all. They stated that it was too complex and had a lot of trouble understanding the meaning of the blue lines between the vertical taste representation bars. For this reason, one more think-aloud study was done with three more variations.

4.2.3 Iteration 3: Final playlist page

Figure 4.12 shows the three variations (A, B and C). Variations A and B are very similar. On the right-hand side, variation A shows the title '*Overall match with playlist*', while variation B shows the title and artist from the selected track again. Variation C uses an accordion element to show the final playlist.

Results

From this final think-aloud study, it was concluded that variation C was the best since it included the accordion-style list. The users preferred this layout since the found the navigation easier to use and there was no left-hand side and right-hand side to both look at.

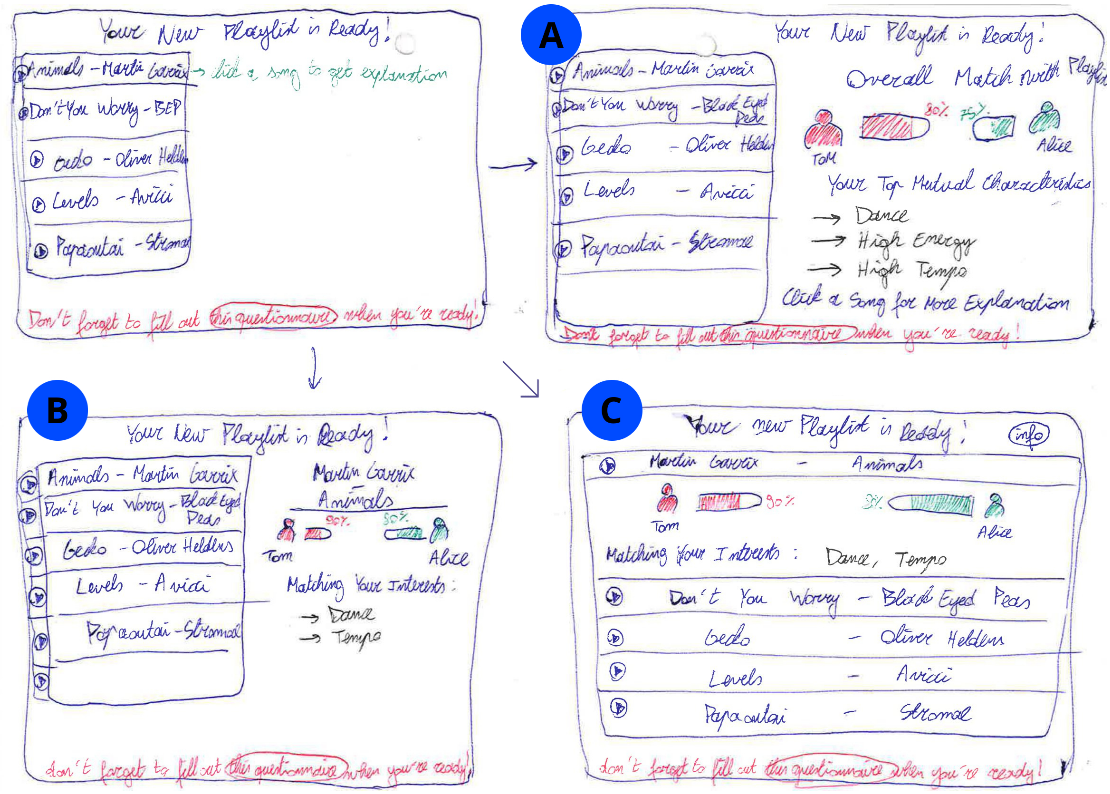


Figure 4.12: Think-aloud study 3: Three more variations

4.2.4 Conclusion

To conclude, the two visualisation designs that were selected were variation B of the first sketch of the second think-aloud study (figure 4.10 B), and variation C of the third think-aloud study (figure 4.12 C). In these two visualisations, the red colour was still changed to purple since the combination of green and red could be confusing. Users could possibly interpret these two visual parts as good and bad. Furthermore, the combination of green and purple is colour-blind-friendly.

Another feature that was still added (in the right top corner) is the '*Add playlist to my Spotify account* button. With this button, users can easily save the newly created playlist for later. Furthermore, the amount of users that saves the playlist could be recorded this way, and later be used in the data analysis 7.2.

As mentioned before, the first visualisation provides a purely user-based explanation, namely the match percentages per selected playlist. The second visualisation provides a content-based explanation, namely the characteristics per song that are matching with the users' interests. The second visualisation also contains the overall match percentage per user, which is a user-based explanation. However, since this design was so popular, it was kept and even added to the user-based explanation as well. The final versions of the visualisations are shown in figure 4.13 and 4.14. In chapter 6 will be explained how these

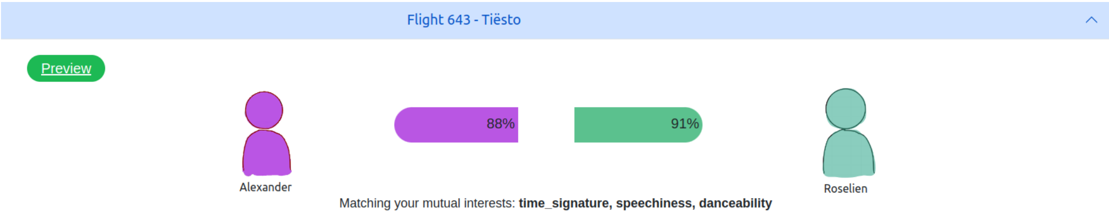


FIGURE 4.13: Visualisation 1: Content-based explanation

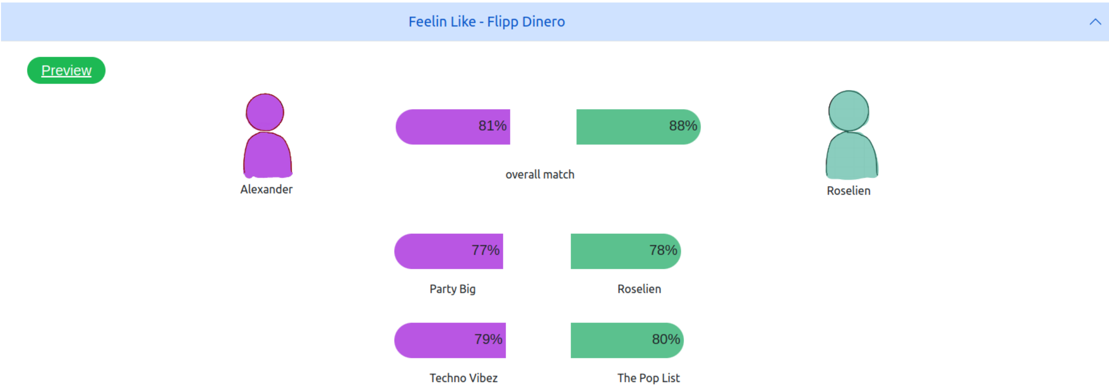


FIGURE 4.14: Visualisation 2: User-based explanation

two different visualisations will be used in the user study to answer the second research question.

4.3 Pre- and post-filtering algorithms

Two algorithms were designed which take the selected playlists of the users as input, and generate a new recommended playlist as output. One algorithm uses pre-filtering, the other uses post-filtering. The final recommended playlist always contains twenty songs.

4.3.1 Pre-filtering algorithm

As explained in 3.3, the pre-filtering algorithm first combines the two playlist selections, after which it generates a final recommended playlist. The pre-filtering algorithm is shown in figure 4.15. To combine the selected playlists, four stages were followed:

1. Vector generation
2. Mutual song selection
3. Mutual artist selection
4. Extra recommendations

Vector generation stage

In the *vector generation* stage, the algorithm first loops over all of the selected playlists from each user to retrieve the songs and their artists. Using the Spotify Web API 5.3, the audio features from each song are retrieved and combined into one vector per song. These audio features contain the following song characteristics: *acousticness, danceability, energy, loudness, mode, speechiness, instrumentalness, liveness, tempo, duration, time signature*. The meaning of these characteristics are shown in table 4.1.
Then, the vector entries are normalised to values between 0 and 1. After this, the average characteristic values from all songs are calculated per playlist. The average values together form a *playlist vector*. Then, for each of these *playlist vectors*, the average value per vector entry is calculated. From this, a global average vector is calculated, called the *user vector*. This vector represents the overall music taste of the user. From this vector, the three song characteristics with the highest value are saved in a *top-3-characteristics* vector. This vector represents the three most fitting audio features for this user's music taste. Hereupon, for each *song vector* and each *playlist vector*, a cosine-similarity score is calculated, resulting in match percentages for each song and each playlist with the overall user's music taste. These match percentages may later be used on the visualisation page (depending on the visualisation type that will be chosen). This process is done for each of both users.

Mutual song selection stage

In the *mutual song selection* stage, the algorithm looks for all of the songs that are present in the playlists selected by user 1, as well as in the selection of user 2. From these songs, seven are added to the final recommended playlist.

Mutual artist selection stage

In the *mutual artist selection* stage, the algorithm looks for all of the artists that are present in both of the users' playlist selections. The algorithm then acquires all the songs from all of the mutual artists in both of the playlist selections. The mutual songs and the songs with the mutual artists make up a combined user playlist.

Extra recommendations stage

Finally, there is a *extra recommendations* stage which acts as a backup. When the combined playlist still does not contain enough songs, for example, when both of the users selected completely different playlists with almost no overlapping songs or artists, the final playlist will complement itself with an equal amount of song recommendations based on the playlist selection of user 1, and playlist selection of user 2. The recommended songs for both users are obtained using the *Get Recommendations* method of the Spotify Web API 5.3. This method requires three input parameters, which it uses to generate the recommendations. The selected songs, artists and genres from each user were used for these input parameters. The inputted genres were retrieved from the *Get Artist* method

Audio feature	**Meaning**
Acousticness	A confidence measure from 0.0 to 1.0 of whether the track is acoustic
Danceability	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity.
Energy	Energy represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
Loudness	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.
Mode	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived.
Speechiness	Detects the presence of spoken words in a track. The more exclusively speech-like the recording, the closer to 1.0 the attribute value.
Instrumentalness	Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal".
Liveness	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live.
Tempo	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
Duration	The duration of the track in milliseconds.
Time signature	An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure).

TABLE 4.1: Audio features and their meaning from the Spotify Web API

from the Spotify API, which returns an object containing the relevant genres for a certain artist. The mutual songs from both users' recommended song lists are then selected and added to the combined playlist. Together, this forms the new final recommended playlist. Finally, the final list of recommended songs is shuffled in a random order using the *Fisher-Yates algorithm*[1].

Final playlist

The final playlist contains mutual songs, songs from mutual artists, and mutual recommended songs from both users. The songs are shuffled in a random order to avoid a final playlist where the first half contains mostly recommended songs for the first user, and the second half contains mostly recommended songs for the second user. The songs from mutual artists (when not empty) ensure that user 1 will be able to explore new songs of an artist they are interested in, which are already present in the playlist selection of user 2, and vice versa. The algorithm is made in such a way that the influence of both users on the final song list will always be balanced. One exception can be made: When for example user 1 has a lot of songs from one particular artist in their playlist selection (e.g. ten songs), and user 2 only has one or a few songs from this artist in their playlist selection (e.g. only two songs). The algorithm will then add these twelve songs to the final playlist, since they have a mutual artist. This way, the final playlist contains more songs from user 1 than from user 2. This however was not expected to happen often. Furthermore, when it would happen, it would also not be a big problem since the first user is probably actually interested in those mutual artist songs.

4.3.2 Post-filtering algorithm

For the post-filtering algorithm, a very different approach was designed to create the final recommended playlist. The post-filtering algorithm is shown in figure 4.16. The second way of recommending happens by first generating a list of individual recommendations per users, and combining these lists afterwards. Two stages were followed to obtain the final playlist:

1. Vector generation
2. Individual song recommendations

Vector generation

This stage is identical to the *vector generation* stage from the pre-filtering algorithm.

Individual song recommendations

The *individual song recommendations* stage is similar to the *extra recommendations* stage from the pre-filtering algorithm. However, there is a slight difference: First, the *user vectors* are compared and the characteristics with the most similar values are saved.

[1] The Fisher-Yates algorithm, also known as the Knuth shuffle, is a popular algorithm used to shuffle the elements of a list in a random order. It does this by iteratively swapping each element with a randomly selected element from the remaining unshuffled portion of the list.

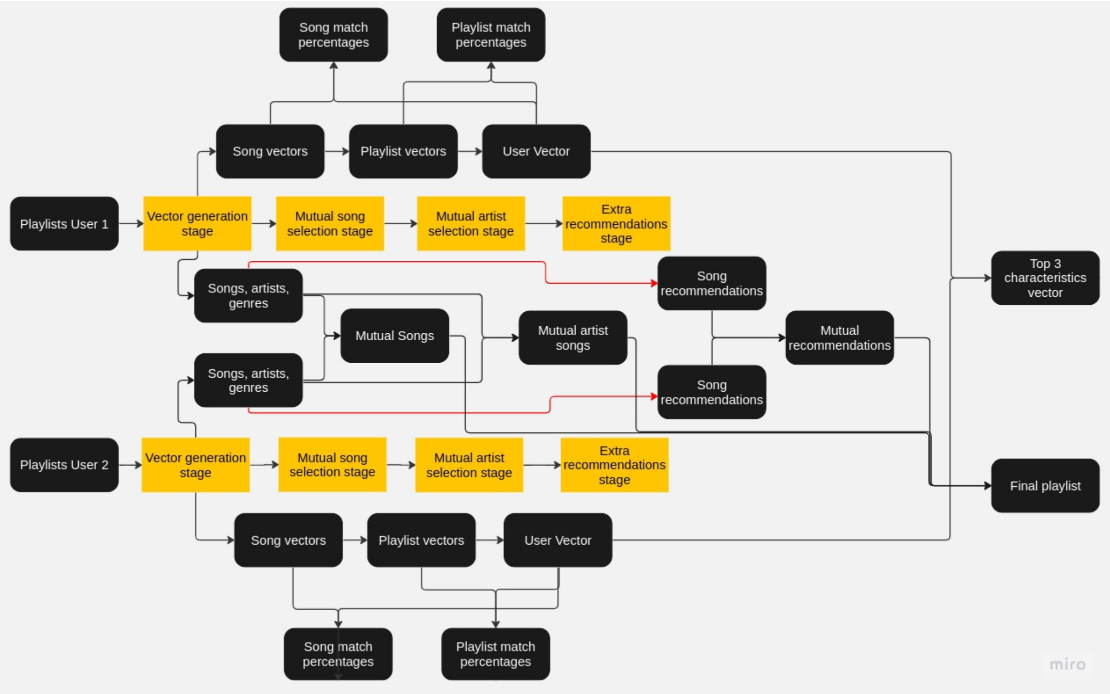


FIGURE 4.15: Pre-filtering algorithm (red arrow indicates Spotify recommend method)

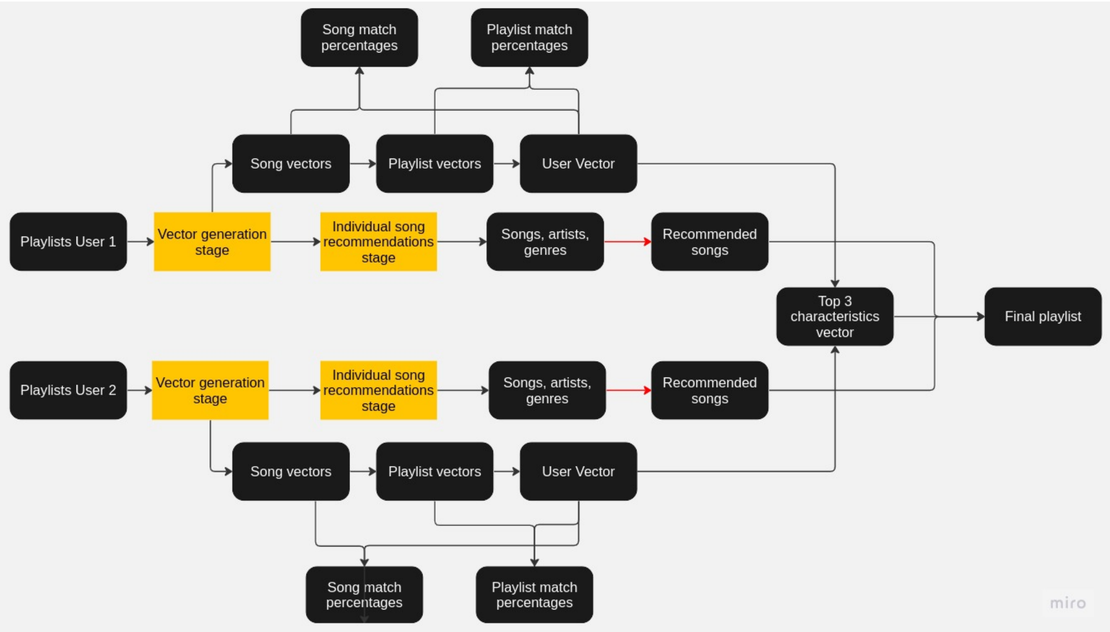


FIGURE 4.16: Post-filtering algorithm (red arrow indicates Spotify recommend method)

Based on these characteristics, the individual song recommendations (generated in the same way as in the *song recommendations* stage from the pre-filtering algorithm) from both users are filtered. Only the twenty recommended songs that match the most on these mutual characteristics are kept. Finally, the final list of recommended songs is shuffled in a random order using the *Fisher-Yates algorithm*.

Final playlist

This post-filtering algorithm first generates individual song recommendations for each of the users. Afterwards, it produces the final recommended playlist by filtering the individual recommended songs, based on each of the users' music tastes (the *user vectors*).

4.3.3 Differences between the algorithms

One of the biggest differences between both algorithms is the amount of novelty of the final song list. In the pre-filtering algorithm, there is a big chance that both of the users already know some of the songs and artists. This is because the intersection of the selected playlists of the users is taken in the beginning. In the post-filtering algorithm however, the final songs list will almost certainly contain only novel songs for both users. This is because in the first stage of the second algorithm, a list of new recommended songs is generated per user (based on the most prominent song characteristics).

Chapter 5

Application Implementation

This chapter consists of three parts about the implementation of the web application: Backend 5.1, Frontend 5.2 and Spotify Web API 5.3. All the source code is available on GitHub. The application is easily reproducible by following the README file. A short demo video of the use of the web application is available on YouTube.

5.1 Backend

5.1.1 Express.js

As a web application framework, Express.js was used. This free and open-source software can be utilised to build RESTful APIs with Node.js. It provides an easy routing system and allows a fast server-side development.

5.1.2 MySQL Database

To store the data from the web application (a unique ID of the database entry, the IDs and names of their selected playlists, the songs and artists in them, the unique code they entered, the user characteristics vectors, a list of new personal recommended songs, etc.) MySQL was used. This relational database management system is widely recognised for its robustness, performance, and user-friendly nature. Furthermore, it works well together with the Express.js framework because both technologies are JavaScript-based. As a graphical user interface, phpMyAdmin was used.

5.1.3 Docker

Since a lot of libraries and packages need to be installed to run the web application, and since the web application had to be run both locally, as well as on the *picasso* server from the KU Leuven, it was decided to dockerise the application. Docker is an open platform to easily package and run applications in an isolated environment called a container. Because the application consisted not only of a web application part (the Express.js app), but also of a MySQL database, two docker containers were needed: one for the web part, and one for the database. Docker Compose made it then possible to host both of these isolated environments on one host.

5.2 Frontend

For building the front-end web pages, PUG was used. With this templating engine, it is possible to generate HTML code by using a syntax that is way more readable (using indentation). CSS was used to create style sheets to give the web pages a nice visual appeal. Bootstrap was used to create the accordion element, and JavaScript was used to let the users interact with the web pages. It was also used for a loading animation.

5.3 Spotify Web API

The Spotify Web API offers many useful functionalities to build a music recommender system. This API brings the possibility to easily collect user data for the input of the recommender algorithm by using the user's Spotify account. Also, their easy-to-use recommender method works very well. However, this method also has one disadvantage: The user of the application should have a (potentially free) Spotify account with personal music data in it. Still, this option seemed to be the best overall for this application. The user study would only include people who have a Spotify account.

5.3.1 Spotify Developer Dashboard

A Spotify application was made, which could be accessed using the Spotify Developer Dashboard. This dashboard gives a nice overview of the web application, showing statistical graphs like the daily and monthly active users, the number of API requests per endpoint, the top countries, and much more. It also gives the possibility to control the user management of your application (shown in fig 5.1). By default, Spotify requires the developer to add a list of names and email addresses of each user in the user management dashboard. Only these users get access (via the email address of their Spotify account) to use the application.

There was one problem however: Only a maximum of 25 users are allowed to be registered in the user management dashboard at the same time. This was a problem for the 80-person user study, explained in section 6. To solve it, a quota extension request could be sent to Spotify, to ask them to expand the maximum user limit. This request was sent but eventually denied by Spotify for the reason: '*You must not analyse the Spotify Content or the Spotify Service for any purpose, including without limitation, benchmarking, functionality, usage statistics, or user metrics*'. Therefore, another solution was found to the problem: The list of 25 users would constantly be updated during the user study. Once two participants had completed the use of the web application, their names would be deleted from the dashboard and be replaced by two new participants.

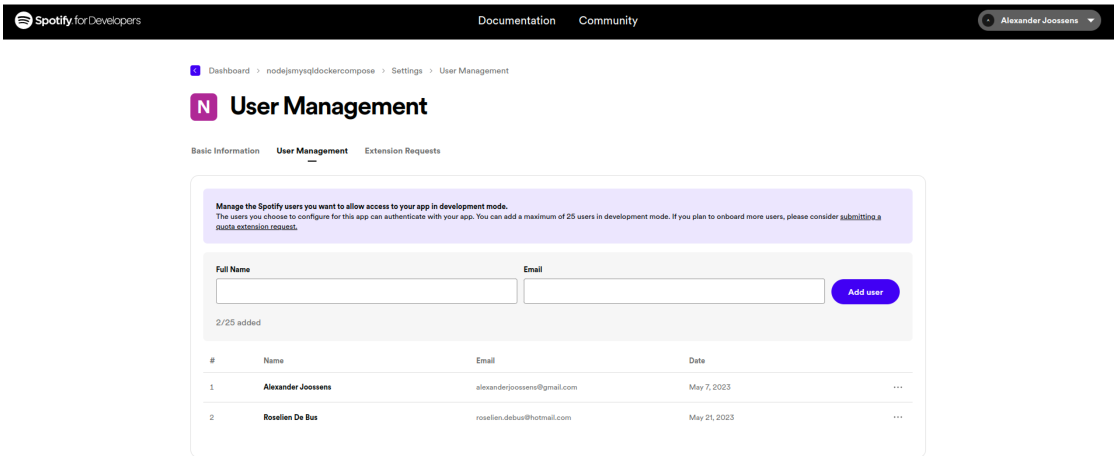


FIGURE 5.1: User Management in the Spotify Developer Dashboard

5.3.2 Authorisation framework

To retrieve the input music data for the group music recommender, it is necessary to ask the user's permission to obtain their Spotify data. Luckily, the Spotify API provides an easy way to do this, by using the OAuth 2.0 authorisation framework. This is the industry standard protocol that allows to approve one application interacting with another on your behalf without giving away a password. A simplified overview of the working of OAuth 2.0 is shown in figure 5.2. The authorisation process involves multiple parties: the Spotify user (referred to as the "End User"), the client application (referred to as "My App"), and the server hosting the protected resources. The permission to obtain the protected resources is granted based on one or multiple scopes. These scopes allow the application to access specific methods, such as reading user profiles, modifying playlists, or streaming content, on behalf of the user. When requesting authorisation, the chosen set of scopes determines the access permissions that the user is prompted to grant. After this, the authorisation server generates an access token that is utilised to make API calls on behalf of the user or application. To make use of this authorisation process, valid client credentials are required: a client ID and a client secret. These could be generated using the Spotify App settings guide.

FIGURE 5.2: OAuth 2.0 authorisation framework (source: Spotify)

5.3.3 Spotify Web API calls

As mentioned in the application design chapter 4.3, The Spotify Web API offers a method to request recommended music tracks. The documentation website mentions: '*Recommendations are generated based on the available information for a given seed entity and matched against similar artists and tracks. If there is sufficient information about the provided seeds, a list of tracks will be returned together with pool size details.*' This API function uses three required parameters: seed_tracks (seed code of some tracks), seed_genres (some genres) and seed_artists (some artists). Furthermore, it is possible to specify 44 other non-required parameters (e.g. min_acousticness, max_energy, target_popularity, etc.), which take values between 0 and 1. This *Get Recommendations* method could therefore be used to retrieve recommendations for the users in the *extra recommendations stage* of the pre-filtering algorithm and in the *individual song recommendations* of the post-filtering algorithm.
Also to make requests for user data like songs and playlists, the Spotify Web API was very useful. The endpoints of the Spotify Web API provide JSON metadata about music artists, tracks, and albums sourced directly from the *Spotify Data Catalogue*. Additionally, the API grants access to user-related data such as playlists and saved music in the user's *Your Music* library. This could be used to get the dynamic content of each user on many of the web pages.
Finally, the Spotify Web API was also used for the method *Create Playlist*. This allowed the creation of a new playlist inside the Spotify account of the user of the web application (after they clicked the button on the final playlist page). All the songs from the final recommended playlist were then added to this new playlist using the *Add Items to Playlist* method.

Chapter 6

Methodology

To answer the two research questions, a between-subjects user study was conducted. The conditions of the experiment obliged the recruitment of pairs of two people to test the web application. There are four possible use case combinations in the web application (shown in table 6.1). Each pair of participants was randomly assigned to one of the four use cases. The participants also filled out a post-test questionnaire about perceived accuracy, fairness, explanation, user satisfaction, intentional behaviour and other aspects. The answers to these questions were analysed. The analysis consisted of CFA tests, two-way ANOVA tests, and post-hoc Tukey HSD tests. From this, information could be gathered about the perceived differences (in accuracy, fairness, etc.) between the four groups, and the research questions could be answered.

	Content-based visual	User-based visual
Pre-filtering	20 users	20 users
Post-filtering	20 users	20 users

TABLE 6.1: Four use case combinations of the web application

6.1 Research Design

6.1.1 Participants

To test the web application, 80 participants (40 groups of 2 persons) were recruited to participate in the research. Most of the participants were found via friends, family and a Facebook post. Of the 80 participants, 90% were students, and 93% were between 18 and 25 years old. As shown in figure 6.1, the number of years that they already used Spotify was, on average, five to six years. Figure 6.2 shows that more than half of the participants use Spotify more than five hours per week. A little less than half of the participants (46%) had already used Spotify Blend mode before to generate a recommended playlist.

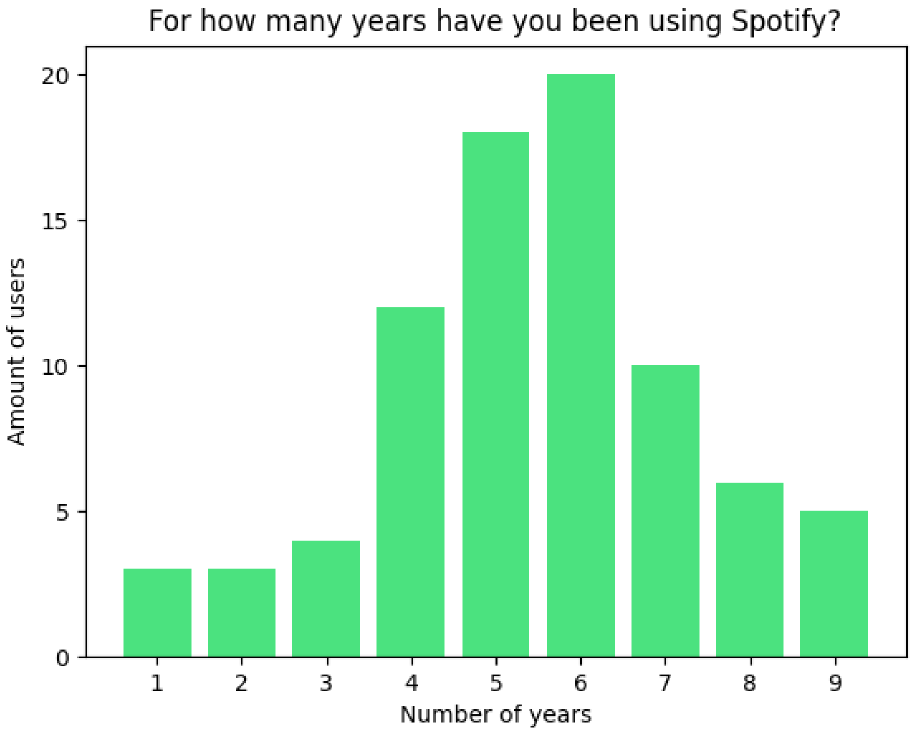


Figure 6.1: Number of years of Spotify usage by participants

6.1.2 User Study Procedure

All of the participant groups followed the same user study flow. First, they received an e-mail asking them for their written permission to use their personal data for this research and also asking for their Spotify email address. This email address was needed to input into the online Spotify Application dashboard of the web application. This way, the participants got permission to use the application and could successfully log in with their Spotify account to use the web application.

Once a pair of participants had sent their written permission and their Spotify email address, they were added to the Spotify application dashboard. After this, they received a second email containing information about the task, to let them better understand the goal of the application. The information was also displayed on one of the first pages of the web application: '*Imagine you want to create a common playlist with a friend for a party. So you want to mix both your party playlists, and the party playlists of your friend. With this application, you can select some of your own favourite playlists, and based on this, a new playlist will automatically be generated for you and your friend! All the steps speak for themselves, you will also get an explanation afterwards.*' Each pair of participants also received a unique code (based on the specific use case they were going to test) to use during the application. Finally, the users were asked (on the final playlist

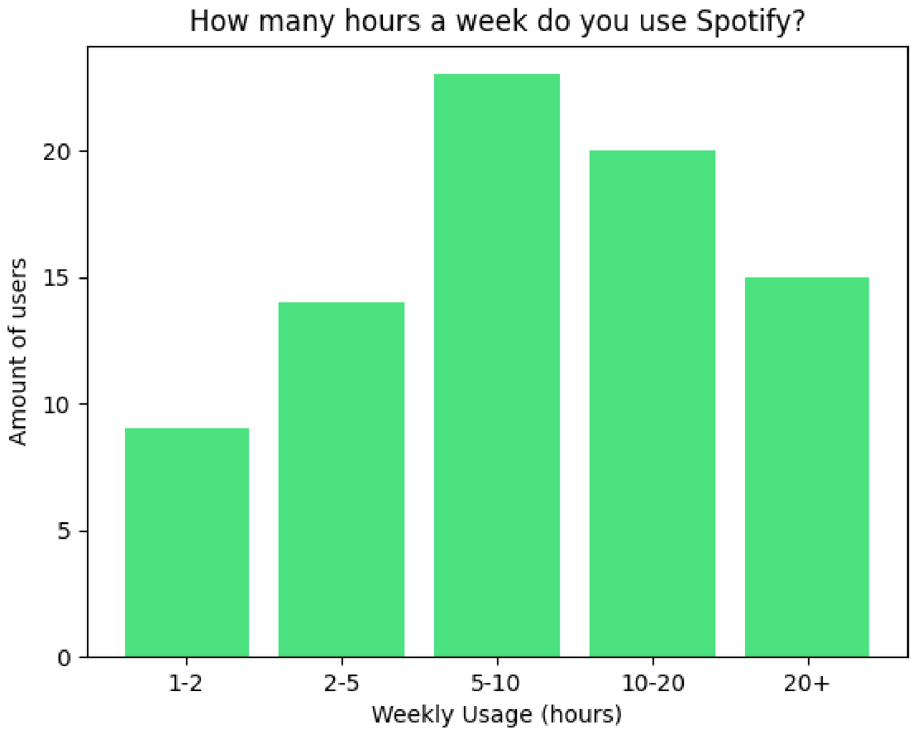


FIGURE 6.2: Number of hours per week of Spotify usage by participants

page) to fill out the post-test questionnaire.

6.1.3 Post-test questionnaire

The research questions focus on five different constructs (perceived fairness, accuracy, explanation, behavioural intentions and ease of use). Therefore, the questionnaire includes relevant questions concerning these constructs. Multiple sources of questions were incorporated:

- General questions (demographic, Spotify usage, etc.)
- Questions from the ResQue model [**?**]
- Questions from NASA-TLX (subjective mental workload assessment)
- Additional questions related to ease of use, behavioural intentions, perceived fairness, accuracy and explanation.

Table 6.2 shows the different constructs with their corresponding questions. All of the questions (except one open question) use a 5-point Likert scale.

Construct	Question
General	Please fill out the unique code that you have received: (Open answer)
General	What is your age? (18-25, 25-30, 30-35, ..., 50-60, 60-70, 70+)
General	Are you a student? (yes/no)
General	For how many years have you been using Spotify? (1,2,3,4,5,6,7,8,9,9+)
General	How many hours a week do you use Spotify? (1-2, 2-5, 5-10, 10-20, 20+)
Speechiness	Have you ever used the Blend/Friend Mix function of Spotify (that automatically creates a mixed playlist for you and a friend) (yes/no)
General	What do you think of the current mixed playlist that Spotify (Blend) generates?
General	Please give a suggestion to improve the application (from this i will see if it is about accuracy or fairness or usability or control)
Perc. accuracy	The recommender gave me good suggestions.
Perc. accuracy	I have the feeling that the songs, artists and genres of the recommended songs were based on the selected playlists.
Perc. accuracy	The recommendation I received better fits my interests than what I may receive from a friend.
Perc. accuracy	The generated playlist matched my interests.
Perc. fairness	I feel like my preferences were taken into account.
Perc. fairness	I feel like the recommendations were balanced between both our interests.
Perc. fairness	The visualisation helps me understand why the songs were recommended to me.
Perc. fairness	The visualisation helps me understand why the songs were recommended to both of us.
Perc. fairness	I think me and my friend both like the playlist
Perc. fairness	I feel like the recommendation algorithm is fair
Perc. fairness	I think this is a good application to make a fair group playlist
Perc. fairness	Fairness is important in group playlists
Ease of use	I became familiar with the recommender system very quickly.
Perc. Explanation	I understood why the songs were recommended to me.
Perc. Explanation	The recommender explains why the songs are recommended to me.
Perc. Explanation	The information provided for the recommended songs is sufficient for me to make a song listen decision
Perc. Explanation	The recommender allows me to tell what I like/dislike.
Behavioural intentions	I will use this recommender again if I need to create a playlist with other people
Behavioural intentions	I will tell my friends about this recommender.
User Experience: NASA TLX	How mentally demanding was the task?
User Experience: NASA TLX	How physically demanding was the task?
User Experience: NASA TLX	How hurried or rushed was the pace of the task?
User Experience: NASA TLX	How successful were you in accomplishing what you were asked to do?
User Experience: NASA TLX	How hard did you have to work to accomplish your level of performance?
User Experience: NASA TLX	How insecure, discouraged, irritated, stressed, and annoyed were you?

TABLE 6.2: Post-test questionnaire

Nevertheless, it is not certain that these questions are actually measuring the corresponding constructs. It is always possible that users interpret some questions differently, leading to answers that are meant for different constructs. This is something that is investigated after the user study, using a Confirmatory Factor Analysis (CFA) (see section 7.2.1). Finally, the number of users that clicked the '*Add to Playlist*' button was also registered. This could give an indication about the use intentions of the users, namely whether they would want to listen to the recommended playlist again later, or not. This is important to measure since the research questions focus on the differences between the four use case groups, with respect to behavioural intentions.

Chapter 7

Results

In this chapter, the results from the user study are described. This consists of the qualitative results and the quantitative results. On the quantitative results, a data analysis was conducted. This consists of a CFA, two-way ANOVA tests, and post-hoc Tukey HSD tests.

7.1 Qualitative results

The only qualitative measure in the user study was the open question in the questionnaires. The question was: '*Please give a suggestion to improve the application*'. Next to suggestions for possible improvements, the answers also provided some general feedback from the participants.
A common recurring theme in the received feedback was that the application provides a nice user experience and that it is very user-friendly. Multiple participants stated that the final playlist was a good combination of the selected playlists, and that the artist matches were great. Furthermore, participants found it a 'fun' project and thought that the application was 'cool'.
However, some of the participants provided negative feedback as well. A recurring feedback theme was that the recommended songs sometimes contained some weird combinations. For example, a final playlist that included multiple different genres. Other participants stated that the final playlist contained music in foreign languages and that they would have preferred to have specific control over the genres and languages of the generated recommendations.
The answers that contained suggestions for further features of the application are discussed in the discussion chapter 8.

7.2 Quantitative results

In this section, the data analysis on both the questionnaire answers and the button click logs is reported.

7.2.1 Confirmatory factor analysis

As mentioned in the qualitative results 7.1, a lot of positive feedback was given on the web application. Not only the open question revealed this, but also the average scores of the other answers in the questionnaire. The average answer of all questions were well above 3/5 (5/5 is the most positive). For the question *I will use this recommender again if I need to create a playlist with other people*, the average score was 3.83. For the question *I will tell my friends about this recommender*, the average score was 3.72. Here, a score of 1 means *Disagree* and a score of 5 means *Agree*.
As mentioned before, the questions from the questionnaire were each intended to aim for a specific construct. However, the perception of the users answering the questions could be different. For example the question '*I think this is a good application to make a fair group playlist*' could both be interpreted to be about the construct 'accuracy', as well as about 'fairness'.
Therefore, a Confirmatory factor analysis (CFA) test was conducted to enable the identification of the optimal question combination for each construct. CFA tests whether the created factors are consistent with the hypothesised model. This fit of the model was assessed using statistical measures such as the Root Mean Square Error of Approximation (RMSEA), Tucker-Lewis Index (TLI), and Comparative Fit Index (CFI)[1]. A model with a good fit was identified (TLI = 0.984, CFI = 0.991, RMSEA = 0.0454). Furthermore, the average variance extracted (AVE) of each construct was examined. The AVEs were all higher than the recommended value of 0.50, indicating adequate convergent validity. Questions not included in the final CFA question list were excluded as they did not effectively measure the intended constructs or led to conflicting results.
The resulting clusters gave a clear separation between most of the questions. The resulting questions per questionnaire are shown in table 7.1. The results of the CFA provided a loading value per question. This value can be interpreted as a weight for the question for the according construct. The higher the loading value, the more influence the question has on the construct.
From the CFA, the loading value per question for each construct was determined. With these values, the average response per construct for each of the four use case combinations could be calculated. The results are shown in figure 7.1. The vertical error bars represent the standard deviation. The average responses can be higher than five, since the score of a construct is a sum over the relevant questions weighted with their loading values. Therefore, it is also not possible to compare different constructs with each other from this graph. However, for each construct, the differences per use case combination are clearly visible.

7.2.2 Two-way ANOVA test

Since the three required conditions were met (homogeneity of variance, independence of observations, normally-distributed dependent variable), two-way ANOVA tests could

[1]Hu and Bentler [36] suggest the values for the fit indices to be: TLI > .95, CFI > .96, and RMSEA < .05, with the upper bound of its 90% CI below 0.10

Construct	Item	Loading
Ease of use	I became familiar with the recommender system very quickly	1.129
Behavioural intentions	I will use this recommender again if I need to create a playlist with other people	1.056
$\alpha = 0.852$	I will tell my friends about this recommender.	0.948
AVE = 0.746		
Perceived explanation	The recommender explains why the songs are recommended to me.	0.959
$\alpha = 0.838$	The information provided for the recommended songs is sufficient for me to make a song listen decision	0.982
AVE = 0.647	I understood why the items were recommended to me	
	The recommender allows me to tell what I like/dislike.	0.743
	The visualisation helps me understand why the songs were recommended to me.	
	The visualisation help me understand why the songs were recommended to both of us.	
Perceived accuracy	The generated playlist matched my interests.	0.740
$\alpha = 0.824$	The recommender gave me good suggestions.	0.714
AVE = 0.702	The recommendation I received better fits my interests than what I may receive from a friend.	
	I have the feeling that the songs, artists and genres of the recommended songs were based on the selected playlists.	
Perceived fairness	I feel like my preferences were taken into account.	0.784
$\alpha = 0.768$	Fairness is important in group playlists	
AVE = 0.650	I feel like the recommendation algorithm is fair	0.586
	I feel like the recommendations were balanced between both our interests.	
	I think me and my friend both like the playlist.	
	I think this is a good application to make a fair group playlist	

TABLE 7.1: Results of the confirmatory factor analysis (CFA), indicating five constructs: Ease of use, Behavioural intentions, Perceived explanation, Perceived accuracy and Perceived fairness. Questions in grey are the ones that were removed from the construct.

Figure 7.1: Average response per construct for every use case combination. The vertical error bars represent the standard deviation.

be conducted. The tests were used to assess whether there are significant main effects of the algorithm type and visualisation type (the independent variables) on each of the constructs (the dependent variable), as well as whether there is a significant interaction effect between these two independent variables. For each construct, a two-way ANOVA test was conducted. All the numerical results are added in the appendix D.

Ease of use: there is no statistically significant interaction between the effects of the algorithm type and the explanation type ($F(1, 76) = 2.16$, $p = 0.146$). Simple main effects analysis showed that the algorithm type did not have a statistically significant effect on the ease of use ($p = 0.922$). Also, the explanation type did not have a statistically significant effect on the ease of use ($p = 0.625$).

Behavioural intentions: there is a statistically significant interaction between the effects of the algorithm type and the explanation type ($F(1, 76) = 12.8$, $p = 0.000602$). Simple main effects analysis showed that the algorithm type did not have a statistically significant effect on the behavioural intentions ($p = 0.377$). Also the explanation type did not have a statistically significant effect on the behavioural intentions ($p = 0.0869$).

Perceived explanation: there is a statistically significant interaction between the effects of the algorithm type and the explanation type ($F(1, 76) = 6.37$, $p = 0.0137$). Simple main effects analysis showed that the algorithm type did not have a statistically significant effect on the perceived explanation ($p = 0.752$). Also the explanation type did not have a statistically significant effect on the perceived explanation ($p = 0.882$).

Perceived accuracy: there is no statistically significant interaction between the effects of the algorithm type and the explanation type ($F(1, 76) = 0.562$, $p = 0.456$). Simple main effects analysis showed that the algorithm type did not have a statistically significant effect on the perceived accuracy ($p = 0.313$). Also the explanation type did not have a statistically significant effect on the perceived accuracy ($p = 0.186$).

Perceived fairness: there is no statistically significant interaction between the effects of the algorithm type and the explanation type ($F(1, 76) = 1.78$, $p = 0.187$). Simple main effects analysis showed that the algorithm type did not have a statistically significant effect on the perceived fairness ($p = 0.396$). Also the explanation type did not have a statistically significant effect on the perceived fairness ($p = 0.252$).

7.2.3 Post-hoc Tukey HSD test

When significant interaction effects were found, post-hoc Tukey HSD (Honestly Significant Difference) tests were conducted. These tests were performed to determine specific pairwise differences between the levels of the independent variables (algorithm type and visual explanation type) that contributed to the observed interaction effect. Therefore it is possible to decide which specific groups differ significantly from each other.

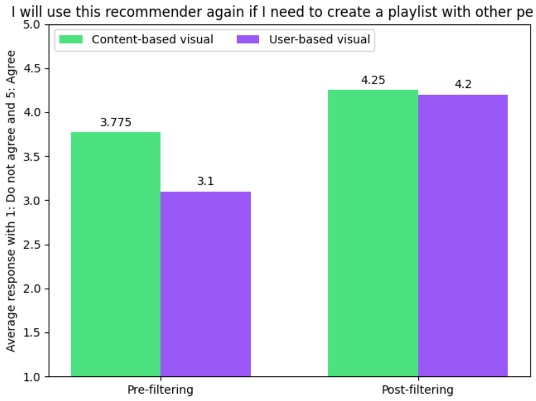
The results of the post-hoc Tukey test, for both the *behavioural intentions* construct and the *perceived explanation* construct are shown in the appendix D. The outcome of the test on the *behavioural intentions* construct showed that the results from the group using the post-filtering algorithm and the content-based visualisation type differ significantly from the group using the pre-filtering algorithm and the user-based visualisation type. This means this former group scored the *behavioural intentions* questions significantly higher than the latter group. The averages of the two *behavioural intentions* questions, are plotted per group in figure 7.2. Even though the loading values (weights) of these questions on the construct are not taken into account in the figure, it is clearly visible that the average score for the combination of post-filtering with the contend-based visualisation is always the highest.

For the *perceived explanation* construct, another post-hoc Tukey HSD test was conducted. The results showed that, again, the group with the post-filtering algorithm and the content-based visualisation type scored significantly higher on the *perceived explanation* construct than the group with the pre-filtering algorithm and user-based visualisation.

The average scores for the *perceived explanation* questions per group are shown in figure 7.3. Again, these plots do not take into account the loading values on the construct. However, from the three plots it is still clear that the group that used post-filtering and the content-based visualisation scored much higher on the *perceived explanation* questions, than the group that used pre-filtering and the user-based visualisation.

FIGURE 7.2: Behavioural intentions questions

(A) Average results per group for question 5



(B) Average results per group for question 25

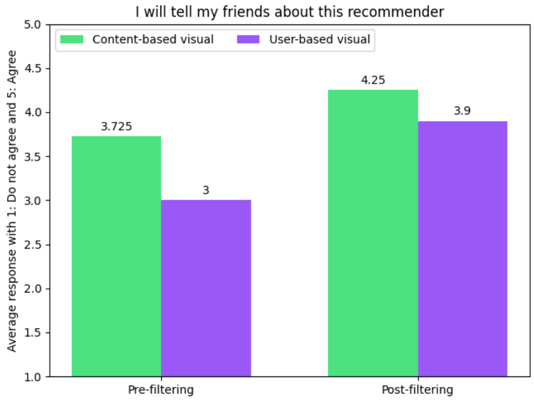
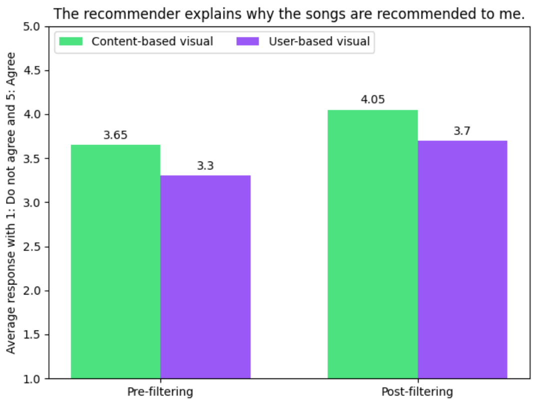
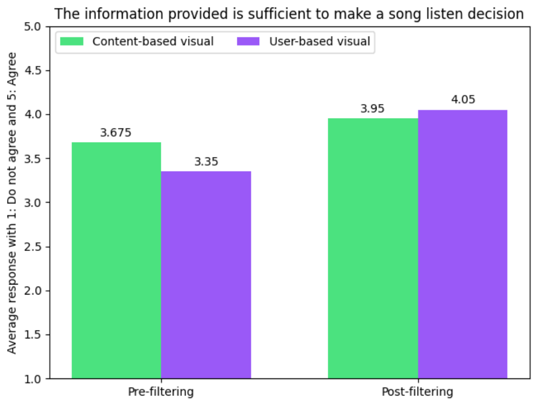


FIGURE 7.3: Perceived explanation questions

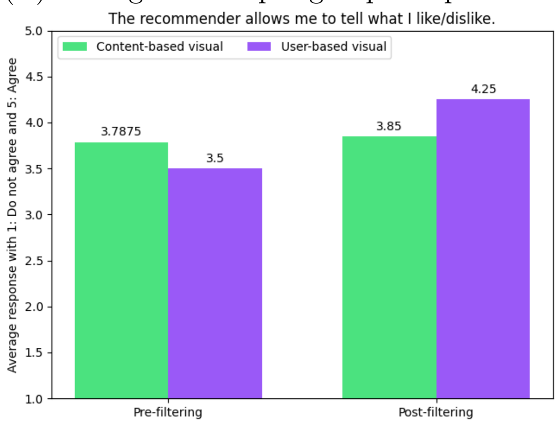
(A) Average results per group for question 4



(B) Average results per group for question 6



(C) Average results per group for question 9



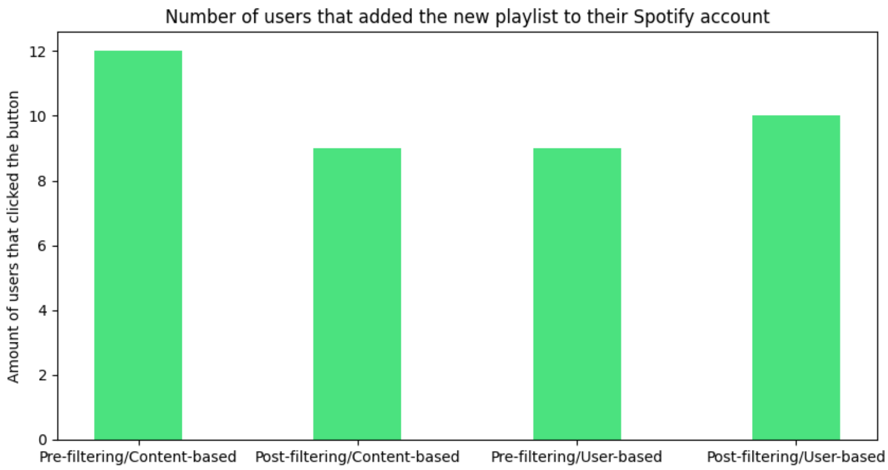


FIGURE 7.4: Number of users that added the new playlist to their Spotify account

7.2.4 Button click logs

As mentioned before, the amount of clicks on the button '*Add playlist to my Spotify account*' was also registered. The results are shown in figure 7.4. In total, 50% of the participants added the generated playlist to their Spotify account. From the group with the pre-filtering algorithm and the content-based visualisation, twelve users added the playlist to their Spotify account. From the group with the pre-filtering algorithm and user-based visualisation, there were nine users, from the group with the post-filtering algorithm and the content-based visualisation there were also nine users, and from the group with the post-filtering algorithm and the user-based visualisation there were ten users. From this data, another two-way ANOVA test was executed. The results showed that there is no statistically significant interaction between the effects of the algorithm type and the explanation type ($F(1, 76) = 0.772$, $p = 0.382$). Simple main effects analysis showed that the algorithm type did not have a statistically significant effect on the amount of button clicks ($p = 0.662$). Also the explanation type did not have a statistically significant effect on the amount of button clicks ($p = 0.662$).

7.2.5 NASA TLX

The cognitive load perceived by the users during the experiment was also measured using the NASA TLX (task load index) workload assessment [32]. This evaluation was included in the post-test questionnaire. The results of a two-way ANOVA test on the *NASA TLX* construct (the average answer of all six NASA TLX questions) gave the following results.
NASA TLX: there is no statistically significant interaction between the effects of the algorithm type and the explanation type ($F(1, 76) = 0.806$, $p = 0.372$). Simple main effects analysis showed that the algorithm type did not have a statistically significant

effect on the NASA TLX questions ($p = 0.795$). Also the explanation type did not have a statistically significant effect on the NASA TLX questions ($p = 0.239$).

Chapter 8

Discussion

This chapter discusses the results of the user study and tries to answer the research questions in light of these outcomes.
The first section summarises the research questions and their respective hypotheses. After this follows a discussion about the results of the user study and a comparison with the initial hypotheses. The final section discusses the limitations.

8.1 Research questions and hypotheses

As explained in chapter 3, this research focuses on two research questions, regarding filtering methods and visual explanation types in a group recommender system. The two research questions are stated again below.

- How do pre- and post-filtering in a group recommendation system influence the perceived accuracy, fairness, explanation and behavioural intentions?
- How do different visual explanations of group recommendations affect the perceived fairness, accuracy, explanation, behavioural intentions, and ease of use?

To summarise, the hypothesis for the first research question is: *The pre-filtering algorithm will score better on perceived fairness and accuracy but lower on behavioural intentions and perceived explanation.*
The hypothesis of the second research question is: *The visualisation with the user-based explanation style will score higher on the perceived fairness and explanation, while the visualisation with the content-based explanation style will score higher on perceived accuracy and behavioural intentions.*

8.2 Discussion user study results

Behavioural intentions

The two-way ANOVA results show that for the *behavioural intentions* construct, the combination of the post-filtering algorithm with the content-based visualisation scored

significantly higher than the combination of the pre-filtering algorithm with the user-based visualisation.
These results show that the hypothesis of the first research question was partly correct. As expected, the post-filtering algorithm leads to a higher score on the *behavioural intentions* construct than pre-filtering, but only in combination with the content-based visualisation and the user-based visualisation respectively. A possible reason for this could be found in the second stage of the post-filtering algorithm. In this *Individual song recommendations* stage, song recommendations are generated based on the users' individual playlists. Only these songs are eligible to be included in the final playlist. It is very likely that these songs are novel, since they were recommended using the Spotify Web API method *Get recommendations*, as explained in 4.3.2. However, the pre-filtering algorithm provides a playlist, based on the intersection of both users' playlists. This means that there are fewer novel songs. This difference in novelty could have had an influence on the *behavioural intentions* construct for the users. If novel songs were something that the users were looking for, they could have scored higher on the *behavioural intentions* questions.
Also the hypothesis for the second research question was partly right about the *behavioural intentions* aspect. The content-based explanation (but in combination with post-filtering) scores higher than the group with user-based explanations (in combination with the pre-filtering algorithm). A possible reason for this outcome is that users could want to use the application again if it shows more direct information about the recommended items. Moreover, the users could have found the top three song characteristics in the visual explanation more useful to understand the origin of the recommendations than the playlist match percentages.

Perceived explanation

For the *perceived explanation* construct, the combination of the post-filtering algorithm with the content-based visualisation type scored significantly higher than the combination of the pre-filtering algorithm with the user-based visualisation. The two-way ANOVA test showed an interaction effect between these two factors, indicating that the choice of algorithm and visualisation type together has a statistically significant impact on the *perceived explanation* construct.
The hypotheses of the research questions were partly correct. As expected, the post-filtering algorithm scored indeed higher on the *perceived explanation* construct than the pre-filtering algorithm. However, it does this in combination with the content-based visualisation, instead of the user-based visualisation mentioned in the hypothesis of the second research question. This was not expected. The idea of the first hypothesis was that the joining of both users' recommended song lists can be done in a transparent way (described in 3.3), and will therefore score higher on the *perceived explanation* aspect. The idea of the second hypothesis was that users could use the user-based explanation to make a direct link with each of the users' interests. Therefore they could immediately judge that the group recommender system provides a good explanation because it is visually shown in the explanation that each of the user's interests were taken into account. This however does not agree with the conclusion of the data analysis. A possible reason for this

could be that the users did not understand the playlist match percentages entirely. As mentioned in the feedback question of the questionnaire, some users found the percentages confusing, and did not understand what they were based on.
These results can be connected to the work of Panniello et al. [56]. In their study, a recommender system using pre-filtering was compared with a recommender system using post-filtering. Although they did not measure the impact on the same constructs, their results show that '*the post-filtering approach reaches better performance than the pre-filtering approach*'. In our study, the post-filtering algorithm also performed better (on the '*behavioural intentions* and '*perceived explanation*'), but in combination with the content-based visualisation alone. As mentioned before, a possible reason for this could be that the playlist match percentages could be perceived as confusing for the user, and therefore a lower score was given when the user-based explanation style was involved.

Perceived fairness, perceived accuracy and ease of use

Unlike the hypotheses of both research questions stated, the algorithm types and visualisation types did not show a statistically significant difference w.r.t. the *perceived fairness* and *perceived accuracy* constructs.
The first hypothesis expected a higher score for the pre-filtering algorithm because the resulting playlist contains a list of songs and artists that both users know for sure. This could lead to a feeling of both of the users that the playlist certainly keeps both of their interests into account. This was apparently not the case, and the users did not have an extra feeling of a more *fair* or *accurate* playlist. Also for the user-based explanation type, the *perceived fairness* and *accuracy* aspects were not scored significantly higher or lower than for the content-based explanation type.

8.3 Limitations

This section describes three limitations of the research.

1. The algorithm that was implemented using post-filtering provides higher chances on generating novel songs than the pre-filtering algorithm. This is because the post-filtering algorithm does not contain the '*mutual song selection* and *mutual artist selection* stages, compared to the pre-filtering algorithm. This difference is not a characteristic of post-filtering techniques in general. It is perfectly possible that pre-filtering algorithms also provide more novel recommendations. It is possible that this contrast between the two implementations of these filtering techniques could have had an impact on the answers of the questionnaire. For example, the *behavioural intentions* construct was more positively received by the group using the post-filtering algorithm (in combination with the content-based explanation). It is imaginable that this higher chance of novel songs could have had an influence on this outcome.

2. It was mentioned in the second email to the users (containing all the information w.r.t. the application and the questionnaire), that both users were not allowed to

discuss the questions with each other while filling out the questions. However, this is something that could not be guaranteed to be followed. Therefore, it is a limitation of this research that could have impacted the answers of the questionnaire.

3. A final limitation of this research involves the visual explanations. As mentioned in section 4.2.3, the content-based visualisation did not exclusively use content-based explanations. Besides the top three audio characteristics (content-based), there was also the overall match percentage bar in the visualisation. This an aspect of the visualisation that is user-based, since it provides a comparison of the music taste of the two users with the recommended song. However, since, this feature was so popular in the think-aloud studies, and the participants found it very useful, it was kept for both visualisations. This could have resulted in an unfair comparison between the two explanation styles.

Chapter 9

Conclusion

9.1 Summary

This research focused on two aspects of group recommender systems, namely filtering methods and explanation styles. A profound literature study provided an overview of the relevant work that has already been done for this research. Especially the studies concerning group recommender systems, explanations, fairness and music group recommender systems provided a solid foundation for the construction of a new web application. This web application uses pre and post-filtering techniques, together with content-based and user-based explanations. Based on a number of selected playlists, the application generates a list of recommended songs and showed them with a visual explanation on a final playlist page.
To ensure an effective design of the web application, an initial questionnaire was used to retrieve the users' expectations. Furthermore, three think-aloud studies were conducted to shape the form of the flow of the application and to determine the best visual explanation pages.
A between-subjects user study was conducted to investigate the influence of these filtering methods and explanation styles on the perceived fairness, accuracy, explanation, behavioural intentions and ease of use for the user. A total of 80 participants were recruited in pairs of two people, to test the web application. They made use of the web application to generate a new recommended playlist, which they could add to their own Spotify account. Afterwards, they filled out a post-test questionnaire, including questions about the five investigated constructs: perceived fairness, accuracy, explanation, behavioural intentions, and ease of use.

9.2 Conclusion

From the qualitative results was concluded that the users had a positive overall user experience and that the final playlist provided a good combination of the selected playlists. However, the playlist generation speed was perceived as slow and the recommended songs sometimes included a mix of different non-matching genres.

To obtain the quantitative results, a data analysis was conducted. Using a confirmatory factor analysis, the relevant questions from the questionnaire w.r.t the five constructs were obtained, together with their loading values. These were used in a two-way ANOVA test to assess whether there were significant main effects of the algorithm type and visualisation type on each of the constructs, as well as whether there is a significant interaction effect between these two independent variables
For the *behavioural intentions* construct and the *perceived explanation* construct, a significant interaction effect was found. First of all, the combination of the post-filtering algorithm and the content-based visualisation type has a positive influence on the *behavioural intentions* construct, relative to the pre-filtering algorithm with the user-based visualisation type. Secondly, the same combination of the post-filtering with the content-based visualisation type has a positive influence on the *perceived explanation* construct.
From these results, the following conclusion was drawn:

- Using a combination of post-filtering (providing only new recommendations) and content-based explanations (like matching song characteristics with the user's taste) can have a positive influence on the *perceived explanation* and the *behavioural intentions*. This is especially true for users who are interested in discovering new songs. Furthermore, showing too many match percentages can become difficult to interpret and confusing for the user.

Finally, the proposed playlist-selection-based approach, the pre and post-filtering algorithms and the user-based and content-based explanations could be a useful addition to Spotify's *Blend* feature. Since the participants of this research indicated a lack of explanation and control possibilities on the recommended songs from the *Blend* playlist, this work's proposed application features could provide a solution for this problem.

9.3 Future work

This research contributed to the field of filtering techniques and visual explanation types in group recommender systems. Providing the user with too many percentages or numbers can lead to confusion and should be avoided. It is important to understand whether users are interested in novel items since this can have a major influence on the *behavioural intentions*.
To further understand the influence of the filtering methods and explanation types in group recommender systems, further studies could be conducted using larger groups of participants. Different group sizes could influence the constructs even more, especially for the fairness aspect, where every additional user plays an important role in the perceived fairness of other users.
Furthermore, the usability of the web application could be improved in a number of ways. The speed of the generation of the recommended songs could be accelerated, and the user-based explanations could be made more clear. Also, the possibility of giving more control to the user on the suggested genres of the tracks could be further implemented.

A final suggestion for future research consists of investigating the effect of different types of explanation styles on the five used constructs. Some specific explanation types include social-based and item popularity-based explanations. Also, the combination of these explanation styles with the proposed pre-filtering and post-filtering algorithm could still be an interesting topic for future research.

Appendices

Appendix A

SMEC

6/5/23, 11:02 AM Register - G-2022-6053

DOSSIER: G-2022-6053

Versie 3.2
Status: Accepted

Datum goedkeuring: 19/12/2022 18:29:15

2.ALGEMEEN

ONDERZOEKERS

Indiener: Alexander Joossens
Faculteit Ingenieurswetenschappen, departement computerwetenschappen
HCI
alexander.joossens@student.kuleuven.be
r0748533

Promotor: Ivania Nadine Donoso Guzmán
computerwetenschappen
HCI
ivania.donoso@kuleuven.be
r0873329

Oefent de promotor een gezondheidszorgberoep uit in de zin van de wet van 10 mei 2015? nee

Andere KU Leuven onderzoekers:

Katrien Verbert

FINANCIER

Wordt het onderzoek gefinancierd (bijvoorbeeld in het kader van een onderzoeksproject/ contract/ mandaat)? nee

Dienst die instaat voor afgesloten of nog af te sluiten contracten:

VOORNAAMSTE / LEIDENDE ETHISCHE COMMISSIE

De voornaamste / leidende ethische commissie: SMEC

3.ONDERZOEK

TITEL, BESCHRIJVING EN DOELSTELLINGEN

Unieke en volledige titel van het onderzoeksproject / protocol:

A Playlist-based Group Music Recommendation System

Beschrijving van het onderzoeksproject / protocol:

The goal is to investigate the usage of a group recommendation system. The application recommends a playlist for groups of users, based on their selected Spotify playlists. Two research questions will be investigated:
- How does pre/post filtering influence user satisfaction, accuracy and fairness?
- How does the degree of explanation in different visualizations in a Group Recommender System affect how much users listen to playlists, their satisfaction, their perceived fairness and accuracy?

Doelstellingen van het onderzoek(sproject):

The objective of the research is to evaluate different explanations for group recommendations, and how the user perceives fairness, accuracy, and other aspects using this application.
This research is very interesting for the AI community since there is more need for research on group recommender systems, focusing on fairness.

https://www.groupware.kuleuven.be/sites/pret/_layouts/15/Print.FormServer.ICTS.aspx?XmlLocation=/sites/pret/Register/G-2022... 1/5

Figure A.1: SMEC part 1

6/5/23, 11:02 AM Register - G-2022-6053

ONDERZOEKSTECHNIEKEN, INSTRUMENTEN & APPARATUUR

Vragenlijst: ja
Niet-gevalideerde vragenlijst: ja

resque_full.pdf
259,12 KB

Online/web gebaseerde activiteiten (al dan niet bestaand uit bovenstaande categorieën): ja
The user usage will be tracked. (Amount of times they listen to a song, which songs they listen, how they interact with the app)

4.VERZAMELEN EN DELEN VAN PERSOONSGEGEVENS IN DE STUDIE

PRIMAIRE/SECUNDAIRE GEGEVENSWERWERKING

Verzamelt u nieuwe gegevens (primaire verwerking) en/of gebruikt u enkel eerder verzamelde gegevens (secundaire verwerking)?
Primaire verwerking: ja
Secundaire verwerking: nee

EXTERNE PARTNERS

Andere niet KU Leuven onderzoekers:

Het betreft een niet-commerciële studie.
Is er naast KU Leuven nog een andere universiteit, onderzoeksinstelling of andere partner betrokken bij het onderzoek? nee

Overeenkomst:

ICF may2022 ENG.odt
24,1 KB

ROLLEN VAN DE BETROKKEN PARTIJEN

Wie bepaalt de doelstellingen van het onderzoek?
Dit wordt binnen KU Leuven bepaald.

Is er een andere partij betrokken die als verwerker in opdracht van KU Leuven (al dan niet samen met anderen) persoonsgegevens verwerkt (zoals bv. data verzamelen; analyses uitvoeren)? nee

DOORGIFTEN EN DELEN VAN PERSOONSGEGEVENS

Zullen de verzamelde persoonsgegevens worden doorgegeven aan/gedeeld met personen/instanties buiten KU Leuven?
ja
Bevindt deze persoon/instantie zich binnen de EU en/of buiten de EU?

PLAATS DATAVERWERKING

Waar zal de dataverzameling worden uitgevoerd?
België: ja

5. DEELNEMERS EN CATEGORIEËN VAN GEGEVENS

CATEGORIEËN VAN BETROKKENEN

Persoonsgegevens die u onderzoekt / verwerkt:

https://www.groupware.kuleuven.be/sites/pret/_layouts/15/Print.FormServer.ICTS.aspx?XmlLocation=/sites/pret/Register/G-2022... 2/5

Figure A.2: SMEC part 2

Objective: people that have been using Spotify for a certain amount of time

We will collect: Name, e-mail address, Login information for Spotify, playlist data from Spotify.

I will use the Spotify API to let the participants login to their Spotify account.

I will use the data from Spotify and use the OAuth2.0 framework. This means that the user has to give consent that I will use their personal data.

The user will grant permission once and I will store the keys in a safe place. (On the server of the university)

Zijn er kwetsbare personen betrokken? nee

Selectiecriteria:

We aim to be as inclusive as possible. (All ages, sexes, characteristics etc. are welcome)
I am going to actively recruit participants at University. Only restriction is that they have to use Spotify for x amount of years.

CATEGORIEËN PERSOONSGEGEVENS

Welke categorieën van gegevens verzamelt u of gebruik u?

Gewone persoonsgegevens: ja

Identificatiegegevens (bv. namen, (e-mail)adressen)

Identificatiegegevens: namen
Identificatiegegevens: e-mailadres

GROOTSCHALIGHEID/ KLEINSCHALIGHEID VAN DE VERWERKINGEN

Wat is het verwachte aantal personen wiens persoonsgegevens worden verzameld?

For 2 conditions with 2 variants: 2x2 matrix of options. For each option I will need minimum 5 couples of persons for the research to be trustworthy.
So I will need 20 couples of 2 persons = 40 persons in total.

Hoe verhoudt deze steekproef zich tot de relevante populatie?

This would be a good sample amount.

Hoeveel gegevens verwerk je van één betrokkene en hoe divers zijn deze gegevens?

Name, email address, Spotify login, Spotify tracking
The user usage will be tracked. (Amount of times they listen to a song, which songs they listen, how they interact with the app)
The participants will answer a questionnaire. This data will also be processed.

Wat is de geografische reikwijdte van de persoonsgegevens die je verwerkt?

I will handle the data with the servers from KU Leuven in Belgium.

6. TECHNISCHE EN ORGANISATORISCHE MAATREGELEN BIJ VERWERKING EN BEHEER VAN GEGEVENS

BEWARING VAN DE GEGEVENS

Waar worden de digitale gegevens bewaard?

Beveiligde netwerkschijf van KU Leuven (bv. I- / J-schijf)
OneDrive gelinkt aan een KU Leuvenaccount

Waar worden papieren gegevens bewaard?

Er zijn geen papieren gegevens (bv. ook de geïnformeerde toestemming wordt digitaal afgenomen).

Wie heeft toegang tot de (persoons)gegevens tijdens de studie?

De betrokken KU Leuvenonderzoekers (zie pagina 2)

Wie heeft toegang tot de (persoons)gegevens na de studie?

De betrokken KU Leuvenonderzoekers, behalve eventuele betrokken studenten. (**)

Hoe lang zullen de (persoons)gegevens bewaard worden na het onderzoek?

Na 10 jaar zal worden beoordeeld of het noodzakelijk is om de (persoons)gegevens nog langer te bewaren. Indien noodzakelijk wordt op dat moment een herinneringsdatum vastgelegd waarop dit opnieuw beoordeeld wordt. Indien verdere bewaring niet meer nodig is, worden de (persoons)gegevens verwijderd.

ANONIMISERING / PSEUDONIMISERING GEGEVENS

FIGURE A.3: SMEC part 3

6/5/23, 11:02 AM Register - G-2022-6053

Is er een ogenblik in uw onderzoek waarop u persoonsgegevens gaat anonimiseren of pseudonimiseren?

Niet geanonimiseerde of niet gepseudonimiseerde gegevens: nee

U gaat zelf persoonsgegevens anonimiseren: ja

The user will log in and give their permission to collect data from their Spotify account on their behalf. Once I have finished the study, I will completely remove the credentials to access their Spotify accounts. I will also remove their names, email addresses and any other personal data.

OVERIGE TECHNISCHE EN ORGANISATORISCHE MAATREGELEN

Neemt u nog andere maatregelen ter bescherming van de privacy van de betrokkenen?

No

7.INFORMATIEVERSTREKKING AAN DE BETROKKENEN/DEELNEMERS

ONDERZOEK DAT MISLEIDING INHOUDT

Zullen de deelnemers misleid worden? nee

INFORMATIEVERSTREKKING AAN DE BETROKKENEN/ DEELNEMERS

Zal de nodig informatie aan de betrokkenen worden verstrekt of werd dit reeds gedaan?

In geval van primaire verwerking: ja

Informatie over verwerking persoonsgegevens EN.pdf
46,54 KB

In geval van secundaire verwerking: nee

/

8. RECHTEN VAN BETROKKENEN EN RECHTMATIGHEID VERWERKING

AFWIJKING OP RECHTEN VAN BETROKKENEN

Wordt uw onderzoek ernstig belemmerd indien de betrokken personen hun recht van inzage, rectificatie, beperking van de verwerking en recht van bezwaar willen uitoefenen?

nee

RECHTMATIGHEID VAN DE VERWERKING

Geselecteerde rechtsbasis:

Universitair onderzoek wordt doorgaans gevoerd in het algemeen belang: ja

9. RISICO-ANALYSE DOOR ONDERZOEKER

Houdt het onderzoek een hoog privacy-risico in voor de betrokkenen?

Indien de gegevens openbaar gemaakt zouden worden, zou dit een grote impact hebben op de betrokkenen? nee
Werkt u met bijzondere categorieën van persoonsgegevens? nee
Verwerkt u persoonsgegevens van kwetsbare groepen? nee
Verwerkt u gegevens op grote schaal? Hou voor het beantwoorden rekening met de absolute hoeveelheid persoonsgegevens, maar ook met de grootte van de steekproef t.a.v. de relevante populatie (zie vraag 4)? nee
Worden de gegevens doorgegeven aan een land buiten de EU dat niet op de 'witte lijst' staat? nee
Gaat u verschillende (bijzondere categorieën van) persoonsgegevens aan elkaar koppelen? nee
Hebben de verwerkingen juridische gevolgen of een gelijkaardig effect voor de betrokkene zoals uitsluiting of discriminatie van de betrokkene? nee
Hebben de verwerkingen het gevolg dat de betrokkene wordt belet om zijn rechten uit te oefenen, of gebruik te maken van een dienst of contract? nee
Gaat u op systematische wijze toezicht houden op personen in openbare ruimten? nee
Dienen de verwerkingen om profielen van personen op te stellen en voorspellingen te maken? nee
Maakt u innovatief gebruik van technologische toepassingen (bijvoorbeeld het gecombineerd gebruiken maken van vingerafdruk en gezichtsherkenning voor toegangscontrole)? nee
Werkt u met niet-gepseudonimiseerde persoonsgegevens? ja

https://www.groupware.kuleuven.be/sites/pret/_layouts/15/Print.FormServer.ICTS.aspx?XmlLocation=/sites/pret/Register/G-2022... 4/5

Figure A.4: SMEC part 4

Appendix B

Informed consent and Information on Data Processing

Informed consent

[This IC form is an example and needs to be adapted to each particular study. Italic text between brackets are instructions for the researchers and need to be deleted from the final draft. Gray boxes indicate that more information needs to be given by the researchers.]

Title of the research:
A Playlist-based Group Music Recommendation System

Name + contact details [*email, phone number, faculty/department/research unit, work address*] of supervisor and researcher(s):

Alexander Joossens [alexander.joossens@student.kuleuven.be, +32483005808, Computer Science Department, HCI research group, Celestijnenlaan 200A box 2402, B-3001 Leuven, BELGIUM]

Ivania Nadine Donoso Guzmán [ivania.donoso@kuleuven.be, Computer Science Department, HCI research group, Celestijnenlaan 200A box 2402, B-3001 Leuven, BELGIUM]

Goal and methodology of the research:
The goal is to investigate the usage of a group recommendation system. The application recommends a playlist for groups of users, based on their selected Spotify playlists.

Duration of the experiment:
4 weeks

I understand what is expected of me during this research.

I know that I will participate in the following trials or tests:
Use the group music recommender application and fill out a questionnaire afterwards.

I know that my participation may be associated to risks or discomforts:
It is possible that I don't like the generated playlist and I have to delete it.

I or others can benefit from this research in the following ways:
I will help the thesis research to investigate the user satisfaction, accuracy and fairness of the application.

My participation offers a contribution to the scientific research. I know that I will not receive any further reward or compensation for my participation.

I understand that my participation to this study is voluntary. I have the right to stop participating at any time. I do not have to give a reason for this and I know that it will not have any negative repercussions for me.

At any time I can also ask to end any further processing of my data and to delete the data that have already been collected.

The results of this study can be used for scientific goals and may be published. My name will not be published. The confidentiality of the data will be protected in all stages of the research. The researchers will take the following measures to protect my privacy:
All the personal user data will be deleted and will never be made public.

I would like to be informed about the results of this research. The researchers may contact me for this purpose using the following e-mail address.

____

Drawn up in duplicate.

Figure B.1: Informed consent part 1

Version April 2023

For questions and for the execution of my rights (access to my data, rectification of the data, …) after my participation I know that I can contact:

alexander.joossens@student.kuleuven.be

More information with regard to privacy in research can be found at https://kuleuven.be/privacy/en/. With further questions about privacy issues I can contact the data protection officer: dpo@kuleuven.be

This study has been reviewed and approved by the Social and Societal Ethics Committee (SMEC) of KU Leuven (*G-2022-6053*). In case of complaints or other concerns with regard to the ethical aspects of this research I can contact SMEC: smec@kuleuven.be

I know that I can contact the individuals/organizations below if I would experience any discomfort or difficulties as a result of some of the subjects that were the topic of this research:

Alexander Joossens (alexander.joossens@student.kuleuven.be)

I have read and understood the information in this document and I have received an answer to all my questions regarding this research. I give my consent to participate.

Date:

Name and signature of the participant | Name and signature of the researcher

Drawn up in duplicate.

FIGURE B.2: Informed consent part 2

KU LEUVEN

GDPR information sheet - last updated May 2020

INFORMATION ON THE PROCESSING OF YOUR PERSONAL DATA

As a result of your participation in the *A Playlist-based Group Music Recommendation System* study, personal data relating to you will be collected and processed. These data will be processed in accordance with the General Data Protection Regulation (GDPR). With this information sheet, we would like to inform you about the use and storage of your data.

In the participant information sheet, you will find further details about the categories of data that will be collected about you during this study. These include personal data such as *age, gender, full name and email address*.

The participant will also have to log in into their Spotify account to use the application, but this login data will not be stored in this research.

Use of your personal data

Only personal data required for the purposes of this study will be collected and processed. More specifically, the study aims to *investigate the usage of a group recommendation system. The application recommends a playlist for groups of users, based on their selected Spotify playlists.* The collected data may possibly be re-used in future studies.

Data collected for this study will be pseudonymised. This means that data that might identify you, such as *your name or your email address*, will be separated from the other data in the study and replaced by a unique random code. In this way, the data can no longer easily be attributed to a specific data subject. Only the researcher can link the data to a specific individual by means of the unique code. However, this will only happen in exceptional cases, for example if you wish to exercise your right to access, rectify or erase your data. You will also not be identified in publications arising from the research.

The data will be processed on the basis of public interest. This means that the research will lead to advances in knowledge and generate insights that (directly or indirectly) benefit society.

Your data will be stored by the researchers for *10 years* after the end of the study at a secure storage location at KU Leuven. After this period, your personal data will be permanently deleted if they are no longer needed for the purposes of the research.

Your rights

You have the right to request more information about the use of your data. In addition, you have the right to access, rectify or erase your data unless exercising these rights would render impossible or seriously impair the achievement of the research objectives.

If you wish to exercise one of these rights, please contact the researchers using the contact details at the bottom of this information sheet.

FIGURE B.3: Informatie over verwerking persoonsgegevens 1

KU LEUVEN

GDPR information sheet - last updated May 2020

Contact details

For the purposes of this research, KU Leuven is the data controller. More specifically, only the researchers involved *Alexander Joossens, Ivania Nadine Donoso Guzmán and Prof. Katrien Verbert* will have access to your personal data. Should you have any specific questions about this study, including the processing of your personal data, please feel free to contact them.

alexander.joossens@student.kuleuven.be

For any further questions and concerns regarding the processing of your personal data, please contact Toon Boon, KU Leuven's data protection officer for research (dpo@kuleuven.be) . Please specify the study concerned by mentioning the title as well as the names of the researchers involved.

If, after contacting the data protection officer, you would still like to lodge a complaint about the use of your personal data, you can contact the Belgian Data Protection Authority (www.gegevensbeschermingsautoriteit.be).

FIGURE B.4: Informatie over verwerking persoonsgegevens 2

Appendix C

Initial questionnaire

Hoe oud ben je? How old are you?
Hoeveel jaar gebruik je al Spotify? For how many years have you been using Spotify?
Hoeveel uur per week gebruik je Spotify? How many hours a week do you use Spotify?
Heb je ooit de Blend/Friend Mix functie van Spotify gebruikt (die automatisch een gemixte playlist voor jou en een vriend creëert)? Have you ever used the Blend/Friend Mix function of Spotify (that automatically creates a mixed playlist for you and a friend)
Voor hoeveel personen zou je een gemixte playlist willen laten genereren? For how many persons would you like to generate a mixed playlist?
Voor welke situaties zou je een blended/gemixte playlist gebruiken? Voel je vrij om een nieuwe situatie toe te voegen als optie. For which situations would you use a blended/mixed playlist? Feel free to add a new situation as an option.
Zou je deze playlist meestal alleen beluisteren of vaker met je vrienden samen? Would you most often listen to this mixed playlist alone or together with your friends?
Wat vind je van de huidige gemixte playlist die Spotify automatisch genereert? What do you think of the current mixed playlist that Spotify generates?
Stel je voor: Je wilt een feest geven met enkele vrienden en je wilt Spotify een playlist laten genereren. Zou je het een leuke feature vinden als iedereen enkele van zijn playlists kan selecteren en Spotify hiervan een mixed playlist maakt?
Imagine: You want to throw a party with some friends and you want Spotify to generate a playlist. Would you like to have a feature that allows everyone to select some of their playlists and let Spotify create a mixed playlist from this?
Momenteel is dit niet mogelijk in Spotify. Zou je graag de functie hebben in Spotify waarmee je de gemixte playlist mee vorm kan geven? Currently this is not possible in Spotify. Would you like to have a feature in Spotify that allows you to have influence on the mixed playlist?
Wat vind je de belangrijkste gemeenschappelijke factoren bij het genereren van een blended/gemixte playlist met vrienden? What do you find the most important mutual factors for the generation of a blended/mixed playlist with friends?
Geef aub enkele andere features/invloeden die je graag zou willen hebben bij het genereren van een blended/gemixte playlist. Please give some other features/influences that you would like to have for the generations of a blended/mixed playlist.

FIGURE C.1: Initial questionnaire

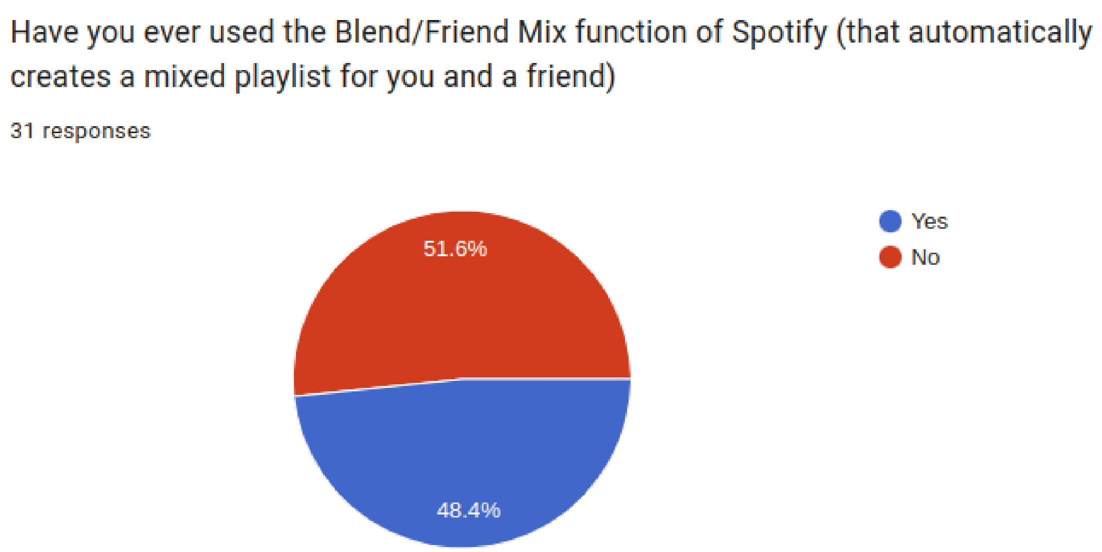


Figure C.2: Have you ever used the Blend/Friend Mix function of Spotify?

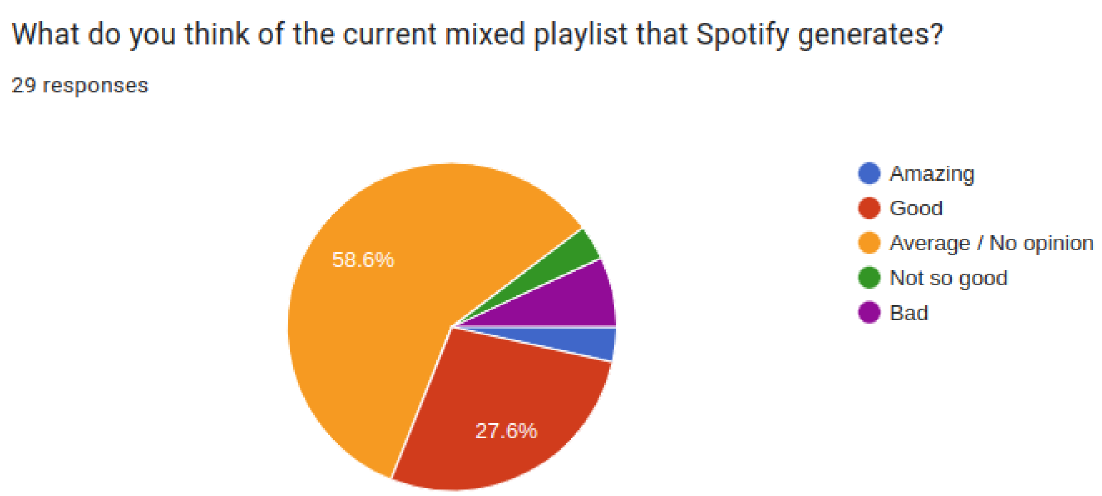


Figure C.3: What do you think of the current mixed playlist that Spotify generates?

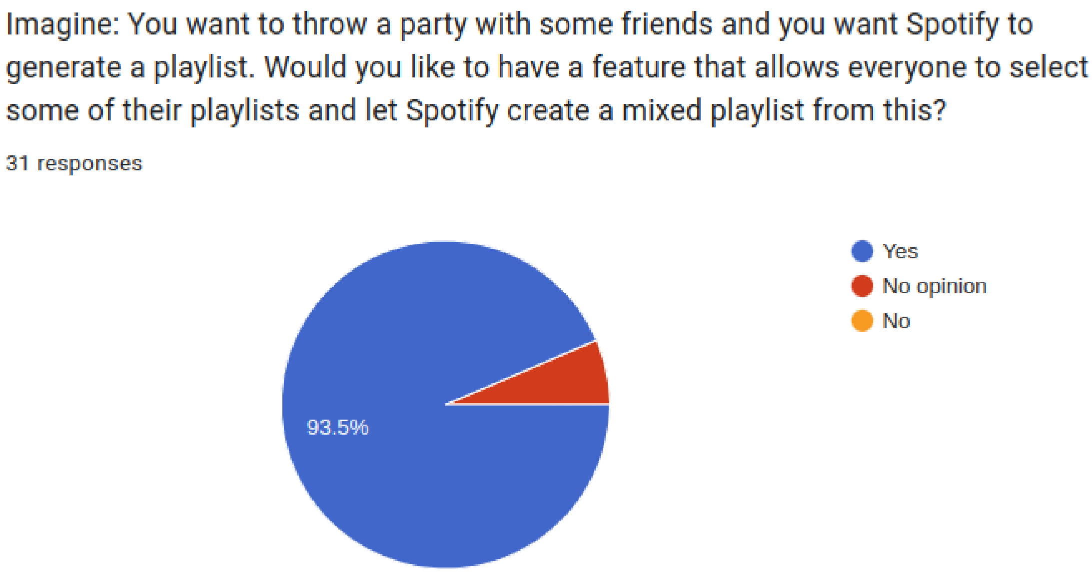


FIGURE C.4: Would you like to have a feature that allows everyone to select some of their playlists and let Spotify create a mixed playlist from this?

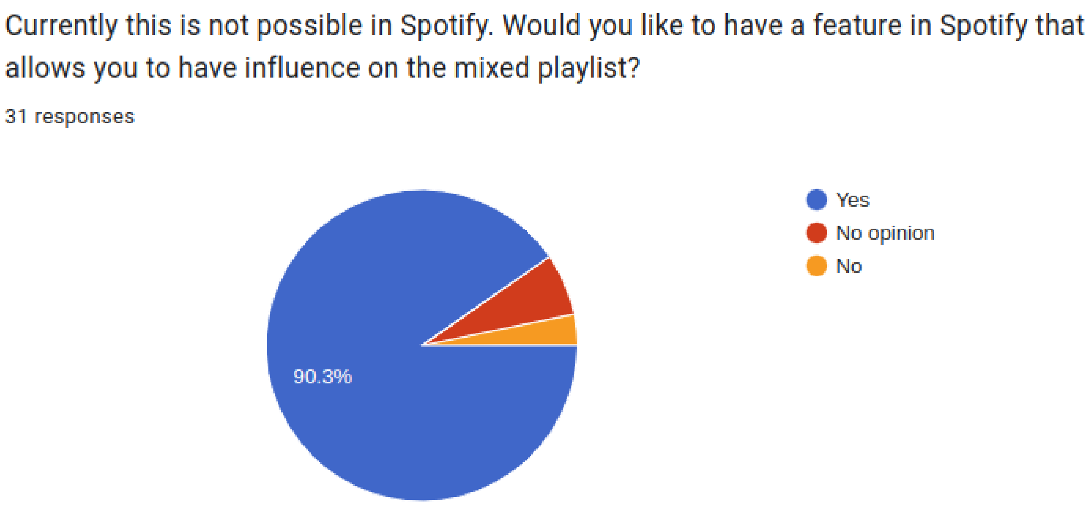


FIGURE C.5: Currently this is not possible in Spotify. Would you like to have a feature in Spotify that allows you to have influence on the mixed playlist?

Geef aub enkele andere features/invloeden die je graag zou willen hebben bij het genereren van een blended/gemixte playlist.

Please give some other features/influences that you would like to have for the generations of a blended/mixed playlist.

9 responses

ones I kept

Mogelijkheid tot het zien uit wie zijn playlist elk nummer in de gemixte playlist komt.

Iets wat ik al ontzettend lang mis in Spotify, is playlists kunnen maken adhv 'moods'. Dit is natuurlijk wel wat moeilijker aangezien iedereen een andere stemming voelt bij een liedje. Maar ik denk dat iedereen wel regelmatig naar een playlist luistert met een bepaalde mood als de bepalende factor van die playlist. Spotify heeft deze "gevoelens-playlists" zelf al gemaakt door basis-emoties te categoriseren met liedjes die zij onder dat gevoel categoriseren. Echter, de gebruiker zelf is nog niet in staat om een eigen playlist aan te maken adhv die emoties.

Hangt ook weer af van de context, maar misschien is het wel leuk om te luisteren naar de meest afgespeelde of favoriet gemaakte nummers/albums van vrienden

volgorde van liedjes bepalen - een 'beheerder' die liedjes kan verwijderen, toevoegen, accepteren,... - tegelijk luisteren met vrienden (zelfde liedje op zelfde tijdstip)

Dat hij een volgorde aanmaakt waarbij de overgang van ene nummer in de andere kei goed is (met die fade in optie)

Figure C.6: Please give some other features/influences that you would like to have for the generations of a blended/mixed playlist

Appendix D

ANOVA and Tukey Results

FIGURE D.1: ANOVA and Tukey test on *behavioural intentions* and *perceived explanations* constructs

(A) ANOVA + Tukey on *behavioural intentions* construct

```
                 sum_sq    df          F    PR(>F)
Algo           2.383956   1.0   0.788829  0.377257
Visual         9.089704   1.0   3.007701  0.086924
Algo:Visual   38.747612   1.0  12.821234  0.000602
Residual     229.682930  76.0        NaN       NaN
tukey result:
     Multiple Comparison of Means - Tukey HSD, FWER=0.05
=============================================================
   group1     group2   meandiff p-adj   lower   upper  reject
-------------------------------------------------------------
PostContent   PostUser   0.7177 0.5623 -0.7263  2.1618  False
PostContent PreContent   1.0466 0.2351 -0.3974  2.4907  False
PostContent    PreUser  -1.0194 0.2565 -2.4635  0.4247  False
   PostUser PreContent   0.3289 0.9323 -1.1152   1.773  False
   PostUser    PreUser  -1.7371 0.0119 -3.1812 -0.2931   True
 PreContent    PreUser  -2.0661 0.0019 -3.5101  -0.622   True
-------------------------------------------------------------
```

(B) ANOVA + Tukey on *perceived explanation* construct

```
                 sum_sq    df         F    PR(>F)
Algo           0.535997   1.0  0.100794  0.751749
Visual         0.117125   1.0  0.022025  0.882413
Algo:Visual   33.850045   1.0  6.365509  0.013725
Residual     404.147295  76.0       NaN       NaN
tukey result:
    Multiple Comparison of Means - Tukey HSD, FWER=0.05
============================================================
   group1     group2   meandiff p-adj   lower  upper  reject
------------------------------------------------------------
PostContent   PostUser   1.2244 0.3418 -0.6911   3.14  False
PostContent PreContent   1.1373 0.4077 -0.7783 3.0528  False
PostContent    PreUser  -0.2402 0.9876 -2.1558 1.6753  False
   PostUser PreContent  -0.0872 0.9994 -2.0027 1.8284  False
   PostUser    PreUser  -1.4647 0.1939 -3.3802 0.4509  False
 PreContent    PreUser  -1.3775 0.2414  -3.293  0.538  False
------------------------------------------------------------
```

FIGURE D.2: ANOVA on *perceived accuracy* and *perceived fairness* constructs

(A) ANOVA on *perceived accuracy* construct

```
                 sum_sq    df         F    PR(>F)
Algo           1.870146   1.0  1.030935  0.313161
Visual         3.226533   1.0  1.778655  0.186298
Algo:Visual    1.020047   1.0  0.562310  0.455646
Residual     137.866227  76.0       NaN       NaN
```

(B) ANOVA on *perceived fairness* construct

```
                 sum_sq    df         F    PR(>F)
Algo           1.120292   1.0  0.728009  0.396211
Visual         2.048648   1.0  1.331290  0.252191
Algo:Visual    2.732282   1.0  1.775542  0.186679
Residual     116.952180  76.0       NaN       NaN
```

Figure D.3: ANOVA on *ease of use* and *NASA TLX* constructs

(a) ANOVA on *ease of use* construct

	sum_sq	df	F	PR(>F)
Algo	0.012498	1.0	0.009611	0.922164
Visual	0.312450	1.0	0.240263	0.625428
Algo:Visual	2.812053	1.0	2.162367	0.145555
Residual	98.834279	76.0	NaN	NaN

(b) ANOVA on *NASA TLX* construct

	sum_sq	df	F	PR(>F)
Algo	1.120292	1.0	0.728009	0.396211
Visual	2.048648	1.0	1.331290	0.252191
Algo:Visual	2.732282	1.0	1.775542	0.186679
Residual	116.952180	76.0	NaN	NaN

Figure D.4: ANOVA on *'Add playlist to my Spotify account* button clicks

(a) ANOVA on *'Add playlist to my Spotify account* button clicks

	sum_sq	df	F	PR(>F)
Algo	0.05	1.0	0.192893	0.661766
Visual	0.05	1.0	0.192893	0.661766
Algo:Visual	0.20	1.0	0.771574	0.382499
Residual	19.70	76.0	NaN	NaN

Bibliography

[1] Babak Joze Abbaschian and Samira Khorshidi. A review of hybrid recommender systems, 2017.

[2] Adiyansjah, Alexander A.S. Gunawan, and Derwin Suhartono. Music recommender system based on genre using convolutional recurrent neural networks. *Procedia Computer Science*, 157:99–109, 1 2019. ISSN 1877-0509. doi: 10.1016/J.PROCS.2019.08.146.

[3] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, 2005.

[4] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alex Tuzhilin. Context-aware recommender systems. *AI Magazine*, 32:67–80, 2011. ISSN 07384602. doi: 10.1609/AIMAG.V32I3.2364. URL https://www.researchgate.net/publication/220605653_Context-Aware_Recommender_Systems.

[5] Fabio Aiolli. A preliminary study on a recommender system for the million songs dataset challenge "bioinfogen: Bioinformatics for personal genomics" view project multiple kernel learning algorithms view project a preliminary study on a recommender system for the million songs dataset challenge, 2013. URL https://www.researchgate.net/publication/256373605.

[6] Mohammed Z. Al-Taie and Seifedine Kadry. Visualization of explanations in recommender systems. *Journal of Advanced Management Science*, 2:140–144, 2014. ISSN 21680787. doi: 10.12720/joams.2.2.140-144.

[7] A Ashu and Jena Debasish. Modeling a rule based movie recommender system for group preferences. 2012.

[8] Francesco Barile, Antonio Caso, and Silvia Rossi. Group recommendation for smart applications: a multi-agent view of the problem. 2014.

[9] Andrea Barraza-Urbina. The exploration-exploitation trade-off in interactive recommender systems. *RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems*, pages 431–435, 8 2017. doi: 10.1145/3109859.3109866. URL https://dl.acm.org/doi/10.1145/3109859.3109866.

[10] Luke Barrington, Reid Oda, and Gert Lanckriet. Human evaluation of music recommender systems. 2009. URL www.last.fm/api.

[11] Asia J. Biega, Krishna P. Gummadi, and Gerhard Weikum. Equity of attention: Amortizing individual fairness in rankings. pages 405–414. Association for Computing Machinery, Inc, 6 2018. ISBN 9781450356572. doi: 10.1145/3209978.3210063.

[12] Ludovico Boratto. Challenges and solutions in group recommender systems, 2017. URL https://www.slideshare.net/LudovicoBoratto/challenges-and-solutions-in-group-recommender-systems.

[13] Ludovico Boratto, Salvatore Carta, and Gianni Fenu. Discovery and representation of the preferences of automatically detected groups: Exploiting the link between group modeling and clustering. *Future Generation Computer Systems*, 64:165–174, 11 2016. ISSN 0167739X. doi: 10.1016/J.FUTURE.2015.10.007.

[14] Svetlin Bostandjiev, John O'Donovan, and Tobias Höllerer. Tasteweights. page 35. ACM Press, 2012. ISBN 9781450312707. doi: 10.1145/2365952.2365964. URL http://dl.acm.org/citation.cfm?doid=2365952.2365964.

[15] Khac Hoai Nam Bui, Xuan Hau Pham, Jason J. Jung, O. Joun Lee, and Min Sung Hong. Context-based traffic recommendation system. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 165:122–131, 2016. ISSN 18678211. doi: 10.1007/978-3-319-29236-6_13. URL https://www.researchgate.net/publication/284398188_Context-Based_Traffic_Recommendation_System.

[16] Giuseppe Carenini and Johanna D. Moore. An empirical study of the influence of argument conciseness on argument effectiveness. pages 150–157, 2000. doi: 10.3115/1075218.1075238. URL https://dl.acm.org/doi/10.3115/1075218.1075238.

[17] Berardina Nadja De Carolis and Sebastiano Pizzutilo. An active environment to manage user adapted interactions, 2007. URL https://www.academia.edu/19222020/An_Active_Environment_to_Manage_User_Adapted_Interactions.

[18] Dennis L. Chao, Justin Balthrop, and Stephanie Forrest. Adaptive radio. page 120. ACM Press, 2005. ISBN 1595932232. doi: 10.1145/1099203.1099224. URL http://portal.acm.org/citation.cfm?doid=1099203.1099224.

[19] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations, 2016. URL https://research.google/pubs/pub45530/.

[20] Amine Dadoun, Michael Defoin-Platel, Thomas Fiig, Corinne Landra, and Raphaël Troncy. How recommender systems can transform airline offer construction and retailing. *Journal of Revenue and Pricing Management*, 20:301–315, 6 2021. ISSN 1477657X. doi: 10.1057/s41272-021-00313-2.

[21] Karlijn Dinnissen and Christine Bauer. Fairness in music recommender systems: A stakeholder-centered mini review. *Frontiers in Big Data*, 5: 913608, 7 2022. ISSN 2624909X. doi: 10.3389/FDATA.2022.913608. URL /pmc/articles/PMC9353048//pmc/articles/PMC9353048/?report=abstracthttps://www.ncbi.nlm.nih.gov/pmc/articles/PMC9353048/.

[22] Gideon Dror, Noam Koenigstein, Yehuda Koren, Markus Weimer, G Dror, Y Koren, and M Weimer. The yahoo! music dataset and kdd-cup'11, 6 2012. ISSN 1938-7228. URL https://proceedings.mlr.press/v18/dror12a.html.

[23] Mehdi Elahi. 2 illustration of cold start problem in recommender systems: New user... | download scientific diagram, 2019. URL https://www.researchgate.net/figure/Illustration-of-Cold-Start-problem-in-recommender-systems-New-user-problem-left-and_fig2_332511384.

[24] Guang Fang, Lei Su, Di Jiang, and Liping Wu. Group recommendation systems based on external social-trust networks. 2018. doi: 10.1155/2018/6709607. URL https://doi.org/10.1155/2018/6709607.

[25] Zeshan Fayyaz, Mahsa Ebrahimian, Dina Nawara, Ahmed Ibrahim, and Rasha Kashef. Recommendation systems: Algorithms, challenges, metrics, and business opportunities. *Applied Sciences (Switzerland)*, 10:1–20, 11 2020. ISSN 20763417. doi: 10.3390/app10217748.

[26] A. Felfernig, N. Tintarev, T. N. T. Trang, and M. Stettinger. Designing explanations for group recommender systems. 2 2021. URL http://arxiv.org/abs/2102.12413.

[27] Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalcic. *Group recommender systems : an introduction*. 2018. ISBN 9783319750668.

[28] Mike Gartrell, Xinyu Xing, Qin Lv, Aaron Beach, Richard Han, Shivakant Mishra, and Karim Seada. Enhancing group recommendation by incorporating social relationship interactions. 2010.

[29] Mustansar Ali Ghazanfar and Adam Prugel-Bennett. A scalable, accurate hybrid recommender system. *3rd International Conference on Knowledge Discovery and Data Mining, WKDD 2010*, pages 94–98, 2010. doi: 10.1109/WKDD.2010.117. URL https://www.researchgate.net/publication/221306167_A_Scalable_Accurate_Hybrid_Recommender_System.

[30] Sofia Gkika and George Lekakos. The persuasive role of explanations in recommender systems, 2014.

[31] Sajal Halder, A. M.Jehad Sarkar, and Young Koo Lee. Movie recommendation system based on movie swarm. *Proceedings - 2nd International Conference on Cloud and Green Computing and 2nd International Conference on Social Computing and Its Applications, CGC/SCA 2012*, pages 804–809, 2012.

doi: 10.1109/CGC.2012.121. URL https://www.researchgate.net/publication/261077583_Movie_Recommendation_System_Based_on_Movie_Swarm.

[32] Sandra G Hart, Moffett Field, California Lowell, and E Staveland. Development of nasa-tlx (task load index)" results of empirical and theoretical research, 1988.

[33] Chen He, Denis Parra, and Katrien Verbert. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications*, 56, 2016. ISSN 09574174. doi: 10.1016/j.eswa.2016.02.013.

[34] Jonathan L Herlocker, Joseph A Konstan, and John Riedl. Explaining collaborative filtering recommendations. 2000. URL http://www.grouplens.org/.

[35] Nyi Nyi Htun, Elisa Lecluse, and Katrien Verbert. Perception of fairness in group music recommender systems. 2021. doi: 10.1145/3397481.3450642.

[36] Li Tze Hu and Peter M. Bentler. Cutoff criteria for fit indexes in covariance structure analysis: Conventional criteria versus new alternatives. *Structural Equation Modeling*, 6:1–55, 1999. ISSN 10705511. doi: 10.1080/10705519909540118.

[37] Jamil Itmazi and Miguel Gea. (pdf) the recommendation systems, types, domains and the ability usage in learning management system., 2006. URL https://www.researchgate.net/publication/281861663_The_Recommendation_Systems_Types_Domains_and_the_Ability_Usage_in_Learning_Management_System.

[38] Anthony Jameson. More than the sum of its members: Challenges for group recommender systems, 2004. URL http://www.movielens.umn.edu/.

[39] George Karypis. Evaluation of item-based top-n recommendation algorithms £. 2001.

[40] Harleen Kaur and Gourav Bathla. Techniques of recommender system. *International Journal of Innovative Technology and Exploring Engineering*, 8:373–379, 7 2019. ISSN 22783075. doi: 10.35940/IJITEE.I1059.0789S19.

[41] Mesut Kaya, Derek Bridge, and Nava Tintarev. Ensuring fairness in group recommendations by rank-sensitive balancing of relevance. pages 101–110. ACM, 9 2020. ISBN 9781450375832. doi: 10.1145/3383313.3412232. URL https://dl.acm.org/doi/10.1145/3383313.3412232.

[42] Shah Khusro, Zafar Ali, and Irfan Ullah. Recommender systems: Issues, challenges, and research opportunities. *Lecture Notes in Electrical Engineering*, 376, 2016. doi: 10.1007/978-981-10-0557-2_112.

[43] Jae Kyeong Kim, Hyea Kyeong Kim, Hee Young Oh, and Young U. Ryu. A group recommendation system for online communities. *International Journal of Information Management*, 30:212–219, 2010. ISSN 02684012. doi: 10.1016/J.IJINFOMGT.2009.09.006. URL https://www.researchgate.net/publication/223588426_A_group_recommendation_system_for_online_communities.

[44] Pigi Kouki, James Schaffer, Jay Pujara, John O'Donovan, and Lise Getoor. Personalized explanations for hybrid recommender systems. volume Part F147615, 2019. doi: 10.1145/3301275.3302306.

[45] Madhusree Kuanr and Puspanjali Mohapatra. Recent challenges in recommender systems: A survey. *Advances in Intelligent Systems and Computing*, 1199:353–365, 2021. ISSN 21945365. doi: 10.1007/978-981-15-6353-9_32.

[46] Yu Liang and Martijn C. Willemsen. Exploring the longitudinal effects of nudging on users' music genre exploration behavior and listening preferences. volume 22, pages 3–13. ACM, 9 2022. ISBN 9781450392785. doi: 10.1145/3523227.3546772. URL https://dl.acm.org/doi/10.1145/3523227.3546772.

[47] Mani Madhukar. Challenges limitation in recommender systems, 2021.

[48] Judith Masthoff. Group recommender systems: Combining individual models, 2011. URL http://link.springer.com/10.1007/978-0-387-85820-3_21.

[49] Joseph F Mccarthy and Theodore D Anagnost. Musicfx: An arbiter of group preferences for computer supported collaborative workouts. 2000.

[50] Brian McFee, Thierry Bertin-Mahieux, Daniel P.W. Ellis, and Gert R.G. Lanckriet. The million song dataset challenge. page 909. ACM Press, 2012. ISBN 9781450312301. doi: 10.1145/2187980.2188222. URL http://dl.acm.org/citation.cfm?doid=2187980.2188222.

[51] Martijn Millecamp, Nyi Nyi Htun, Yucheng Jin, and Katrien Verbert. Effects of personal characteristics on music recommender user interfaces. pages 101–109. Association for Computing Machinery, Inc, 7 2018. ISBN 9781450355896. doi: 10.1145/3209219.3209223.

[52] Thi Ngoc, Trang Tran, Müslüm Atas, Alexander Felfernig, Viet Man Le, Ralph Samer, Martin Stettinger, Man Le, Jennifer B Sartor, Theo D ' Hondt, and Wolfgang De. Towards social choice-based explana-tions in group recommender systems. 2019. doi: 10.1145/3320435.3320437. URL https://doi.org/10.1145/3320435.3320437.

[53] Reza Barzegar Nozari and Hamidreza Koohi. A novel group recommender system based on members' influence and leader impact. 205:106296, 2020. doi: 10.1016/j.knosys.2020.106296. URL https://doi.org/10.1016/j.knosys.2020.106296.

[54] A. Ant Ozok, Quyin Fan, and Anthony F. Norcio. Design guidelines for effective recommender system interfaces based on a usability criteria conceptual model: Results from a college student population. *Behaviour and Information Technology*, 29:57–83, 1 2010. ISSN 0144929X. doi: 10.1080/01449290903004012.

[55] Sasmita Panigrahi, Rakesh Ku Lenka, and Ananya Stitipragyan. A hybrid distributed collaborative filtering recommender engine using apache spark. volume 83, pages 1000–1006. Elsevier, 2016. doi: 10.1016/j.procs.2016.04.214.

[56] Umberto Panniello, Michele Gorgoglione, and Cosimo Palmisano. Comparing pre-filtering and post-filtering approach in a collaborative contextual recommender system: An application to e-commerce. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5692 LNCS:348–359, 2009. ISSN 16113349. doi: 10.1007/978-3-642-03964-5_32/COVER. URL https://link.springer.com/chapter/10.1007/978-3-642-03964-5_32.

[57] Alexis Papadimitriou, Panagiotis Symeonidis, and Yannis Manolopoulos. A generalized taxonomy of explanations styles for traditional and social recommender systems. *Data Mining and Knowledge Discovery*, 24, 2012. ISSN 13845810. doi: 10.1007/s10618-011-0215-0.

[58] Bansari Patel, Palak Desai, and Urvi Panchal. Methods of recommender system: A review. volume 2018-January, pages 1–4. IEEE, 3 2017. ISBN 978-1-5090-3294-5. doi: 10.1109/ICIIECS.2017.8275856. URL http://ieeexplore.ieee.org/document/8275856/.

[59] Toon De Pessemier, Simon Dooms, and Luc Martens. Design and evaluation of a group recommender system. 2012.

[60] W Pnnz, M Jarke, Y Rogers, K Schmidt, V Wulf, Mark O'connor, Dan Cosley, Joseph A Konstan, and John Riedl. Polylens: A recommender system for groups of users. pages 199–218, 2001.

[61] Pearl Pu and Li Chen. Trust building with explanation interfaces. 2006. URL www.activedecisions.com.

[62] Pearl Pu, Li Chen, and Rong Hu. A user-centric evaluation framework for recommender systems. *RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems*, pages 157–164, 2011. doi: 10.1145/2043932.2043962. URL https://dl.acm.org/doi/10.1145/2043932.2043962.

[63] Javier Pérez-Marcos and Vivian López Batista. Recommender system based on collaborative filtering for spotify's users. *Advances in Intelligent Systems and Computing*, 619:214–220, 2017. ISSN 21945357. doi: 10.1007/978-3-319-61578-3_22. URL https://www.researchgate.net/publication/318511102_Recommender_System_Based_on_Collaborative_Filtering_for_Spotify's_Users.

[64] James Schaffer, Prasanna Giridhar, Debra Jones, Tobias Höllerer, Tarek Abdelzaher, and John O'donovan. Getting the message? a study of explanation interfaces for microblog data analysis. volume 2015-January, 2015. doi: 10.1145/2678025.2701406.

[65] Markus Schedl, Hamed Zamani, Ching-Wei Chen, Yashar Deldjoo, · Mehdi Elahi, and Mehdi Elahi. Current challenges and visions in music recommender systems research. *International Journal of Multimedia Information Retrieval*, 7:95–116, 2018. doi: 10.1007/s13735-018-0154-2. URL https://doi.org/10.1007/s13735-018-0154-2.

[66] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction second edition, in progress. 2018.

[67] Nava Tintarev and Judith Masthoff. A survey of explanations in recommender systems. pages 801–810, 2007. ISBN 1424408326. doi: 10.1109/ICDEW.2007.4401070.

[68] Nava Tintarev and Judith Masthoff. *Recommender Systems Handbook*. Springer US, 10 2011. ISBN 978-0-387-85819-7. doi: 10.1007/978-0-387-85820-3. URL http://link.springer.com/10.1007/978-0-387-85820-3.

[69] Nava Tintarev and Judith Masthoff. Evaluating the effectiveness of explanations for recommender systems: Methodological issues and empirical studies on the impact of personalization. *User Modeling and User-Adapted Interaction*, 22:399–439, 10 2012. ISSN 09241868. doi: 10.1007/s11257-011-9117-5.

[70] Chun-Hua Tsai and Peter Brusilovsky. Explaining recommendations in an interactive hybrid social recommender. volume Part F147615, pages 391–396. ACM, 3 2019. ISBN 9781450362726. doi: 10.1145/3301275.3302318. URL https://dl.acm.org/doi/10.1145/3301275.3302318.

[71] Christian Villavicencio, Silvia Schiaffino, J Andrés DÍAZ-PACE, and Ariel Monteserin. A group recommendation system for movies based on mas, 2016. URL http://adcaij.usal.es.

[72] Wei Wang, Guangquan Zhang, and Jie Lu. Hierarchy visualization for group recommender systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49:1152–1163, 6 2019. ISSN 21682232. doi: 10.1109/TSMC.2017.2699690.

[73] Xiangshi Wang, Lei Su, Qihang Zhou, and Liping Wu. Group recommender systems based on members' preference for trusted social networks. *Security and Communication Networks*, 2020:1–11, 5 2020. ISSN 1939-0114. doi: 10.1155/2020/1924140. URL https://www.hindawi.com/journals/scn/2020/1924140/.

[74] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. A survey on the fairness of recommender systems. *ACM Transactions on Information Systems*, 37: 45, 6 2022. doi: 10.1145/3547333. URL http://arxiv.org/abs/2206.03761http://dx.doi.org/10.1145/3547333.

[75] Lin Xiao, Zhang Min, Zhang Yongfeng, Gu Zhaoquan, Liu Yiqun, and Ma Shaoping. Fairness-aware group recommendation with pareto-efficiency. pages 107–115. ACM, 8 2017. ISBN 9781450346528. doi: 10.1145/3109859.3109887. URL https://dl.acm.org/doi/10.1145/3109859.3109887.

[76] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. 7 2017. doi: 10.1145/3285029. URL http://arxiv.org/abs/1707.07435http://dx.doi.org/10.1145/3285029.

A playlist-based Group Recommender System

ALEXANDER JOOSSENS, Department of Computer Science, KU Leuven, Belgium

Fig. 1. Homepage of the playlist-based group recommender system

Recommender systems have a profound influence on our choices and preferences, shaping our decisions in various domains. However, their lack of transparency remains a persistent challenge. This research focuses on filtering algorithms and visual explanations in a group music recommender system. Users actively steer recommendations through playlist selection. A user study with 80 participants reveals the positive impact of post-filtering and content-based explanations on perceived explanation and behavioural intentions. Findings have implications for improving commercial music applications like Spotify Blend.

1 INTRODUCTION

In today's digital era, the large amount of information and options available can make it difficult for users to find what they need. Recommender systems are essential tools designed to help with this issue. They play a major role in the decisions of the movies we watch, the restaurants we visit or the music we listen to.

A crucial challenge encountered in numerous recommender systems is their perceived black box nature, leaving users unaware of the underlying basis of their recommendations. Showing the user an explanation about the algorithm behind the recommender system has been proven to have a positive influence on the user's user experience [1, 3].

User-based and content-based explanations are two approaches used in recommender systems to provide insights about recommendations. User-based explanations focus on users' preferences and behaviours, considering the choices and actions of similar users. Content-based explanations, on the other hand, emphasise the attributes and characteristics of the recommended items.

Author's address: Alexander Joossens, Department of Computer Science, KU Leuven, Belgium.

Also in group recommender systems, this is an important challenge. In such systems, the traditional objective is twofold: to provide relevant recommendations to each individual in the group and to ensure that all users have an equal say in the final outcome. However, providing the users of a group recommender system with (visual) explanations can be more complex than for individual recommender systems. Since they have to account for more users, the visual presentations can quickly become cluttered and messy. This is confirmed by Htun et al. [4], who state that scalability is an important aspect of visualisations. They also state that a lot depends on the number of users of the application.

Another concept that plays an important role on the user's perceptions is the use of pre- and post-filtering. In pre-filtering, input data is adjusted or weighted before being used in the recommender algorithm. For instance, individual user data can be combined into a single data set. In post-filtering, data changes occur after the recommender algorithm is executed. For example, recommended item lists can be filtered to align with the interests of all users equally.

This paper aims to build and evaluate a new group music recommender system using the Spotify Web API, focusing on different filtering algorithms and visual explanations of the generated playlist. Users can actively influence the system by selecting their playlists, steering the recommendations.

The two research questions with their respective hypotheses are described in Section 2. After this follows an explanation about the used definitions of the researched evaluation factors in Section 3. The application design is discussed in Section 4, including the questionnaire, think-aloud studies, application flow, visualisations, and algorithm details. Implementation details are described in Section 5, covering the backend, frontend, and Spotify Web API. Section 6 presents the user study, including research design, results, and data analysis. The discussion of the research results is provided in Section 7, followed by the conclusion in Section 8.

2 RESEARCH QUESTIONS AND HYPOTHESES

The two research questions are:

- How do pre- and post-filtering in a group recommendation system influence the perceived accuracy, fairness, explanation and behavioural intentions?
- How do different visual explanations of group recommendations affect the perceived fairness, accuracy, explanation, behavioural intentions, and ease of use?

The hypotheses are formulated as:

- *The pre-filtering algorithm will score better on perceived fairness and accuracy but lower on behavioural intentions and perceived explanation*
- *The visualisation with the user-based explanation style will score higher on the perceived fairness and explanation, while the visualisation with the content-based explanation style will score higher on perceived accuracy and behavioural intentions*

3 DEFINITIONS OF EVALUATION FACTORS

Here we discuss our used definitions for perceived accuracy, fairness, explanation, behavioural intentions, and ease of use in a group recommender system.

Perceived accuracy: For this evaluation factor we use the definition of Felfernig et al. [2], who defines accuracy in a group recommender system as three metrics: '*(1) classification metrics that evaluate to which extent a recommender is able to determine items of relevance (interest) for the user, (2) error metrics that evaluate how well a recommender predicts ratings, and (3) ranking metrics that help to evaluate how well a recommender predicts the importance ranking of items*'.

Perceived fairness: Fairness in a group recommender system is the extent of balance between group members' specific utilities [2]. Htun et al. [4] define it as the degree of imbalance between individual utilities within the group.

Behavioural intentions: This concept refers to users' intentions and actions after using the system. It includes factors such as the intention to recommend the system to friends [6].

Perceived explanation: An explanation in recommender systems justifies the recommendations to help users understand the relevance of the items [7].

Ease of use: Ease of use can be influenced by many aspects of the application. Two important factors include page layout and navigation [5]. For our study, we investigate the ease of use w.r.t. different visualisation types.

4 APPLICATION DESIGN

The research questions were answered by conducting a between-subjects user study. Two group recommender algorithms (pre- and post-filtering) and two visual explanation types (content-based and user-based) were made. An evaluation was conducted with 80 participants, randomly assigned to different scenarios. The goal was to assess the impact on accuracy, fairness, explanation, behaviour, and ease of use. A post-test questionnaire based on the ResQue model [6] was used.

4.1 Initial questionnaire

An initial questionnaire was designed to gather insights into participants' thoughts on Spotify's *Blend* feature, which generates a new playlist for two users. By identifying the weaknesses of Blend, the new web application aimed to improve upon those aspects.

4.2 Iteration 1: Application flow

A first think-aloud study was conducted to decide the exact application flow, and the layout of the web pages. The homepage featured a simple design with an image, title, and a Spotify connect button. The authorisation page followed Spotify's standard, requesting users' permission to access their Spotify data by providing their username and password.

Once authorised, users' playlists are displayed on the playlist selection page along with a personalised welcome message and profile picture. Two options for verification pages were considered for connecting users. The first option allows the user to select a follower from their list to create a playlist with. The second option requires entering a unique code received for the research. Depending on the chosen option, two scenarios can occur. If the first user enters the code, they will be redirected to a '*Waiting for your friend*' page until the second user completes the steps. The second user, once finished, immediately sees the final playlist page. The first user receives an email with a link to the final playlist page.

A preview button is provided next to each recommended song on the final playlist, allowing users to listen to a 30-second preview. After reviewing the final playlist and explanatory visualisations, users can click the A '*Add to my Spotify Account*' button to add the recommended playlist to their personal Spotify account.

During the think-aloud study, improvements were made, including enlarging the connect button and clarifying the authorisation process. Most users preferred the follower selection option, but due to API limitations, unique codes were used instead.

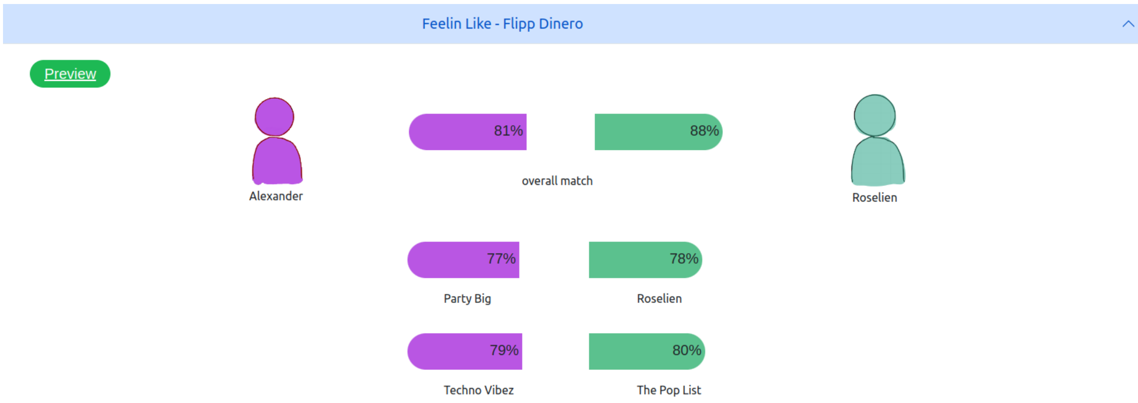


Fig. 2. Visualisation with user-based explanation

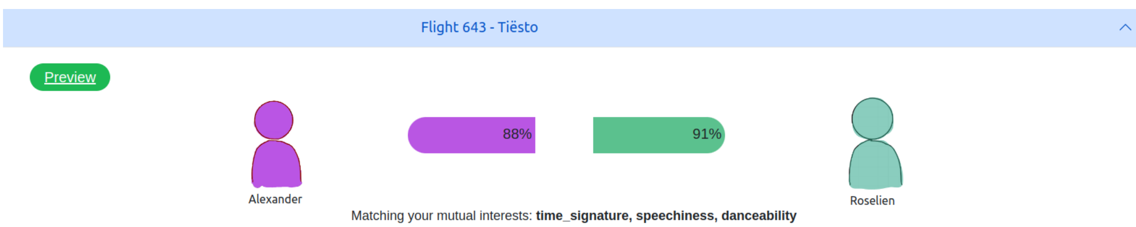


Fig. 3. Visualisation with content-based explanation

4.3 Iteration 2: Final playlist page with user-based explanation

In this iteration, six sketches were made for the final recommended playlist page, containing multiple types of visual explanation. Different styles of user-based and content-based explanations were sketched, and researched through another think-aloud study. From the resulting feedback of the participants, one of the sketches was kept. The sketch contained a user-based explanation, showing the match percentages for each selected playlist w.r.t. the recommended track, and also an overall match percentage of each user's music profile with the recommended track. The final design is shown in figure 2. It was decided that a third iteration was needed for the content-based visual explanation.

4.4 Iteration 3: Final playlist page with content-based explanation

Three more sketches were made and tested through a third think-aloud study. The final design that was selected is shown in figure 3. The visualisation shows a content-based explanation by displaying the top three matching audio characteristics from the recommended track with the user's music profile. Since the overall match percentage feature from the design from iteration 2 was so popular, this feature was kept.

4.5 Pre- and post-filtering algorithms

Two playlist recommendation algorithms were developed, using pre-filtering and post-filtering. Both algorithms take the selected playlists of the users as input and generate a new recommended playlist as output, consisting of twenty songs.

The pre-filtering algorithm combines the selected playlists and goes through four stages: vector generation, mutual song selection, mutual artist selection, and extra recommendations. In the vector generation stage, the algorithm retrieves song data and audio features from the selected playlists, normalises the values, and calculates average characteristic values per playlist and user. Cosine-similarity scores are then calculated for each song and playlist, representing the match percentage with the user's music taste. The mutual song selection stage adds songs that are present in both users' playlists to the recommended playlist. The mutual artist selection stage includes songs from artists that are present in both playlists. If the combined playlist still contains less than twenty songs, the extra recommendations stage complements it with recommended songs based on each user's playlist selection. The final playlist is shuffled to ensure balance between user influences.

The post-filtering algorithm takes a different approach. It goes through two stages: vector generation and individual song recommendations. The vector generation stage is similar to pre-filtering, while the individual song recommendations stage filters the recommended songs based on the most similar characteristics between users. The final playlist is created by combining the filtered individual recommendations and shuffling the songs.

5 APPLICATION IMPLEMENTATION

The source code of the application is available on GitHub. A demo video of the web application is available on YouTube.

The backend of the application uses Express.js as the web application framework, MySQL as the database management system, and Docker for containerisation. The frontend section uses PUG for building web pages, CSS for styling, Bootstrap for the accordion element, and JavaScript for interactivity. The Spotify Web API was used for various API calls, like the *Get Recommendations* method. Finally, the OAuth 2.0 authorisation framework was used to access the users' Spotify data.

6 METHODOLOGY

To answer the two research questions, a between-subjects user study was conducted. A total of 80 participants were recruited in pairs of two, to test the web application under four different use case combinations (using pre- or post-filtering and content- or user-based explanations).

After using the application, the participants filled out a post-test questionnaire, including questions that were derived from various sources, including the ResQue model and NASA-TLX. Each question corresponded to one of five constructs: perceived fairness, accuracy, explanation, behavioural intentions, and ease of use. The questionnaire data was also supplemented with the number of users who clicked the "Add to Playlist" button.

The data analysis included a confirmatory factor analysis (CFA) test to identify the optimal question combination for each construct. A two-way ANOVA test examined the main effects of algorithm type and visualisation type, as well as the interaction effect. Post-hoc Tukey HSD tests were performed to determine specific group differences in the interaction effect.

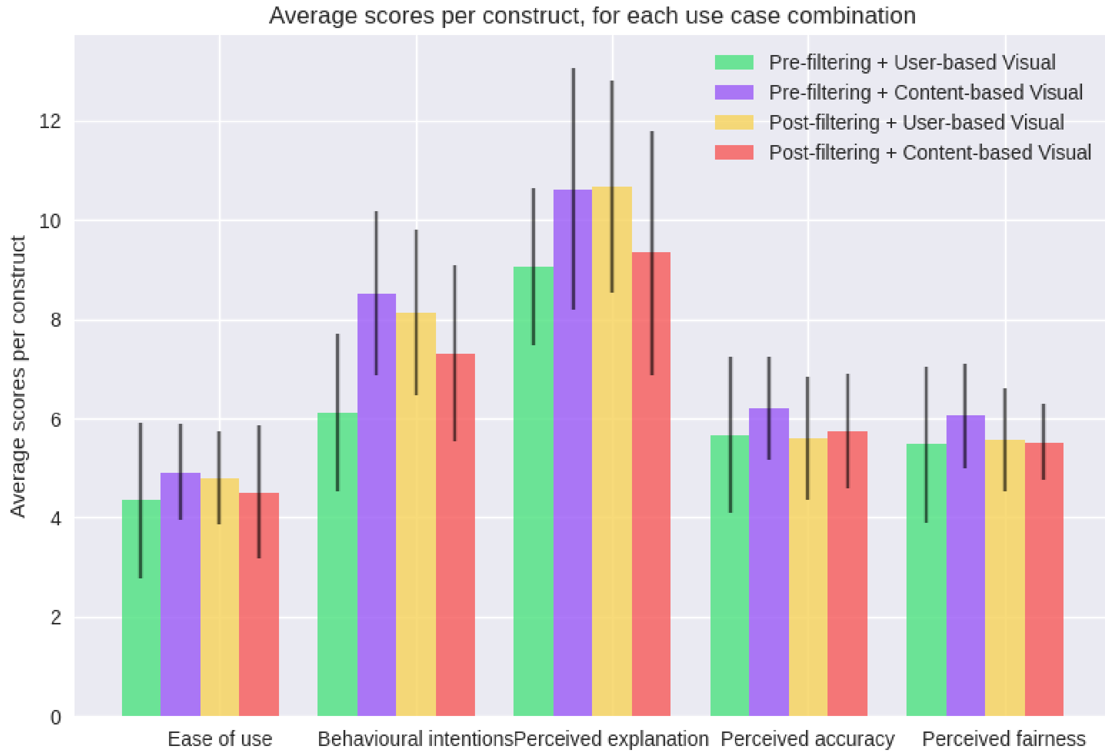


Fig. 4. Average response per construct for every use case combination. The vertical error bars represent the standard deviation.

6.1 Results

6.1.1 Qualitative results. Participants provided feedback through an open-ended question in the questionnaire. The feedback generally indicated that the application provided a positive user experience and was user-friendly. Participants appreciated the final playlist and artist matches. However, some participants expressed negative feedback, citing instances of unusual song combinations and foreign language music in the recommended playlists. They also expressed a desire for more control over the genres and languages of the recommendations.

6.1.2 Quantitative results. After weighing a selection of questions from the CFA with their loading values, five constructs were made, as shown in fig 4. The results from the two-way ANOVA tests indicated some interesting results.

For *Ease of use*, the main effects of algorithm and visualisation type did not reach statistical significance ($p > 0.05$). No significant interaction effect was observed.

Regarding *Behavioural Intentions*, a significant interaction effect was found ($p = 0.000602$). A subsequent post-hoc Tukey HSD test showed that the group using post-filtering and content-based visualisation had significantly higher scores on average on the *behavioural intentions* construct compared to the group using pre-filtering and/or user-based visualisation.

There was a significant interaction effect observed for the construct *Perceived explanation* (p-value = 0.013725). A subsequent post-hoc Tukey HSD revealed that the group using the post-filtering algorithm and content-based visualisation had a higher average score on the *perceived explanation* construct compared to the group using the pre-filtering algorithm and content-based visualisation.

For the *Perceived accuracy and fairness* constructs, no significant main effects or interaction effects were observed.

ANOVA tests on the button click logs and cognitive load assessments showed no significant effects either.

7 DISCUSSION

The first research question explores how pre- and post-filtering in a group recommender system affect perceived accuracy, fairness, explanation, and behavioural intentions. The second research question investigates the impact of different visual explanations on perceived fairness, accuracy, explanation, behavioural intentions, and ease of use.

The study findings indicate that the combination of post-filtering with content-based visualisation leads to higher scores in the *behavioural intentions* construct compared to pre-filtering with user-based visualisation. This partly aligns with the hypotheses of both research questions. The use of post-filtering, particularly in the '*individual song recommendations*' stage, likely resulted in more novel song recommendations, positively influencing users' behavioural intentions. It does this however only in combination with the content-based explanation. Additionally, the content-based explanation in the post-filtering group scored higher in perceived explanation, possibly due to users preferring more direct information about recommended items.

The statistical analysis also revealed an interaction effect between the filtering and visualisation type, indicating a significant impact on *perceived explanation*. While the hypothesis for the first research question was partly supported, the hypothesis for the second research question did not align with the findings. Contrary to expectations, the post-filtering algorithm combined with the content-based visualisation outperformed the user-based visualisation. This unexpected result may be attributed to users' limited understanding of playlist match percentages, as indicated by some participants finding them confusing.

The research involves three limitations. Firstly, the implemented post-filtering algorithm generated more novel songs compared to the pre-filtering algorithm, which may have influenced the questionnaire responses. Secondly, the restriction on discussing questionnaire questions between users could not be guaranteed, potentially impacting the answers. Thirdly, the visual explanations included a user-based element alongside the content-based ones, potentially creating an unfair comparison between the two explanation styles.

8 CONCLUSION

This research explored filtering methods and explanation styles in group recommender systems. A web application was developed and tested in a user study. From the results of the data analysis, it can be concluded that combining post-filtering with content-based explanations positively impacts perceived explanation and behavioural intentions, particularly for users interested in discovering new songs. Additionally, presenting an excessive number of match percentages can lead to confusion and difficulties in interpretation for users.

Moreover, the proposed playlist-selection-based approach, along with pre- and post-filtering algorithms, and user-based and content-based explanations, could serve as a valuable enhancement to Spotify's *Blend* feature.

For future research, exploring the effects of different explanation styles on the five constructs is suggested. This could involve investigating social-based and item popularity-based explanations and combining them with the proposed pre-filtering and post-filtering algorithms.

REFERENCES

[1] Mohammed Z. Al-Taie and Seifedine Kadry. 2014. Visualization of Explanations in Recommender Systems. *Journal of Advanced Management Science* 2 (2014), 140–144. Issue 1. https://doi.org/10.12720/joams.2.2.140-144

[2] Alexander Felfernig, Ludovico Boratto, Martin Stettinger, and Marko Tkalcic. 2018. *Group recommender systems : an introduction.* 173 pages.

[3] Chen He, Denis Parra, and Katrien Verbert. 2016. Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities. *Expert Systems with Applications* 56 (2016). https://doi.org/10.1016/j.eswa.2016.02.013

[4] Nyi Nyi Htun, Elisa Lecluse, and Katrien Verbert. 2021. Perception of Fairness in Group Music Recommender Systems. *International Conference on Intelligent User Interfaces, Proceedings IUI.* https://doi.org/10.1145/3397481.3450642

[5] A. Ant Ozok, Quyin Fan, and Anthony F. Norcio. 2010. Design guidelines for effective recommender system interfaces based on a usability criteria conceptual model: Results from a college student population. *Behaviour and Information Technology* 29 (1 2010), 57–83. Issue 1. https://doi.org/10.1080/01449290903004012

[6] Pearl Pu, Li Chen, and Rong Hu. 2011. A user-centric evaluation framework for recommender systems. *RecSys'11 - Proceedings of the 5th ACM Conference on Recommender Systems* (2011), 157–164. https://doi.org/10.1145/2043932.2043962

[7] Nava Tintarev and Judith Masthoff. 2012. Evaluating the effectiveness of explanations for recommender systems: Methodological issues and empirical studies on the impact of personalization. *User Modeling and User-Adapted Interaction* 22 (10 2012), 399–439. Issue 4-5. https://doi.org/10.1007/s11257-011-9117-5