TU WIEN Informatics

Evaluating the Fairness of News Recommender Algorithms Within Detected User Communities

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Data Science

eingereicht von

Bernhard Steindl, BSc
Matrikelnummer 01529136

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Assistant Prof. Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Mitwirkung: Projektass. Dipl.-Ing. Thomas Elmar Kolb, BSc

Wien, 18. Oktober 2024

Bernhard Steindl	Julia Neidhardt

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

TU WIEN Informatics

Evaluating the Fairness of News Recommender Algorithms Within Detected User Communities

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Data Science

by

Bernhard Steindl, BSc
Registration Number 01529136

to the Faculty of Informatics

at the TU Wien

Advisor: Assistant Prof. Mag.a rer.nat. Dr.in techn. Julia Neidhardt
Assistance: Projektass. Dipl.-Ing. Thomas Elmar Kolb, BSc

Vienna, October 18, 2024

Bernhard Steindl	Julia Neidhardt

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Bernhard Steindl, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 18. Oktober 2024

Bernhard Steindl

Acknowledgements

First and foremost, I am very grateful to my parents for their continuous support and encouragement, as well as for making my education possible. My appreciation extends to my brother, who often gives me good advice and motivates me to complete my studies.

I thank the team of the Christian Doppler Lab for Recommender Systems at TU Wien Informatics and lab partners for providing the dataset used in my study and for the cooperation. In particular, many thanks to Julia Neidhardt and Thomas Elmar Kolb for their ongoing support throughout the project.

Kurzfassung

Diese Arbeit befasst sich mit dem Problem der ungerechten Behandlung verschiedener Gruppen von Nutzer:innen hinsichtlich der Empfehlungen, die sie von Empfehlungsalgorithmen erhalten. Empfehlungssysteme, englisch „Recommender Systems“ (RS), sind Algorithmen, die Nutzer:innen die Objekte vorschlagen, die aufgrund ihrer Interaktionshistorie mit Objekten am ehesten von Interesse sind. RS werden in vielfältigen Bereichen eingesetzt, etwa zur Empfehlung von Musik auf Musik-Streaming-Plattformen. Frühere Studien haben gezeigt, dass RS, basierend auf kollaborativem Filtern, besonders empfindlich auf Datenungleichgewicht reagieren, was zu weniger relevanten Empfehlungen für bestimmte Gruppen von Nutzer:innen führt. Bei solchen Studien zur Gruppenfairness werden Benutzer:innen nach einem sensiblen Attribut gruppiert, das meist auf ihren Eigenschaften oder demografischen Merkmalen basiert, und die gerechte Behandlung dieser Gruppen wird dann untersucht. Diese Arbeit untersucht die Fairness von Empfehlungen, indem sie die Unterschiede zwischen Gruppen von Nutzer:innen und verschiedenen Algorithmen für kollaboratives Filtern quantifiziert, wobei ein Datensatz der österreichischen Online-Nachrichtenplattform „DER STANDARD“ verwendet wird. Im Gegensatz zu ähnlichen Arbeiten handelt es sich bei diesen Gruppen von Benutzer:innen um große Gemeinschaften oder Communitys, die in einem aus Interaktionsdaten modellierten Netzwerk entdeckt werden. Graphen mit verschiedenen Beziehungen zwischen Nutzer:innen werden anhand von Daten zu Klicks auf Nachrichtenartikel der Nutzer:innen und deren Community-Aktivitäten, wie Forumsbeiträge, Beitragsabstimmungen und Follower:innen, erstellt. Die Communitys in diesen Graphen werden mit zwei Algorithmen zur Erkennung von Gemeinschaften identifiziert. Der Graph mit der höchsten Übereinstimmung zwischen den Community-Erkennungsalgorithmen bei der Zuordnung von Nutzer:innen zu Gemeinschaften wird ausgewählt, und die Partition mit der höchsten Qualität wird verwendet, um Nutzer:innen basierend auf den erkannten Gemeinschaften zu gruppieren. Es wird untersucht, inwieweit verschiedene RS unterschiedliche Empfehlungen für Nutzer:innen liefern, wobei der Schwerpunkt auf den Unterschieden zwischen großen Communitys liegt. Diese Studie stellt Instabilitäten in den Community-Erkennungsalgorithmen fest, aufgrund von erheblichen Schwankungen bei Konsens-Bewertungen für Graphpartitionen sowie bei der Größe und Anzahl der erkannten Gemeinschaften, wenn Netzwerke anhand verschiedener Schwellenwerte für die Kantengewichtung gefiltert werden. Die Empfehlungen für die Nutzer:innen in den erkannten Communitys variieren erheblich abhängig vom Algorithmus für kollaboratives Filtern und der Evaluierungsmetrik.

Abstract

This work addresses the problem of unfair treatment of different user groups in the recommendations they receive from recommendation algorithms. Recommender systems (RS) are algorithms that suggest items to a user that are most likely to be of interest, based on the user's interaction history with items. Recommendation algorithms are used in a variety of domains, such as recommending music on music streaming platforms. Collaborative filtering RS have been shown in previous studies to be particularly sensitive to data imbalance, resulting in less relevant recommendations for certain user groups. In such group fairness studies, users are grouped according to a sensitive user attribute, typically based on user traits or demographics, and the equitable treatment of these groups is then examined. This thesis explores the fairness of recommendations by quantifying variations between user groups and different collaborative filtering RS, using a dataset from the Austrian online news platform "DER STANDARD". Unlike related work, these user groups are large user communities discovered in a user network modelled from user interaction data. Graphs with different user relationships are built using data on users' clicks on news articles and users' community activities, such as postings, votes for postings and users' followers. User communities in these networks are identified using two community detection algorithms. The graph with the highest agreement between community detection algorithms in assigning users to communities is selected, and the partition with the highest quality is used to group users based on the communities detected. The extent to which various RS provide different recommendations to users is evaluated, with a focus on differences between large user communities. This study identifies instability in community detection algorithms by observing considerable variations in consensus scores for graph partitions, as well as in the size and number of communities discovered when networks are filtered using different edge weight thresholds. The recommendations to users in the detected user communities vary considerably depending on the collaborative filtering RS and the evaluation metric.

Contents

CHAPTER 1

Introduction

This introductory chapter provides a description of the research problem, the aim of this thesis and the research questions. It also outlines the methodological approach that is followed to address the research questions.

1.1 Motivation and Problem Statement

Recommender systems (RS) have become an essential part of our modern lives and are one of the most widely used applications of machine learning. Vast amounts of data are available on the internet, and recommender systems enable users to cope with the digital information overload. Recommendation algorithms are used, for example, by music streaming platforms to suggest music to users that they may like based on their previous consumption. RS assist users by suggesting items that are tailored to their interests and preferences. Recommender systems help users discover new items and simplify the decision-making process. Nowadays, recommender algorithms affect many aspects of our lives, influencing our choices of what to buy, watch or read, who to connect with and where to travel on holiday.

Despite the benefits that recommendation systems offer, they are also associated with risks [JZ22]. Societal risks of RS include that certain user groups may be disadvantaged in terms of the utility of recommendations. Additionally, especially on news sites, filter bubbles or echo chambers are an issue, where biased recommendations primarily reflect the existing interests and opinions of individuals or user groups [JZ22]. Excessive personalization of recommendations may lead to a lack of content diversity [SL23]. In the case of news recommender systems, for example, there may be a tendency for recommendations to favour certain political ideologies over others, making it difficult for users to find a balanced range of viewpoints [LCX+23]. Accuracy and beyond-accuracy metrics, such as diversity, novelty and serendipity, have historically been used to evaluate RS [EDBD22]. For example, recommended news should cover a variety of topics or

viewpoints, be recent and unfamiliar to users, and be unexpected but valuable. More recently, concerns about the fairness of recommendations have also become an important topic in the literature [EDBD22]. Individuals or user groups should be similarly satisfied with the recommendations they receive.

In a paper by Deldjoo et al. [DJB+23] that surveyed publications in recent years on the topic of fairness in RS, it was found that most papers in this area contribute new fairness-aware algorithms, but few studies compare the results of RS. Group fairness studies examine the equitable treatment of different groups of individuals. For this purpose, subjects are usually assigned to either a protected or an unprotected group on the basis of a sensitive attribute. Commonly used attributes are based on demographics or traits that are not under the control of the user, such as gender, ethnicity or age [DJB+23]. According to Ekstrand et al. [EDBD22] it is not uncommon for studies to focus on only one attribute. Furthermore, fairness is often simplified in research by assuming that the sensitive attribute used to categorize individuals is binary, which is not realistic in practice because multiple features make up a person's identity. Combining multiple attributes or generalizing beyond two groups is rarely done [EDBD22]. For example, Ekstrand et al. [ETA+24] compared differences in recommendation utility when users were divided into groups by either gender or age. An alternative to using known sensitive attributes is to compute them based on user interaction data [DJB+23], as done by Li et al. [LCF+21], who divided users into two groups based on their level of activity on an e-commerce platform.

Few empirical studies explore user group fairness in recommender systems, and those that do often focus on sensitive user attributes based on demographics, such as gender, while rarely considering domain-specific aspects. However, recommender algorithms should also account for the fairness of general user groups, such as those with similar behaviours, but research on this is limited. In practice, less active users or those with specific interests should also receive good recommendations. This thesis investigates the fairness of recommendations for users in the news domain. Given that filter bubbles are fostered by biased recommendations for user groups, this work addresses the problem of unfair treatment of different user communities in the recommendations generated by recommender systems, where users within each community share similar behaviour.

1.2 Aim of the Work

The aim of this work is to quantify the extent of variation in the accuracy and diversity of recommendations between user groups and recommendation algorithms, in order to investigate the fairness of recommendations for different user communities, each characterized by users with similar patterns of behaviour. These user groups are derived from communities discovered in a user network modelled from user interaction data, and these user communities are assumed to share common interests or behaviours. To achieve this aim, user network construction, community detection and news recommendation are necessary steps to analyse the differences in recommendations between user communities.

One of the challenges is to develop the network design, that is to define the relationships between users, which are represented as nodes in a graph. For example, links between nodes in a graph might be established based on whether one user follows another on an online platform. User networks with diverse user relationships are constructed and the differences in graph partitions obtained by community detection algorithms are analysed. The number and size of communities identified in a network by community detection algorithms varies by network and is not predefined.

The recommendations to users of different recommendation algorithms are evaluated, with a focus on the group fairness of large behavioural user communities identified in a selected user network. While this work examines group fairness within behavioural communities, it is recognized that this issue is complex to model and has been simplified for this thesis, with many factors, such as the dynamic nature of user preferences, not taken into account. This study investigates the extent of unfairness across user groups in recommendations by comparing traditional recommendation algorithms and a fairness-aware recommender algorithm using accuracy and beyond-accuracy evaluation metrics.

This thesis uses a dataset from the Austrian online news platform "DER STANDARD"[1]. It includes the metadata of news articles, as well as data on news article clicks, forum postings on articles, votes on postings and follow connections of users.

Research Questions

Specifically, the following research questions are answered.

- RQ1: To what extent does the selection of different types of user interactions for constructing networks influence key metrics such as community size and group centrality within the network?
- RQ2: What variations exist in the accuracy and diversity of recommended content across identified user communities when employing non-fairness-aware recommendation algorithms?
- RQ3: To what extent does a fairness-aware recommendation algorithm improve the equitable distribution of accurate and diverse recommendations across user communities?

1.3 Methodological Approach

The applied research methods of the thesis follow a widely used data science methodology called "Cross-Industry Standard Process for Data Mining" (CRISP-DM [CCK+00, MCF+21]). In the following, the methodological steps of CRISP-DM are enumerated, and the specific actions for this work are described.

[1] https://www.derstandard.at

1. Business understanding

   In this step, the research problem is determined, the current situation is assessed, which then leads to the definition of objectives and the development of a project plan. A literature review is carried out, inspired by the suggestions of Kitchenham et al. [KPBB+09]. The literature is searched for state of the art fairness-aware recommender algorithms and graph community detection algorithms.

2. Data understanding

   The provided dataset is explored in this step.

3. Data preparation

   The dataset is preprocessed, which includes data filtering, data consistency checks and data exploration. User networks with diverse user relationships are constructed. For example, a graph is created where the nodes are users and weighted edges exist when a user positively votes for another user's forum postings, with the weight indicating the frequency of this interaction. Two graph community detection algorithms are applied to the constructed networks and differences in graph partitions are analysed. The user network with the highest agreement for the identified user communities between both community detection algorithms is selected, and a graph partition is chosen according to the assessment by partition quality functions. Users are then assigned to user groups based on the large user communities identified in the selected network. A user-item interaction matrix is created that records users' views of news articles, which is then used to recommend news to users.

4. Modelling

   In the modelling phase, a fairness-aware and several traditional recommender algorithms are trained using the training set of the user-item interaction matrix. The user's membership of a user group is considered to be a protected user attribute. Model hyperparameter optimization is performed using the validation set.

5. Evaluation

   The recommendations generated for users by different recommendation algorithms are evaluated on the test set using accuracy (NDCG, precision, recall) and beyond-accuracy (coverage, diversity, novelty) metrics. Evaluation results are analysed for users overall and at the level of the large user communities detected in the selected graph. In particular, the evaluation focuses on quantifying the average discrepancy in the mean evaluation scores of the user communities and the effect of the recommender models in amplifying existing imbalances in the data with respect to the distribution of the user communities.

6. Deployment

   This step serves to document the research results.

1.4 Structure of the Work

The remainder of this thesis is structured as follows. Chapter 2 explains the state of the art of community detection in graphs and recommender systems, with a focus on fairness-aware recommender algorithms, and discusses related work. The following Chapter 3 describes the experiment design, including a description of the dataset and the approach used for data preprocessing, user network construction, user community detection, and news recommendation. Chapter 4 presents and interprets the key results of this research. Chapter 5 concludes the thesis with a summary of the research contribution, a discussion of the research results and an outlook for future work.

CHAPTER 2

State of the Art

This thesis investigates the extent to which the accuracy and diversity of recommendations vary for user communities detected in a user network modelled from user interaction data. User networks with diverse user relationships are constructed and the differences in the graph partitions obtained by two community detection algorithms are analysed. The user network with the highest agreement for the identified user communities between both community detection algorithms is selected, and a graph partition is chosen according to the assessment by partition quality functions. The recommendations to users of different recommendation algorithms are evaluated, with a focus on the recommendation fairness of large user communities identified in the selected network. This chapter explains the state of the art of community detection in graphs and recommender systems, with a focus on fairness-aware recommender algorithms, and discusses related work.

2.1 Community Detection in Graphs

In this thesis, networks of users are constructed based on user interactions on a news platform, and community detection algorithms are used to identify the membership of users to a community within a network. Graphs consist of vertices, which correspond to users in this work, and edges, when there is a connection between two users. Clusters in graphs can be groups of friends or people who share common interests or activities and community detection algorithms are concerned with their discovery. According to Fortunato et al. [FH16], a classical view of the notion of community is that they are well separated, with higher edge density within clusters than between clusters. Communities can also be defined as groups of nodes that are more likely to be connected to each other than to members of other groups [FH16]. Recent literature on community detection in graphs includes the work of Lancichinetti et al. [LF09], Fortunato [For10], Barabási et al. [BP16], Newman [New18] and Fortunato et al. [FH16, FN22]. A partition is a division of a network into groups, with each node belonging to only one cluster. In

community detection, the number and size of communities are not predefined. Examining all the partitions of a large network is computationally infeasible because the number of partitions grows exponentially or faster with the network size. Fortunately, dedicated community detection algorithms with polynomial runtime complexity have been developed [BP16]. Although there is no clear answer as to which method should be used [FH16], the `Louvain` algorithm and `Infomap` are among the leading methods with reasonable accuracy and good scalability for detecting clusters in networks [BP16, LF09, New18]. The runtime complexity of both methods is about $\mathcal{O}(n \log n)$ for sparse graphs, where n is the number of vertices in the graph [New18].

`Infomap`, by Rosvall and Bergstrom [RB08], is an information-theoretic approach for community detection. It can be applied to weighted graphs, both undirected and directed, and it has been extended to detect hierarchical community structures [RB11]. This algorithm is inspired by the idea of viewing a network as an information flow and compressing the amount of information needed to describe the dynamics of that flow. `Infomap` seeks the partition that minimizes the map equation to identify communities within a graph. The map equation [RAB09] leverages the modular structures of a network and measures the per-step average minimal information required to track the movements of a random walker on a network, which is equal to the entropy of a random walk.

The `Louvain` algorithm, proposed by Blondel et al. [BGLL08], is an agglomerative, hierarchical community detection algorithm that greedily maximizes the Newman-Girvan modularity. The Newman-Girvan modularity [NG04] measures the partition quality of a particular graph by comparing the original graph with a random graph that has the same degree sequence. Modularity compares the edge density of subgraphs in the original graph with that expected in a random graph to determine whether the connections within communities are stronger than they would be by chance, since random graphs are assumed to have no community structure. In the `Louvain` method, vertices are first allocated to individual clusters and then each vertex is iteratively assigned to the neighbouring community that leads to a positive maximum gain in modularity. This is repeated until no further improvements are possible. In the next phase, the clusters found are converted into nodes and the edge weights are aggregated to create a more compact graph, representing a network partition or hierarchy level. These two phases, named a “pass”, may then be repeated on the last partition found until a level with maximum modularity is reached, yielding increasingly larger clusters in subsequent passes. This method can be applied to weighted graphs and has been extended by Dugué and Perez [DP22] to directed networks.

2.2 Traditional Recommender Systems

This thesis evaluates different recommendation algorithms used to generate news article recommendations for users, with a focus on the recommendation fairness of large user communities. Recommender systems (RS) refer to software, or algorithms, whose purpose is to provide a user with suggestions of items that are most likely to be of interest to

the user. Items generally denote the objects that a system recommends to users, and typically a recommender algorithm suggests items of a certain type [RRS22]. RS are applied in various domains and the recommended items include, for example, films, books, people, holidays or news articles. The items recommended to a user by an RS are usually personalized, and the user receives a ranked list of the suggested items sorted according to their presumed relevance to the user. Recommendation algorithms predict the most suitable items based on the user's preferences, either explicitly expressed by the user or implicitly derived from the user's interactions with the system. An example of explicit user feedback is a rating for a product, while implicit feedback refers to user interaction, such as visiting a product page on a website. Recommendation algorithms can be categorized based on the technique used, with collaborative filtering, content-based and hybrid RS being among the most common, where hybrid systems combine different techniques to overcome the limitations of individual methods [RRS22].

A content-based RS suggests items to a user that are similar to those liked in the past, where item similarity is calculated based on item characteristics and the user's previous interactions with items. In order to recommend items to a target user, a content-based RS does not require data from other users, but item characteristics must be extracted for each item based on the item content. For example, music genre could be a feature of an item in the music recommendation domain, and if a user has rated music of a certain genre positively, the system can learn to recommend other music of that genre [RRS22].

Instead of depending on content data, a collaborative filtering recommender system suggests items to a target user by relating the target user's interacted items to the item interaction history of other users, leveraging patterns in user-item interactions. Approaches for collaborative filtering can be divided into neighbourhood-based and model-based methods. Neighbourhood-based collaborative filtering directly uses user feedback on items to predict recommendations for unseen items by considering the items most similar to those a target user has interacted with or the users most similar to the target user. Model-based approaches use user-item feedback to learn a predictive model that captures the characteristics of users and items and their relation, which is then used to predict new items [NNDK22]. Various model-based collaborative filtering techniques have been proposed, such as matrix factorization models, which decompose a matrix of user-item interactions into low-rank latent user and item factors, that represent user preferences and item characteristics in a common latent space, allowing prediction of a user's preference for an item [KRB22].

With the revival of neural networks, deep learning has also attracted attention in the context of recommender systems [ZTY+22]. In particular, collaborative filtering tasks have been transformed into a corresponding deep learning solution [RRS22]. Deep learning can be used in recommender systems to learn representations of users, items, and their interactions, capturing the underlying patterns and relationships in the data. In deep learning, a hierarchy of neural networks is used to perform machine learning. A neural network is a computational model composed of layers of interconnected neurons, and it is considered deep if it includes multiple layers between the input and output

layers. Data is transferred between layers by neurons, with data flow and transformation controlled by activation functions. The parameters of deep learning models are learned by a process called back-propagation, which adjusts the parameters of a neural network by using the error of a loss function to update them as it propagates backwards through the network. Based on the organization of the neurons, different neural network architectures are distinguished, including multilayer perceptrons, convolutional neural networks, recurrent neural networks, autoencoders, generative adversarial networks and graph neural networks. Various methods based on deep learning have been developed for RS, and recommendation algorithms using deep learning techniques have become the main research focus [RRS22]. Among these, variational autoencoders (VAE) have only recently emerged as a prominent method for recommender systems, and the number of publications using VAE for recommendations has increased considerably in the past few years, according to a survey by Liang et al. [LPL+24]. A variational autoencoder is a deep generative model used in recommender systems for learning latent representations of users and items by using Bayesian inference to model user-item interactions as probabilistic distributions, with an encoder that maps input data into a probabilistic latent space and a decoder that reconstructs the data [LPL+24].

Recent literature by Karimi et al. [KJJ18], Raza et al. [RD22] and Wu et al. [WWHX23] summarize the state of the art for news recommendation algorithms and specific characteristics associated with the application of RS in the news domain. An analysis of studies shows that in academic settings, collaborative filtering is the most commonly used approach in the literature on recommender systems, while all other approaches are used much less frequently [KJJ18]. In contrast, content-based RS are mainly used for news recommendations, followed by hybrid RS and collaborative filtering recommendation algorithms [KJJ18, RD22]. News recommender systems (NRS) differ from other recommendation domains, such as film recommendations, in several ways. A key difference lies in the importance of recency and the short lifespan of news articles. News articles are very dynamic, and their relevance quickly diminishes as new events occur and new articles are published. This requires NRS algorithms to prioritize the most recent content and continuously update recommendations. Strategies to address the recency aspect of news have been proposed, such as pre-filtering for recent articles, incorporating timeliness into the recommendation model, or using post-filtering techniques to prioritize newer content. In addition, the interest of users in news can change based on context, such as the time of day, current events, the user's device or location. NRS must be highly adaptable, balancing long-term user interests with short-term changes driven by events such as upcoming elections or breaking news. Another challenge for NRS is the permanent item cold-start problem, where the frequent publication and short lifespan of news articles, combined with limited interaction data, reduce the effectiveness of collaborative filtering, although the adoption of a content-based RS can help mitigate this issue. Furthermore, news recommender systems need to focus on more than just accuracy and ensure diversity, novelty and serendipity in recommendations. This means providing a mix of different stories, the latest news and unexpected but interesting articles, which prevents users from seeing too many similar stories and keeps the content engaging and relevant [KJJ18].

The aim of this work is to quantify the extent of variation in accuracy and diversity of recommendations between user communities detected in a user network. This study focuses on collaborative filtering algorithms for two reasons. Unlike content-based RS, which require item features to generate recommendations, collaborative filtering can rely on implicit feedback, such as clicks on news articles, which is available for this study. Collaborative filtering algorithms are particularly sensitive to data bias because they rely on user interaction data from both the target user and other users, often resulting in less relevant recommendations for certain user groups, as shown for example in studies by Yao et al. [YH17], Ekstrand et al. [ETA+24], Melchiorre et al. [MZS20, MRP+21] and Li et al. [LCF+21]. This thesis compares several traditional collaborative filtering algorithms based on nearest neighbours, matrix factorization and variational autoencoder, as well as a fairness-aware recommender model.

2.3 Fairness-Aware Recommender Systems

In contrast to traditional recommendation methods, some algorithms have been developed that inherently incorporate fairness into their recommendations [WMZ+23]. This thesis uses a fairness-aware recommendation model in addition to conventional recommendation algorithms to investigate possible improvements in fairness metrics for user communities detected in a user network. Recent literature reviews on fairness in recommendations are conducted by Wang et al. [WMZ+23], Li et al. [LCX+23], Deldjoo et al. [DJB+23] and Jin et al. [JWZ+23]. According to Wang et al. [WMZ+23], fairness in recommender systems can be divided into process fairness, which addresses the fairness of the recommendation process itself, and outcome fairness, which refers to the fairness of the recommendations provided by the system. Outcome fairness includes group fairness, which focuses on achieving fairness between different groups, and individual fairness, which ensures similar outcomes for similar individuals. In addition, outcome fairness can also be understood in terms of different concepts of fairness. Researchers have different opinions on the definition of fairness and several concepts have been proposed. Most research emphasizes ideas such as consistent fairness, which requires that similar groups to be treated similarly, and calibrated fairness, which seeks to allocate recommendations to each group in proportion to their merits [WMZ+23].

Methods for fair recommendations aim to address biases and ensure fairness in recommender systems through various approaches, which can be broadly categorized into pre-processing, in-processing and post-processing techniques. Pre-processing methods aim to modify the input data to reduce bias before it is used by a recommender system. In-processing methods involve incorporating fairness constraints directly into RS through techniques such as regularization and adversarial learning. Regularization modifies the model's objective function to balance accuracy and fairness by penalizing unfair outcomes. Adversarial learning aims to reduce bias by training a model that challenges the system to learn fair representations, minimizing the influence of sensitive data on recommendations. Post-processing methods adjust recommendations after generation, such as re-ranking the results to ensure a fair distribution across different user groups [WMZ+23].

Collaborative filtering algorithms capture consumption patterns, and even if sensitive user attributes are not provided directly by users, models can learn them indirectly from users' interactions with items [SL23, WMZ+23]. These learned biases may influence the recommendations of RS to further separate the content offered to different user groups [GPR+22]. Adversarial learning can be used, for example, to learn fair representations of users by removing implicit encoded sensitive information. Adversarial learning involves training an adversary model, often a neural network, alongside the main model to detect and penalize biases, forcing the main model to adapt and reduce its reliance on sensitive information. Recent work by Li et al. [LCX+21], Wu et al. [WWW+21] and Ganhör et al. [GPR+22] addresses this issue. Li et al. [LCX+21] propose a framework for achieving personalized counterfactual fairness in RS by allowing users to specify sensitive attributes they care about and generating user representations that are independent of these attributes through adversarial learning, demonstrating its effectiveness with both shallow and deep recommender models. Wu et al. [WWW+21] introduce a technique to decompose the user interest model into bias-aware and bias-free user representations through adversarial learning and orthogonality regularization. Ganhör et al. [GPR+22] propose an RS called "Adversarial Variational Auto-Encoder with Multinomial Likelihood", which integrates adversarial training in a variational autoencoder to reduce biases encoded in the learned user representations with respect to a protected user attribute.

This work investigates the fairness of user groups when using collaborative filtering RS. Recommendations to users are compared for traditional collaborative filtering algorithms based on nearest neighbours, matrix factorization and variational autoencoders. In addition to conventional recommendation systems, this work also uses a fairness-aware RS that supports more than two user groups, since the number of communities identified in a graph is usually not limited to two. This work compares the fairness-aware recommender model proposed by Ganhör et al. [GPR+22] with a non-fairness-aware variational autoencoder for collaborative filtering.

2.4 Empirical Studies on User Group Fairness

Related work includes a study by Yao et al. [YH17], which examined recommendation fairness in the education domain for user groups defined by gender and STEM preferences. Ekstrand et al. [ETA+24] investigated the fairness of music and film recommendations for users grouped by gender and age. The study by Melchiorre et al. [MZS20] explored fairness in music recommendations by examining prejudices against user groups with different personality traits. Melchiorre et al. [MRP+21] analysed recommendation fairness in the music domain between male and female users and proposed the fairness metrics RecGap, which quantifies the differences in recommendation performance between user groups, and the Compounding Factor, which measures the extent to which an RS amplifies data bias. Li et al. [LCF+21] studied recommendation fairness for active and less active user groups on an e-commerce platform. Unlike previous work, this paper explores user fairness in recommendations for user communities detected in a user network modelled from user interaction data.

CHAPTER 3

Experiment Design

In this chapter a detailed description of the design of the experiments performed in this thesis is provided. The widely used data science methodology "Cross-Industry Standard Process for Data Mining" (CRISP-DM [CCK+00, MCF+21]) is followed to conduct the experiments. First, the dataset used is described and then the preprocessing of the data is explained. An overview of the user networks that are constructed in the course of this work is given thereafter. This chapter concludes with an explanation of the procedure for the detection of user communities in these networks and the recommendation of news.

3.1 Dataset

Anonymized data from the Austrian online news platform "DER STANDARD" is used in this work[1]. This online medium offers daily news coverage and is free to access. News in various sections, such as domestic, international, economic, web, sports, science, are published. On the homepage users find an overview of current and recently published content, displayed with a headline, and sometimes combined with an excerpt and an image. In Figure 3.1 a screenshot of the "DER STANDARD" homepage is shown. At the top of the website, links to overview pages for certain news sections (e.g., international news) are shown. By clicking on one item of the overview, users can read its content.

Each news article page consists of news content and a discussion section below the article. The latter allows registered users who decided to create a community identity to exchange with others. In the discussion section, users can create postings, respond to postings and share their opinion by voting for a posting either positive or negative. Screenshots of the content and discussion area of a news article are shown in Figure 3.2 and Figure 3.3.

[1] https://www.derstandard.at

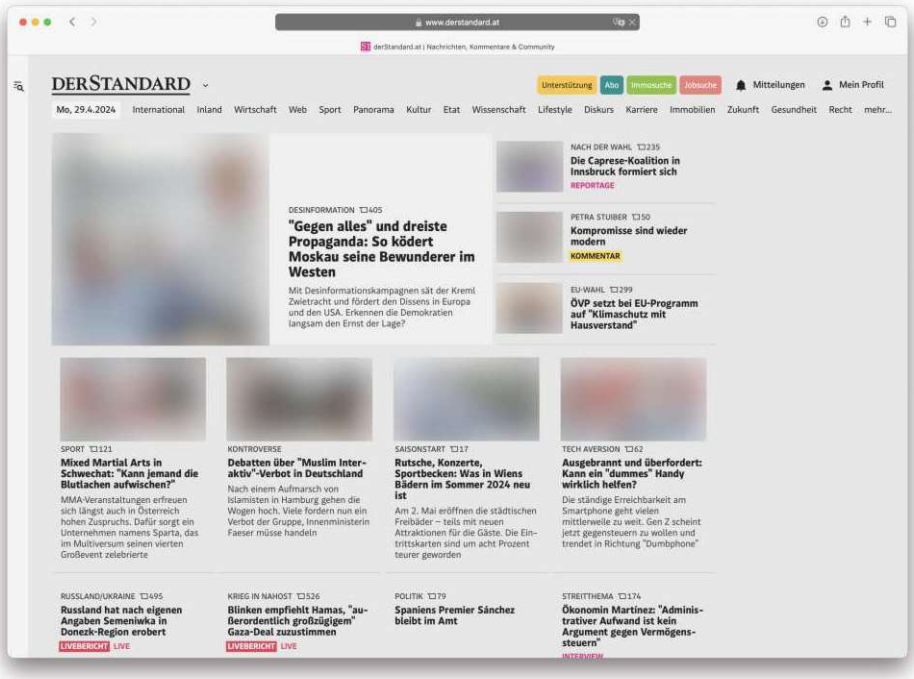

Figure 3.1: Screenshot of the "DER STANDARD" homepage from 29 April 2024

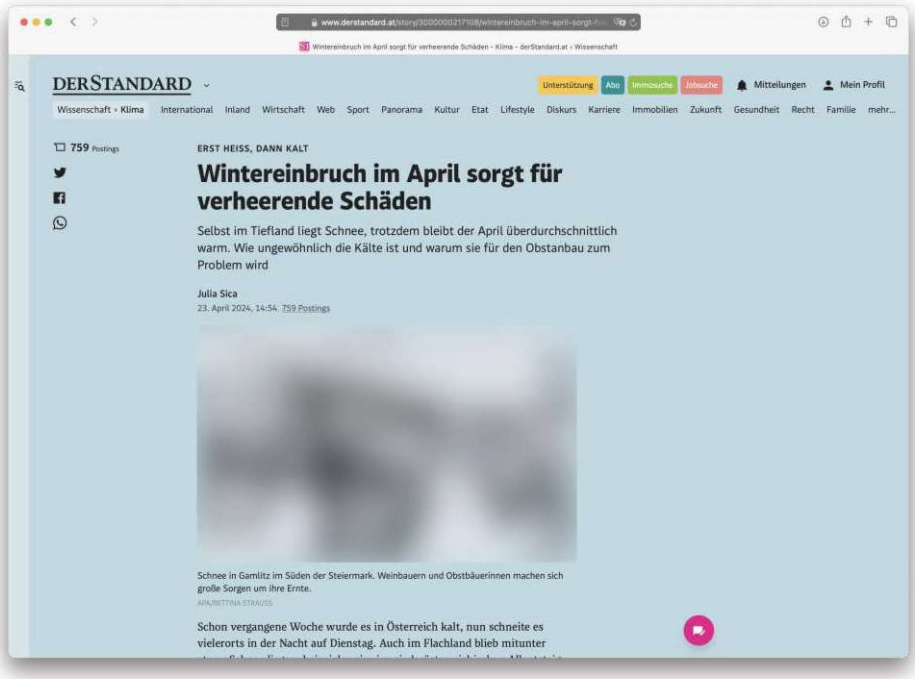

Figure 3.2: Screenshot of the content of a news article from "DER STANDARD"

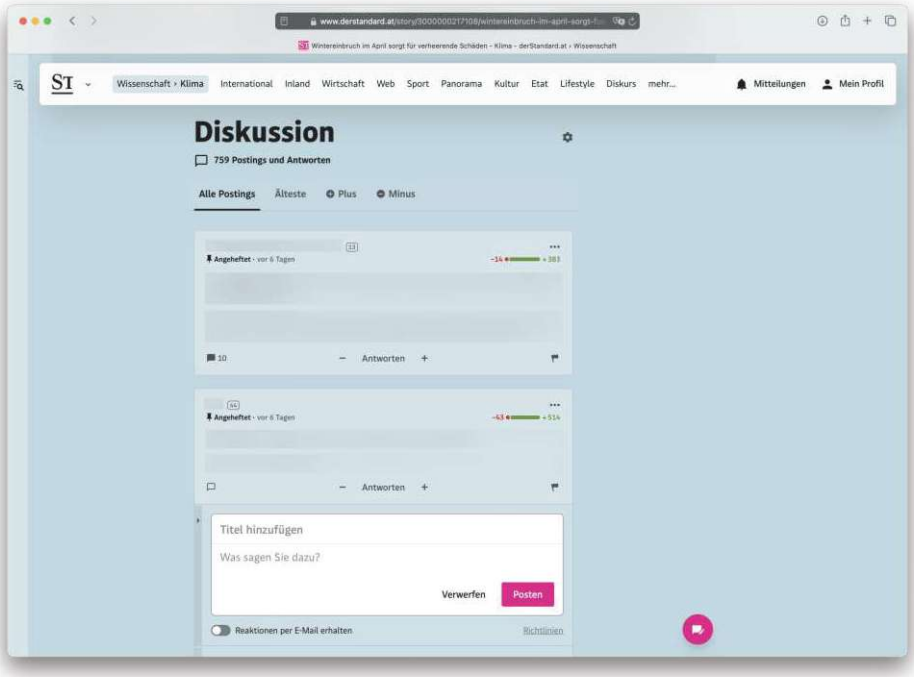

Figure 3.3: Screenshot of the discussion section of a news article from "DER STANDARD"

Clicking on a user's community identity name in the discussion area shows the corresponding user profile. On a user's profile page, the postings created and the number of followers are displayed. A screenshot of a user's profile page is depicted in Figure 3.4.

Registered users with a community identity can also follow other users in order not to miss any community activity. When clicking on the number of followers on the user profile page, the community identity names of all followers appear. Figure 3.5 shows a screenshot of a list of followers of an active user with a community identity.

Besides discussions to news articles, users can also interact with others in a dedicated discussion forum, which allows for an exchange on diverse topics. In this forum, users can discuss with others on blogs, columns, commentaries, debates or even ask for their own forum page on a certain topic.

The website of "DER STANDARD" also offers other forms of information and entertainment. User can engage with live reports, videos, podcasts, recipes, or crossword and Sudoku puzzles. Additionally, the platform integrates a job and real estate search.

In this study, data on users, news articles, users' views of news articles and community activities, which include postings to news articles, votes on postings and user follow connections, are used. After an initial data exploration, the dataset is preprocessed as part of the overall experiment workflow. A simplified flow chart of the experiment is provided in Figure 3.6, outlining the key steps of the procedure.

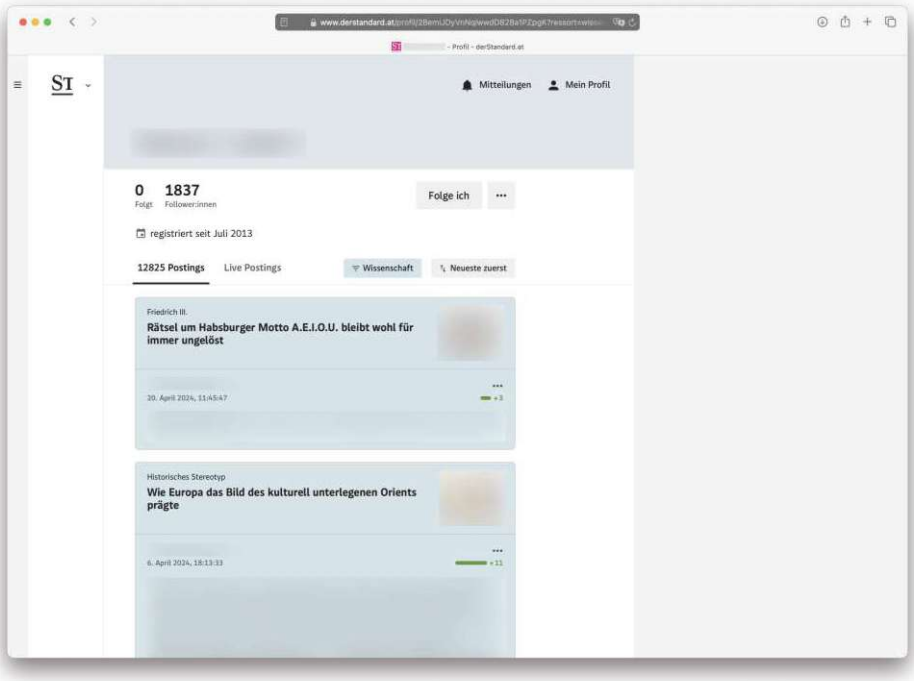

Figure 3.4: Screenshot of a user profile page on the “DER STANDARD” webpage

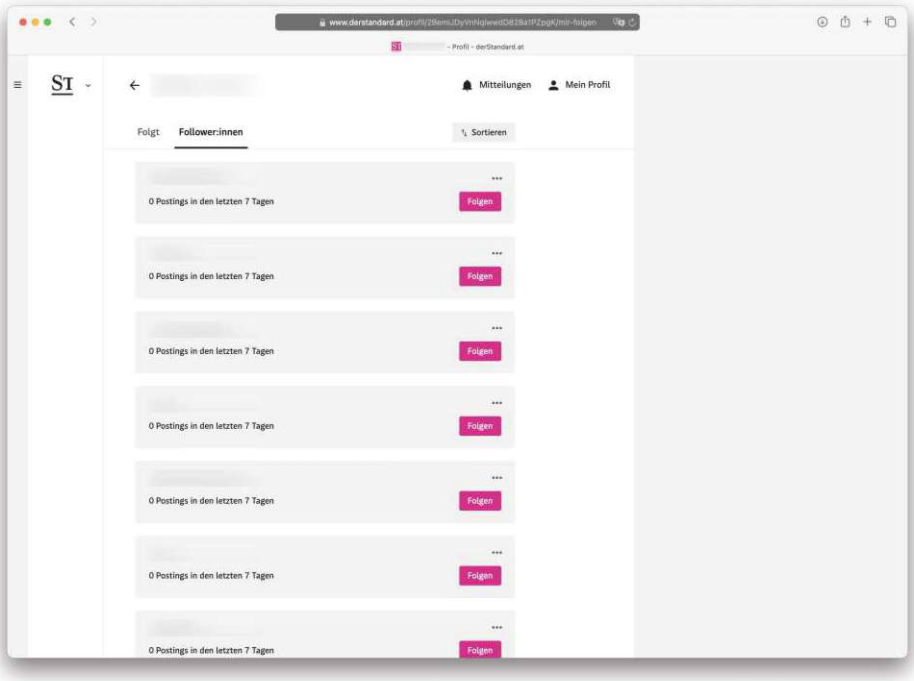

Figure 3.5: Screenshot of the follower list of a user on the “DER STANDARD” webpage

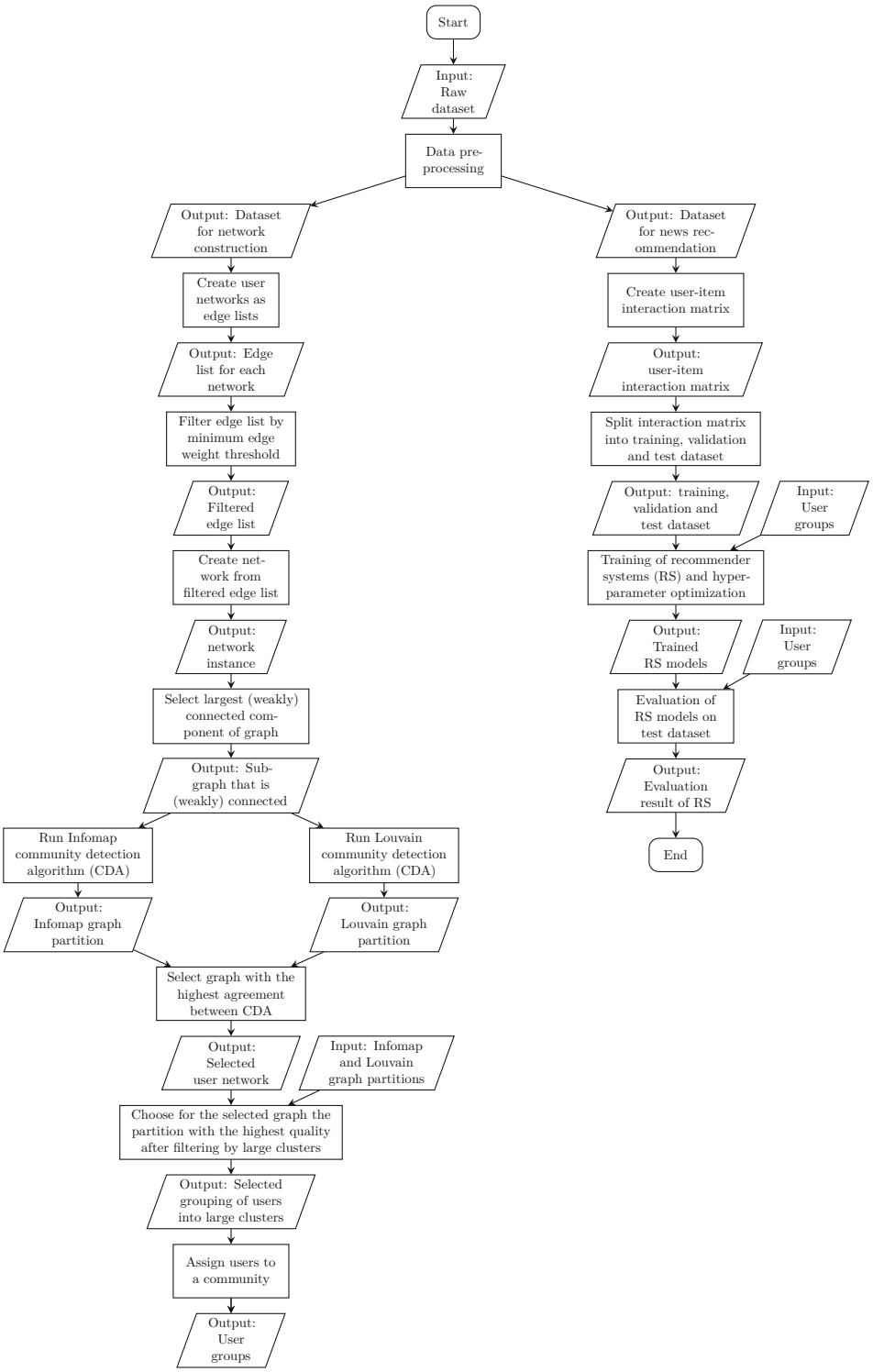

Figure 3.6: Simplified flow chart of the experimental workflow

3.2 Data Preprocessing

In this work, the news recommendation fairness is evaluated within detected communities in a network of users. There are various possibilities to construct a network, where graph nodes are users and an edge between them represents some user relationship. This work utilizes users' views of news pages and users' community activities, in the form of postings, votes and user follow connections, for modelling the relationship in networks. Recommender algorithms rely on data about users' views of news articles. The focus of this study are the recommendation of news articles for registered and active users of the online news platform "DER STANDARD". Several data processing steps are applied to the raw data before the preprocessed data is used as input for network construction and news recommendation. These data preprocessing steps are described in this section.

3.2.1 Time-based Filtering

Data in the period from January 2017 to August 2019 is used. More precisely, data from January 2017 to December 2018 is used for constructing networks and detecting user communities. Subsequent data from January 2019 to August 2019 is used for training and evaluation of news recommender algorithms. Assumptions in this study are that a user's community membership does not change in the course of the observation period and a user belongs to only one community. The data is split and filtered, so that user interactions, such as views of news articles, posting submissions and votes for postings, do not overlap in both periods. News articles are used exclusively in one of the two time frames. This time-based filtering is done based on the dates of news article publications, posting submissions, vote creations and users' views of news articles. The most recent user follow connections are considered up to the end of the network construction time period. Interactions with the content and discussion area of news articles peak in the vicinity of the publication date and subsequently decline the following days. To take this diminishing interaction effect into account, the news articles are filtered such that the respective publication date must be one month earlier than the respective time frame end. That is, only news articles are used with a date of publication in the period from January 2017 to November 2018 for the network construction. Analogously, news articles with a publication date from January 2019 to July 2019 are used for the news recommendation.

3.2.2 News Article Filtering

As explained in Section 3.1, the Austrian online news platform "DER STANDARD" offers users a broad spectrum of information and entertainment formats. However, this study uses only published news articles from the editors of "DER STANDARD" and news agencies. Advertising pages, overview pages, live reports, videos, slideshows, podcasts, recipes, quizzes, crossword and Sudoku puzzles, as well as pages in the dedicated discussion forum, like blogs, columns, commentaries, debates and user created forum pages, are not taken into account. Pages that regularly appear as summaries of several news pages, such as morning briefings, are also filtered out. Regularly published pages that provide

an overview and recommendations for upcoming radio and television programmes are excluded. News articles from the following news sections ("Ressorts") from the "DER STANDARD" news website are used: Wirtschaft, International, Panorama, Web, Sport, Kultur, Etat, Wissenschaft, Inland, dieStandard, Lifestyle, Gesundheit, Bildung, Reisen, Karriere, Immobilien, AutoMobil, Familie, Zukunft and Recht. In addition, a news article must have at least 100 views from different users in order to be considered. Outliers in news articles are removed by requiring the corresponding number of user views, postings and votes to be below or at a quantile value of 0.997. With this quantile-based outlier removal, news pages are ignored if they have either the highest 0.3% of the number of postings, votes, or user views.

3.2.3 User Filtering

Users do not have to sign up, but can freely access pages of the online news platform. Registered user of the "DER STANDARD" website can only use community features, such as submitting postings, voting for postings, or following other users, once they have created a community identity by entering a unique community name. Only registered users with a community identity are used in this study because modelling networks requires activity data on the community features. In the past, users were able to change their community name on the online news platform, which resulted in another community identity being linked to the user account. Should a user have more than one community identity, only the active one's community activity data is used. Geolocation data is used for retaining only users whose primary country of news article interactions is Austria during the whole observation period. The main target group are users who read and interact with online news articles on a regular basis. Ignored are user accounts that are only used for a short period of time, such as fake accounts that attempt to manipulate public opinion with coordinated, often negative postings about a political candidate ahead of an election. Exclusively active users are taken into account, i.e., users with at least eight interactions with distinct news articles every 90 days over the entire time period considered. The minimum number of interactions per time interval is chosen to account for those users who may be on holiday for a few weeks and do not consume news online. Outlier users are removed by stipulating that the corresponding number of news article views, postings and votes for postings must be below or at a quantile value of 0.997. Additionally, users must have interacted with a minimum of 120 and 30 different news articles during the network construction and news recommendation time frames, respectively. This user filtering procedure ensures that the selected users are active during the time frames of network construction and news recommendation.

3.2.4 User Follow Connection Filtering

Other users can be followed on the platform of "DER STANDARD" to stay informed about their new postings. Follow connections can be either unidirectional or bidirectional. Naturally, follow connections can change over time. Users can change their mind and unfollow persons they previously followed. When constructing networks using data on

users' follow connections, the most recent relationships up to the end of the network construction period are used. Follow connections are only kept for the selected active users. Of course, user follow connections that link one and the same user are not intended.

3.2.5 Posting Filtering

Registered users can create postings below news articles or reply to other users' postings. Postings can contain a headline or a comment, or both. Only the postings of filtered users who also have an active community identity are kept. The postings of filtered out news articles are also ignored. Due to unknown inconsistencies in the database, a posting is filtered out if there is no click interaction recorded between the author of the posting and the corresponding news article to which the posting is associated. A simple content-based posting filtering procedure removes postings with trivial or low informational content, such as those that merely express emotions or acknowledgements, as they do not contribute meaningfully to the conversation. Examples of such low informational content are "Aha.", "haha", ":-D", "rofl", or "Oha!". Filtering out non-informative postings speeds up subsequent computations, reduces data noise and ensures a focus on conversations that provide valuable contributions to the discussion. Content-based posting filtering operates on the headline and comment of postings after removing control characters, replacing consecutive space characters with a single space and stripping whitespace. Either the posting headline or comment must be at least 10 characters long, as well as contain 8 characters and 5 distinct letters of the German alphabet. If a posting is removed during preprocessing, the posting's child postings are not removed.

3.2.6 Vote Filtering

Users on the online news platform "DER STANDARD" can cast their vote for postings. Votes can either be positive or negative. Only the votes of active users that remain after filtering news articles and postings are used. A vote is removed if there is no interaction between the vote creator and the corresponding news article to which the vote is linked.

3.2.7 User Clicks of News Articles Filtering

In this study, user views of news articles, also known as page impressions, are considered for registered active users and filtered news articles. Users can interact with a news page for varying lengths of time and at different occasions or visits. User data on page views and page interactions provide an estimate of how long a user stays on a news article page. However, knowing the page views of a user is no indication of whether the user has actually read the news articles or the postings in the related discussion area. Page views are ignored if a user inadvertently clicks or stays very briefly on a news page on the platform. More precisely, a user's page impression is ignored if the total dwell time of the user on the news page is less than 15 seconds. No distinction is made between the time the user spends with the content and the discussion area of a news article.

3.3 Network Construction

Preprocessing the data as described in Section 3.2 results in two data sets, one for the network construction and the other for the news recommendation. Several possibilities exist to design networks in which the nodes are users and the edges represent a certain relationship between the users. The preprocessed dataset for network construction includes user data on postings, votes for postings, follow connections and news article views and allows for the creation of different graph variants. This section provides a definition of the graphs used, followed by a description of the modelled networks.

3.3.1 Graph Definition

Weighted simple graphs that allow neither self-loops nor multiple edges are used in this work. A simple graph G is formally defined as an ordered pair $G = (V, E)$, where V is a set of vertices, or nodes, and E is a set whose elements are edges, or links. If the edges of a graph have a direction, it is referred to as directed, otherwise as undirected. An edge that connects a vertex with itself, is called a self-loop. Multiple edges in a graph are distinct edges that join the same pair of nodes, having the same direction in directed graphs and disregarding direction in undirected graphs. In an undirected graph, the set $E \subseteq \{\{u, v\} \mid (u, v) \in V^2 \wedge u \neq v\}$ contains unordered pairs of distinct vertices. For a directed graph, links in $E \subseteq \{(u, v) \mid (u, v) \in V^2 \wedge u \neq v\}$ are ordered pairs of different nodes. A weighted simple graph is one in which edges are assigned weights, or values, by a weight function w. In this study, $w : E \to \mathbb{R}^+$ assigns positive weights to each edge.

3.3.2 Modelled Networks

This section describes the various networks modelled in this work. An edge list is used as a data structure to represent a graph as a list of edges. This work models users as graph nodes and each link is defined by its start and end vertex and an edge weight. The weight of each edge reflects the strength of the relationship between the connected nodes. A higher edge weight may indicate a more frequent interaction, a stronger relation or greater similarity between users. All networks considered use non-binary edge weights to avoid losing information about the connection quality. Among the networks created are both undirected and directed graphs.

Network of Users Who Vote on Postings

In the discussion area of a news article, users can vote either positively or negatively on the postings of other users. In the *Network of Users Who Vote on Postings*, a weighted link is established from the user who casts a vote to the author of the posting. The edge weight results from the number of positive votes minus the negative votes. A link between two users is only created if the total weight is strictly positive. Both a directed and an undirected variant of this network are created. If there are edges with opposite directions between two nodes in the directed network, their weights are added together in the undirected network.

Network of Users Who Reply to Postings

Besides casting votes on postings, users can also reply to postings of other users in the discussion area of a news article. When a user replies to another user's posting, an edge weighted according to the number of replies is created in the *Network of Users Who Reply to Postings*. This network is created in a directed and an undirected variant. If edges with opposite directions exist between two nodes in the directed graph, their weights are added in the undirected network.

Network Combining Users' Votes and Posting Replies

This network is based on the *Network of Users Who Vote on Postings* and the *Network of Users Who Reply to Postings*. The link weight is a weighted linear combination of the number of votes and posting replies, and the contribution of each is determined by the reciprocal of the respective proportion. A directed and an undirected graph are created. For the undirected network the respective undirected variants of the *Network of Users Who Vote on Postings* and the *Network of Users Who Reply to Postings* are used.

Network Combining Users' Votes, Posting Replies and Followers

Similar to the *Network Combining Users' Votes and Posting Replies*, this graph combines user interactions via votes and posting replies, while also incorporating users' followers. Users can follow other users to keep up to date with their postings on the news platform. A directed, weighted graph is created where the weighting of the edges between users is a weighted linear combination of the number of votes on postings, posting replies and the presence of a follow connection. The weights in the linear combination are calculated from the reciprocal of the respective proportions of the individual interaction types.

Network on Voting Behaviour of Users

For the *Network on Voting Behaviour of Users*, the voting habits of user pairs who have voted for the same postings by other users are modelled. The link weight for a pair of users is calculated based on the postings they both voted for by subtracting the number of times the users voted differently from the number of times they voted the same. Two users vote the same way if they both vote positively on a posting or both vote negatively, otherwise they vote differently. This network is implemented as a weighted, undirected graph and edges are only formed if the total edge weight is strictly positive.

Network on Posting Behaviour of Users

The *Network on Posting Behaviour of Users* is designed to model the posting behaviour of pairs of users who have posted in the discussion area of the same news articles. A weighted, undirected network is constructed in which the link weight reflects the number of news articles under which both users have posted at least once, ignoring multiple posting by users to the same news article.

Network of Users by Similarity of News Views

In contrast to the other networks, which utilize data on users' votes, postings or followers, the *Network of Users by Similarity of News Views* uses data about the news article views of users. In this weighted, undirected network a link between a pair of users reflects their similarity in terms of the news articles they viewed. The similarity of a user pair is quantified in this work by considering the news articles viewed by each user as a set and computing the similarity of the two sets using a structural similarity measure.

Verma and Aggarwal [VA20] conducted a theoretical and empirical analysis of several structural similarity measures akin to the Jaccard index. The authors suggest using the Salton's cosine index or the Overlap Coefficient, as these provided higher accuracy in their experiments on collaborative recommendations.

For this network, three different variants are explored by using either the Jaccard index, the Salton's cosine index or the Overlap coefficient as similarity measure. Let A and B be sets of news articles viewed by two users.

The Jaccard index measures the similarity between two finite sets A and B and is defined as the size of the intersection divided by the size of the union of the sets.

$$\text{Jaccard Index}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

The Salton's cosine index (aka Salton's Index) measures the similarity of sets as the ratio of the size of the intersection of two sets to the square root of the product of the sizes of the two sets.

$$\text{Salton's cosine index}(A, B) = \frac{|A \cap B|}{\sqrt{|A| \times |B|}}$$

The Overlap coefficient measures also the similarity between two sets and is defined as the ratio of the intersection size to the smaller of the two sets.

$$\text{Overlap coefficient}(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)}$$

All three described similarity measures take on values in the interval $[0, 1]$, and a higher value indicates a greater similarity between the sets.

After the construction of various networks, the subsequent step is to use community detection algorithms to identify community structure within these graphs.

3.4 User Community Detection

The result of the network construction, as described in Section 3.3, is an edge list for each graph created. An edge list contains triples consisting of the source user, the target user and the weight of the edge between them. The aim of identifying user communities is to determine which user community the users in a particular network are assigned to. In this section, the procedure for detecting user communities in networks is described.

3.4.1 Procedure for User Community Detection

The procedure for detecting user communities in a graph consists of several steps. At the beginning, a NetworkX graph instance is created based on an edge list. NetworkX [HSS08] is a Python library used for the creation, manipulation, and study of the structure, dynamics, and functions of networks. In this work, the NetworkX version 3.2 is used. The type of NetworkX graph to be created, either Graph or DiGraph, is selected according to whether the graph is undirected or directed. The weight of an edge connecting two users in a graph is encoded as an edge attribute in the NetworkX instance. Before the network is created, the edge list may be filtered using a threshold value for the edge weight. In this study, several edge weight thresholds per graph are examined to analyse the impact on the detected community structures. For community detection, a subgraph of the NetworkX graph instance is used, resulting from the largest connected component for an undirected graph or the largest weakly connected component for a directed graph. Various network measures are calculated for the graph used to identify user communities, including measures of node degree, distance and edge weights.

Two graph partitions are obtained by running the two community detection algorithms `Infomap` and `Louvain`. A graph partition divides the nodes of a graph into disjoint subsets, ensuring each vertex belongs to exactly one community. The consensus between two graph partitions is quantified with the Normalized Mutual Information (NMI) and the Rand Index, which are commonly used partition similarity measures [FH16].

Although a graph partition may contain communities of different sizes, this study focuses on evaluating the fairness of recommendations for large communities with a minimum size of 750 users. However, after filtering the original graph partition by a minimum community size threshold, the result is no longer a partition of the graph. In addition to calculating partition similarity scores for the original partitions, partition consensus scores are also calculated for the graph partitions resulting from the original partitions by combining all users in communities with less than 750 users into a single group. The latter is intended to emphasize the agreement in the assignment of users to large communities between the two community detection algorithms. Network measures, such as the number of nodes, edges, node degree, group centrality and edge weights, are computed for each large community within graphs that are filtered to include only users from these large communities. Additionally, the quality of a partitioned graph, containing only users from large communities, is quantified using the partition quality functions coverage, modularity and performance.

This community detection procedure is applied to all modelled networks, each filtered with different minimum edge weight thresholds, to analyse how varying these thresholds affects the community detection results. A network filtered by a specific edge weight threshold is selected based on the highest agreement between two community detection algorithms, as measured by the Normalized Mutual Information (NMI) score. For this network, one of the two graph partitions is selected based on an evaluation with partition quality functions.

This concludes the community detection procedure used in this thesis. The following sections provide a more detailed explanation of specific aspects of this approach.

3.4.2 Edge Weight Filtering

The edges in the graph are only slightly filtered, as higher edge weight thresholds are associated with a greater loss of nodes and edges. For the variants of the *Network of Users by Similarity of News Views*, the values of the 0.85, 0.9 and 0.95 quantile of the respective similarity measure are used as threshold values for the edge weights in order to reduce the graph density to below 0.2. In networks where edge weights are determined by a weighted linear combination of interaction counts, using the reciprocals of the respective interaction proportions as weights, edges are filtered by retaining only those whose weights are multiples of 1, 2, 3 or 4 times the smallest reciprocal proportion. For all other networks, the impact of edge weight filtering on community detection is analysed using thresholds of 1, 2, 3 and 4.

3.4.3 Network Measures

In the following, the network measures are presented, which are calculated on the graphs used for community detection. These measures are divided into general graph measures and specific measures for either undirected or directed graphs. Most network measures are calculated with the NetworkX package and a few with the igraph [CN06] Python package version 0.11.3 due to computational performance reasons. Unless otherwise specified, the NetworkX package is used for the calculation of network measures.

The formal definition of a graph from Section 3.3.1 is used to describe the network measures. Consider a weighted, simple graph $G = (V, E)$, where V is a set of vertices and E is a set of edges. The edges can either be undirected or directed and each edge is assigned a weight by a weight function $w : E \to \mathbb{R}^+$.

Network Measures in General

The following network measures are calculated for both undirected and directed graphs.

Number of Nodes The number of nodes in the graph, denoted as |V|.

Number of Edges The number of edges in the graph, represented by |E|.

Global Clustering Coefficient (Transitivity) The global clustering coefficient, or transitivity, C quantifies the extent of clustering, as it measures the probability that two neighbours of a node are connected. Transitivity is defined as the ratio of the number of closed triples, or triangles, to the total number of connected triples (both open and closed) in a graph. A connected triple in a graph consists of three nodes where at least two of them are connected by edges. A triangle is a connected triple where all three vertices are mutually connected by edges, forming a closed loop. In contrast, an open triple is a connected triple with only two edges. The factor three in the numerator arises because each triangle is counted three times when counting the connected triples.

$$C = \frac{(\text{number of triangles}) \times 3}{(\text{number of connected triples})}$$

When calculating this measure, directed graphs are regarded as undirected graphs. The implementation of this measure in the igraph package is used.

Distance Measures Before defining the average distance and the diameter, consider the distance $d(i, j)$ in a graph as the shortest path length in terms of the number of edges between nodes i and j. If no such path exists, the distance is defined as $d(i, j) := \infty$.

The average distance in a graph is the arithmetic mean of the finite distances between all pairs of vertices. It is calculated by summing the distances between all pairs of different nodes that are reachable from each other and dividing by the number of such pairs.

$$\text{Average Distance} = \frac{\sum_{i,j \in V,\, i \neq j,\, d(i,j) < \infty} d(i, j)}{|\{(i, j) \mid i, j \in V,\, i \neq j,\, d(i, j) < \infty\}|}$$

The diameter of a graph is defined as the longest distance between any pair of different vertices for which a path exists. To calculate the diameter of a graph, the shortest path between each pair of nodes is determined, and the diameter is then the length of the longest of these shortest paths.

$$\text{Diameter} = \max_{i,j \in V,\, i \neq j,\, d(i,j) < \infty} d(i, j)$$

These distance measures do not take into account the weights assigned to the edges. When calculating these measures, directed graphs are regarded as undirected graphs. For calculating the distances of all node pairs, the igraph package is used.

Edge Weight Measures The minimum, maximum and mean value of the edge weights are calculated for the network. The edge weight minimum $w_{\min}$ is the smallest weight assigned to any edge in the graph, while the edge weight maximum $w_{\max}$ is the largest weight assigned to any edge in the graph. The edge weight mean $\overline{w}$ is the arithmetic average of the edge weights in the network.

$$w_{\min} = \min_{e \in E} w(e) \qquad w_{\max} = \max_{e \in E} w(e) \qquad \overline{w} = \frac{1}{|E|} \sum_{e \in E} w(e)$$

For obtaining all edge weights in a graph the igraph package is used.

Network Measures for Undirected Graphs

Each of the subsequent network measures is calculated for undirected graphs only.

Node Degree Measures For an undirected graph, the minimum, maximum, and average node degree are calculated. Let $k_i = |\{j \in V \mid \{i,j\} \in E\}|$ represent the degree of a vertex i in an undirected graph, defined as the number of edges connected to i.

The minimum degree $k_{\min}$ is the smallest degree of any node in the graph, while the maximum degree $k_{\max}$ is the largest. The average degree $\langle k \rangle$ is the arithmetic mean of all degrees k_i, or twice the number of edges divided by the number of nodes in the graph.

$$k_{\min} = \min_{i \in V} k_i \qquad k_{\max} = \max_{i \in V} k_i \qquad \langle k \rangle = \frac{1}{|V|} \sum_{i=1}^{|V|} k_i = \frac{2|E|}{|V|}$$

Density The density ρ for a graph is the ratio of the number of edges $|E|$ to the maximum possible number of edges that could exist in a complete graph with the same number of vertices. For an undirected graph, the density is calculated as follows.

$$\rho = \frac{2|E|}{|V|(|V|-1)}$$

Network Measures for Directed Graphs

All the following network measures are only calculated for directed graphs.

Node Degree Measures For a directed graph, the minimum, maximum, and average node degree are calculated. Each vertex has two degrees in a directed graph. The in-degree $k_i^{\text{in}} = |\{j \in V \mid (j,i) \in E\}|$ denotes the number of directed connections coming to a vertex i and the out-degree $k_i^{\text{out}} = |\{j \in V \mid (i,j) \in E\}|$ is the number of directed outgoing edges of i. The degree k_i of a node i in a directed graph is the sum of its in-degree and out-degree: $k_i = k_i^{\text{in}} + k_i^{\text{out}}$.

The minimum in-degree $k_{\min}^{\text{in}}$ is the smallest in-degree of any node in the graph, while the maximum in-degree $k_{\max}^{\text{in}}$ is the largest. Analogously, the minimum out-degree $k_{\min}^{\text{out}}$ is the smallest out-degree of any node in the graph, while the maximum out-degree $k_{\max}^{\text{out}}$ is the largest.

$$k_{\min}^{\text{in}} = \min_{i \in V} k_i^{\text{in}} \qquad k_{\max}^{\text{in}} = \max_{i \in V} k_i^{\text{in}} \qquad k_{\min}^{\text{out}} = \min_{i \in V} k_i^{\text{out}} \qquad k_{\max}^{\text{out}} = \max_{i \in V} k_i^{\text{out}}$$

The average in-degree $\langle k^{\text{in}} \rangle$ and the average out-degree $\langle k^{\text{out}} \rangle$ in a directed network are equal and defined as the arithmetic mean of either all in-degrees or all out-degrees. Simplified, the average in-degree $\langle k^{\text{in}} \rangle$ and average out-degree $\langle k^{\text{out}} \rangle$ are calculated as the number of edges divided by the number of nodes.

$$\langle k^{\text{in}} \rangle = \frac{1}{|V|} \sum_{i=1}^{|V|} k_i^{\text{in}} = \langle k^{\text{out}} \rangle = \frac{1}{|V|} \sum_{i=1}^{|V|} k_i^{\text{out}} = \frac{|E|}{|V|}$$

Reciprocity The reciprocity r in a directed graph quantifies the proportion of edges that are bidirectional. If there is a directed edge from node i to node j and an edge from j to i in a directed network, then these edges are said to be reciprocated. The reciprocity is the ratio of reciprocated edges to the total number of edges in the graph.

$$r = \frac{|\{(i,j) \in E \mid (j,i) \in E\}|}{|E|}$$

Density The density ρ for a graph is the ratio of the number of edges $|E|$ to the maximum possible number of edges that could exist in a complete graph with the same number of vertices. For a directed graph, the density is calculated as follows.

$$\rho = \frac{|E|}{|V|(|V|-1)}$$

3.4.4 Detecting Communities With the Infomap Algorithm

One method utilized in this study to detect communities in a graph is the `Infomap` algorithm. The implementation of the algorithm in version 2.7.1 of the infomap [EHR23] Python package is used. `Infomap` is a network clustering method based on the Map equation. For this work, the algorithm is run with the standard arguments, except for the parameters "num_trials", which is set to 20, and "flow_model", which is set to "directed" in the case of a directed graph and "undirected" for an undirected graph. The weighted graph as a NetworkX instance is loaded by the algorithm, the community detection is performed, which then returns the assignment of the nodes to the top-level modules, i.e., the partition of the graph.

3.4.5 Detecting Communities With the Louvain Algorithm

This study also utilizes the `Louvain` algorithm to identify the community structure in a network. The implementation of the algorithm from the NetworkX [HSS08] Python package version 3.2 is used. The agglomerative community detection algorithm `Louvain` greedily maximizes the Newman-Girvan modularity and returns a hierarchy of partitions. Two functions for identifying community structure based on the `Louvain` algorithm are provided in the NetworkX package, which return either the partition of the graph with the highest modularity or the partitions of all levels in the hierarchy.

Lancichinetti and Fortunato [LF09], who tested several community detection algorithms on various benchmark graphs, suggest using the `Infomap` and `Louvain` algorithms because they perform well and have low computational complexity. However, the authors clarified in an erratum [LF14] that in their experiments they used the partition from the first pass, resulting from the initial aggregations of nodes into clusters, instead of the partition with the maximum modularity, as otherwise the performance would have been very poor. The issue of selecting a partition of the `Louvain` partition hierarchy was also discussed in [FH16]. Based on these findings, this work also uses the partition resulting after the first `Louvain` pass rather than the partition with the highest modularity. As with the `Infomap` method, the implementation of `Louvain` in the NetworkX package also takes into account whether the graph is directed or undirected. Default function arguments are used to obtain the graph partition with the `Louvain` algorithm.

3.4.6 Graph Partition Similarity Measures

Two community detection algorithms are applied to a given network to obtain two variants for a partition of the graph. Graph partition similarity measures are used to quantify the similarity of two partitions of nodes in a graph. Such measures can be used for comparing the results of a community detection algorithm with a ground truth partition. According to Fortunato and Hric [FH16], there are several ways to quantify the similarity of partitions. In this work, the Normalized Mutual Information (NMI) and the Rand Index are used to evaluate the agreement between the two algorithms in assigning nodes to communities. Both the NMI and the Rand Index can be used as a consensus measure because they are symmetrical with respect to their arguments, which means that swapping the arguments does not change the score. The values for the NMI and the Rand Index lie in the range $[0, 1]$. A low value indicates minimal agreement or shared information between the partitions, while a high value signifies strong consensus. The scikit-learn [PVG+11] Python package version 1.4 is used to compute the Normalized Mutual Information and the Rand Index between two graph partitions.

Normalized Mutual Information (NMI)

Mutual information (MI) is an information-theoretic measure of the dependence between two random variables. It quantifies the shared information of the variables as the amount of information conveyed about one random variable through the observation of the other random variables. If two random variables are independent, knowing one variable gives no information about the other, so the mutual information is zero. But, if there is a dependence between two random variables, the value of one random variable provides information about the other, resulting in a positive mutual information value.

In the context of community detection, the mutual information is used to measure the similarity of two partitions. Based on the notations of [New18] and [FH16], let us consider two graph partitions $\mathcal{X} = (X_1, X_2, ..., X_{q_X})$ and $\mathcal{Y} = (Y_1, Y_2, ..., Y_{q_Y})$, with q_X and q_Y clusters, respectively. Each vertex is assigned to a group in both of the partitions.

Consider n the total number of vertices, n_i^X and n_j^Y the number of vertices in clusters X_i and Y_j, and n_{ij} the number of vertices shared by clusters X_i and Y_j: $n_{ij} = |X_i \cap Y_j|$.

The mutual information $MI(\mathcal{X};\mathcal{Y})$ measures the reduction in uncertainty about partition $\mathcal{X}$ due to knowing $\mathcal{Y}$, or the amount of information shared between partitions $\mathcal{X}$ and $\mathcal{Y}$:

$$MI(\mathcal{X};\mathcal{Y}) = H(\mathcal{X}) - H(\mathcal{X}|\mathcal{Y})$$

where $H(\mathcal{X})$ is the Shannon entropy of partition $\mathcal{X}$ and $H(\mathcal{X}|\mathcal{Y})$ is the conditional entropy of $\mathcal{X}$ given $\mathcal{Y}$. The entropy $H(\mathcal{X})$ is the total information contained in partition $\mathcal{X}$:

$$H(\mathcal{X}) = -\sum_{i=1}^{q_X} P(X_i) \log P(X_i)$$

where $P(X_i) = n_i^X/n$ is the probability of assigning a node to group X_i in partition $\mathcal{X}$. The conditional entropy $H(\mathcal{X}|\mathcal{Y})$ quantifies the additional information needed to describe the cluster assignments of partition $\mathcal{X}$, given knowledge of the assignments in $\mathcal{Y}$:

$$H(\mathcal{X}|\mathcal{Y}) = -\sum_{j=1}^{q_Y} P(Y_j) \sum_{i=1}^{q_X} P(X_i|Y_j) \log P(X_i|Y_j)$$

where $P(X_i|Y_j) = n_{ij}/n_j^Y$ represents the probability that a node is in group X_i in partition $\mathcal{X}$, given it is assigned to group Y_j in partition $\mathcal{Y}$.

Without normalization, the mutual information can range from 0 to $\min(H(\mathcal{X}), H(\mathcal{Y}))$. To make this measure comparable between different networks, it is usually normalized to the interval $[0, 1]$. The Normalized Mutual Information $NMI(\mathcal{X};\mathcal{Y})$ is most commonly computed by dividing the mutual information $MI(\mathcal{X};\mathcal{Y})$ by the arithmetic average of the entropies of partitions $\mathcal{X}$ and $\mathcal{Y}$:

$$NMI(\mathcal{X};\mathcal{Y}) = \frac{2MI(\mathcal{X};\mathcal{Y})}{H(\mathcal{X}) + H(\mathcal{Y})}$$

Rand Index

The Rand Index is commonly used to measure the similarity between two partitions. It is determined by calculating the number of pairs of vertices that are either in the same or in different communities, taking both partitions into account. Based on the notations of [FH16], let us consider two partitions $\mathcal{X} = (X_1, X_2, ..., X_{q_X})$ and $\mathcal{Y} = (Y_1, Y_2, ..., Y_{q_Y})$ of a network, with q_X and q_Y clusters, respectively. Consider n as the total number of vertices, with each vertex assigned to a group in both partitions.

Define a_{00} as the number of node pairs that are in different communities in both partitions and a_{11} as the number of node pairs that are in the same community in both partitions.

The symbols a_{10} and a_{01} indicate the number of pairs of nodes that are in the same community in $\mathcal{X}$ ($\mathcal{Y}$) but in different communities in $\mathcal{Y}$ ($\mathcal{X}$), respectively.

The Rand Index $R(\mathcal{X}, \mathcal{Y})$ is defined as the ratio of the number of vertex pairs correctly assigned in both partitions (i.e., either in the same or in different groups) to the total number of pairs:

$$R(\mathcal{X}, \mathcal{Y}) = \frac{a_{11} + a_{00}}{a_{11} + a_{01} + a_{10} + a_{00}} = \frac{2(a_{11} + a_{00})}{n(n-1)}$$

Values of the Rand Index lie in the interval $[0, 1]$, where a large value represents a high agreement between the partitions $\mathcal{X}$ and $\mathcal{Y}$ in the assignment of node pairs to groups.

3.4.7 Community Structure Network Measures

Running a community detection algorithm on a network produces a partition of the graph's vertex set, dividing the nodes into distinct groups, or communities. Each node is assigned to a specific group, with the aim of identifying clusters where nodes are more densely connected within the community than to those in other communities. The number of communities identified and their size varies depending on the graph and the algorithm used to detect communities. This work focuses on large communities of at least 750 users identified in a graph.

Various network measures commonly applied in network analysis are used to characterize the large user communities detected by a particular community detection algorithm. In the following, the network measures used to describe the community structure are presented. The network measures used include those presented in Section 3.4.3, along with additional measures. Measures of community structure are calculated for a subgraph containing only users who belong to large communities.

For each community, the number of nodes (or users), the number of edges within the same community (intra-community edges) and the number of edges connecting users to a different community (inter-community edges) are determined. On group level, the graph density and the global cluster coefficient (transitivity) are calculated. The minimum, maximum and mean value of the weights of intra-community edges are computed. In the case of an undirected network, the minimum degree, the maximum degree and the average degree are determined based on intra-community edges. For directed graphs, the minimum in-degree, the maximum in-degree, the minimum out-degree, the maximum out-degree, as well as the average in-degree and the average-out-degree are calculated for intra-community edges. In addition, the reciprocity is calculated for each user community in a directed graph.

When the graph partition contains several large communities, group centrality measures are also computed. In network analysis, centrality measures quantify the importance or influence of nodes within a network. Everett and Borgatti [EB99] extended the centrality measures of degree, closeness and betweenness to be applicable to groups of individuals.

In this study, the group degree centrality and group closeness centrality are computed using the NetworkX Python package. For computational reasons, the calculation of the group betweenness centrality is omitted. Subsequently, the group centrality measures computed for each user community are described.

Group Degree Centrality Group degree centrality of a group of nodes is the proportion of nodes outside the group (non-group nodes) that are connected to vertices within the group. The group degree centrality for a group of nodes S in a graph is defined as

$$\text{Group Degree Centrality} = \frac{|\bigcup_{i \in S} N(i) \setminus S|}{|V| - |S|}$$

where $N(i)$ is the set of adjacent vertices (neighbours) of node i.

For a directed graph, the group out-degree centrality and group in-degree centrality are calculated. Group out-degree centrality measures the directed outgoing links from vertices within the community to non-group nodes, while group in-degree centrality quantifies the directed incoming links from non-group nodes to group vertices.

Values of the group degree centrality lie in the interval $[0, 1]$, whereby a value of zero means that there is no connection between nodes outside the group and vertices in the group. In contrast, a high group centrality indicates that many non-group nodes are connected with at least one vertex within the group.

Group Closeness Centrality Group closeness centrality of a group of vertices measures the closeness of the group to the other nodes in the network. It is the inverse of the average shortest distance from nodes in group S to all nodes outside of group S. The group closeness centrality for a group of nodes S in a graph is defined as

$$\text{Group Closeness Centrality} = \frac{|V \setminus S|}{\sum_{j \in V \setminus S} d_{S,j}}$$

where $d_{S,j} = min_{i \in S}\, d(i, j)$ is the minimum finite distance from any vertex i in group S to the node j.

Edge weights are not taken into account when calculating the distance between two nodes. For directed graphs, the incoming distances are computed using directed edges from nodes outside of group S to the nodes in group S.

Group closeness centrality values are in the interval $[0, 1]$. A group closeness centrality value of zero means that there is no connection between at least one node in group S and any vertex outside the group. Higher group closeness centrality values indicate that the group S is very close to all other non-group nodes in the graph in terms of the number of edges or shortest paths.

3.4.8 Partition Quality Functions

Partition quality functions quantify the quality of a graph partition in terms of some criterion. Different partitions of a graph can be evaluated by comparing the quality function scores of the partitions. While high partition quality scores generally indicate good partitions, determining the best partition depends on the concept of a community and the quality function used. Some quality functions used in the literature are the coverage, modularity and performance [For10].

After running two community detection algorithms on a given network, two partitions of the graph are obtained. Then, a subgraph of the original graph used for community detection is created by retaining only communities that meet a minimum user size. This means that only users who belong to a large community are kept in the subgraph. In this work, partition quality functions are then used to quantify the quality of the partition of a graph consisting exclusively of users belonging to large communities. All partition quality functions are calculated with the NetworkX Python package for this study. Descriptions of the partition quality functions coverage, performance and modularity are given next.

The following notation is used for a formal definition of the partition quality functions. Consider a partition $\mathcal{P} = (P_1, P_2, ..., P_{q_P})$ of a graph $G = (V, E)$ with q_P clusters and let c_i denote the community to which node i belongs. In this network different types of edges are present. Intra-community edges $E_{\text{intra}} = \{(i,j) \in E \mid c_i = c_j\}$ are edges where both endpoints belong to the same community in the network. Inter-community edges $E_{\text{inter}} = \{(i,j) \in E \mid c_i \neq c_j\}$ connect nodes from different communities. Inter-community non-edges $E_{\text{non-inter}} = \{(i,j) \notin E \mid c_i \neq c_j\}$ refer to pairs of nodes from distinct communities that are not directly linked.

Coverage

The coverage of a graph partition $\mathcal{P}$ is the ratio of the number of intra-community edges to the total number of edges in the graph.

$$\text{Coverage}(\mathcal{P}) = \frac{|E_{\text{intra}}|}{|E|}$$

Values of the graph partition quality coverage lie in the interval $[0, 1]$. A higher value indicates that a larger proportion of the edges in the graph are within identified communities. If all edges in the graph exist only between vertices within the same community, the coverage of the graph partition would reach its maximum value of one.

Performance

The performance of a graph partition $\mathcal{P}$ is the number of intra-community edges plus inter-community non-edges divided by the total number of potential edges in the network. The total number of potential edges E_{complete} is $|V|(|V|-1)/2$ in an undirected graph and $|V|(|V|-1)$ in a directed graph, respectively.

$$\text{Performance}(\mathcal{P}) = \frac{|E_{\text{intra}}| + |E_{\text{non-inter}}|}{|E_{\text{complete}}|}$$

Values of the graph partition quality performance are within the interval $[0, 1]$. Unlike the coverage of a graph partition, the performance of $\mathcal{P}$ not only considers intra-community edges but also emphasizes the presence or absence of edges between nodes from different communities. The performance of a graph partition may differ from its coverage. For instance, if a graph partition contains a single cluster, the coverage of the partition would reach its maximum value of one, but the performance score would be relatively low. This is because all edges and vertices are within one cluster, and there are no inter-community pairs of vertices.

Modularity

The modularity of Newman and Girvan [NG04] is a popular partition quality function based on the idea that a random graph is expected to lack a community structure, and hence, clusters can be identified by comparing the actual edge density in a subgraph to the expected density when vertices are connected at random. Specifically, the modularity measures the difference between the fraction of edges within communities and the expected fraction of such edges in a random graph with the same degree distribution [For10].

If the number of edges within a community do not deviate from the expected number of edges in a random network, the modularity is zero. A higher positive modularity score indicates a graph with a strong community structure. Modularity values are strictly less than one and are generally in the range from 0.3 to 0.7 [NG04]. A modularity value above about 0.3 is a good indicator of a significant community structure in a graph [CNM04].

Although originally defined for undirected, unweighted graphs, modularity was extended to graphs with positively weighted edges [New04] and directed links [LN08]. In this study, weighted graphs are used, and therefore the modularity is computed by considering the edge weights instead of just the edge count. Specifically, the sum of the weights of edges is used instead of the number of edges, and the weighted node degree is used instead of the node degree in the calculation of modularity. The modularity function of the NetworkX package is used in this work.

Consider the following notation for a formal description of modularity. The weight of the edge connecting vertices i and j is represented by A_{ij}. The sum of all edge weights in the graph is denoted by m. The weighted degree of node i is k_i in an undirected graph, while for directed graphs, k_i^{in} and k_j^{out} denote the weighted in-degree and out-degree of nodes i and j, respectively. The Kronecker delta function $\delta(c_i, c_j)$ is 1 if nodes i and j are in the same community and 0 otherwise. L_c represents the sum of weights of the edges within community c, and k_c is the sum of the weighted degrees of the vertices in community c. For directed graphs, k_c^{in} and k_c^{out} are the sums of the weighted in-degrees and out-degrees of the nodes in community c, respectively.

The expected weight for an edge between the vertices i and j in an undirected graph is $(k_i k_j)/(2m)$ and in a directed graph $(k_i^{\text{in}} k_j^{\text{out}})/m$, respectively, if links in the graph are randomly established and node degrees are respected.

The modularity Q for a weighted, undirected graph is defined as follows [NG04, CNM04].

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) = \sum_{c=1}^{q_P} \left[\frac{L_c}{m} - \left(\frac{k_c}{2m} \right)^2 \right]$$

In the first formula of modularity, the sum is over vertex pairs and the only contribution comes from those node pairs belonging to the same cluster, whereas the rewritten, second modularity formula sums over the q_P clusters. In the second modularity formula, the first term of each summand is the proportion of the sum of the edge weights within community c, and the second term is the expected fraction of the sum of the weighted degrees of nodes within cluster c, if the graph were a random graph with the same degree distribution as the original network [FH16].

Similarly, the modularity Q for a weighted, directed graph is defined as follows [LN08].

$$Q = \frac{1}{m} \sum_{ij} \left(A_{ij} - \frac{k_i^{\text{in}} k_j^{\text{out}}}{m} \right) \delta(c_i, c_j) = \sum_{c=1}^{q_P} \left[\frac{L_c}{m} - \frac{k_c^{\text{in}} k_c^{\text{out}}}{m^2} \right]$$

3.4.9 Assigning Users to a Community

Once a community structure is determined for a specific network, news recommendation algorithms are then trained and evaluated. All registered, active users on the news platform of "DER STANDARD" are taken into account when recommending news, with particular attention being paid to the fairness of recommendations for large user communities. Large communities are defined as those with 750 or more users. One peculiarity is that the fairness-aware algorithm used in this study (`ADV-VAE`) requires information about the assignment of each user to a community during model training.

It is possible that not all users are present in the graph used for community detection. Reasons for this include users not being part of the constructed network edge list due to a lack of interactions (e.g., users not voting on postings at all), users being omitted after filtering the graph based on an edge weight threshold, or users not being in the largest (weakly) connected component.

Users are assigned a label that indicates their affiliation to a community. Each user is only assigned to one community. Those users who are not included in the graph used for community detection for the selected network are labelled "Not in network". Users who are part of the original graph partition but are in communities that do not reach the minimum community size required to be classified as large are collectively assigned the label "Small Sized Community". This means that all users who belong to small communities are combined into one group. All other users who are part of a large community are labelled accordingly to indicate their membership of this community.

3.5 News Recommendation

This study investigates the extent to which various recommendation algorithms provide different recommendations for news articles depending on a user's group membership in a constructed user network. These user groups are derived from detected communities in a selected network of users. A graph and partition are selected based on the consensus between two community detection algorithms and the partition quality. All registered, active users on the online news platform of "DER STANDARD" are taken into account when recommending news articles. Each user is associated with a user group, even if the user does not belong to a large user community or is not part of the constructed network, as described in the previous Section 3.4.9. User group membership is considered a protected user attribute. News articles are recommended to users based on their clicks on news articles. Recommendations are evaluated for all users as a whole, with a particular focus on the fairness of recommendations received by users in large user communities of at least 750 users discovered in the selected graph. This section presents the experiment design used in this study for recommending news articles to users.

3.5.1 Procedure for News Recommendation

In this section, the procedure for training and evaluating news recommendation algorithms is described in detail. This work follows the experiment design used in the studies of Melchiorre et al. [MRP+21] and Ganhör et al. [GPR+22] for the recommendation task. At first, the dataset obtained from the data preprocessing described in Section 3.2 is prepared for news article recommendation. Data on users' news article views is used to construct a binary user-item interaction matrix, with rows representing users and columns representing news articles. If a user has viewed a news article, the corresponding entry in this matrix has a value of one, otherwise it is zero. For the training and evaluation of the recommender algorithms, the user group of each user, to which the user is assigned based on the detected user communities in a particular network, is used as a protected user attribute. Data about the news section (or "Ressort") to which a news article is associated is used from the dataset to evaluate the diversity of recommendations.

A wide variety of data splitting strategies have been proposed in the literature on recommender systems [MMMO20]. For the experiments conducted in the present study, the dataset is split based on a user splitting strategy. The reader is referred to the work of Melchiorre et al. [MRP+21], who also used this user splitting strategy and described it with an appealing visualization. Accordingly, users are divided into a training, validation and test set in a 60:20:20 ratio. Training users and their entire history of interactions with news articles are used to train the recommendation algorithms. The recommendations are evaluated either on the validation users or test users by providing 80% of the users' items sampled uniformly at random as input to the recommender algorithms and using the remaining 20% as ground truth for calculating the recommendation evaluation measures. This means that the items for the evaluation procedure are randomly divided into a training and a test set in a ratio of 80:20 per user.

In order to obtain an estimate of the effectiveness of the recommendations for all users, cross-validation with five folds is performed, following the approach used by Melchiorre et al. [MRP+21], but with the folds stratified to ensure balanced representation of the user groups defined by the protected user attribute. More specifically, the users are divided into five subsets of (roughly) equal size, three of which are used for training (60% of users), one for validation (20% of users) and one for testing (20% of users). For each of the five different partitions of users, the procedure for training and evaluating the recommendation algorithms is carried out as described before. These subsets are systematically and cyclically rotated five times, so that each user ends up in a test set once. Even though the number of users may be unevenly distributed in different user groups, the training data is not upsampled in this work.

Among the recommender systems studied in this work are several traditional collaborative filtering algorithms and one fairness-aware recommendation algorithm, where the latter aims to provide fairer recommendations by minimizing the sensitive information encoded in the learned latent user representations with respect to a protected user attribute. These news recommendation algorithms are compared with two baseline algorithms, which provide either random or the most popular news articles.

Once a recommender system has been trained, it provides a list of recommended news articles, sorted according to their presumed relevance to the user. News articles that the user has already interacted with are typically removed from this recommendation list. The effectiveness of the recommendation system is then evaluated based on the news article interactions of the validation users. Optimal values for the tunable parameters, or hyperparameters, of a recommendation algorithm are determined using a grid search over different hyperparameter value combinations and the achieved recommendation performance according to a recommendation evaluation metric.

After hyperparameter optimization and model selection, the performance of the recommendation algorithms is assessed on the test set using various recommendation evaluation measures of accuracy and beyond-accuracy. The recommendation results in terms of the accuracy and beyond-accuracy metrics of the RS algorithms are reported and discussed for all users overall and for each of the large user communities. In addition, user community fairness metrics are used to quantify both the difference between the mean recommendation results of user groups and the extent to which a recommendation algorithm reinforces existing imbalances in the training data with respect to the distribution of user groups.

The following sections describe in more detail the recommendation algorithms, model selection and hyperparameter optimization, as well as the measures used to evaluate the recommendations for users and user communities.

3.5.2 Recommender System Algorithms

This study investigates the extent to which various recommendation algorithms for implicit data provide different recommendations for news articles depending on a user's group membership in a constructed network. Among the algorithms studied are traditional

collaborative filtering algorithms based on nearest neighbours, matrix factorization and variational autoencoder, as well as a fairness-aware recommendation algorithm based on a variational autoencoder with adversarial training. All algorithms are applied to a binary user-item interaction matrix. The recommendation algorithms used are briefly described in the following.

- Random Items (`RAND`) serves as a non-personalized baseline that returns items selected uniformly at random from the entire item collection.
- Popular Items (`POP`) is as a non-personalized method that recommends items ranked based on their popularity among the training users. The popularity of each news article is measured by the number of views it has received from distinct users.
- Item k-Nearest Neighbours (`ITEM-KNN`) [SKKR01, DK04] is an item-based collaborative filtering algorithm that recommends items based on item similarities and past user interactions. The intuition behind this approach is that a user will be interested in recommended items similar to those the user liked before. In the model-building phase the similarity between all pairs of items are computed using a similarity function. Then, for each item, only the top-k most similar items (nearest neighbours) are retained in an item-item similarity matrix. In the subsequent recommendation phase, the most similar items to those the user has interacted with are used to generate recommendations. The recommendation scores of items are calculated by combining the item similarity values with the user's item interaction history. Finally, the top-K items with the highest scores are selected as the recommendations for the user. This work uses the cosine similarity for computing item similarities.
- Alternating Least Squares (`ALS`) [HKV08] is a matrix factorization method, a collaborative filtering technique that decomposes the user-item interaction matrix into low-rank latent user and item factors, that represent user preferences and item characteristics in a common latent space, allowing the prediction of a user's preference for an item by calculating the dot product of the corresponding user and item factors. ALS alternates between optimizing the user and item factors iteratively through regularized least squares for implicit user-item feedback data, while taking into account different levels of confidence in the observed interactions.
- Bayesian Personalized Ranking (`BPR`) [RFGS09] uses Bayesian inference to provide a model-agnostic optimization criterion and learning algorithm for predicting personalized item rankings. BPR directly learns to rank item pairs for each user by maximizing the difference in recommendation scores between items users have interacted with and those they have not. BPR employs a stochastic gradient descent algorithm with bootstrap sampling to optimize the ranking criterion. Various collaborative filtering algorithms can be integrated with BPR to optimize model parameters for personalized ranking. Matrix factorization is used as core predictor within the BPR framework to learn user and item factors in this work.

- Sparse Linear Method (SLIM) [NK11] is a linear model that generates recommendations for items to a user by aggregating the items from a user's past interaction history with the coefficients of a sparse item-item similarity matrix. This matrix is learned by solving a constrained ℓ_1-norm and ℓ_2-norm regularized optimization problem. Coefficients in the learned item-item similarity matrix represent relations between items. The j-th column in the item-item similarity matrix is computed by solving a regularized regression problem, where the j-th column of the user-item interaction matrix is estimated from the other columns.

- Variational Autoencoder with Multinomial Likelihood (VAE) [LKHJ18] is a generative recommender model that adopts a neural network architecture and uses variational inference for model optimization. The input to the model is a vector of item interaction data of one user, and the output is a vector of preference probabilities over all items for that user. As it is an autoencoder, it consists of an encoder and decoder. Unlike a traditional autoencoder, the encoder of a variational autoencoder outputs the parameters of a Gaussian distribution for the latent variables, while the decoder reconstructs the user's item interaction history as closely as possible from a sample of this latent distribution. For the encoder and decoder, a symmetrical architecture with multilayer perceptrons is used, and their parameters are learned by maximizing the likelihood of the observed user-interaction data under the generative model by optimizing the evidence lower bound (ELBO) using gradient-based optimization methods. After model training, personalized item recommendations are generated by feeding a user's item interactions into the model and selecting the top-K items sorted in decreasing order of their predicted preference probabilities from the model's output. The VAE model used in this work is an extension of variational autoencoders for collaborative filtering of implicit feedback data, utilizing a multinomial likelihood distribution to model the data and controlling regularization in the optimization objective with a parameter that is tuned by annealing.

- Adversarial Variational Autoencoder with Multinomial Likelihood (ADV-VAE) [GPR+22] is an extension of the VAE model used in this work that integrates an adversarial component to reduce bias in recommendations and to provide fairer recommendations. The key difference between ADV-VAE and VAE is the inclusion of an adversarial network that attempts to predict a specific protected user attribute from the latent representations produced by the encoder. During model training, the model is presented with the item interaction history of a user as well as a label indicating the user's affiliation to a user group, the protected attribute. The adversarial network, implemented as a feedforward network, learns to identify a user's protected attribute from the latent representation. Simultaneously, the model is trained to optimize the objective of the variational encoder, the evidence lower bound, and to minimize the sensitive information encoded in the latent representation concerning the protected attribute.

In this work, CPU-based implementations of the algorithms ITEM-KNN, ALS and BPR from the implicit [Fre23] Python package version 0.7.2 are used. CPU-based implementation of the SLIM algorithm in the "recommendation_systems_fairness" GitHub repository[2] provided by Melchiorre et al. [MRP+21] and GPU-based implementations of the VAE and ADV-VAE models in the "adv-multvae" GitHub repository[3] provided by Ganhör et al. [GPR+22] are used.

3.5.3 Model Selection and Hyperparameter Optimization

At first, a manual investigation of different hyperparameter configurations is performed to get an impression of the performance of the recommendation algorithms on the validation set. Then, the hyperparameters of the studied algorithms are selected by a grid search over various parameter values. The Table B.1 in the appendix lists the examined values of the hyperparameters and fixed parameters, along with a description of the parameters for each recommender algorithm. The best hyperparameter values are determined based on the best average results of an evaluation metric across the validation set folds.

This evaluation metric used for model selection is NDCG@10 for all models with tuneable hyperparameters, except for the ADV-VAE. The best hyperparameters found for the VAE, which are common with the ADV-VAE, are fixed for the ADV-VAE, and the hyperparameters specific to the ADV-VAE are selected based on the lowest value of the average balanced accuracy (BAcc) of the adversarial network in predicting the actual protected attribute of users across data batches. In a classification setting, the balanced accuracy is the average of the recall values calculated for each class, with values in the range $[0, 1]$, where zero means no correct classifications and one indicates that the classifier is correct for every class.

This work follows the suggestion of Ganhör et al. [GPR+22] to use BAcc for selecting the best ADV-VAE model. It seems that Ganhör et al. selected the best ADV-VAE model based on the lowest BAcc value and not on the highest NDCG value as for the VAE. However, in our opinion, the description of the model selection in their study is not entirely clear, as the exact model hyperparameters they tuned were not specified. Our experience with ADV-VAE shows that optimizing its hyperparameters, including both those specific to ADV-VAE and those shared with the VAE, using solely BAcc leads to the selection of a model with very poor recommendation performance in terms of NDCG. For this reason, we decided to optimize only the hyperparameters specific to the ADV-VAE with BAcc and to set the common hyperparameters with the VAE to values resulting from the selection of the VAE model using NDCG.

Early stopping is performed for the VAE and ADV-VAE models, i.e., the model with the best performance across the training epochs is selected based on the validation results. Following validation, the selected model with the best performance is subsequently assessed on the test set with various recommendation evaluation measures.

[2] https://github.com/CPJKU/recommendation_systems_fairness/tree/4808295
[3] https://github.com/CPJKU/adv-multvae/tree/2dfcb41

3.5.4 Recommendation Evaluation Measures

A news recommendation algorithm generates for each user being evaluated a ranked list of news articles, ordered by their estimated relevance for the user according to the algorithm. In general, the algorithm's recommendations are compared with users' actual views of news articles for evaluation. In this study, the effectiveness of the recommender system algorithms is evaluated using three measures of accuracy: normalized discounted cumulative gain (NDCG), precision and recall. Recommendations are also evaluated using three beyond-accuracy measures: coverage, diversity and novelty. These recommendation evaluation measures are calculated for a ranked recommendation list up to the position K. The length of the recommendation list provided may vary depending on the requirements and the context or domain in which a recommendation system is used. Despite computing the recommendation evaluation measures for the top-K recommended news articles, where $K = \{5, 10, 20, 50\}$, this study focuses on the top $K = 10$ recommendations and the results for the other evaluation levels are documented in the appendix. Next, the measures used to evaluate the recommendations are explained.

The following notation is used to define the evaluation metrics for recommendations. Given a user u, the symbol $\hat{R}(u)$ denotes a ranked recommendation list of items created by a recommender algorithm that is truncated to a length of K, i.e., $|\hat{R}(u)| = K$. Consider $\hat{R}_i(u)$ as the item recommended at the i-th position for user u. Let $R(u)$ represent a ground-truth set of items, such as news articles, with which the user u has interacted.

Normalized Discounted Cumulative Gain (NDCG) Normalized Discounted Cumulative Gain (NDCG) measures how useful the items in a top-K recommendation list are for a user by evaluating the utility gain of each item in the list based on its position and relevance for the user. The utility of each recommended item, such as a news article, is discounted by a logarithmic factor based on its rank in the list. The utility of a recommendation list, the discounted cumulative gain (DCG), is the cumulative utility gain of the individual recommended items. The DCG is usually normalized by the ideal DCG (IDCG), which is achieved by the ideal ranking that contains all relevant items for the user sorted by descending order of relevance [GSY22, Ren22]. In this work, the relevance of a news article to a user u is determined by whether u actually viewed the news article. The NDCG, DCG and IDCG are defined as follows.

$$\text{NDCG@}K(u) = \frac{\text{DCG@}K(u)}{\text{IDCG@}K(u)}$$

$$\text{DCG@}K(u) = \sum_{i=1}^{K} \frac{\mathbf{1}_{R(u)}(\hat{R}_i(u))}{\log_2(i+1)} \qquad \text{IDCG@}K(u) = \sum_{i=1}^{\min(K,|R(u)|)} \frac{1}{\log_2(i+1)}$$

where the indicator function $\mathbf{1}_{R(u)}(\hat{R}_i(u)) = 1$ if $\hat{R}_i(u) \in R(u)$ and 0 otherwise.

Values of NDCG lie in the range $[0, 1]$, where a NDCG value of one indicates that the relevant items are perfectly ranked, while a value of zero means no relevant items are in the recommendation list.

Precision Precision of a recommendation list for a user u is the proportion of recommended news article that are relevant out of the total number of news articles recommended [HKTR04]. Precision only considers whether an item (such as a news article) is relevant or not, without taking into account the rank or position of the item in the recommendation list.

$$\text{Precision@}K(u) = \frac{|\hat{R}(u) \cap R(u)|}{K}$$

Precision takes values in $[0, 1]$, where zero means none of the recommended items are relevant and one means that all recommended items are relevant to the user, respectively.

Recall Recall of a top-K recommendation list for a user u is the ratio of recommended news article that are relevant to the total number of relevant news articles available for that user [HKTR04]. Like precision, recall also does not take into account the rank of the items in the recommendation list.

$$\text{Recall@}K(u) = \frac{|\hat{R}(u) \cap R(u)|}{|R(u)|}$$

Recall values are in the range of $[0, 1]$, where a high recall value means that many of the user's available relevant items are recommended, while a low recall value means few of the user's available relevant items are recommended.

Coverage The extent to which the generated recommendations cover the catalogue of available items, in the case of this work news articles, is quantified with the coverage or also referred to as catalogue coverage [KB17]. Unlike the other measures used in this study to evaluate recommendations, coverage is evaluated for a collective of users. Coverage is defined as the proportion of news articles that are recommended at least once to a set of users $\mathcal{U}$.

$$\text{Coverage@}K(\mathcal{U}) = \frac{|\bigcup_{u \in \mathcal{U}} \hat{R}(u)|}{|\mathcal{I}|}$$

where $\mathcal{I}$ represents the catalogue of all news articles.

Coverage values are in the range $[0, 1]$, where high coverage shows that the recommender system is able to recommend a greater variety of items, while low coverage indicates that only few items are recommended.

Diversity Diversity refers to the variety and distinctiveness within a collection of items [CHV22]. In this study, diversity is computed for each user as the normalized Shannon entropy on the level of news sections (or "Ressorts"), following the work of Melchiorre et al. [MRP+21]. Diversity of a recommendation list for user u is defined as

$$\text{Diversity@}K(u) = -\frac{1}{\log_2 |S_u|} \sum_{s_i \in S_u} p(s_i) \log_2 p(s_i)$$

where S_u is the set of the unique news sections to which the top-K news articles recommended to user u belong, and $p(s_i)$ is the proportion of news articles in the top-K recommendations that are classified with news section s_i.

Diversity takes on values in the interval $[0, 1]$. A value of one means that every news article recommended to the user u is from a different news section, and zero indicates that all top-K recommended news articles are associated with the same news section.

Novelty A recommended item is considered novel to a user if the item is unknown to the user. The novelty of an item is commonly estimated using the inverse of its popularity among users of the recommender system [KB17]. For example, the popularity of news articles can be estimated by the number of views by different users, and news articles with low popularity are more likely to be new to a target user. This definition of novelty is also referred to as global long-tail novelty because it considers novel items to be rare items in the long tail of the popularity distribution, independent of any individual user [CHV22, KB17]. In this study, the novelty of recommendations is measured using a popularity-based item novelty scheme called "Expected Popularity Complement" (EPC) proposed by Vargas and Castells [VC11].

The popularity of a news article i respectively the probability of i being seen $P(seen \mid i)$ is estimated by the fraction of users used to train a recommender algorithm U_{train} who actually interacted with i.

$$P(seen \mid i) = \frac{|\{u \in U_{\text{train}} \mid i \in R_u\}|}{|U_{\text{train}}|}$$

The novelty of a top-K recommendation list for user u is

$$\text{Novelty@}K(u) = C \sum_{i=1}^{K} rd(i)[1 - P(seen \mid \hat{R}_i(u))]$$

where $rd(i) = 1/\log_2(i+1)$ is a rank discount and $C = 1/\sum_{i=1}^{K} rd(i)$ is a normalizing constant. This novelty metric does not take into account whether an item is relevant for the user, but it is rank-sensitive. In this work, the novelty of the recommended articles decreases by a logarithmic factor of the rank, as is the case with the NDCG measure.

In general, an item with a high novelty score is an item that few users have interacted with, whereas an item with a low novelty score is a popular item. The novelty measure used in this work takes values in $[0, 1]$. A novelty value of zero indicates that all recommended items are well-known and have been seen by all users, while the maximum value of one indicates that all recommended items are entirely new and have not been seen by any users. The rank discount function ensures that items ranked higher in the recommendation list contribute more to the EPC value.

3.5.5 User Community Recommendation Fairness Measures

This work adopts the RecGap and the Compounding Factor recommendation fairness measures proposed in the work of Melchiorre et al. [MRP+21]. Since these measures are applicable for evaluating the recommendation fairness among an arbitrary number of user groups, they are used in this work to evaluate the fairness of large user communities discovered in a network of users constructed from user interaction data. Both measures can be computed for any evaluation measure that can be calculated at user level and aggregated at user group level, such as NDCG, or diversity. RecGap quantifies the gap or difference between the average recommendation evaluation scores of the various user groups. The Compounding Factor measures the extent to which a recommendation algorithm reinforces existing imbalances in the training data with regard to the distribution of user groups.

Consider a recommender algorithm, which, if necessary, was trained using a training set of users and their interactions with items. Then, the predicted recommendations for users in a test set are evaluated separately for each user in a certain user community using various recommendation evaluation measures based on accuracy and beyond-accuracy. In the following, the RecGap and Compounding Factor are computed for such an evaluation measure μ and defined for a set G of user groups that correspond to the large user communities detected in a network within this work.

RecGap – Measuring Recommendation Unfairness Among User Groups

As in the work of Melchiorre et al. [MRP+21], a recommender system is considered fair in this study if it performs equally well on any evaluation measure for all user groups. RecGap [MRP+21] quantifies the average disparity between user groups with regard to a particular recommendation evaluation measure μ, such as NDCG. More precisely, RecGap^{μ} is computed as the mean of the absolute differences in the evaluation measure μ between all pairs of user groups, calculated based on the average evaluation scores of users within each group.

$$\text{RecGap}^{\mu} = \frac{\sum_{(g,g') \in G_{\text{pair}}} \left| \frac{\sum_{u \in U_g} \mu(u)}{|U_g|} - \frac{\sum_{u' \in U_{g'}} \mu(u')}{|U_{g'}|} \right|}{|G_{\text{pair}}|}$$

where $G_{\text{pair}} = \{(g, g') \mid g, g' \in G,\ g \neq g'\}$ is the set of pairs of different user groups, U_g is the set of users in group g and $\mu(u)$ is the score of the evaluation measure μ for user u.

RecGap returns a value that is either zero or positive. A value of zero indicates that the recommender algorithm is fair, whereas a higher positive value represents the degree of unfairness in recommendations among user groups for a certain evaluation measure μ.

Compounding Factor – Measuring Amplified User Group Imbalances

Consider a recommendation algorithm that is to some degree unfair, applied to a dataset with different user groups of varying sizes. Like the work by Melchiorre et al. [MRP+21], this study also expects the performance gains from the algorithm to be proportional to the size of each group. Fairness is assessed by quantifying the gains for each group according to an evaluation metric μ and comparing them to their population size. If the gains are disproportionate, it indicates the algorithm is compounding the initial population bias present in the data. More precisely, Melchiorre et al. [MRP+21] compare the distribution of the user groups of the population with a distribution that quantifies an aggregate of the evaluation scores of the metric μ.

The population distribution B is the proportion of users in each group. For instance, in a user population with two groups, a population distribution $B = [0.8, 0.2]$ means that 80% of users belong to the first group and 20% to the second user group.

The metric scores distribution C^μ over user groups represents the portion of the score of the metric μ regarding each group, and it is denoted by $C^\mu = \{c_g^\mu \mid g \in G\}$. Each element c_g^μ of C^μ is a probability, defined as the sum of the evaluation scores of all users in group g divided by the sum of the evaluation scores of all users across all groups:

$$c_g^\mu = \frac{\sum_{u \in U_g} \mu(u)}{\sum_{g' \in G} \sum_{u' \in U_{g'}} \mu(u')}$$

where U_g represents the set of users in group g and $\mu(u)$ is the score of the evaluation measure μ for user u.

Ultimately, the difference between the two probability distributions is characterized with the Compounding Factor measure. The Compounding Factor CompFct^μ regarding an evaluation measure μ is the Kullback-Leibler (KL) divergence of the population distribution B from the metric scores distribution C^μ.

$$\text{CompFct}^\mu = \text{KL}\left(B \parallel C^\mu\right) = \sum_{g \in G} B(g) \log\left(\frac{B(g)}{C^\mu(g)}\right)$$

The value of CompFct^μ is non-negative. A value of zero indicates that the two probability distributions are identical, while a higher Compounding Factor shows a greater amplification of the imbalance of the user groups in recommendations with respect to μ by the recommender algorithm and an intensification of the population bias.

CHAPTER 4

Experimental Results

The key results of the experiments performed are presented and interpreted in this chapter. Information about the dataset obtained from the preprocessing of the data is given at the beginning. Subsequently, the results of identifying communities in different user networks with two community detection algorithms are presented. The effect of filtering the network edges by a minimum edge weight value on the clustering result is examined. An analysis of the consensus of the graph partitions is performed, followed by a discussion of the number and size of the detected user communities. In particular, the chosen network and graph partition for categorizing users into user groups, which are used to evaluate news recommendation algorithms, are explained. The chapter concludes with a summary of the overall recommendation performance of different news recommendation algorithms and an analysis of recommendation fairness at the level of user communities.

4.1 Data Preprocessing

Various data preprocessing steps are applied to the raw data, including time-based filtering and filtering of news articles, users, postings, votes, as well as users' views of news articles and follow relations. The data preprocessing results in two datasets, one for the construction of user networks and the other for the recommendation of news articles. The data records for the former task are from an earlier time period than those for the latter task. Data used for network construction is separate and not shared with data used for news recommendations. Users' views of news articles are used to recommend news articles. Various user networks with diverse user relationships are created based on users' postings, votes, follow relationships and views of news articles. However, users' postings, votes and views of news articles from both time periods are used to filter for active users and to remove outliers.

After data preprocessing, the dataset comprises 12,724 registered and active users of the Austrian online news platform "DER STANDARD". Data on a total of 24,508 follow

Table 4.1: Number of data records per entity after data preprocessing for the network construction and news recommendation periods, as well as for the entire period. The number of news articles, postings, users, users' follow relations, users' views of news articles and votes are listed.

Entity	Network Construction Period	News Recommendation Period	Whole Period
News Articles	58,076	16,495	74,571
Postings	5,200,254	1,666,114	6,866,368
Users	—	—	12,724
Users' Follow Relations	24,508	—	24,508
Users' News Article Views	27,238,923	8,463,142	35,702,065
Votes	15,548,772	5,157,591	20,706,363

relations is available for these users and is used to design the relations in a specific user network. The dataset also contains 58,076 news articles for the network construction time frame and 16,495 for the recommendation period. Without counting duplicates, there are 27,238,923 and 8,463,142 views from different users for these news articles in the graph construction and news recommendation periods respectively. Postings and votes used to create certain networks of users account for 5,200,254 and 15,548,772 records respectively. Table 4.1 summarizes the number of records per entity after data preprocessing.

In Figure 4.1 the distribution of the "DER STANDARD" news sections ("Ressorts") of the news articles used for the network construction and news recommendation is presented as a grouped bar plot. The data in the bar chart is sorted by the number of news articles used for the network construction. News articles of the news sections "Web", "Panorama" and "Sport" are most frequently used for both time frames. The subsequent sections "International", "Wirtschaft" and "Inland" have fewer but still considerable numbers of articles. Beyond these, the number of news articles drops considerably and the values for the other sections are much lower.

4.2 User Community Detection

The `Infomap` and `Louvain` algorithms are run to identify communities, or clusters, of users in all the weighted networks of users constructed. In these networks, the users are the nodes and the edges represent a particular user relationship with a certain intensity or magnitude. The effect of filtering graph edges by a minimum edge weight threshold on the clustering result is also investigated. Community detection is performed for each constructed network and variants of the graphs whose edges are filtered with an edge weight threshold. Both directed and undirected graphs are used as input for community detection, depending on the network design. Two graph partitions per network are obtained from both community detection algorithms. For 44 out of 45 different user networks created, both community detection algorithms yielded partitions with at least one large user community with a minimum size of 750 users.

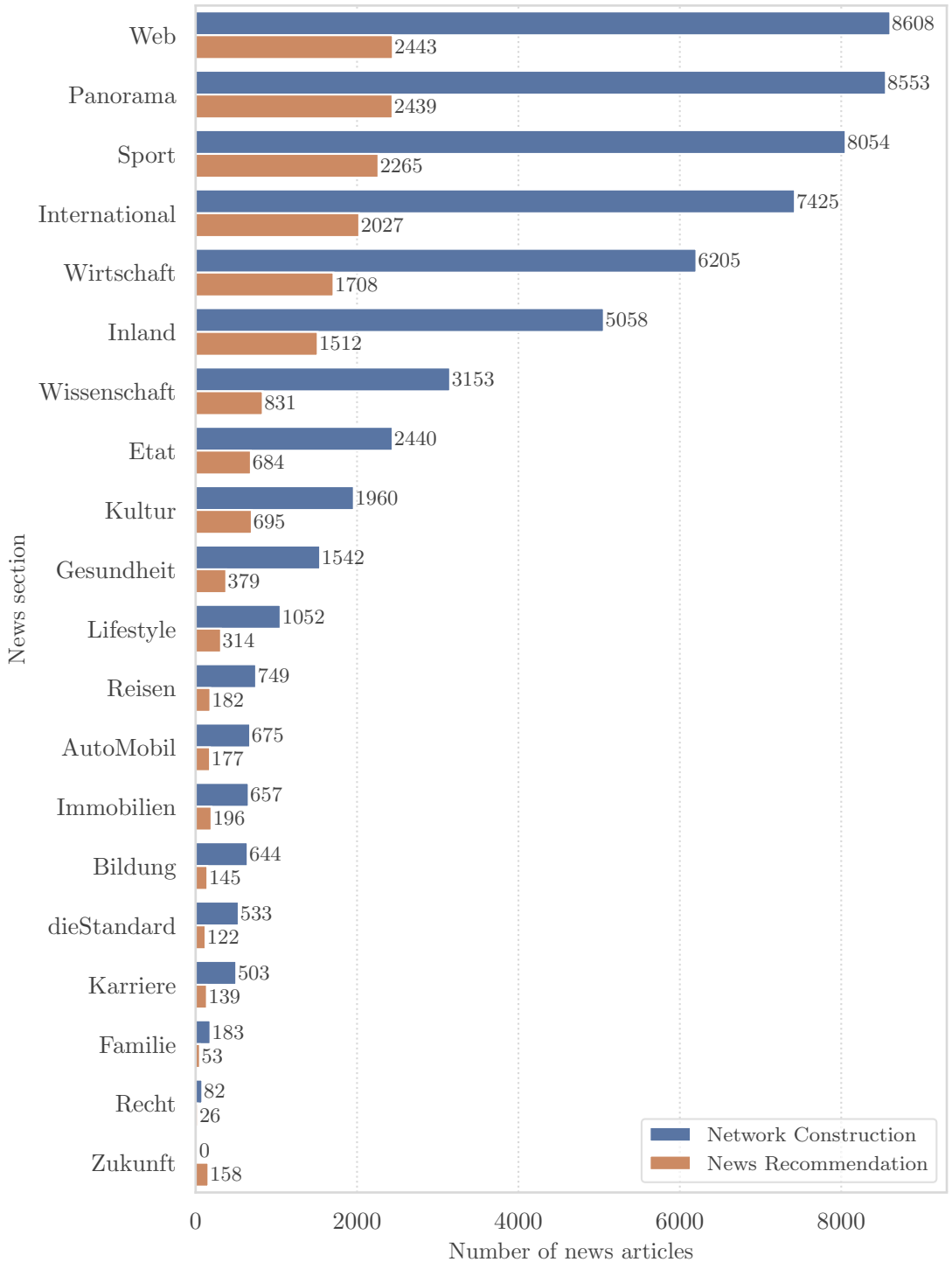

Figure 4.1: Distribution of the "DER STANDARD" news sections ("Ressorts") of the news articles used for network construction and news recommendation

4.2.1 Analysis of Consensus in Graph Partitions

Graph partition similarity measures are used to quantify the consensus between the `Infomap` and `Louvain` algorithms in identifying community structure in a network. The Normalized Mutual Information (NMI) and the Rand Index are used as consensus measures in this work. For each network and edge weight threshold, used to filter edges, the Table 4.2 shows the Normalized Mutual Information and Rand Index consensus scores for the partitions obtained by the `Infomap` and `Louvain` algorithms. It turns out that the agreement between the two community detection algorithms in assigning users to communities is very low for most of the user networks constructed. The distribution of NMI scores shows that 28 networks have an NMI below 0.1, 34 graphs below 0.3, 38 graphs below 0.4, and 42 graphs below 0.5. Only the undirected *Network of Users Who Vote on Postings*, whose edges are filtered by either an edge weight threshold of three or four, have high NMI scores of 0.66 and 0.73, respectively. Similarly, the distribution of Rand Index values reveals that 21 networks have a Rand Index below 0.3, 35 graphs below 0.4, and 39 graphs below 0.6. The undirected *Networks of Users Who Vote on Postings* with a filtered minimum edge weight of either three or four also achieve the highest Rand Index of 0.84 and 0.89. Using an edge weight threshold of either two or three for the directed variant of this network, the Rand Index is 0.78 and 0.84. In between lies the directed *Network of Users Who Reply to Postings* with a minimum edge weight of three and a value of 0.73 for the Rand Index. For those graphs where both undirected and directed variants are constructed, the consensus between the two community detection algorithms is generally not consistent, but network-dependent.

Interestingly, the choice of the minimum edge weight threshold used to remove edges in the graph before community detection definitely has an effect on the partitions obtained by `Infomap` and `Louvain` for some graphs. This can be illustrated with the undirected *Network of Users Who Vote on Postings*. Both community detection algorithms have the best agreement on the communities detected in this network when using an edge weight threshold of three or four (NMI scores: 0.66 and 0.73, Rand Index scores: 0.84 and 0.89). However, when the minimum edge weight in this graph is either one or two, the consensus between the `Infomap` and `Louvain` algorithms is very low (NMI scores: 0.01 and 0.07, Rand Index scores: 0.27 and 0.28). One might except that the more edges are removed, the greater the agreement between the algorithms, but this is not always the case, as the directed variant of this network shows. For the directed *Network of Users Who Vote on Postings*, the NMI and Rand Index increase when using an edge weight threshold of either two or three, but with a minimum edge weight of four the consensus scores drop to values close to those when using an edge weight threshold of one. This illustrates that the communities detected by community detection algorithms can vary considerably depending on the minimum edge weight threshold chosen.

Another interesting observation is the effect of integrating users' follow connections with their votes and posting replies on the partition consensus scores. In the directed *Network Combining Users' Votes and Posting Replies*, the weight of edges between users is modelled with a weighted linear combination of the number of votes and posting replies, and the

contribution of each is determined by the inverse of the respective proportions. The weights in the directed *Network Combining Users' Votes, Posting Replies and Followers* are calculated by including whether a user follows another user in the linear combination. Since the number of follow connections is the smallest share compared to the number of votes and posting replies, an existing follow relationship between users contributes the most to a link's weight. The graph partitions obtained by `Infomap` and `Louvain` for the directed *Network Combining Users' Votes and Posting Replies* show a very poor agreement and hardly any difference for the studied edge weight thresholds (NMI scores: 0.01–0.05, Rand Index scores: 0.23–0.26). However, when users' follower relationships are included for the directed *Network Combining Users' Votes, Posting Replies and Followers*, both consensus measures increase to moderately low values (NMI scores: 0.31–0.41, Rand Index scores: 0.35–0.55).

In addition to calculating partition similarity scores for the original partitions, partition consensus scores are also calculated for the graph partitions resulting from the original partitions by combining all users in communities with less than 750 users into a single group. The latter is intended to emphasize the agreement between the two community detection algorithms in assigning users to large communities. Table A.3 in the appendix shows the Normalized Mutual Information (NMI) and Rand Index consensus scores for the graph partitions when small user communities are grouped together. The partition consensus scores are similar to the original graph partitions, but generally even lower across networks. There are 40 out of 44 graphs with an NMI below 0.16 and a Rand Index below 0.54. As for the original partitions, the undirected *Networks of Users Who Vote on Postings*, whose edges are filtered by an edge weight threshold of either three or four, have the highest agreement in assigning users to communities among the `Infomap` and `Louvain` algorithms (NMI scores: 0.62 and 0.69, Rand Index scores: 0.83 and 0.88). The directed variants of this network with a minimum edge weight of either two or three follow with a moderate consensus among the community detection algorithms (NMI scores: 0.41 and 0.47, Rand Index scores: 0.76 and 0.79). An alternative to using graph partition similarity measures to quantify the consensus between two community detection algorithms with respect to the detected community structure is to compare the number and size of detected user communities.

4.2.2 Number and Size of Detected Communities

One expectation of this work is to find more than one large community of users for which to evaluate the utility of news recommendations. This study defines a large user community as one with 750 or more users. User networks are created based on the interaction data of 12,724 users of the "DER STANDARD" news platform. However, some users may not be part of certain networks that are created, for example, if they do not post in the discussion area of news articles. The Table A.2 in the appendix shows the number of detected communities, categorized by size intervals, for each modelled network, whose edges may have been filtered using an edge weight threshold. In general, the number and size of communities detected varies between user networks, with many

Table 4.2: This table shows the Normalized Mutual Information (NMI) and Rand Index consensus scores for graph partitions obtained by the `Infomap` and `Louvain` algorithms for the constructed user networks, which may have been filtered with an edge weight threshold. All scores are rounded to the second digit. Highest values are shown in bold.

Network Name	Edge Weight Threshold	NMI	Rand Index
Undirected Network of Users Who Vote on Postings	4.00	**0.73**	**0.89**
Undirected Network of Users Who Vote on Postings	3.00	0.66	0.84
Directed Network of Users Who Vote on Postings	3.00	0.49	0.84
Directed Network of Users Who Reply to Postings	3.00	0.48	0.73
Directed Network of Users Who Vote on Postings	2.00	0.42	0.78
Directed Network Combining Users' Votes, Posting Replies and Followers	3.44	0.41	0.55
Directed Network Combining Users' Votes, Posting Replies and Followers	4.59	0.38	0.44
Undirected Network of Users Who Reply to Postings	4.00	0.37	0.49
Directed Network Combining Users' Votes, Posting Replies and Followers	2.30	0.37	0.50
Directed Network Combining Users' Votes, Posting Replies and Followers	1.15	0.31	0.35
Undirected Network of Users Who Reply to Postings	3.00	0.28	0.38
Directed Network of Users Who Reply to Postings	2.00	0.27	0.39
Undirected Network of Users Who Reply to Postings	2.00	0.19	0.33
Undirected Network of Users by Similarity of News Views (Jaccard Index)	0.07	0.17	0.32
Undirected Network of Users by Similarity of News Views (Jaccard Index)	0.05	0.11	0.36
Undirected Network of Users by Similarity of News Views (Jaccard Index)	0.06	0.10	0.28
Undirected Network of Users by Similarity of News Views (Salton's Index)	0.14	0.07	0.28
Undirected Network of Users Who Vote on Postings	2.00	0.07	0.28
Undirected Network of Users by Similarity of News Views (Salton's Index)	0.12	0.06	0.35
Undirected Network of Users by Similarity of News Views (Salton's Index)	0.11	0.06	0.35
Directed Network Combining Users' Votes and Posting Replies	4.59	0.05	0.24
Directed Network Combining Users' Votes and Posting Replies	3.44	0.05	0.26
Undirected Network on Voting Behaviour of Users	4.00	0.04	0.31
Undirected Network Combining Users' Votes and Posting Replies	4.59	0.04	0.23
Directed Network of Users Who Reply to Postings	1.00	0.04	0.23
Directed Network Combining Users' Votes and Posting Replies	2.29	0.03	0.23
Undirected Network Combining Users' Votes and Posting Replies	3.44	0.03	0.24
Undirected Network Combining Users' Votes and Posting Replies	2.30	0.03	0.23
Undirected Network of Users Who Reply to Postings	1.00	0.02	0.23
Directed Network of Users Who Vote on Postings	1.00	0.02	0.27
Undirected Network of Users Who Vote on Postings	1.00	0.01	0.27
Directed Network Combining Users' Votes and Posting Replies	1.15	0.01	0.24
Directed Network of Users Who Vote on Postings	4.00	0.01	0.28
Undirected Network on Voting Behaviour of Users	3.00	0.00	0.31
Undirected Network of Users by Similarity of News Views (Overlap Coef.)	0.18	0.00	0.35
Undirected Network of Users by Similarity of News Views (Overlap Coef.)	0.20	0.00	0.35
Undirected Network on Posting Behaviour of Users	4.00	0.00	0.27
Undirected Network Combining Users' Votes and Posting Replies	1.15	0.00	0.25
Undirected Network on Posting Behaviour of Users	3.00	0.00	0.27
Undirected Network on Posting Behaviour of Users	2.00	0.00	0.27
Undirected Network on Posting Behaviour of Users	1.00	0.00	0.28
Undirected Network on Voting Behaviour of Users	2.00	0.00	0.33
Undirected Network on Voting Behaviour of Users	1.00	0.00	0.39
Undirected Network of Users by Similarity of News Views (Overlap Coef.)	0.24	0.00	0.26

more small communities identified than large ones. In fact, small communities of up to a size of about 50 users, and especially those with fewer than 10 users, are the most common. Communities with 250 to 1,000 users are the least frequent. Special cases are the undirected *Network on Posting Behaviour of Users* and the undirected *Network of Users by Similarity of News Views (Overlap Coef.)*, where `Infomap` groups all users of the respective network into one user community, while `Louvain` identifies three and four large user communities, respectively. These two networks only have large user communities. Both community detection algorithms seem to agree that the directed *Network Combining Users' Votes, Posting Replies and Followers* in particular has many small communities with a size between $[0, 50]$. The `Infomap` community detection algorithm finds only one large community with a size of more than 1,000 users in almost all created networks. In contrast, the `Louvain` algorithm identifies three or four user communities with a size of more than 750 users in nearly all graphs, except in some rare cases where it detects one and two large communities once and five large communities three times. But, in the undirected *Networks of Users Who Vote on Postings*, `Louvain` identifies four and three large communities for minimum edge weights of three and four, respectively, while `Infomap` detects three. In the directed variants of this network, the `Infomap` algorithm finds three large user communities, while `Louvain` identifies four, using an edge weight threshold of either two or three. After analysing the consensus scores of the graph partitions along with the number and size of the detected user communities, a network and graph partition are selected, which are then used to evaluate the fairness of news recommendations for user communities.

4.2.3 Selection of a Network and Graph Partition

Based on the results of performing the user community detection for different networks of users, a network and graph partition is selected. The selected graph and its respective partition are used to assign users to user groups, which are then at the centre of the evaluation of the news recommendation algorithms. An important concern in this work is not to select just any network of users and partition. Instead, a network is selected for which ideally more than one large user community is found and for which there is a high degree of agreement between the two community detection algorithms.

A network whose edges may have been filtered by an edge weight threshold is selected based on the highest agreement between the `Infomap` and `Louvain` community detection algorithms, quantified by the Normalized Mutual Information (NMI) score. Both the NMI and the Rand Index clearly show that both community detection algorithms return the most similar graph partitions for the undirected *Network of Users Who Vote on Postings*. Using an edge weight threshold of either three or four for this network, the NMI values are 0.66 and 0.73 and the Rand Index scores are 0.84 and 0.89 for the original graph partitions obtained. When evaluating the graph partitions resulting from the original partitions by combining all small communities into a single group, these networks also have the highest consensus scores of all networks (NMI scores: 0.62 and 0.69, Rand Index scores: 0.83 and 0.88).

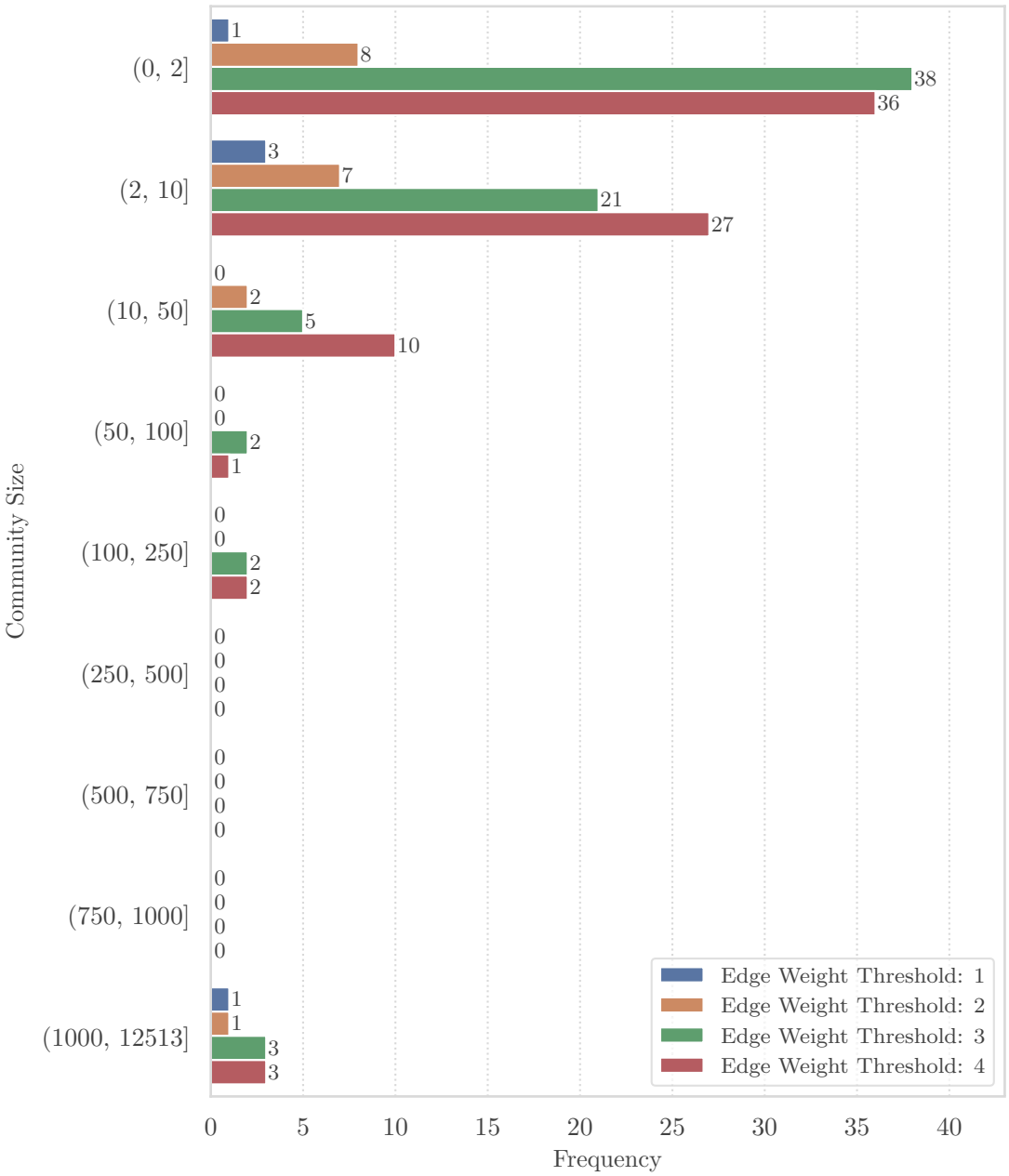

Figure 4.2: Distribution of the size of the user communities detected by the `Infomap` algorithm in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with different edge weight thresholds.

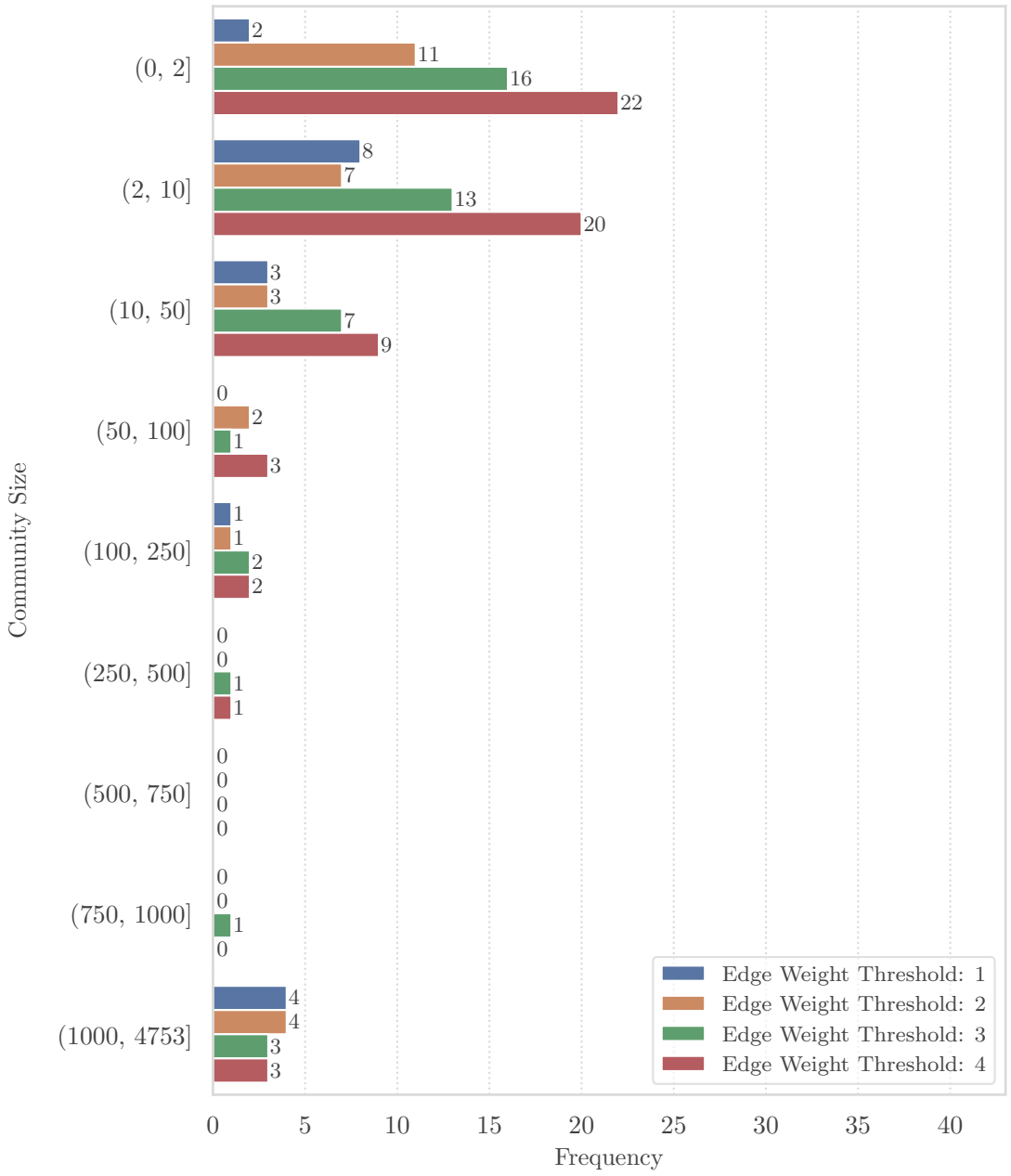

Figure 4.3: Distribution of the size of the user communities detected by the `Louvain` algorithm in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with different edge weight thresholds.

The investigation of the number and size of detected communities shows that the `Infomap` algorithm typically returns only a single, large community across all networks created. However, the undirected *Network of Users Who Vote on Postings* is a special case, as `Louvain` identifies four and three large communities for minimum edge weights of three and four, respectively, while `Infomap` detects three. Figure 4.2 and Figure 4.3 visualize the distribution of the size of user communities detected by the `Infomap` and `Louvain` algorithms in the undirected *Network of Users Who Vote on Postings* for different edge weight thresholds. It is noticeable in the bar plots that `Infomap` finds smaller communities up to a size of ten more often than `Louvain` for this network, especially when the graph has a minimum edge weight of three or four. There seems to be a tendency for the `Infomap` algorithm to find more communities the higher the minimum edge weight used to filter the edges. Both community detection algorithms hardly find any user clusters with a size between 250 and 1,000 users. Looking at the larger communities with more than 1,000 users, it is noticeable that `Infomap` finds only one community each for the networks with a minimum edge weight threshold of either one or two, while `Louvain` finds four clusters each, but when the edge weight threshold is three or four, both algorithms find three communities each.

Table 4.3: This table describes the undirected *Network of Users Who Vote on Postings* whose edges are filtered using different edge weight thresholds with various graph measures.

Network Measure	**Network Filtered by Edge Weight**			
	1	**2**	**3**	**4**
Number of Nodes	12,535	11,745	10,424	9,092
Number of Edges	5,535,637	1,932,500	1,003,722	621,288
Graph Density	0.0705	0.0280	0.0185	0.0150
Node Degree Minimum	1	1	1	1
Node Degree Maximum	6,740	4,564	3,404	2,667
Node Degree Mean	883.23	329.08	192.58	136.67
Edge Weight Minimum	1	2	3	4
Edge Weight Maximum	2,790	2,790	2,790	2,790
Edge Weight Mean	2.11	4.18	6.20	8.17
Global Clustering Coefficient	0.29	0.26	0.25	0.24
Average Distance	1.98	2.26	2.42	2.50
Diameter	4	5	6	7

The community structure found varies for networks with different edge weights. Therefore, it is of interest to analyse the variants of the undirected *Network of Users Who Vote on Postings* whose edges have been filtered with different edge weight thresholds using measures typically used in social network analysis. Table 4.3 lists various graph measures for this network using different edge weight thresholds. The undirected *Network of Users Who Vote on Postings*, without any edge weight filtering, contains 12,535 users and 5,535,637 edges, which corresponds to a graph density of 0.0705. In this network with an edge weight threshold of one, some users are connected to only one other user, while others are connected to at most 6,740 other users, and the mean node degree is around 883.23. The edge weights in this graph variant are on average 2.11, with a range of one to 2,790. The global clustering coefficient is 0.29, the average distance is 1.98 and

the graph diameter is four. It is obvious and can be seen from the table that filtering the graph according to a minimum edge weight value leads to the loss of nodes and edges. For example, using an edge weight threshold of four reduces the number of users in the network to 9,092 and the number of links to 621,288, giving a graph density of 0.015 approximately. The higher the minimum edge weight in the undirected *Network of Users Who Vote on Postings*, the lower the graph density, the mean node degree and the global clustering coefficient, but the higher the average edge weight, the average distance and the graph diameter. A complete description of all modelled networks using graph measures is provided in Table A.1 in the appendix.

Table 4.4: Network properties of the user communities detected by the `Infomap` and `Louvain` algorithms in the undirected *Network of Users Who Vote on Postings* when a minimum edge weight of four is required. Several network measures are computed for the communities of the graph, which is filtered by users belonging to communities with a minimum size of 750 users. Both community detection algorithms find three large user communities for this network, which are named A, B and C in order of size. The large user communities identified in the graph are described with different community structure parameters. For each user community, a subgraph in the network, the number of nodes, intra-community edges and inter-community edges are computed. The graph density, the global clustering coefficient and the parameters for node degrees and edge weights are calculated based on intra-community edges. The importance or influence of user groups is quantified using group centrality measures of degree and closeness.

Community Structure Parameter	**Detected Communities by Algorithm**					
	Infomap			**Louvain**		
	A	**B**	**C**	**A**	**B**	**C**
Number of Nodes	4,694	2,173	1,214	3,957	2,252	1,612
Number of Intra-Community Edges	254,114	133,290	26,193	213,616	133,654	38,031
Number of Inter-Community Edges	135,821	117,986	64,433	130,298	113,496	86,624
Graph Density	0.0231	0.0565	0.0356	0.0273	0.0527	0.0293
Global Clustering Coefficient	0.27	0.37	0.30	0.29	0.37	0.28
Node Degree Minimum	1	1	1	1	1	1
Node Degree Maximum	2,069	1,140	660	1,918	1,158	780
Node Degree Mean	108.27	122.68	43.15	107.97	118.70	47.18
Edge Weight Minimum	4	4	4	4	4	4
Edge Weight Maximum	1,179	2,790	1,836	1,179	2,790	1,836
Edge Weight Mean	7.67	9.84	11.21	7.83	9.78	10.16
Group Degree Centrality	0.79	0.69	0.60	0.77	0.68	0.64
Group Closeness Centrality	0.83	0.76	0.71	0.81	0.76	0.74

The undirected *Network of Users Who Vote on Postings*, whose edges are filtered by a minimum edge weight of four, is chosen to be used for categorizing users into groups, because it has the highest agreement between the `Infomap` and `Louvain` algorithms according to the Normalized Mutual Information and the Rand Index consensus scores, and both algorithms detected three large user communities. In order to assign users to a user group, it is also necessary to select one of the two graph partitions. Table 4.4 compares the graph partitions obtained from `Infomap` and `Louvain`, focusing on the

large communities of this network. More precisely, the graph is filtered to include only communities with at least 750 users, with large user communities viewed as subgraphs and analysed using various network analysis measures. In this table, it is noticeable that the network properties of the three large communities detected by both algorithms are of similar magnitude. Suppose the three communities discovered are named A, B and C, in order of size. The number of users in these communities varies, with 4,694 users in community A, 2,173 in community B and 1,214 users in community C, according to `Infomap`. Similarly, the `Louvain` algorithm finds that community A has 3,957 users, community B has 2,252 users, while C has 1,612 users. The number of intra-community edges and inter-community edges for the two community algorithms is also similar across the three communities. Interestingly, the number of inter-community edges in community C is more than twice as high as the number of intra-community edges, whereas in the other communities intra-community edges predominate by a factor of about 1.1–1.8. The higher level of inter-community edges in community C could be due to several reasons. Community detection algorithms may have misclassified users in this community. In addition, the members of community C may have weak cohesion, i.e., they don't form a tightly-knit group. Alternatively, this community may overlap with other communities or have common members, leading to increased links between communities. Community C could also play a bridging role, connecting several other communities. However, the group centralities of degree and closeness are higher for communities A and B than for community C, suggesting that community C is less central or connected than the other communities. Group centralities tend to decrease as the size of the community decreases. Although community A has more intra-community edges, its lower density and global clustering coefficient relative to community B indicate that it has proportionally fewer internal connections and is less cohesive, i.e., the nodes are more loosely connected. Community C's density and global clustering coefficient are of similar order of magnitude to community A. The average node degree is highest in community B and of a similar scale to community A, while community C has the lowest value, which is less than half as large. However, the smaller the community, the larger the average edge weight of the graph. The consistent detection of similar clusters by both algorithms according to the network analysis also confirms that these clusters are a stable and genuine part of the network.

The similarity of the `Infomap` and `Louvain` graph partitions of the undirected *Network of Users Who Vote on Postings*, whose edges are filtered by a minimum edge weight of four, is also evident from the partition quality scores. For the selected network, the `Louvain` partition has better performance (0.63 vs. 0.58) and marginally higher modularity (0.36 vs. 0.35) than the `Infomap` partition, although `Infomap` has slightly better coverage (0.72 vs. 0.70). The Table A.4 in the appendix shows the scores of the partition quality functions for all modelled networks. Since the `Louvain` graph partition for the undirected *Network of Users Who Vote on Postings* has slightly better modularity and performance, this graph partition is chosen as the basis for categorizing users into groups. Figure 4.5 shows a visualization of this network, where the nodes are coloured according to the graph partition obtained by the `Louvain` algorithm.

4.2.4 Categorization of Users Into Groups

Users are categorized into user groups based on the user communities detected by the `Louvain` algorithm in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered by a minimum edge weight threshold of four. Both the `Infomap` and `Louvain` community detection algorithms find three large user communities with a size of at least 750 users in this network, and the agreement in assigning users to user clusters between the algorithms is the highest compared to other constructed networks. The `Louvain` graph partition is used over the `Infomap` partition because of its slightly higher partition quality, quantified in terms of modularity and performance.

Each of the 12,724 users is assigned a label. The undirected *Network of Users Who Vote on Postings*, whose edges are filtered by a minimum edge weight threshold of four, comprises a total of 9,092 users. Of these, 3,957 users ($\approx 43.52\%$) belong to the largest community A, another 2,252 users ($\approx 24.77\%$) belong to the second-largest community B, followed by 1,612 users ($\approx 17.73\%$) who are in community C. The remaining 1,271 users ($\approx 13.98\%$) in the graph are detected as part of smaller clusters and are grouped together in a group named "Small communities". All other 3,632 users who are not included in the graph used for community detection for the selected network are categorized in the "Not in network" user group. Figure 4.4 visualizes the categorization of users into user groups.

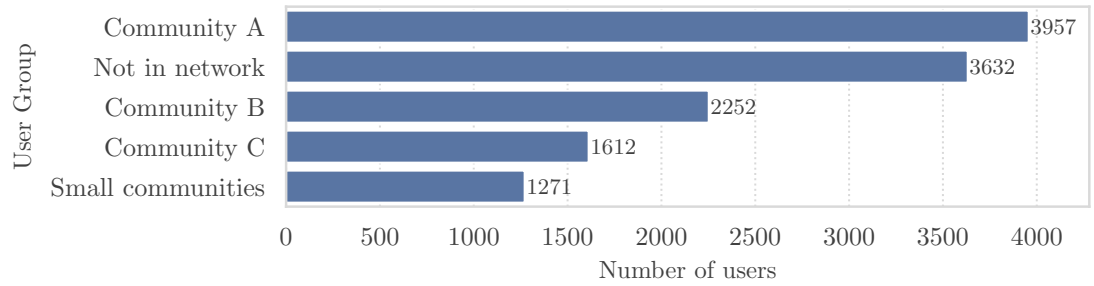

Figure 4.4: Distribution of the assignment of users to user groups based on the user communities detected by the `Louvain` algorithm in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. Each user is assigned to a community. Users not included in the graph used for community detection are placed in the "Not in network" group. Communities A, B and C are the three largest communities of users detected in the graph. All users belonging to the remaining small communities are grouped together.

These users are then used to recommend news articles, where a user's membership of a user group is considered the protected user attribute. Different recommender algorithms are trained and the recommendations they generate for users are evaluated using accuracy and beyond-accuracy recommendation metrics. This work evaluates the extent to which various recommendation algorithms provide users with different recommendations for news articles, with a particular focus on the differences between large user communities detected in a specific network constructed from user interaction data.

Figure 4.5: Visualization of the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. This network has a total of 9,092 nodes and 621,288 edges. The nodes are coloured according to the graph partition obtained by the `Louvain` algorithm. Orange nodes represent users belonging to the largest community with a size of 3,957 users ($\approx$ 43.52%), green nodes belong to the second-largest community with a size of 2,252 users ($\approx$ 24.77%), followed by the purple nodes which are part of a community with 1,612 users ($\approx$ 17.73%), and the remaining 1,271 users ($\approx$ 13.98%) from smaller communities are coloured grey. The graph is created with Gephi version 0.10.1 using the OpenOrd graph layout algorithm.

4.3 News Recommendation

This study investigates the extent to which various recommendation algorithms provide different recommendations for news articles depending on a user's group membership in a constructed network. The user group to which a user belongs to is derived from detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered by a minimum edge weight threshold of four. User group membership is considered a protected user attribute. Recommendations are evaluated for all users as a whole, with a particular focus on the fairness of recommendations received by users in large user communities of at least 750 users discovered in this network.

This news recommendation experiment uses a binary user-item interaction matrix that records which of the 16,495 news articles each of the 12,724 users has interacted with. Excluding duplicate article views, the dataset contains a total of 8,463,142 interactions with news articles, which corresponds to a density of about 0.04.

This chapter presents the results of the news recommendation experiment and discusses the findings. The overall performance of the recommendation algorithms is presented first. This is followed by the results of measuring the fairness of recommendations for user communities detected in a network constructed from user interaction data. Only the results for ranked recommendation lists of length $K = 10$ are reported in this chapter, while results for other list lengths (5, 20 and 50) can be found in the appendix B.

4.3.1 Overall Recommendation Performance

The overall results of news recommendations for all users are presented in this section. Recommendation performance is measured using metrics of accuracy (NDCG, precision and recall) and beyond-accuracy (coverage, diversity and novelty). Table 4.5 shows the mean evaluation scores at level 10 for all users collectively for the RS algorithms studied.

Overall, SLIM achieves the best performance in terms of accuracy-based metrics, while ITEM-KNN and BPR are the worst performers among the personalized RS algorithms. After SLIM, the ALS, VAE and ADV-VAE models have the next best performance for NDCG, precision and recall. The baseline algorithms are the worst for accuracy metrics.

The RAND model has the highest scores for coverage, diversity and novelty. The POP baseline has the second-best diversity, but, as expected, the lowest coverage and novelty. In terms of personalized recommendation systems, ITEM-KNN has the lowest coverage and novelty, but provides the most diverse recommendations. The BPR algorithm, which like ITEM-KNN performs the worst on accuracy-based metrics, has the highest coverage and novelty among personalized recommenders. Notably, the coverage of the BPR model far exceeds that of any other personalized recommender system. Although VAE and ADV-VAE achieve the next best coverage after BPR, these models score the lowest overall in terms of diversity. The algorithms ALS and SLIM show mediocre performance for the coverage metric, BPR, ALS and SLIM for the diversity metric, and ALS, SLIM, VAE and ADV-VAE for the novelty metric.

Table 4.5: Overall results of accuracy and beyond-accuracy metrics at level 10 for the RS algorithms studied. The values show the mean evaluation scores for all users collectively, averaged over five cross-validation test folds. Results are rounded to the fourth digit.

Model	NDCG	Precision	Recall	Coverage	Diversity	Novelty
RAND	0.0084	0.0085	0.0006	**0.7875**	**0.9446**	**0.9612**
POP	0.0868	0.0860	0.0067	0.0024	0.9360	0.7132
ITEM-KNN	0.1617	0.1547	0.0142	0.0335	0.8644	0.7508
BPR	0.1871	0.1729	0.0167	0.5352	0.7472	0.9075
ALS	0.2887	0.2710	0.0246	0.1467	0.7551	0.8171
SLIM	**0.3105**	**0.2900**	**0.0264**	0.1792	0.7482	0.8334
VAE	0.2877	0.2693	0.0248	0.2472	0.7016	0.8268
ADV-VAE	0.2770	0.2602	0.0237	0.2645	0.7169	0.8282

These results show that the performance differences between VAE and ADV-VAE are negligible. The values for the accuracy and beyond-accuracy metrics are mediocre compared to the other algorithms analysed, except for diversity, where the performance is the worst. Similar performance is shown by SLIM and ALS, with the former performing slightly better in terms of accuracy, and both also having good beyond-accuracy scores. The BPR model, like the ITEM-KNN model, has lower accuracy than the other models, with the former being slightly better, but BPR achieves the best novelty and by far the best coverage with good diversity, while ITEM-KNN has the worst novelty and by far the worst coverage, but still the best diversity.

4.3.2 User Community Recommendation Fairness

The focus of this section is to present and discuss the results of the studied news recommendation algorithms for large communities with at least 750 users detected in the undirected *Network of Users Who Vote on Postings* with a minimum edge weight of four. Recommendation results are summarized for all users in large communities and aggregated at the community level for all evaluation metrics used. In addition, the results of the user communities are also evaluated using the RecGap and the Compounding Factor recommendation fairness measures proposed by Melchiorre et al. [MRP+21]. RecGap quantifies the gap between the average evaluation metric scores of different user groups. The Compounding Factor measures the extent to which a recommendation algorithm reinforces existing imbalances in the training data with respect to the population distribution of user groups by comparing it with the distribution of metric scores, an aggregation of the evaluation scores per user group. This population distribution of user groups refers to the proportion of the three detected large communities in a network and is defined as $B = [0.5059, 0.2879, 0.2061]$. The metric scores distribution over user groups represents the share of scores for a particular evaluation metric that are distributed across user groups. If a particular user group is gaining metric scores disproportionately compared to the distribution of the user group population, then the respective RS model has amplified the population bias towards that group, which is reflected by the Compounding Factor. More precisely, the Compounding Factor for an evaluation metric

is the Kullback-Leibler divergence of the population distribution from the metric scores distribution. Both fairness measures are calculated for each of the recommendation evaluation metrics used, i.e., NDCG, precision, recall, coverage, diversity and novelty. Tables 4.6–4.11 show the recommendation results for each user community and evaluation metric. In the following, the results for the personalized recommender systems studied are discussed.

The tables show that, on average, the algorithms perform as well for users in communities as for all users. Scores for users in communities are on average higher for NDCG and precision, but lower for coverage compared to scores for all users. Statements about general recommendation performance also apply to the subset of users in communities.

When analysing the recommendation performance per user community, trends can be identified that appear to be related to the size of the community. In the undirected *Network of Users Who Vote on Postings*, whose edges are filtered by a minimum edge weight threshold of four, three communities with 7,821 users are detected. Of these, 3,957 users belong to the largest community A, another 2,252 users belong to the second-largest community B, followed by 1,612 users who are included in community C. The smaller the community size, the higher the NDCG, precision, recall and novelty, but the lower the coverage. The diversity of recommendations is lowest for community C and similar for communities A and B across all recommendation models, with slightly higher diversity for community B. However, the differences in recommendations between the communities also depend on the recommendation algorithm, and the difference between communities A and B is often small. Depending on the metric used for evaluation, there are differences in the utility of the recommendations for users in different communities.

The RecGap measure is used to quantify the average discrepancy in the mean evaluation scores of the user communities with a single number. In general, the RecGap for diversity is the highest compared to the other metrics. For accuracy-based metrics, the better the recommendation algorithm performs, the larger the RecGap tends to be. The same applies to the coverage metric, where the higher the score, the greater the gap in recommendation performance between user communities. Recommendation diversity and novelty show no clear trend for RecGap in relation to recommendation model performance. The SLIM model has the best performance for accuracy-based metrics, but also has the highest degree of unfairness for these, according to the RecGap. However, ITEM-KNN has the lowest RecGap for NDCG, followed by BPR and ADV-VAE. In terms of precision, ITEM-KNN has the smallest RecGap, followed by BPR, ADV-VAE and VAE. For recall, ADV-VAE, followed by ITEM-KNN and BPR, have the lowest RecGap, but in general, except for SLIM, the differences between the algorithms are small. The BPR model achieves by far the best item coverage for recommendations, but this is associated with the highest RecGap, while ITEM-KNN, followed by ALS and SLIM, have the worst coverage and also the lowest RecGap. While ITEM-KNN has the highest diversity, it has by far the lowest RecGap, while SLIM and VAE have the highest RecGap. It is similar with the novelty, where BPR has the best results but the lowest RecGap, while SLIM has the highest RecGap. For ADV-VAE, the RecGap is slightly smaller for accuracy-based

metrics, where the metric scores are marginally lower compared to the VAE model. The RecGap for ADV-VAE is slightly larger for coverage and lower for diversity and novelty, while the scores for these metrics are actually slightly higher when compared to VAE. In summary, SLIM has the largest RecGap for all evaluation metrics, except for coverage, where BPR has the highest RecGap.

The Compounding Factor provides a complementary view of the unfairness of recommendations by showing the effect of the model in amplifying imbalances in the data with respect to the distribution of user communities. For the accuracy-based metrics, the Compounding Factor is highest for ITEM-KNN, BPR and SLIM, while it is lowest for ADV-VAE, VAE and ALS, which have good accuracy. The situation is less clear for beyond-accuracy metrics. The Compounding Factor for coverage is highest for BPR, where the model also performs best, and lowest for ITEM-KNN. Notably, the Compounding Factor for coverage and BPR is by far the highest compared to all other models and metrics. Although ITEM-KNN has the highest diversity, the Compounding Factor is the lowest, and instead SLIM and VAE have the highest value. The Compounding Factor of the novelty metric is the lowest compared to the other evaluation metrics and the differences between the RS are negligible. It is noticeable that the Compounding Factor of ADV-VAE is slightly lower than that of VAE across all evaluation metrics. Both ALS and ADV-VAE show only a moderate amplification of data imbalances when compared to the other algorithms studied. The VAE model has a mediocre Compounding Factor, except for diversity, where it is among the highest. The BPR is among the recommender algorithms with the highest Compounding Factor for accuracy-based metrics and coverage, but the lowest for novelty. As with the RecGap measure, the Compounding Factor of SLIM is clearly among the highest, except for coverage.

This concludes the chapter on the results of the news recommendations. There is no single best recommender algorithm, as recommendation performance varies by evaluation metric. As shown, the recommendation utility for users in different communities can vary greatly, demonstrating once again the need to evaluate recommendation algorithms not only on the basis of all users, but also per user group. In this study, the algorithms that perform the worst on accuracy-based metrics perform best on some beyond-accuracy metrics. In general, the SLIM recommendation algorithm is the best in terms of the accuracy-based metrics used, while the recommendation fairness of the user communities is among the worst. The ALS and ADV-VAE are among the algorithms with good recommendation performance, with exceptions, while also having the lowest degree of unfairness with respect to different user communities across all evaluation metrics. The results for the fairness-aware ADV-VAE model, whose architecture combines a variational autoencoder with adversarial learning, mostly show performance differences that are worse for accuracy-based metrics and better for beyond-accuracy metrics, but the fairness of user communities is slightly better compared to the VAE model results, although the differences are smaller than expected.

Table 4.6: NDCG@10 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0099	0.0099 / 0.0095 / 0.0104	0.0006	0.0006	0.5070 / 0.2772 / 0.2158
POP	0.1010	0.1109 / 0.0834 / 0.1013	0.0184	0.0105	0.5556 / 0.2377 / 0.2068
ITEM-KNN	0.1865	0.1719 / 0.1734 / 0.2407	0.0458	0.0140	0.4664 / 0.2677 / 0.2659
BPR	0.2137	0.1913 / 0.2089 / 0.2753	0.0560	**0.0148**	0.4530 / 0.2815 / 0.2655
ALS	0.3301	0.3045 / 0.3214 / 0.4052	0.0671	0.0091	0.4667 / 0.2803 / 0.2529
SLIM	**0.3566**	0.3263 / 0.3424 / 0.4510	**0.0831**	0.0120	0.4629 / 0.2765 / 0.2606
VAE	0.3280	0.3030 / 0.3208 / 0.3998	0.0645	0.0086	0.4673 / 0.2816 / 0.2512
ADV-VAE	0.3162	0.2950 / 0.3069 / 0.3812	0.0575	0.0074	0.4720 / 0.2795 / 0.2485

Table 4.7: Precision@10 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0101	0.0104 / 0.0095 / 0.0105	0.0007	0.0012	0.5170 / 0.2699 / 0.2131
POP	0.0998	0.1075 / 0.0862 / 0.0998	0.0143	0.0064	0.5452 / 0.2486 / 0.2062
ITEM-KNN	0.1788	0.1673 / 0.1600 / 0.2330	0.0486	**0.0154**	0.4736 / 0.2578 / 0.2686
BPR	0.1973	0.1766 / 0.1923 / 0.2552	0.0524	0.0152	0.4528 / 0.2806 / 0.2666
ALS	0.3104	0.2866 / 0.3023 / 0.3804	0.0625	0.0090	0.4671 / 0.2804 / 0.2525
SLIM	**0.3340**	0.3062 / 0.3184 / 0.4239	**0.0785**	0.0123	0.4638 / 0.2746 / 0.2616
VAE	0.3079	0.2860 / 0.2983 / 0.3751	0.0594	0.0083	0.4700 / 0.2789 / 0.2511
ADV-VAE	0.2977	0.2793 / 0.2846 / 0.3611	0.0545	0.0078	0.4747 / 0.2752 / 0.2500

Table 4.8: Recall@10 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0007	0.0007 / 0.0006 / 0.0007	0.0001	0.0026	0.5313 / 0.2629 / 0.2058
POP	0.0067	0.0072 / 0.0060 / 0.0064	0.0008	0.0050	0.5460 / 0.2573 / 0.1967
ITEM-KNN	0.0138	0.0128 / 0.0131 / 0.0173	0.0030	**0.0108**	0.4690 / 0.2727 / 0.2582
BPR	0.0163	0.0149 / 0.0163 / 0.0197	0.0031	0.0083	0.4634 / 0.2883 / 0.2484
ALS	0.0241	0.0224 / 0.0246 / 0.0275	0.0034	0.0048	0.4703 / 0.2942 / 0.2355
SLIM	**0.0259**	0.0237 / 0.0261 / 0.0312	**0.0050**	0.0085	0.4619 / 0.2896 / 0.2484
VAE	0.0240	0.0222 / 0.0245 / 0.0274	0.0034	0.0049	0.4694 / 0.2951 / 0.2355
ADV-VAE	0.0231	0.0218 / 0.0232 / 0.0261	0.0028	0.0035	0.4781 / 0.2890 / 0.2329

Table 4.9: Coverage@10 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.6131**	0.3820 / 0.2389 / 0.1777	**0.1362**	**0.0676**	0.6471 / 0.2303 / 0.1226
POP	0.0024	0.0023 / 0.0019 / 0.0020	0.0002	0.0048	0.5464 / 0.2630 / 0.1906
ITEM-KNN	0.0248	0.0160 / 0.0146 / 0.0137	0.0016	0.0029	0.5351 / 0.2787 / 0.1863
BPR	0.4271	0.2790 / 0.1834 / 0.1285	0.1003	0.0650	0.6403 / 0.2396 / 0.1201
ALS	0.1303	0.0925 / 0.0783 / 0.0642	0.0189	0.0141	0.5666 / 0.2732 / 0.1602
SLIM	0.1449	0.1001 / 0.0843 / 0.0642	0.0239	0.0200	0.5744 / 0.2754 / 0.1502
VAE	0.2075	0.1311 / 0.1000 / 0.0803	0.0339	0.0269	0.5940 / 0.2577 / 0.1482
ADV-VAE	0.2214	0.1398 / 0.1063 / 0.0885	0.0342	0.0243	0.5914 / 0.2560 / 0.1526

Table 4.10: Diversity@10 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.9450**	0.9452 / 0.9449 / 0.9445	0.0005	0.0000	0.5061 / 0.2879 / 0.2060
POP	0.9331	0.9326 / 0.9327 / 0.9349	0.0015	0.0000	0.5057 / 0.2878 / 0.2065
ITEM-KNN	0.8706	0.8756 / 0.8860 / 0.8368	0.0328	0.0003	0.5089 / 0.2930 / 0.1981
BPR	0.7655	0.8085 / 0.8172 / 0.5880	0.1528	0.0115	0.5343 / 0.3074 / 0.1583
ALS	0.7678	0.8037 / 0.8137 / 0.6153	0.1322	0.0083	0.5296 / 0.3052 / 0.1652
SLIM	0.7515	0.8042 / 0.8159 / 0.5320	**0.1892**	**0.0190**	0.5414 / 0.3126 / 0.1459
VAE	0.7068	0.7525 / 0.7740 / 0.5008	0.1821	0.0190	0.5386 / 0.3153 / 0.1461
ADV-VAE	0.7194	0.7565 / 0.7714 / 0.5554	0.1439	0.0111	0.5321 / 0.3088 / 0.1592

Table 4.11: Novelty@10 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.9613**	0.9609 / 0.9617 / 0.9615	0.0005	0.0000	0.5058 / 0.2881 / 0.2062
POP	0.7135	0.7138 / 0.7131 / 0.7135	0.0005	0.0000	0.5061 / 0.2878 / 0.2061
ITEM-KNN	0.7510	0.7410 / 0.7466 / 0.7813	0.0269	**0.0003**	0.4993 / 0.2863 / 0.2144
BPR	0.9073	0.9023 / 0.9103 / 0.9154	0.0087	0.0000	0.5032 / 0.2889 / 0.2080
ALS	0.8192	0.8083 / 0.8222 / 0.8418	0.0223	0.0002	0.4992 / 0.2890 / 0.2118
SLIM	0.8314	0.8172 / 0.8348 / 0.8615	**0.0295**	0.0003	0.4973 / 0.2891 / 0.2136
VAE	0.8329	0.8213 / 0.8358 / 0.8575	0.0242	0.0002	0.4989 / 0.2889 / 0.2122
ADV-VAE	0.8344	0.8250 / 0.8347 / 0.8571	0.0214	0.0002	0.5002 / 0.2880 / 0.2117

CHAPTER 5

Conclusion

This chapter concludes the thesis with a summary of the research contribution, highlighting the key findings. The following discussion reflects on the research results. An outlook for future work is also provided, suggesting potential areas for further investigation and possible directions for further research.

5.1 Summary

This work addresses the problem of unfair treatment of different user groups in the recommendations they receive from recommendation algorithms. Recommender systems (RS) are software algorithms that suggest items to a user that are most likely to be of interest, based on the user's interaction history with items. Recommendation algorithms are used in a variety of domains, such as recommending music on music streaming platforms or products on e-commerce platforms. Collaborative filtering RS capture the patterns of users' interactions with items and are particularly sensitive to data imbalance, resulting in less relevant item recommendations for certain user groups, as shown in previous studies by Ekstrand et al. [ETA+24], Melchiorre et al. [MZS20, MRP+21] and Li et al. [LCF+21], for example. In such group fairness studies, users are grouped according to a sensitive user attribute, typically based on user traits or demographics, such as gender or age, and the equitable treatment of these groups is then examined. This thesis explores the fairness of recommendations for users in the news domain, more specifically in the recommendation of news articles, using a dataset from the Austrian online news platform "DER STANDARD". The aim of this work is to quantify the extent of variation in accuracy and diversity of recommendations between user groups and different collaborative filtering recommendation algorithms. Unlike related work, these user groups are large behavioural user communities with a minimum size of 750 users discovered in a user network modelled from user interaction data. To achieve this aim, user network construction, user community detection and news recommendation are necessary steps.

After preprocessing the dataset, data on users' views of news articles and users' community activities in discussion areas, in the form of postings, votes and user follow connections, are used to model the relationships between users, represented as nodes, in networks. The following weighted simple graphs with different user relations are constructed.

- Directed and undirected *Network of Users Who Vote on Postings*
- Directed and undirected *Network of Users Who Reply to Postings*
- Directed and undirected *Network Combining Users' Votes and Posting Replies*
- Directed *Network Combining Users' Votes, Posting Replies and Followers*
- Undirected *Network on Voting Behaviour of Users*
- Undirected *Network on Posting Behaviour of Users*
- Undirected *Network of Users by Similarity of News Views (Jaccard Index)*
- Undirected *Network of Users by Similarity of News Views (Overlap Coef.)*
- Undirected *Network of Users by Similarity of News Views (Salton's Index)*

Both the `Infomap` and `Louvain` algorithms are used to identify communities in all of these networks, and each of the community detection algorithms returns a graph partition. In addition, the effect of filtering the edges of a graph with different minimum edge weight thresholds before community detection on the clustering result is analysed.

For 44 out of 45 different network variants created, both community detection algorithms yielded partitions with at least one large user community with a size of at least 750 users. It turns out that the consensus between `Infomap` and `Louvain` in assigning users to communities is very low for most of the user networks constructed according to the Normalized Mutual Information (NMI) and the Rand Index. The distribution of NMI scores shows that 28 networks have an NMI below 0.1, 34 graphs below 0.3, 38 graphs below 0.4, and 42 graphs below 0.5. Similarly, the distribution of Rand Index values reveals that 21 networks have a Rand Index below 0.3, 35 graphs below 0.4, and 39 graphs below 0.6. The choice of the minimum edge weight threshold used to remove edges in the graph before community detection definitely affects the partitions obtained by `Infomap` and `Louvain` for some graphs. Instability in community detection algorithms is indicated by considerable differences in the consensus scores of graph partitions, as well as in the size and number of user communities found when filtering a network with different edge weight thresholds. The `Infomap` community detection algorithm finds only one large community with a size of more than 1,000 users in almost all the networks created. In contrast, the `Louvain` algorithm identifies three or four user communities with a size of more than 750 users in nearly all graphs. However, both the `Louvain` and `Infomap` algorithms detect three large communities in the undirected *Network of Users*

Who Vote on Postings, whose edges are filtered by a minimum edge weight of four. This graph also has the highest agreement between community detection algorithms with an NMI score of 0.73 and a Rand Index of 0.89, so this network is selected for categorizing users into groups. An evaluation of the quality of the partitions for the selected network shows that the `Louvain` partition has better performance (0.63 vs. 0.58) and marginally higher modularity (0.36 vs. 0.35) than the `Infomap` partition, although `Infomap` has slightly better coverage (0.72 vs. 0.70). The `Louvain` graph partition is therefore used for assigning users to groups, as it has a slightly higher partition quality than the `Infomap` partition. A characterization of the three large user communities with network analysis measures for the selected network also shows a high similarity between the two community detection algorithms.

Users are categorized into user groups based on the large user communities detected by the `Louvain` algorithm in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered by a minimum edge weight threshold of four. A total of 12,724 active users of the news platform of "DER STANDARD" are used for evaluating the fairness of news recommendations. However, the selected network consists of only 9,092 users, as some users do not vote on postings in the discussion area of news articles, or are removed from the network due to edge weight filtering. Of these, 3,957 users ($\approx 43.52\%$) belong to the largest community A, another 2,252 users ($\approx 24.77\%$) belong to the second-largest community B, followed by 1,612 users ($\approx 17.73\%$) who are in community C. The remaining 1,271 users ($\approx 13.98\%$) in the graph are detected as part of smaller clusters and are grouped together in a group named "Small communities". All other 3,632 users who are not included in the graph used for community detection for the selected network are categorized in the "Not in network" user group. A user's membership of a user group is considered the protected user attribute.

This work then evaluates the extent to which various recommendation algorithms provide users with different recommendations for news articles, with a particular focus on the differences between large user communities detected in a specific network constructed from user interaction data. Two baseline algorithms that provide either random or the most popular news articles are compared with traditional collaborative filtering algorithms based on nearest neighbours, matrix factorization and variational autoencoders. Additionally, a fairness-aware RS is compared, which aims to provide fairer recommendations by minimizing the sensitive information encoded in the learned latent user representations with respect to a protected user attribute that must be provided during model training. Recommendations to users are evaluated using accuracy (NDCG, precision, recall) and beyond-accuracy (coverage, diversity, novelty) metrics. Recommendation results are reported and discussed for all users overall and for each of the large user communities identified in the selected network modelled from user interaction data. The RecGap and Compounding Factor fairness measures proposed by Melchiorre et al. [MRP+21] are used to quantify both the difference between the mean recommendation results of large detected user communities and the extent to which a recommendation model reinforces imbalances in the training data with respect to the distribution of user communities.

In general, the results for personalized RS do not show a single best algorithm, as the recommendation performance varies depending on the evaluation metric. The SLIM recommendation method based on neighbourhood-learning achieves the best accuracy, and the ALS model based on matrix factorization, the variational autoencoder VAE and the ADV-VAE, which integrates adversarial learning in a variational autoencoder, have the next best accuracy. The ITEM-KNN, a nearest neighbour method that uses cosine similarity as a distance metric, and the BPR model, which is based on matrix factorization, are the worst personalized RS in terms of accuracy metrics. While ITEM-KNN provides recommendations with the lowest novelty value and by far the worst coverage, it also provides the most diverse recommendations. The BPR model achieves the highest novelty and covers by far the largest number of items in the catalogue. Both VAE and ADV-VAE score the lowest overall in terms of diversity. Negligible differences are observed in the recommendations of the fairness-aware ADV-VAE compared to the VAE model. The metric scores tend to be higher for NDCG and precision and lower for coverage for users in the three large communities compared to the scores of all users. Trends in recommendation performance related to the size of the three large discovered communities are apparent. The smaller the community size, the higher the NDCG, precision, recall and novelty, but the lower the coverage. The diversity of recommendations is lowest for the third-largest community and similar for the other two communities.

An assessment of fairness using RecGap shows that the better the RS for accuracy-based metrics and coverage, the higher the degree of unfairness between user communities. The RecGap for diversity is clearly the highest compared to the other metrics, with SLIM and VAE showing the largest gap between communities and ITEM-KNN the smallest. The BPR model has the lowest RecGap for the novelty metric and SLIM has the highest.

The Compounding Factor as a complementary view of recommendation fairness is highest for ITEM-KNN, BPR and SLIM and lowest for ADV-VAE, VAE and ALS for accuracy-based metrics. The Compounding Factor for coverage is highest for BPR and lowest for ITEM-KNN. In terms of diversity, the Compounding Factor is lowest for ITEM-KNN and highest for SLIM and VAE. The Compounding Factor of the novelty metric is the lowest compared to the other metrics and the differences between the RS are negligible.

In summary, the recommendations to users in large user communities detected in a network modelled from user interaction data can vary considerably depending on the collaborative filtering algorithm and evaluation metric used. In particular, algorithms that perform well on certain metrics have an increased risk of being unfair to certain groups of users. Both ALS and ADV-VAE generally show good recommendation performance and have the lowest degree of unfairness towards specific user communities according to the RecGap and Compounding Factor measures, but the differences in the fairness measures between RS may be small. The fairness-aware ADV-VAE exhibits only marginal differences in recommendation metrics and tends to perform slightly better on measures of user community fairness than the VAE model in this study.

Research Contribution

This thesis contributes to research by answering the following research questions.

- RQ1: To what extent does the selection of different types of user interactions for constructing networks influence key metrics such as community size and group centrality within the network?

  In this work, several user networks with different types of user relationships are constructed. The `Infomap` and `Louvain` algorithms, the former focusing on information flow and the latter on optimizing modularity in a graph, are used to detect user communities in a network. The effect of filtering edges of graphs with different minimum edge weights on the clustering result is also analysed. Section 4.2.1 discusses the consensus between community detection algorithms in assigning users to communities, and Section 4.2.2 examines the number and size of user communities detected in the graphs. Various measures of network analysis are used to characterize the user networks modelled and the communities discovered in the graphs. Section 4.2.3 discusses the effects of edge weight filtering on the selected network for grouping users and the differences between `Infomap` and `Louvain` for the large detected communities using network analysis measures.

  It turns out that the consensus between the two community detection algorithms in assigning users to communities is very low for most of the user networks constructed according to the Normalized Mutual Information and the Rand Index. The choice of the minimum edge weight threshold used to remove edges in the graph before community detection definitely affects the partitions obtained by `Infomap` and `Louvain` for some graphs. Instability in community detection algorithms is indicated by considerable differences in the consensus scores of graph partitions, as well as in the size and number of user communities found when filtering graphs with different edge weight thresholds.

- RQ2: What variations exist in the accuracy and diversity of recommended content across identified user communities when employing non-fairness-aware recommendation algorithms?

  This work evaluates the extent to which various collaborative filtering algorithms provide users with different recommendations for news articles, focusing on the differences between large user communities detected in a network of users. Several traditional algorithms based on nearest neighbours, matrix factorization and variational autoencoders are compared. Recommendations are evaluated using accuracy (NDCG, precision, recall) and beyond-accuracy (coverage, diversity, novelty) metrics. The RecGap and Compounding Factor are used to quantify the fairness of recommendations for users in large communities. Section 4.3.1 discusses the overall recommendation results, and Section 4.3.2 presents the results at the level of large communities with at least 750 users detected in a specific network.

Recommendations to users in large user communities detected in a network modelled from user interaction data can vary considerably depending on the collaborative filtering algorithm and evaluation metric used. Trends in recommendation performance related to the size of the three large discovered communities are apparent. The smaller the community size, the higher the NDCG, precision, recall and novelty, but the lower the coverage. The diversity of recommendations is lowest in the third-largest community and similar in the other two. RS that perform well on certain metrics have an increased risk of being unfair to certain groups of users.

- RQ3: To what extent does a fairness-aware recommendation algorithm improve the equitable distribution of accurate and diverse recommendations across user communities?

  In addition to conventional recommender algorithms, this work also uses a fairness-aware RS, which aims to provide fairer recommendations by minimizing the sensitive information encoded in the learned latent user representations with respect to a protected user attribute that must be provided during model training. This work uses the "Adversarial Variational Auto-Encoder with Multinomial Likelihood" fairness-aware recommender model proposed by Ganhör et al. [GPR+22], which integrates adversarial learning into a variational autoencoder. Results of the recommendation evaluation of this fairness-aware RS are also discussed in Section 4.3.1 for all users overall and in Section 4.3.2 for each of the large user communities.

  This fairness-aware RS exhibits only marginal differences in recommendation metrics compared to a variational autoencoder and tends to perform slightly better on measures of user community fairness than the VAE model in this study.

5.2 Discussion

Traditional collaborative filtering RS are designed for personalization by suggesting items that users are most likely to engage with, based on their past preferences and those of similar users. This type of recommender system aims to provide highly accurate recommendations by tailoring content to users' interests. For example, if a user prefers sports news, a traditional RS might show the user mainly sports-related articles. In the short term, some users may be satisfied with recommendations of conventional RS and spend more time on the platform, as the content is highly relevant and aligned with their specific interests. However, a drawback is that traditional RS tend to reinforce users' existing views and interests, as recommendations are based on their past behaviour on the platform. Excessive personalization can foster filter bubbles, where users' access to diverse content and opposing viewpoints is limited. For example, in news recommendations, this bias may prevent users from encountering different political ideologies. A lack of diversity in political news recommendations poses societal risks because people are less exposed to various points of view. They can become trapped in narrow world views, contributing to a more divided and intolerant society. Some individuals may even be drawn to extreme radical views and become hostile towards other groups. Apart from

societal risks, diversity in news recommendations helps, among other reasons, to prevent individual users from becoming bored with similar content by offering a variety of topics. Additionally, diverse recommendations can benefit platform providers by exposing users to a wider range of content, such as less popular or niche items that might otherwise remain unseen by users.

On the other hand, many fairness-aware recommender systems, such as the one used in this study, aim to reduce the influence of sensitive user attributes, like users' affiliation with interest groups or demographic characteristics, on recommendations. The concept of fair recommendations is to ensure that recommendation utility, as measured by metrics such as accuracy and diversity, is more equitably distributed across different groups. For instance, some user groups may receive highly accurate recommendations with low diversity, while others may be shown more varied content. Fairness-aware algorithms can reduce the influence of factors like user group membership on recommendations, ensuring that certain groups are not disadvantaged by receiving less diverse content. As a result, users are presented with more balanced and diverse news, including articles that differ from their existing views, helping them to break out of their filter bubbles. For example, a user who frequently reads articles supporting conservative political viewpoints would also be recommended articles from other parts of the political spectrum, such as socialist or progressive perspectives. However, in the short term, some users may feel less satisfied with recommendations from fairness-aware recommender systems, as they may seem less relevant or aligned with their views. This could potentially lead to some users spending less time on the platform, which could be a concern for both users and platform providers. But over time, as users are exposed to a wider range of viewpoints and become more informed, some users may appreciate the variety of recommended content, build trust and engage more with the platform. Ultimately, access to diverse news content can encourage critical thinking, foster a culture of tolerance and dialogue across different perspectives, help to reduce polarization in society and promote a better informed readership. One could argue that platform providers have an ethical obligation to consider fairness in their recommender systems, as this can help to reduce social divisions and promote a more open-minded society. While this may lead to lower user engagement and a potential short-term revenue impact due to less personalized recommendations, the long-term societal benefits could outweigh these immediate concerns. However, if a news platform builds a reputation for providing inclusive and diverse news, it can attract more customers looking for in-depth news coverage.

One of the challenges of using fairness-aware recommender systems is their greater complexity, as they require techniques that balance personalization with fairness. This study explores the integration of adversarial learning into a recommender model based on a variational autoencoder, with the aim of learning fairer user representations with respect to a sensitive user attribute. Various recommendation systems have been proposed in the literature that also take diversity and fairness into account, which makes it more difficult to decide on a system. Recommender algorithms can be evaluated using various metrics, but there are few measures in the literature that specifically assess the fairness of

recommendations, especially when considering more than two groups. Beyond selecting which fairness-aware recommender algorithm to use, another challenge for platform providers is deciding which sensitive attributes should be considered to ensure more diverse and fair representations. In this study, user communities, where users within each community exhibit similar behaviour, are detected in a user network modelled from user interaction data, and a user's membership in a community is the only attribute considered. These user clusters are identified in a weighted undirected network, where users are represented as nodes and an edge is established between two users if one has voted for the other's post in the discussion forum of the news platform, with the edge weight reflecting the frequency of their interactions. This is just one of many ways to analyse clusters of users based on similar behaviour. However, the approach has certain limitations, such as the fact that the cluster analysis in this study focuses only on registered users who vote or post in the platform's discussion forum. This raises the question of the extent to which users who do not participate in forum activities benefit from fairer recommendations. It is also likely that users belong to several interest groups rather than just one, as assumed for simplicity in this study. It is important to bear in mind that users' interests change over time and their profiles need to be updated regularly based on their current membership of user groups. To reduce the influence of group membership on recommendations, one approach is to group users based on similar interests, while another is to minimize the impact of demographic characteristics on recommendations. Deciding which sensitive attributes to use in fairness-aware RS is certainly not easy for platform providers, and it may even be worth considering allowing users to decide individually. Depending on the domain in which the recommender algorithms are used, additional specific challenges may arise. The news domain presents unique challenges that need to be considered when using recommender systems, such as the short lifespan of news articles and their frequent publication. Since collaborative filtering recommender systems suffer from the item cold-start problem, which refers to the difficulty of recommending new items, it may be necessary not only to use a collaborative filtering RS that provides diverse recommendations, but also to integrate a content-based RS to address this issue.

This study contributes to the recommender systems research community by emphasizing the importance of not only improving recommendation accuracy but also ensuring fairness and diversity, while highlighting some challenges involved in balancing these objectives. In conclusion, it is worth investing in the research of recommender algorithms that provide more diverse and balanced recommendations, as in the case of news recommendations, where they can promote exposure to a wider range of perspectives, reduce social polarization, and foster a more informed and inclusive society. However, the implementation of such systems requires careful consideration of the ethical, societal and technical challenges that arise in the respective domain of application.

5.3 Limitations and Future Work

While this thesis provides valuable insights into the challenge of using user interaction data to construct user networks, the varying results of two community detection algorithms

and their sensitivity to edge filtering with different minimum edge weight thresholds, as well as the user group fairness of recommendations provided by collaborative filtering algorithms in the news domain, it also has limitations. Like other research, this study faces certain constraints that may have influenced the results or the generalizability of the findings. This section outlines some of these limitations and offers an outlook for future work, suggesting possible directions for subsequent research.

A challenge of this study is the skew of user interaction data, such as clicks on news articles, submissions of postings or votes for postings. The distribution of user activity is very uneven, with a small group of users being highly active, while most others are much less engaged. This data skew complicates the analysis of user clusters, as they may mainly reflect the behaviour of these highly active users. The data imbalance can also introduce bias into recommendation algorithms, causing them to focus more on the preferences of the most active users, limiting their generalizability to the user population.

A limitation of this study is that the large user communities identified within a network based on user interaction data are only characterized using measures of network analysis. While these measures provide valuable insights into the network's structure, they do not capture the variety of user behaviour. Future research could address this by examining how these user communities relate to specific news reading behaviours or community activities. This could involve analysing the topics of news articles and postings that users interact with to understand the specific interests and behaviours within each community.

Users most frequently click on news articles that are placed at the top of the news platform's home page, and news articles often change throughout the day. Therefore, it is very likely that two users will read the same articles if they visit the news platform at the same time. One issue could be that clustering users by similarity of their clicked news articles or community activities could lead to user communities that do not necessarily share the same interests, but could be the result of interacting with the platform's content at similar times. The short lifespan and frequent publication of news articles is a challenge for cluster analysis of users in the news domain.

This thesis is limited by its focus on text-based news articles and user activity in the discussion areas below these articles. It does not take into account content in the dedicated discussion forum, such as blogs, columns, commentaries, debates and user-created forum pages, where user engagement might differ considerably. In particular, users are most active in the dedicated discussion forum, which is not included in this work. Furthermore, the study only considers interactions with text-based news articles, although the platform offers multimodal content, including videos and podcasts. As a result, the findings might not fully reflect how users interact with different content on the platform. Future work could extend the scope to different content types and also to forum pages.

A limitation of this study is that it focuses only on the interactions of registered users who have created a community identity. A considerable number of users who interact with the platform anonymously or without using community features are excluded. However, most users interact with the platform without creating an account. Since the user

communities detected are based on data of users' votes for postings, the study focuses on the behaviour of active users who engage in these community activities, while neglecting how non-registered users and less active registered users interact with the content. This results in an incomplete understanding of the overall user interaction on the platform.

This study identified several challenges, in particular the varying results of two community detection algorithms and their sensitivity to edge filtering with different minimum edge weight thresholds. The poor consensus observed between the algorithms suggests that they may be capturing different aspects of the network, leading to inconsistent results. Future research could focus on analysing the differences between these algorithms in more detail and investigating the reasons for their different results. In addition, it would be valuable to investigate methods to help select appropriate edge weight thresholds for edge filtering and to better track changes in node assignment across different thresholds. Furthermore, alternative methods of user clustering could be explored.

The study's reliance on implicit feedback in the form of clicks on news articles to provide recommendations is a limitation, as a recorded click does not guarantee that the user has actually read or engaged with the news article. For example, a user may open an article but then move the browser window in the background to do something else, so the click alone may not accurately reflect the user's interest or actual content consumption.

The fairness-aware recommender algorithm `ADV-VAE` used in this study aims to provide fairer recommendations by reducing the sensitive information encoded in the learned latent user representations with respect to a protected user attribute that must be provided during model training. In future work, the actual effect of bias mitigation of this approach could be measured and compared with a non-fairness-aware model, such as the `VAE` model. As noted by the authors of the `ADV-VAE` model, this study also found that a limitation of their model is the difficulty in selecting a model that both improves recommendation performance and reduces the bias associated with a protected user attribute during training. As the `ADV-VAE` model currently only supports bias mitigation of a single protected attribute, future work could adapt the model to support sensitive information reduction of multiple attributes. In addition, a comparison with alternative fairness-aware recommender algorithms could provide insights into which methods are more applicable for providing fairer recommendations.

This thesis quantifies the fairness of recommendations for different user groups based on the average metric values as well as the RecGap and Compounding Factor fairness measures per user group and evaluation metric. Future work could focus on assessing differences in recommendation performance with further fairness measures applicable to the number of user groups studied.

In this study, the diversity of recommendations is evaluated using the Shannon entropy of news sections. While this approach provides a useful measure of broad content diversity, it does not take into account the finer granularity within these sections. Future work could explore alternative methods of measuring diversity at a more granular level, such as analysing the variety of article topics within news sections, thereby providing a

more nuanced understanding of the diversity in recommendations. Additionally, future research could consider assessing diversity based on the user needs model for news[1], which categorizes content according to different reader motivations. This model is built around four main drivers. Users' need for knowledge is met by providing updates and factual content to keep them informed. The need for understanding is satisfied through explanations, context and perspectives to help users grasp the significance of events. Emotional engagement comes from stories that move, inspire or entertain. Finally, motivation to act is supported by offering practical advice, opportunities to connect with others and ways to participate in meaningful activities. By analysing how well recommendations align with these varied motivations, the system can ensure that it is not only diverse in news sections and topics but also effective in delivering what people want from the news.

The present study does not address the various unique aspects that news recommender systems must consider when providing recommendations on a news platform, such as the importance of recency, the short lifespan of news articles, user interests that change based on context, the persistent item cold-start problem, and the need to balance additional quality factors such as diversity, novelty and serendipity. Instead, this study focuses on evaluating the fairness of recommendations for users in large communities using conventional collaborative filtering algorithms and comparing them with a fairness-aware recommender algorithm designed to reduce the sensitive information encoded in the learned latent user representations. The study aims to quantify the fairness of recommendations for collaborative filtering techniques, which are known to be sensitive to imbalances in the training data. Future work could explore the integration of techniques that address the unique aspects of news recommender systems with various fairness-aware recommender algorithms to improve both relevance and fairness of recommendations. Content-based recommender systems, which use either manually created features like news metadata and article topics, or neural networks to learn representations from article text, can be integrated to overcome the limitations of collaborative filtering algorithms.

[1] https://smartocto.com/blog/explaining-user-needs

APPENDIX A

User Community Detection

Table A.1: This table describes each modelled network, whose edges may have been filtered using an edge weight threshold, with various graph measures. These measures include the number of nodes and edges, the graph density, the mean node degree, the average edge weight, the global clustering coefficient, the average distance and the graph diameter.

Network Name	Edge Weight Threshold	Number of Nodes	Number of Edges	Graph Density	Node Degree Mean	Edge Weight Mean	Global Clustering Coefficient	Average Distance	Diameter
Directed Network of Users Who Vote on Postings	1.00	12,535	6,105,250	0.0389	487.06	1.93	0.30	1.98	4
	2.00	11,660	1,914,247	0.0141	164.17	3.97	0.25	2.27	5
	3.00	10,252	950,809	0.0090	92.74	5.97	0.23	2.45	6
	4.00	8,873	573,512	0.0073	64.64	7.93	0.22	2.54	6
Undirected Network of Users Who Vote on Postings	1.00	12,535	5,535,637	0.0705	883.23	2.11	0.29	1.98	4
	2.00	11,745	1,932,500	0.0280	329.08	4.18	0.26	2.26	5
	3.00	10,424	1,003,722	0.0185	192.58	6.20	0.25	2.42	6
	4.00	9,092	621,288	0.0150	136.67	8.17	0.24	2.50	7
Directed Network of Users Who Reply to Postings	1.00	11,643	1,247,835	0.0092	107.17	1.39	0.17	2.43	5
	2.00	8,577	241,682	0.0033	28.18	2.99	0.13	2.91	7
	3.00	5,915	87,934	0.0025	14.87	4.72	0.11	3.14	7
Undirected Network of Users Who Reply to Postings	1.00	11,643	996,951	0.0147	171.25	1.73	0.17	2.43	5
	2.00	10,125	313,706	0.0061	61.97	3.33	0.13	2.79	6
	3.00	8,061	135,724	0.0042	33.67	5.08	0.11	3.01	7
	4.00	6,484	73,635	0.0035	22.71	6.83	0.10	3.16	8
Directed Network Combining Users' Votes and Posting Replies	1.15	12,540	6,961,250	0.0443	555.12	3.89	0.32	1.97	4
	2.29	12,241	2,957,605	0.0197	241.61	7.59	0.27	2.19	5
	3.44	12,067	2,066,396	0.0142	171.24	9.88	0.25	2.29	5
	4.59	11,967	1,726,637	0.0121	144.28	11.15	0.22	2.33	5
Undirected Network Combining Users' Votes and Posting Replies	1.15	12,540	6,104,458	0.0776	973.60	4.39	0.32	1.97	4
	2.30	12,243	2,672,075	0.0357	436.51	8.57	0.28	2.18	5
	3.44	12,068	1,825,574	0.0251	302.55	11.47	0.26	2.28	5
	4.59	11,967	1,489,253	0.0208	248.89	13.29	0.23	2.33	5
Directed Network Combining Users' Votes, Posting Replies and Followers	1.15	12,551	6,967,492	0.0442	555.13	5.83	0.32	1.97	4
	2.30	12,277	2,965,720	0.0197	241.57	12.15	0.27	2.19	5
	3.44	12,125	2,075,475	0.0141	171.17	16.38	0.25	2.29	5
	4.59	12,043	1,736,362	0.0120	144.18	18.90	0.22	2.34	5

Continued on the next page

Table A.1: Continued from the previous page

Network Name	Edge Weight Threshold	Number of Nodes	Number of Edges	Graph Density	Node Degree Mean	Edge Weight Mean	Global Clustering Coefficient	Average Distance	Diameter
Undirected Network on Voting Behaviour of Users	1.00	12,377	14,177,590	0.1851	2,290.96	6.33	0.49	1.83	4
	2.00	11,902	8,035,149	0.1135	1,350.22	10.40	0.46	1.94	4
	3.00	11,275	5,638,645	0.0887	1,000.20	13.97	0.44	1.99	4
	4.00	10,712	4,339,848	0.0756	810.28	17.25	0.43	2.03	4
Undirected Network on Posting Behaviour of Users	1.00	11,981	25,347,901	0.3532	4,231.35	6.31	0.67	1.65	3
	2.00	11,528	16,157,018	0.2432	2,803.09	9.34	0.63	1.77	4
	3.00	10,940	12,023,553	0.2009	2,198.09	11.86	0.62	1.83	4
	4.00	10,376	9,576,761	0.1779	1,845.94	14.12	0.60	1.86	4
Undirected Network of Users by Similarity of News Views (Jaccard Index)	0.05	10,872	12,130,220	0.2053	2,231.46	0.07	0.65	1.95	7
	0.06	10,075	8,086,799	0.1594	1,605.32	0.08	0.61	2.04	9
	0.07	8,646	4,043,390	0.1082	935.32	0.09	0.56	2.18	9
Undirected Network of Users by Similarity of News Views (Overlap Coef.)	0.18	12,724	12,131,307	0.1499	1,906.84	0.23	0.28	1.85	2
	0.20	12,724	8,086,885	0.0999	1,271.12	0.25	0.21	1.90	2
	0.24	12,724	4,043,406	0.0500	635.56	0.29	0.12	1.95	3
Undirected Network of Users by Similarity of News Views (Salton's Index)	0.11	11,846	12,130,209	0.1729	2,047.98	0.14	0.57	1.87	5
	0.12	11,210	8,086,799	0.1287	1,442.78	0.15	0.53	1.95	5
	0.14	9,976	4,043,395	0.0813	810.62	0.17	0.48	2.06	7

Table A.2: Distribution of the size of the communities detected in the constructed user networks. The table shows the number of detected communities categorized by size intervals for each modelled network, whose edges may have been filtered using an edge weight threshold. Each interval represents a specific range of community sizes.

Network Name	Edge Weight Threshold	Community Detection Algorithm	Distribution of Detected Communities by Size								
			(0, 2]	(2, 10]	(10, 50]	(50, 100]	(100, 250]	(250, 500]	(500, 750]	(750, 1000]	(1000, ∞)
Directed Network of Users Who Vote on Postings	1.00	Infomap	12	14	0	0	0	0	0	0	1
		Louvain	1	10	3	0	1	0	0	0	4
	2.00	Infomap	530	79	24	2	3	0	0	0	3
		Louvain	9	11	7	1	1	0	0	1	3
	3.00	Infomap	585	98	32	6	4	0	0	1	2
		Louvain	21	17	5	3	2	0	0	0	4
	4.00	Infomap	0	2	0	0	0	0	0	0	1
		Louvain	25	19	11	3	2	1	0	0	3
Undirected Network of Users Who Vote on Postings	1.00	Infomap	1	3	0	0	0	0	0	0	1
		Louvain	2	8	3	0	1	0	0	0	4
	2.00	Infomap	8	7	2	0	0	0	0	0	1
		Louvain	11	7	3	2	1	0	0	0	4
	3.00	Infomap	38	21	5	2	2	0	0	0	3
		Louvain	16	13	7	1	2	1	0	1	3
	4.00	Infomap	36	27	10	1	2	0	0	0	3
		Louvain	22	20	9	3	2	1	0	0	3
Directed Network of Users Who Reply to Postings	1.00	Infomap	15	7	3	1	0	0	0	0	1
		Louvain	7	9	7	1	1	1	0	1	4
	2.00	Infomap	116	88	10	1	2	1	0	0	1
		Louvain	52	62	30	5	5	2	1	1	3
	3.00	Infomap	67	107	45	2	5	1	0	0	1
		Louvain	71	90	58	8	3	0	2	2	0

Continued on the next page

Table A.2: Continued from the previous page

Network Name	Edge Weight Threshold	Community Detection Algorithm	Distribution of Detected Communities by Size								
			(0, 2]	(2, 10]	(10, 50]	(50, 100]	(100, 250]	(250, 500]	(500, 750]	(750, 1000]	(1000, ∞)
Undirected Network of Users Who Reply to Postings	1.00	Infomap	12	2	1	0	0	0	0	0	1
		Louvain	10	10	8	1	1	0	0	0	5
	2.00	Infomap	81	43	7	0	2	1	0	0	1
		Louvain	37	29	23	3	4	1	1	0	4
	3.00	Infomap	100	70	9	0	2	1	0	0	1
		Louvain	50	55	31	5	4	1	2	2	2
	4.00	Infomap	85	104	21	1	2	1	0	0	1
		Louvain	61	77	58	3	5	4	2	1	0
Directed Network Combining Users' Votes and Posting Replies	1.15	Infomap	4	1	0	0	0	0	0	0	1
		Louvain	4	9	5	1	0	1	0	0	4
	2.29	Infomap	21	3	1	2	0	0	0	0	1
		Louvain	6	9	3	0	1	1	1	0	4
	3.44	Infomap	18	4	2	2	0	0	0	0	1
		Louvain	4	11	5	0	2	0	1	0	4
	4.59	Infomap	14	3	1	2	0	0	0	0	1
		Louvain	10	10	5	1	2	1	0	1	4
Undirected Network Combining Users' Votes and Posting Replies	1.15	Infomap	0	0	0	0	0	0	0	0	1
		Louvain	4	12	2	1	1	0	0	0	4
	2.30	Infomap	7	0	1	0	0	0	0	0	1
		Louvain	9	10	4	0	1	1	1	0	4
	3.44	Infomap	10	0	1	0	0	0	0	0	1
		Louvain	7	11	6	0	2	2	0	0	4
	4.59	Infomap	8	1	2	0	0	0	0	0	1
		Louvain	10	13	5	1	2	1	1	0	4

Continued on the next page

Table A.2: Continued from the previous page

Network Name	Edge Weight Threshold	Community Detection Algorithm	Distribution of Detected Communities by Size								
			(0, 2]	(2, 10]	(10, 50]	(50, 100]	(100, 250]	(250, 500]	(500, 750]	(750, 1000]	(1000, ∞)
Directed Network Combining Users' Votes, Posting Replies and Followers	1.15	Infomap	143	97	13	3	2	0	0	0	1
		Louvain	263	353	65	1	5	1	0	0	3
	2.30	Infomap	158	132	25	0	3	1	0	0	1
		Louvain	255	352	76	1	4	0	0	0	3
	3.44	Infomap	135	162	29	1	4	1	0	0	1
		Louvain	257	388	64	2	4	0	0	0	3
	4.59	Infomap	133	159	35	0	3	1	0	0	1
		Louvain	254	351	69	1	5	0	0	0	4
Undirected Network on Voting Behaviour of Users	1.00	Infomap	0	0	0	0	0	0	0	0	1
		Louvain	0	0	1	0	0	0	0	0	3
	2.00	Infomap	0	0	0	0	0	0	0	0	1
		Louvain	0	0	1	0	0	0	0	0	4
	3.00	Infomap	0	1	0	0	0	0	0	0	1
		Louvain	0	0	1	0	0	0	0	0	4
	4.00	Infomap	2	1	1	0	0	0	0	0	1
		Louvain	1	1	1	0	0	0	0	0	4
Undirected Network on Posting Behaviour of Users	1.00	Infomap	0	0	0	0	0	0	0	0	1
		Louvain	0	0	0	0	0	0	0	0	4
	2.00	Infomap	0	0	0	0	0	0	0	0	1
		Louvain	0	0	0	0	0	0	0	0	4
	3.00	Infomap	0	0	0	0	0	0	0	0	1
		Louvain	0	0	0	0	0	0	0	0	4
	4.00	Infomap	0	0	0	0	0	0	0	0	1
		Louvain	0	0	0	0	0	0	0	0	4

Continued on the next page

Table A.2: Continued from the previous page

Network Name	Edge Weight Threshold	Community Detection Algorithm	Distribution of Detected Communities by Size								
			(0, 2]	(2, 10]	(10, 50]	(50, 100]	(100, 250]	(250, 500]	(500, 750]	(750, 1000]	(1000, ∞)
Undirected Network of Users by Similarity of News Views (Jaccard Index)	0.05	Infomap	9	15	3	0	0	0	0	0	1
		Louvain	6	6	2	0	0	0	0	0	3
	0.06	Infomap	10	15	4	0	0	0	0	0	1
		Louvain	3	7	2	0	0	0	0	0	4
	0.07	Infomap	7	12	6	1	0	1	0	0	1
		Louvain	2	7	3	0	0	0	0	0	4
Undirected Network of Users by Similarity of News Views (Overlap Coef.)	0.18	Infomap	0	0	0	0	0	0	0	0	1
		Louvain	0	0	0	0	0	0	0	0	3
	0.20	Infomap	0	0	0	0	0	0	0	0	1
		Louvain	0	0	0	0	0	0	0	0	3
	0.24	Infomap	0	0	0	0	0	0	0	0	1
		Louvain	0	0	0	0	0	0	0	0	4
Undirected Network of Users by Similarity of News Views (Salton's Index)	0.11	Infomap	3	4	2	0	0	0	0	0	1
		Louvain	1	1	2	0	0	0	0	0	3
	0.12	Infomap	7	4	1	0	0	0	0	0	1
		Louvain	2	2	1	0	0	0	0	0	3
	0.14	Infomap	3	9	2	0	0	0	0	0	1
		Louvain	3	3	1	0	0	0	0	0	4

Table A.3: This table shows, for each constructed user network, the Normalized Mutual Information (NMI) and Rand Index consensus scores for the graph partitions resulting from the original partitions obtained by the `Infomap` and `Louvain` algorithms by combining all users in communities with less than 750 users into a single group. All scores are rounded to the second digit. Highest values are shown in bold.

Network Name	Edge Weight Threshold	NMI	Rand Index
Undirected Network of Users Who Vote on Postings	4.00	**0.69**	**0.88**
Undirected Network of Users Who Vote on Postings	3.00	0.62	0.83
Directed Network of Users Who Vote on Postings	3.00	0.47	0.79
Directed Network of Users Who Vote on Postings	2.00	0.41	0.76
Directed Network Combining Users' Votes, Posting Replies and Followers	3.44	0.15	0.52
Directed Network Combining Users' Votes, Posting Replies and Followers	2.30	0.14	0.49
Directed Network Combining Users' Votes, Posting Replies and Followers	4.59	0.12	0.42
Undirected Network of Users Who Reply to Postings	3.00	0.11	0.44
Directed Network of Users Who Reply to Postings	2.00	0.11	0.44
Undirected Network of Users Who Reply to Postings	2.00	0.11	0.36
Directed Network Combining Users' Votes, Posting Replies and Followers	1.15	0.10	0.36
Undirected Network of Users by Similarity of News Views (Jaccard Index)	0.07	0.07	0.32
Undirected Network of Users by Similarity of News Views (Jaccard Index)	0.05	0.07	0.36
Undirected Network of Users by Similarity of News Views (Jaccard Index)	0.06	0.06	0.28
Undirected Network of Users by Similarity of News Views (Salton's Index)	0.12	0.05	0.35
Undirected Network of Users by Similarity of News Views (Salton's Index)	0.11	0.05	0.35
Undirected Network of Users by Similarity of News Views (Salton's Index)	0.14	0.04	0.28
Undirected Network of Users Who Reply to Postings	4.00	0.04	0.53
Directed Network of Users Who Reply to Postings	3.00	0.04	0.51
Undirected Network on Voting Behaviour of Users	4.00	0.04	0.31
Undirected Network of Users Who Vote on Postings	2.00	0.03	0.28
Directed Network Combining Users' Votes and Posting Replies	4.59	0.02	0.24
Directed Network Combining Users' Votes and Posting Replies	2.29	0.02	0.24
Directed Network Combining Users' Votes and Posting Replies	3.44	0.02	0.27
Directed Network of Users Who Reply to Postings	1.00	0.01	0.24
Undirected Network Combining Users' Votes and Posting Replies	4.59	0.01	0.24
Undirected Network Combining Users' Votes and Posting Replies	3.44	0.01	0.24
Undirected Network Combining Users' Votes and Posting Replies	2.30	0.01	0.23
Undirected Network of Users Who Vote on Postings	1.00	0.01	0.27
Undirected Network of Users Who Reply to Postings	1.00	0.01	0.23
Directed Network of Users Who Vote on Postings	1.00	0.00	0.27
Directed Network Combining Users' Votes and Posting Replies	1.15	0.00	0.25
Directed Network of Users Who Vote on Postings	4.00	0.00	0.30
Undirected Network on Voting Behaviour of Users	3.00	0.00	0.31
Undirected Network of Users by Similarity of News Views (Overlap Coef.)	0.18	0.00	0.35
Undirected Network of Users by Similarity of News Views (Overlap Coef.)	0.20	0.00	0.35
Undirected Network on Posting Behaviour of Users	4.00	0.00	0.27
Undirected Network Combining Users' Votes and Posting Replies	1.15	0.00	0.25
Undirected Network on Posting Behaviour of Users	3.00	0.00	0.27
Undirected Network on Posting Behaviour of Users	2.00	0.00	0.27
Undirected Network on Posting Behaviour of Users	1.00	0.00	0.28
Undirected Network on Voting Behaviour of Users	2.00	0.00	0.33
Undirected Network on Voting Behaviour of Users	1.00	0.00	0.39
Undirected Network of Users by Similarity of News Views (Overlap Coef.)	0.24	0.00	0.26

Table A.4: This table shows, for each constructed user network, the scores of the partition quality functions coverage, modularity and performance for the graph partitions resulting from the original partitions obtained by the `Louvain` and `Infomap` algorithms by keeping only communities that satisfy a minimum size of 750 users. This quantifies the quality of the graph partitions, consisting only of the large user communities. All scores are rounded to the second digit.

Network Name	**Edge Weight Threshold**	**Community Detection Algorithm**	**Coverage**	**Modularity**	**Performance**
Directed Network of Users Who Vote on Postings	1.00	`Infomap`	1.00	0.00	0.04
		`Louvain`	0.45	0.23	0.72
	2.00	`Infomap`	0.70	0.29	0.50
		`Louvain`	0.59	0.30	0.67
	3.00	`Infomap`	0.74	0.36	0.57
		`Louvain`	0.61	0.34	0.71
	4.00	`Infomap`	1.00	0.00	0.01
		`Louvain`	0.72	0.38	0.62
Undirected Network of Users Who Vote on Postings	1.00	`Infomap`	1.00	0.00	0.07
		`Louvain`	0.44	0.23	0.72
	2.00	`Infomap`	1.00	0.00	0.03
		`Louvain`	0.54	0.29	0.71
	3.00	`Infomap`	0.70	0.31	0.55
		`Louvain`	0.63	0.33	0.68
	4.00	`Infomap`	0.72	0.35	0.58
		`Louvain`	0.70	0.36	0.63
Directed Network of Users Who Reply to Postings	1.00	`Infomap`	1.00	0.00	0.01
		`Louvain`	0.39	0.18	0.75
	2.00	`Infomap`	1.00	0.00	0.00
		`Louvain`	0.53	0.33	0.74
	3.00	`Infomap`	1.00	0.00	0.01
		`Louvain`	0.82	0.33	0.50

Continued on the next page

Table A.4: Continued from the previous page

Network Name	Edge Weight Threshold	Community Detection Algorithm	Coverage	Modularity	Performance
Undirected Network of Users Who Reply to Postings	1.00	Infomap	1.00	0.00	0.01
		Louvain	0.37	0.18	0.76
	2.00	Infomap	1.00	0.00	0.01
		Louvain	0.46	0.25	0.74
	3.00	Infomap	1.00	0.00	0.01
		Louvain	0.55	0.36	0.74
	4.00	Infomap	1.00	0.00	0.01
		Louvain	1.00	0.00	0.01
Directed Network Combining Users' Votes and Posting Replies	1.15	Infomap	1.00	0.00	0.04
		Louvain	0.38	0.18	0.72
	2.29	Infomap	1.00	0.00	0.02
		Louvain	0.42	0.21	0.73
	3.44	Infomap	1.00	0.00	0.01
		Louvain	0.43	0.21	0.71
	4.59	Infomap	1.00	0.00	0.01
		Louvain	0.38	0.20	0.75
Undirected Network Combining Users' Votes and Posting Replies	1.15	Infomap	1.00	0.00	0.08
		Louvain	0.37	0.17	0.71
	2.30	Infomap	1.00	0.00	0.04
		Louvain	0.41	0.20	0.72
	3.44	Infomap	1.00	0.00	0.03
		Louvain	0.42	0.20	0.72
	4.59	Infomap	1.00	0.00	0.02
		Louvain	0.43	0.21	0.71

Continued on the next page

Table A.4: Continued from the previous page

Network Name	Edge Weight Threshold	Community Detection Algorithm	Coverage	Modularity	Performance
Directed Network Combining Users' Votes, Posting Replies and Followers	1.15	Infomap	1.00	0.00	0.05
		Louvain	0.47	0.20	0.65
	2.30	Infomap	1.00	0.00	0.02
		Louvain	0.60	0.29	0.48
	3.44	Infomap	1.00	0.00	0.02
		Louvain	0.61	0.31	0.48
	4.59	Infomap	1.00	0.00	0.01
		Louvain	0.43	0.32	0.72
Undirected Network on Voting Behaviour of Users	1.00	Infomap	1.00	0.00	0.19
		Louvain	0.47	0.24	0.59
	2.00	Infomap	1.00	0.00	0.11
		Louvain	0.45	0.24	0.66
	3.00	Infomap	1.00	0.00	0.09
		Louvain	0.47	0.25	0.68
	4.00	Infomap	1.00	0.00	0.08
		Louvain	0.48	0.26	0.69
Undirected Network on Posting Behaviour of Users	1.00	Infomap	1.00	0.00	0.35
		Louvain	0.30	0.09	0.58
	2.00	Infomap	1.00	0.00	0.24
		Louvain	0.32	0.09	0.64
	3.00	Infomap	1.00	0.00	0.20
		Louvain	0.32	0.10	0.66
	4.00	Infomap	1.00	0.00	0.18
		Louvain	0.33	0.10	0.67

Continued on the next page

Table A.4: Continued from the previous page

Network Name	Edge Weight Threshold	Community Detection Algorithm	Coverage	Modularity	Performance
Undirected Network of Users by Similarity of News Views (Jaccard Index)	0.05	Infomap	1.00	0.00	0.21
		Louvain	0.48	0.16	0.65
	0.06	Infomap	1.00	0.00	0.17
		Louvain	0.43	0.18	0.72
	0.07	Infomap	1.00	0.00	0.12
		Louvain	0.48	0.22	0.74
Undirected Network of Users by Similarity of News Views (Overlap Coef.)	0.18	Infomap	1.00	0.00	0.15
		Louvain	0.50	0.16	0.65
	0.20	Infomap	1.00	0.00	0.10
		Louvain	0.52	0.19	0.65
	0.24	Infomap	1.00	0.00	0.05
		Louvain	0.49	0.22	0.74
Undirected Network of Users by Similarity of News Views (Salton's Index)	0.11	Infomap	1.00	0.00	0.17
		Louvain	0.49	0.17	0.66
	0.12	Infomap	1.00	0.00	0.13
		Louvain	0.52	0.20	0.66
	0.14	Infomap	1.00	0.00	0.08
		Louvain	0.49	0.23	0.73

APPENDIX B

News Recommendation

Table B.1: Investigated hyperparameters and fixed parameters of the RS algorithms ITEM-KNN, BPR, ALS, SLIM, VAE, and ADV-VAE. The table shows for each model the explored values of the hyperparameters, the best values found in a grid search, and the fixed parameters, along with descriptions of the parameters. The best hyperparameter values are determined based on the best mean recommendation evaluation scores across all validation sets and when for the protected user attribute the group is used to which a user is assigned based on the detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four.

Model	Type	Parameter	Explored Values	Best Value	Description
ITEM-KNN	Search	k	{5, 10, 15, 20, 25, 50, 75, 100}	5	Number of neighbours to include when calculating the item-item similarity matrix
BPR	Search	factors	{50, 100, 250, 500}	250	Dimensionality of latent user and item factors to compute
		iterations	{100, 500, 1000}	500	Number of training epochs to use when fitting the data
		learning_rate	{1e-3, 1e-4}	1e-3	Learning rate to apply for SGD updates during training
		regularization	{1e-4, 1e-5}	1e-5	Regularization factor to use
ALS	Search	alpha	{1, 2, 4, 6}	2	Weight to give to positive examples
		factors	{100, 125, 150}	125	Dimensionality of latent user and item factors to compute
		iterations	{15, 50, 100}	100	Number of ALS iterations to use when fitting data
		regularization	{1e-2, 1e-3}	1e-3	Regularization factor to use
SLIM	Fixed	max_iter	50	—	Maximum number of elastic net iterations to perform
	Search	alpha	{2e-1, 1e-1, 5e-2, 1e-2, 5e-3, 1e-3, 5e-4, 1e-4}	1e-2	Controls the overall regularization strength in elastic net
		l1_ratio	{2e-1, 1.5e-1, 1e-1, 5e-2, 2e-2}	5e-2	Elastic net mixing parameter which controls the balance between ℓ_1 and ℓ_2 regularization
VAE	Fixed	beta_cap	1	—	Maximum value of beta in beta annealing process
		beta_patience	5	—	Number of steps without improvement in the validation score, after which beta annealing should be stopped
		latent_dropout_rate	0	—	Dropout to apply on the latent space (before the decoder)
		n_epochs	150	—	Number of training epochs
		normalize_gradients	false	—	Whether to clip the gradients of the model's parameters
		normalize_inputs	true	—	Whether the input to the encoder should be normalized
		opt_learning_rate	5e-4	—	Adam learning rate for optimizing VAE model parameters
		opt_weight_decay	1e-4	—	Adam weight decay (ℓ_2 regularization) for optimizing VAE model parameters

Continued on the next page

Table B.1: Continued from the previous page

Model	Type	Parameter	Explored Values	Best Value	Description
	Search	beta_steps	{20, 40, 60}	20	Maximum number of beta update steps in beta annealing
		input_dropout_rate	{1e-1, 5e-1}	1e-1	Dropout to apply on the input (before encoder)
		p_dims	{[200], [500], [800], [1000], [200, 1000]}	[1000]	Defines the linear layer structure of the decoder network. The first element in the list is the dimensionality of the latent space. The decoder's output dimensionality (i.e., the number of items) is automatically appended to the list as the last element. The linear layer structure of the encoder is the reverse of that of the decoder.
ADV-VAE	Fixed	adv_earlystop	true	—	Whether to perform early stopping on the balanced accuracy of the adversarial network in predicting the actual protected attribute of users
		adv_earlystop_min_epochs	50	—	Minimum number of epochs before early stopping on the adversarial balanced accuracy
		adv_in_use	true	—	Whether adversarial training should be performed
		adv_latent_dropout	1e-1	—	Dropout to apply before the adversarial network
		adv_loss_weight	1	—	Scaling factor for the adversarial loss when added to the objective loss of the VAE
		adv_opt_learning_rate	1e-4	—	Adam learning rate for optimizing adversarial network model parameters
		adv_opt_weight_decay	1e-4	—	Adam weight decay (ℓ_2 penalty) for optimizing adversarial network model parameters
		adv_warmup	true	—	Whether to train the VAE and the adversarial network separately for a certain number of epochs before training them together
		adv_warmup_n_epochs	15	—	Number of epochs to train the VAE and the adversarial network separately before training them together
		beta_cap	1	—	Maximum value of beta in beta annealing process
		beta_patience	5	—	Number of steps without improvement in the validation score, after which beta annealing should be stopped

Continued on the next page

Table B.1: Continued from the previous page

Model	Type	Parameter	Explored Values	Best Value	Description
		beta_steps	20	—	Maximum number of beta update steps in beta annealing
		input_dropout_rate	1e-1	—	Dropout to apply on the input (before encoder)
		latent_dropout_rate	0	—	Dropout to apply on the latent space (before the decoder)
		n_epochs	150	—	Number of training epochs
		normalize_gradients	false	—	Whether to clip the gradients of the model's parameters
		normalize_inputs	true	—	Whether the input to the encoder should be normalized
		opt_learning_rate	5e-4	—	Adam learning rate for optimizing VAE model parameters
		opt_weight_decay	1e-4	—	Adam weight decay (ℓ_2 regularization) for optimizing VAE model parameters
		p_dims	[1000]	—	Defines the linear layer structure of the decoder network. The first element in the list is the dimensionality of the latent space. The decoder's output dimensionality (i.e., the number of items) is automatically appended to the list as the last element. The linear layer structure of the encoder is the reverse of that of the decoder.
	Search	adv_dims	{[ ], [10]}	[ ]	Defines the linear layer structure of an adversarial network. The adversarial network's output dimensionality (i.e., the number of user groups) is automatically appended to the list as the last element.
		adv_grad_scaling	{100, 200, 400, 600, 800}	800	The gradient reversal scaling factor is used in the gradient reversal layer between the latent space and the adversarial network. It negatively scales the gradients by this factor during the backward pass when optimizing the model. This factor controls the removal of sensitive information about the protected attribute of users from the learned latent representations.
		adv_n_adv	{1, 3}	1	Number of adversarial networks to train in parallel. The loss and the balanced accuracy of the adversaries is averaged.

Table B.2: Overall results of accuracy and beyond-accuracy metrics at level 5 for the RS algorithms studied. The values show the mean evaluation scores for all users collectively, averaged over five cross-validation test folds. Results are rounded to the fourth digit.

Model	NDCG	Precision	Recall	Coverage	Diversity	Novelty
RAND	0.0083	0.0085	0.0003	**0.5381**	**0.9625**	**0.9612**
POP	0.0881	0.0875	0.0034	0.0015	0.9372	0.7102
ITEM-KNN	0.1711	0.1654	0.0078	0.0183	0.8640	0.7433
BPR	0.2076	0.1972	0.0096	0.3593	0.7513	0.9007
ALS	0.3150	0.3028	0.0141	0.1057	0.7487	0.8119
SLIM	**0.3400**	**0.3250**	**0.0151**	0.1246	0.7390	0.8304
VAE	0.3157	0.3036	0.0144	0.1749	0.6854	0.8259
ADV-VAE	0.3019	0.2903	0.0136	0.1854	0.7018	0.8277

Table B.3: Overall results of accuracy and beyond-accuracy metrics at level 20 for the RS algorithms studied. The values show the mean evaluation scores for all users collectively, averaged over five cross-validation test folds. Results are rounded to the fourth digit.

Model	NDCG	Precision	Recall	Coverage	Diversity	Novelty
RAND	0.0083	0.0083	0.0012	**0.9545**	**0.9254**	**0.9612**
POP	0.0836	0.0816	0.0128	0.0041	0.9019	0.7196
ITEM-KNN	0.1533	0.1455	0.0261	0.0600	0.8388	0.7615
BPR	0.1661	0.1494	0.0282	0.7261	0.7193	0.9146
ALS	0.2597	0.2376	0.0421	0.1985	0.7359	0.8234
SLIM	**0.2777**	**0.2524**	**0.0447**	0.2504	0.7295	0.8370
VAE	0.2592	0.2369	0.0423	0.3353	0.6905	0.8281
ADV-VAE	0.2489	0.2277	0.0405	0.3597	0.7063	0.8290

Table B.4: Overall results of accuracy and beyond-accuracy metrics at level 50 for the RS algorithms studied. The values show the mean evaluation scores for all users collectively, averaged over five cross-validation test folds. Results are rounded to the fourth digit.

Model	NDCG	Precision	Recall	Coverage	Diversity	Novelty
RAND	0.0083	0.0082	0.0029	**0.9995**	**0.8952**	**0.9613**
POP	0.0650	0.0550	0.0240	0.0092	0.8385	0.7644
ITEM-KNN	0.1403	0.1273	0.0548	0.1298	0.7942	0.7819
BPR	0.1411	0.1196	0.0551	0.9200	0.6809	0.9240
ALS	0.2218	0.1917	0.0826	0.2932	0.7102	0.8336
SLIM	**0.2344**	**0.2008**	**0.0858**	0.3748	0.7054	0.8436
VAE	0.2210	0.1906	0.0820	0.4733	0.6751	0.8312
ADV-VAE	0.2134	0.1849	0.0794	0.5088	0.6913	0.8314

Table B.5: NDCG@5 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0099	0.0097 / 0.0099 / 0.0103	0.0004	0.0004	0.4970 / 0.2882 / 0.2147
POP	0.1024	0.1153 / 0.0784 / 0.1045	0.0246	**0.0190**	0.5694 / 0.2204 / 0.2102
ITEM-KNN	0.1973	0.1786 / 0.1897 / 0.2537	0.0501	0.0141	0.4580 / 0.2769 / 0.2651
BPR	0.2371	0.2120 / 0.2329 / 0.3045	0.0617	0.0146	0.4524 / 0.2828 / 0.2647
ALS	0.3590	0.3310 / 0.3498 / 0.4407	0.0731	0.0092	0.4665 / 0.2805 / 0.2530
SLIM	**0.3894**	0.3560 / 0.3760 / 0.4900	**0.0893**	0.0116	0.4626 / 0.2780 / 0.2594
VAE	0.3580	0.3278 / 0.3540 / 0.4379	0.0733	0.0092	0.4633 / 0.2847 / 0.2521
ADV-VAE	0.3434	0.3189 / 0.3375 / 0.4116	0.0618	0.0072	0.4700 / 0.2830 / 0.2471

Table B.6: Precision@5 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0103	0.0103 / 0.0100 / 0.0104	0.0003	0.0001	0.5087 / 0.2818 / 0.2095
POP	0.1009	0.1117 / 0.0794 / 0.1042	0.0215	0.0151	0.5603 / 0.2267 / 0.2130
ITEM-KNN	0.1912	0.1753 / 0.1768 / 0.2504	0.0501	**0.0158**	0.4638 / 0.2663 / 0.2699
BPR	0.2248	0.2007 / 0.2207 / 0.2897	0.0594	0.0150	0.4516 / 0.2827 / 0.2656
ALS	0.3451	0.3186 / 0.3365 / 0.4220	0.0689	0.0088	0.4672 / 0.2808 / 0.2520
SLIM	**0.3729**	0.3420 / 0.3574 / 0.4704	**0.0855**	0.0117	0.4641 / 0.2760 / 0.2600
VAE	0.3444	0.3162 / 0.3385 / 0.4221	0.0706	0.0092	0.4645 / 0.2829 / 0.2526
ADV-VAE	0.3303	0.3086 / 0.3199 / 0.3982	0.0597	0.0074	0.4727 / 0.2789 / 0.2484

Table B.7: Recall@5 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0003	0.0003 / 0.0003 / 0.0004	0.0000	0.0004	0.5066 / 0.2792 / 0.2142
POP	0.0034	0.0038 / 0.0028 / 0.0033	0.0007	0.0116	0.5633 / 0.2381 / 0.1986
ITEM-KNN	0.0076	0.0069 / 0.0073 / 0.0097	0.0018	**0.0129**	0.4616 / 0.2757 / 0.2627
BPR	0.0094	0.0086 / 0.0094 / 0.0113	0.0018	0.0084	0.4627 / 0.2888 / 0.2485
ALS	0.0137	0.0127 / 0.0139 / 0.0157	0.0020	0.0049	0.4700 / 0.2936 / 0.2364
SLIM	**0.0147**	0.0134 / 0.0149 / 0.0178	**0.0029**	0.0088	0.4601 / 0.2914 / 0.2485
VAE	0.0138	0.0128 / 0.0143 / 0.0160	0.0021	0.0057	0.4660 / 0.2964 / 0.2376
ADV-VAE	0.0131	0.0123 / 0.0132 / 0.0148	0.0017	0.0037	0.4763 / 0.2902 / 0.2336

Table B.8: Coverage@5 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.3783**	0.2141 / 0.1279 / 0.0932	**0.0806**	**0.0799**	0.6590 / 0.2241 / 0.1169
POP	0.0015	0.0014 / 0.0012 / 0.0012	0.0001	0.0030	0.5382 / 0.2714 / 0.1904
ITEM-KNN	0.0137	0.0087 / 0.0076 / 0.0077	0.0008	0.0033	0.5396 / 0.2666 / 0.1939
BPR	0.2710	0.1651 / 0.1034 / 0.0735	0.0611	0.0729	0.6503 / 0.2318 / 0.1179
ALS	0.0921	0.0630 / 0.0520 / 0.0413	0.0145	0.0188	0.5759 / 0.2702 / 0.1538
SLIM	0.0994	0.0669 / 0.0541 / 0.0398	0.0181	0.0274	0.5875 / 0.2701 / 0.1424
VAE	0.1439	0.0883 / 0.0644 / 0.0504	0.0253	0.0353	0.6069 / 0.2520 / 0.1411
ADV-VAE	0.1512	0.0924 / 0.0678 / 0.0556	0.0245	0.0302	0.6014 / 0.2511 / 0.1475

Table B.9: Diversity@5 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.9624**	0.9622 / 0.9620 / 0.9636	0.0011	0.0000	0.5058 / 0.2878 / 0.2064
POP	0.9361	0.9360 / 0.9339 / 0.9395	0.0037	0.0000	0.5059 / 0.2873 / 0.2069
ITEM-KNN	0.8759	0.8787 / 0.8932 / 0.8451	0.0320	0.0003	0.5075 / 0.2936 / 0.1989
BPR	0.7723	0.8179 / 0.8279 / 0.5828	0.1634	0.0130	0.5358 / 0.3086 / 0.1555
ALS	0.7632	0.8024 / 0.8131 / 0.5974	0.1438	0.0100	0.5319 / 0.3068 / 0.1613
SLIM	0.7458	0.8045 / 0.8167 / 0.5027	**0.2093**	0.0243	0.5457 / 0.3153 / 0.1389
VAE	0.6894	0.7394 / 0.7637 / 0.4630	0.2005	**0.0248**	0.5426 / 0.3190 / 0.1384
ADV-VAE	0.7034	0.7449 / 0.7602 / 0.5220	0.1588	0.0145	0.5358 / 0.3112 / 0.1530

Table B.10: Novelty@5 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.9614**	0.9609 / 0.9620 / 0.9618	0.0007	0.0000	0.5057 / 0.2881 / 0.2062
POP	0.7104	0.7106 / 0.7100 / 0.7104	0.0004	0.0000	0.5061 / 0.2878 / 0.2061
ITEM-KNN	0.7434	0.7339 / 0.7384 / 0.7737	0.0265	**0.0003**	0.4995 / 0.2860 / 0.2145
BPR	0.9005	0.8950 / 0.9035 / 0.9100	0.0099	0.0000	0.5028 / 0.2889 / 0.2083
ALS	0.8140	0.8032 / 0.8173 / 0.8361	0.0219	0.0002	0.4992 / 0.2891 / 0.2117
SLIM	0.8284	0.8144 / 0.8318 / 0.8579	**0.0290**	0.0003	0.4974 / 0.2891 / 0.2134
VAE	0.8323	0.8207 / 0.8353 / 0.8566	0.0239	0.0002	0.4989 / 0.2890 / 0.2121
ADV-VAE	0.8342	0.8244 / 0.8346 / 0.8573	0.0219	0.0002	0.5001 / 0.2881 / 0.2118

Table B.11: NDCG@20 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0098	0.0098 / 0.0097 / 0.0100	0.0002	0.0001	0.5053 / 0.2838 / 0.2109
POP	0.0970	0.1067 / 0.0803 / 0.0964	0.0176	0.0104	0.5567 / 0.2385 / 0.2048
ITEM-KNN	0.1766	0.1646 / 0.1622 / 0.2260	0.0426	0.0131	0.4718 / 0.2644 / 0.2638
BPR	0.1893	0.1692 / 0.1843 / 0.2457	0.0510	**0.0156**	0.4522 / 0.2803 / 0.2675
ALS	0.2973	0.2758 / 0.2865 / 0.3652	0.0596	0.0090	0.4694 / 0.2774 / 0.2532
SLIM	**0.3193**	0.2935 / 0.3042 / 0.4036	**0.0734**	0.0118	0.4651 / 0.2744 / 0.2605
VAE	0.2964	0.2752 / 0.2865 / 0.3621	0.0579	0.0086	0.4699 / 0.2783 / 0.2518
ADV-VAE	0.2846	0.2671 / 0.2746 / 0.3416	0.0496	0.0070	0.4748 / 0.2778 / 0.2474

Table B.12: Precision@20 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0099	0.0099 / 0.0097 / 0.0100	0.0002	0.0001	0.5088 / 0.2829 / 0.2083
POP	0.0944	0.1030 / 0.0802 / 0.0933	0.0152	0.0081	0.5518 / 0.2445 / 0.2037
ITEM-KNN	0.1679	0.1587 / 0.1503 / 0.2152	0.0433	0.0136	0.4781 / 0.2577 / 0.2642
BPR	0.1703	0.1520 / 0.1651 / 0.2224	0.0469	**0.0163**	0.4517 / 0.2792 / 0.2692
ALS	0.2729	0.2540 / 0.2615 / 0.3353	0.0542	0.0090	0.4709 / 0.2759 / 0.2532
SLIM	**0.2914**	0.2689 / 0.2752 / 0.3692	**0.0668**	0.0120	0.4669 / 0.2720 / 0.2611
VAE	0.2722	0.2544 / 0.2600 / 0.3331	0.0525	0.0086	0.4728 / 0.2750 / 0.2522
ADV-VAE	0.2614	0.2468 / 0.2491 / 0.3145	0.0451	0.0071	0.4777 / 0.2743 / 0.2480

Table B.13: Recall@20 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0013	0.0013 / 0.0012 / 0.0012	0.0001	0.0011	0.5245 / 0.2807 / 0.1947
POP	0.0127	0.0140 / 0.0111 / 0.0117	0.0019	0.0080	0.5575 / 0.2520 / 0.1904
ITEM-KNN	0.0254	0.0240 / 0.0240 / 0.0309	0.0046	0.0080	0.4779 / 0.2715 / 0.2507
BPR	0.0276	0.0250 / 0.0279 / 0.0337	0.0058	**0.0099**	0.4577 / 0.2908 / 0.2515
ALS	0.0414	0.0388 / 0.0413 / 0.0477	0.0059	0.0046	0.4747 / 0.2874 / 0.2378
SLIM	**0.0441**	0.0408 / 0.0435 / 0.0532	**0.0082**	0.0077	0.4679 / 0.2839 / 0.2482
VAE	0.0413	0.0388 / 0.0415 / 0.0472	0.0057	0.0043	0.4749 / 0.2893 / 0.2358
ADV-VAE	0.0397	0.0377 / 0.0396 / 0.0446	0.0046	0.0030	0.4811 / 0.2871 / 0.2318

Table B.14: Coverage@20 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.8505**	0.6178 / 0.4213 / 0.3240	**0.1959**	0.0477	0.6243 / 0.2423 / 0.1334
POP	0.0041	0.0040 / 0.0033 / 0.0035	0.0005	0.0058	0.5499 / 0.2572 / 0.1929
ITEM-KNN	0.0456	0.0309 / 0.0287 / 0.0252	0.0038	0.0043	0.5372 / 0.2842 / 0.1786
BPR	0.6187	0.4403 / 0.3083 / 0.2157	0.1497	**0.0536**	0.6258 / 0.2493 / 0.1249
ALS	0.1796	0.1331 / 0.1158 / 0.0966	0.0243	0.0108	0.5585 / 0.2764 / 0.1651
SLIM	0.2084	0.1509 / 0.1273 / 0.0993	0.0344	0.0180	0.5720 / 0.2746 / 0.1534
VAE	0.2864	0.1892 / 0.1499 / 0.1208	0.0456	0.0221	0.5846 / 0.2634 / 0.1520
ADV-VAE	0.3079	0.2022 / 0.1605 / 0.1349	0.0449	0.0187	0.5802 / 0.2621 / 0.1577

Table B.15: Diversity@20 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.9255**	0.9255 / 0.9254 / 0.9258	0.0003	0.0000	0.5059 / 0.2879 / 0.2062
POP	0.8956	0.8942 / 0.8971 / 0.8969	0.0019	0.0000	0.5052 / 0.2884 / 0.2064
ITEM-KNN	0.8437	0.8483 / 0.8602 / 0.8094	0.0338	0.0003	0.5087 / 0.2936 / 0.1977
BPR	0.7368	0.7771 / 0.7878 / 0.5665	0.1475	0.0114	0.5336 / 0.3079 / 0.1585
ALS	0.7448	0.7751 / 0.7910 / 0.6060	0.1233	0.0073	0.5265 / 0.3058 / 0.1677
SLIM	0.7310	0.7765 / 0.7908 / 0.5355	**0.1702**	**0.0157**	0.5375 / 0.3115 / 0.1510
VAE	0.6932	0.7328 / 0.7548 / 0.5100	0.1632	0.0154	0.5348 / 0.3135 / 0.1517
ADV-VAE	0.7075	0.7376 / 0.7545 / 0.5678	0.1245	0.0082	0.5275 / 0.3071 / 0.1654

Table B.16: Novelty@20 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.9613**	0.9611 / 0.9616 / 0.9615	0.0003	0.0000	0.5058 / 0.2880 / 0.2061
POP	0.7202	0.7206 / 0.7195 / 0.7200	0.0008	0.0000	0.5063 / 0.2877 / 0.2061
ITEM-KNN	0.7613	0.7511 / 0.7579 / 0.7912	0.0267	0.0003	0.4992 / 0.2866 / 0.2142
BPR	0.9143	0.9098 / 0.9169 / 0.9216	0.0079	0.0000	0.5035 / 0.2888 / 0.2078
ALS	0.8254	0.8146 / 0.8282 / 0.8479	0.0222	0.0002	0.4993 / 0.2889 / 0.2117
SLIM	0.8353	0.8214 / 0.8381 / 0.8655	**0.0294**	**0.0003**	0.4975 / 0.2889 / 0.2136
VAE	0.8340	0.8226 / 0.8366 / 0.8584	0.0239	0.0002	0.4990 / 0.2888 / 0.2121
ADV-VAE	0.8349	0.8259 / 0.8350 / 0.8571	0.0208	0.0001	0.5005 / 0.2880 / 0.2116

Table B.17: NDCG@50 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0098	0.0098 / 0.0097 / 0.0098	0.0001	0.0000	0.5063 / 0.2868 / 0.2069
POP	0.0733	0.0791 / 0.0640 / 0.0723	0.0101	0.0059	0.5456 / 0.2512 / 0.2032
ITEM-KNN	0.1603	0.1506 / 0.1478 / 0.2017	0.0360	0.0112	0.4753 / 0.2654 / 0.2593
BPR	0.1594	0.1430 / 0.1553 / 0.2055	0.0417	**0.0147**	0.4538 / 0.2805 / 0.2657
ALS	0.2523	0.2357 / 0.2420 / 0.3074	0.0477	0.0082	0.4727 / 0.2762 / 0.2511
SLIM	**0.2673**	0.2476 / 0.2537 / 0.3348	**0.0581**	0.0108	0.4686 / 0.2733 / 0.2581
VAE	0.2505	0.2345 / 0.2415 / 0.3023	0.0452	0.0074	0.4737 / 0.2776 / 0.2487
ADV-VAE	0.2418	0.2287 / 0.2319 / 0.2877	0.0394	0.0062	0.4785 / 0.2762 / 0.2453

Table B.18: Precision@50 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0097	0.0097 / 0.0097 / 0.0096	0.0000	0.0000	0.5064 / 0.2883 / 0.2054
POP	0.0620	0.0659 / 0.0559 / 0.0610	0.0067	0.0036	0.5379 / 0.2595 / 0.2026
ITEM-KNN	0.1474	0.1399 / 0.1335 / 0.1853	0.0345	0.0114	0.4802 / 0.2607 / 0.2590
BPR	0.1369	0.1228 / 0.1323 / 0.1781	0.0368	**0.0156**	0.4538 / 0.2782 / 0.2680
ALS	0.2214	0.2079 / 0.2099 / 0.2707	0.0418	0.0084	0.4751 / 0.2730 / 0.2520
SLIM	**0.2324**	0.2166 / 0.2176 / 0.2917	**0.0500**	0.0109	0.4717 / 0.2696 / 0.2587
VAE	0.2195	0.2071 / 0.2085 / 0.2653	0.0388	0.0074	0.4774 / 0.2735 / 0.2491
ADV-VAE	0.2125	0.2024 / 0.2007 / 0.2538	0.0354	0.0065	0.4819 / 0.2720 / 0.2461

Table B.19: Recall@50 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	0.0030	0.0031 / 0.0030 / 0.0028	0.0002	0.0015	0.5250 / 0.2853 / 0.1897
POP	0.0232	0.0249 / 0.0219 / 0.0209	0.0027	0.0042	0.5430 / 0.2715 / 0.1855
ITEM-KNN	0.0541	0.0516 / 0.0513 / 0.0640	0.0085	0.0059	0.4831 / 0.2729 / 0.2441
BPR	0.0541	0.0497 / 0.0545 / 0.0646	0.0100	**0.0076**	0.4643 / 0.2897 / 0.2460
ALS	0.0817	0.0776 / 0.0810 / 0.0927	0.0101	0.0034	0.4807 / 0.2853 / 0.2339
SLIM	**0.0851**	0.0800 / 0.0832 / 0.1004	**0.0136**	0.0058	0.4753 / 0.2815 / 0.2432
VAE	0.0804	0.0765 / 0.0802 / 0.0904	0.0093	0.0030	0.4813 / 0.2870 / 0.2317
ADV-VAE	0.0780	0.0750 / 0.0770 / 0.0869	0.0080	0.0024	0.4862 / 0.2841 / 0.2297

Table B.20: Coverage@50 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.9902**	0.9018 / 0.7317 / 0.6107	0.1941	0.0169	0.5755 / 0.2657 / 0.1588
POP	0.0090	0.0087 / 0.0083 / 0.0081	0.0004	0.0006	0.5202 / 0.2812 / 0.1986
ITEM-KNN	0.1024	0.0742 / 0.0678 / 0.0567	0.0116	0.0074	0.5459 / 0.2840 / 0.1701
BPR	0.8576	0.6978 / 0.5432 / 0.3947	**0.2020**	**0.0334**	0.5975 / 0.2648 / 0.1377
ALS	0.2671	0.2083 / 0.1875 / 0.1599	0.0322	0.0073	0.5479 / 0.2807 / 0.1714
SLIM	0.3185	0.2413 / 0.2137 / 0.1699	0.0476	0.0125	0.5585 / 0.2814 / 0.1601
VAE	0.4162	0.2924 / 0.2419 / 0.1960	0.0643	0.0171	0.5734 / 0.2700 / 0.1566
ADV-VAE	0.4488	0.3131 / 0.2600 / 0.2183	0.0632	0.0143	0.5692 / 0.2691 / 0.1617

Table B.21: Diversity@50 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.8955**	0.8946 / 0.8966 / 0.8962	0.0013	0.0000	0.5054 / 0.2883 / 0.2063
POP	0.8409	0.8443 / 0.8364 / 0.8389	0.0053	0.0000	0.5080 / 0.2864 / 0.2056
ITEM-KNN	0.7988	0.8087 / 0.8162 / 0.7503	0.0439	0.0007	0.5122 / 0.2942 / 0.1936
BPR	0.6968	0.7368 / 0.7424 / 0.5348	0.1384	0.0116	0.5350 / 0.3068 / 0.1582
ALS	0.7177	0.7451 / 0.7589 / 0.5928	0.1107	0.0063	0.5253 / 0.3045 / 0.1702
SLIM	0.7074	0.7480 / 0.7604 / 0.5337	**0.1511**	**0.0130**	0.5350 / 0.3095 / 0.1555
VAE	0.6765	0.7109 / 0.7311 / 0.5160	0.1434	0.0122	0.5316 / 0.3112 / 0.1572
ADV-VAE	0.6908	0.7159 / 0.7295 / 0.5753	0.1028	0.0058	0.5243 / 0.3041 / 0.1717

Table B.22: Novelty@50 results for the RS algorithms studied and detected user communities in the undirected *Network of Users Who Vote on Postings*, whose edges are filtered with an edge weight threshold of four. The values in the columns "All" and "Communities A / B / C" represent the mean evaluation scores of all users in the communities collectively and for each user community individually, averaged over five cross-validation test folds. The RecGap quantifies the mean disparity between the average evaluation scores of the user communities. The Compounding Factor shows the effect of the model in amplifying data imbalances. The metric scores distribution over user communities is shown in the rightmost column. The population distribution of user communities is $B = [0.5059, 0.2879, 0.2061]$. Only users in communities with a minimum size of 750 users are considered. The RecGap and the Compounding Factor are calculated using the combined metric scores of all test folds. Values are rounded to the fourth digit.

Model	All	Communities A / B / C	RecGap	Comp. Factor	Score Distribution Communities
RAND	**0.9615**	0.9614 / 0.9615 / 0.9615	0.0001	0.0000	0.5059 / 0.2880 / 0.2061
POP	0.7691	0.7720 / 0.7646 / 0.7683	0.0049	0.0000	0.5078 / 0.2863 / 0.2059
ITEM-KNN	0.7816	0.7719 / 0.7787 / 0.8093	0.0250	0.0002	0.4997 / 0.2869 / 0.2134
BPR	0.9236	0.9198 / 0.9257 / 0.9299	0.0067	0.0000	0.5039 / 0.2886 / 0.2075
ALS	0.8354	0.8254 / 0.8377 / 0.8568	0.0210	0.0002	0.4999 / 0.2887 / 0.2114
SLIM	0.8419	0.8288 / 0.8445 / 0.8706	**0.0279**	**0.0003**	0.4980 / 0.2888 / 0.2131
VAE	0.8366	0.8259 / 0.8387 / 0.8601	0.0228	0.0002	0.4995 / 0.2887 / 0.2119
ADV-VAE	0.8369	0.8288 / 0.8365 / 0.8572	0.0189	0.0001	0.5011 / 0.2878 / 0.2111

List of Figures

List of Tables

Bibliography

[BGLL08] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, October 2008.

[BP16] Albert-László Barabási and Márton Pósfai. *Network Science*. Cambridge university press, Cambridge, 2016.

[CCK+00] Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rüdiger Wirth. CRISP-DM 1.0: Step-by-step data mining guide, 2000.

[CHV22] Pablo Castells, Neil Hurley, and Saúl Vargas. Novelty and Diversity in Recommender Systems. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 603–646. Springer US, New York, NY, 2022.

[CN06] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[CNM04] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, December 2004.

[DJB+23] Yashar Deldjoo, Dietmar Jannach, Alejandro Bellogin, Alessandro Difonzo, and Dario Zanzonelli. Fairness in recommender systems: Research landscape and future directions. *User Modeling and User-Adapted Interaction*, April 2023.

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

[DP22] Nicolas Dugué and Anthony Perez. Direction matters in complex networks: A theoretical and applied study for greedy modularity optimization. *Physica A: Statistical Mechanics and its Applications*, 603:127798, October 2022.

[EB99] M. G. Everett and S. P. Borgatti. The centrality of groups and classes. *The Journal of Mathematical Sociology*, 23(3):181–201, January 1999.

[EDBD22] Michael D. Ekstrand, Anubrata Das, Robin Burke, and Fernando Diaz. Fairness in Recommender Systems. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 679–707. Springer US, New York, NY, 2022.

[EHR23] Daniel Edler, Anton Holmgren, and Martin Rosvall. The MapEquation software package, April 2023.

[ETA+24] Michael D. Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D. Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera. All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation and effectiveness. In Sorelle A. Friedler and Christo Wilson, editors, *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, volume 81 of *Proceedings of Machine Learning Research*, pages 172–186. PMLR, 2018-02-23/2018-02-24.

[FH16] Santo Fortunato and Darko Hric. Community detection in networks: A user guide. *Physics Reports*, 659:1–44, November 2016.

[FN22] Santo Fortunato and Mark E. J. Newman. 20 years of network community detection. *Nature Physics*, 18(8):848–850, August 2022.

[For10] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, February 2010.

[Fre23] Ben Frederickson. Fast python collaborative filtering for implicit datasets, September 2023.

[GPR+22] Christian Ganhör, David Penz, Navid Rekabsaz, Oleg Lesota, and Markus Schedl. Unlearning Protected User Attributes in Recommendations with Adversarial Training. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2142–2147, Madrid Spain, July 2022. ACM.

[GSY22] Asela Gunawardana, Guy Shani, and Sivan Yogev. Evaluating Recommender Systems. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 547–601. Springer US, New York, NY, 2022.

[HKTR04] Jonathan L. Herlocker, Joseph A. Konstan, Loren G. Terveen, and John T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, January 2004.

[HKV08] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, Pisa, Italy, December 2008. IEEE.

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11–15, Pasadena, CA USA, 2008.

[JWZ+23] Di Jin, Luzhi Wang, He Zhang, Yizhen Zheng, Weiping Ding, Feng Xia, and Shirui Pan. A survey on fairness-aware recommender systems. *Information Fusion*, 100:101906, December 2023.

[JZ22] Dietmar Jannach and Markus Zanker. Value and Impact of Recommender Systems. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 519–546. Springer US, New York, NY, 2022.

[KB17] Marius Kaminskas and Derek Bridge. Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems. *ACM Transactions on Interactive Intelligent Systems*, 7(1):1–42, March 2017.

[KJJ18] Mozhgan Karimi, Dietmar Jannach, and Michael Jugovac. News recommender systems – Survey and roads ahead. *Information Processing & Management*, 54(6):1203–1227, November 2018.

[KPBB+09] Barbara Kitchenham, O. Pearl Brereton, David Budgen, Mark Turner, John Bailey, and Stephen Linkman. Systematic literature reviews in software engineering – A systematic literature review. *Information and Software Technology*, 51(1):7–15, January 2009.

[KRB22] Yehuda Koren, Steffen Rendle, and Robert Bell. Advances in Collaborative Filtering. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 91–142. Springer US, New York, NY, 2022.

[LCF+21] Yunqi Li, Hanxiong Chen, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. User-oriented Fairness in Recommendation. In *Proceedings of the Web Conference 2021*, pages 624–632, Ljubljana Slovenia, April 2021. ACM.

[LCX+21] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, and Yongfeng Zhang. Towards Personalized Fairness based on Causal Notion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1054–1063, Virtual Event Canada, July 2021. ACM.

[LCX+23] Yunqi Li, Hanxiong Chen, Shuyuan Xu, Yingqiang Ge, Juntao Tan, Shuchang Liu, and Yongfeng Zhang. Fairness in Recommendation: Foundations, Methods and Applications. *ACM Transactions on Intelligent Systems and Technology*, page 3610302, July 2023.

[LF09] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117, November 2009.

[LF14] Andrea Lancichinetti and Santo Fortunato. Erratum: Community detection algorithms: A comparative analysis [Phys. Rev. E 80, 056117 (2009)]. *Physical Review E: Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 89(4):049902, April 2014.

[LKHJ18] Dawen Liang, Rahul G. Krishnan, Matthew D. Hoffman, and Tony Jebara. Variational Autoencoders for Collaborative Filtering. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, pages 689–698, Lyon, France, 2018. ACM Press.

[LN08] E. A. Leicht and M. E. J. Newman. Community Structure in Directed Networks. *Physical Review Letters*, 100(11):118703, March 2008.

[LPL+24] Shangsong Liang, Zhou Pan, Wei Liu, Jian Yin, and Maarten De Rijke. A Survey on Variational Autoencoders in Recommender Systems. *ACM Computing Surveys*, 56(10):1–40, October 2024.

[MCF+21] Fernando Martinez-Plumed, Lidia Contreras-Ochando, Cesar Ferri, Jose Hernandez-Orallo, Meelis Kull, Nicolas Lachiche, Maria Jose Ramirez-Quintana, and Peter Flach. CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 33(8):3048–3061, August 2021.

[MMMO20] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Exploring Data Splitting Strategies for the Evaluation of Recommendation Models. In *Fourteenth ACM Conference on Recommender Systems*, pages 681–686, Virtual Event Brazil, September 2020. ACM.

[MRP+21] Alessandro B. Melchiorre, Navid Rekabsaz, Emilia Parada-Cabaleiro, Stefan Brandl, Oleg Lesota, and Markus Schedl. Investigating gender fairness of recommendation algorithms in the music domain. *Information Processing & Management*, 58(5):102666, September 2021.

[MZS20] Alessandro B. Melchiorre, Eva Zangerle, and Markus Schedl. Personality Bias of Music Recommendation Algorithms. In *Fourteenth ACM Conference on Recommender Systems*, pages 533–538, Virtual Event Brazil, September 2020. ACM.

[New04] M. E. J. Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, November 2004.

[New18] Mark Newman. *Networks*. Oxford University Press, July 2018.

[NG04] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, February 2004.

[NK11] Xia Ning and George Karypis. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *2011 IEEE 11th International Conference on Data Mining*, pages 497–506, Vancouver, BC, Canada, 2011. IEEE.

[NNDK22] Athanasios N. Nikolakopoulos, Xia Ning, Christian Desrosiers, and George Karypis. Trust Your Neighbors: A Comprehensive Survey of Neighborhood-Based Methods for Recommender Systems. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 39–89. Springer US, New York, NY, 2022.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RAB09] M. Rosvall, D. Axelsson, and C. T. Bergstrom. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, November 2009.

[RB08] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123, January 2008.

[RB11] Martin Rosvall and Carl T. Bergstrom. Multilevel Compression of Random Walks on Networks Reveals Hierarchical Organization in Large Integrated Systems. *PLoS ONE*, 6(4):e18209, April 2011.

[RD22] Shaina Raza and Chen Ding. News recommender system: A review of recent progress, challenges, and opportunities. *Artificial Intelligence Review*, 55(1):749–800, January 2022.

[Ren22] Steffen Rendle. Item Recommendation from Implicit Feedback. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 143–171. Springer US, New York, NY, 2022.

[RFGS09] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, Uai '09, pages 452–461, Arlington, Virginia, USA, 2009. AUAI Press.

[RRS22] Francesco Ricci, Lior Rokach, and Bracha Shapira. Recommender Systems: Techniques, Applications, and Challenges. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 1–35. Springer US, New York, NY, 2022.

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

[SL23] Markus Schedl and Elisabeth Lex. 3 Fairness of information access systems. In Mirjam Augstein, Eelco Herder, and Wolfgang Wörndl, editors, *Personalized Human-Computer Interaction*, pages 59–78. De Gruyter, July 2023.

[VA20] Vijay Verma and Rajesh Kumar Aggarwal. A comparative analysis of similarity measures akin to the Jaccard index in collaborative recommendations: Empirical and theoretical perspective. *Social Network Analysis and Mining*, 10(1):43, December 2020.

[VC11] Saúl Vargas and Pablo Castells. Rank and relevance in novelty and diversity metrics for recommender systems. In *Proceedings of the Fifth ACM Conference on Recommender Systems*, pages 109–116, Chicago Illinois USA, October 2011. ACM.

[WMZ+23] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma. A Survey on the Fairness of Recommender Systems. *ACM Transactions on Information Systems*, 41(3):1–43, July 2023.

[WWHX23] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. Personalized News Recommendation: Methods and Challenges. *ACM Transactions on Information Systems*, 41(1):1–50, January 2023.

[WWW+21] Chuhan Wu, Fangzhao Wu, Xiting Wang, Yongfeng Huang, and Xing Xie. Fairness-aware News Recommendation with Decomposed Adversarial Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4462–4469, May 2021.

[YH17] Sirui Yao and Bert Huang. Beyond Parity: Fairness Objectives for Collaborative Filtering. In *Advances in Neural Information Processing Systems*, volume 30, pages 2922–2931, 2017.

[ZTY+22] Shuai Zhang, Yi Tay, Lina Yao, Aixin Sun, and Ce Zhang. Deep Learning for Recommender Systems. In Francesco Ricci, Lior Rokach, and Bracha Shapira, editors, *Recommender Systems Handbook*, pages 173–210. Springer US, New York, NY, 2022.