Soha Pervez

VISUALIZATIONS AND EXPLANATIONS FOR SEQUENTIAL GROUP RECOMMENDATIONS

Faculty of Information Technology and Communication Sciences (ITC)
Master's thesis
October 2023

Abstract

Soha Pervez: VISUALIZATIONS AND EXPLANATIONS FOR SEQUENTIAL GROUP RECOMMENDATIONS
Master's thesis
Tampere University
Master's Degree Programme in Information Technology and Communication Sciences
October 2023

The popularity of sequential recommendations is on the rise these days. It is important for the system not to treat each round of recommendations as an independent activity; rather, it should store information about previous encounters. More and more people are creating groups for activities, which makes group recommendation systems more popular. It frequently happens, however, that recommenders are unable to find the most useful data pieces. This flaw is addressed by explaining why specific suggestions are given. This work proposes visualizations for recommendations generated by SQUIRREL, A Framework for Sequential Group Recommendations through Reinforcement Learning. We explored three why questions using the 20M MovieLens dataset. Explanations rely on the aggregation method used for the last iteration for a particular group, combined with single-user and group recommendations. The Graphical User Interface framework incorporates visualizations and explanations. We have used three test cases and are able to provide explanations personalized for each group.

Keywords: recommendation, sequentialrecommendation, visualization, graphicaluserinterface.

The originality of this thesis has been checked using the Turnitin Originality Check service.

Contents

1 Introduction

Everything we do these days, from online shopping to watching a movie on Netflix, has a recommendation system on its back end. These tools aid users in sorting through the enormous quantity of information accessible and selecting options that suit their preferences and interests. The reason for this is that recommendation systems play a huge role in ensuring the satisfaction of end-users. As a result, recommendation systems play a very crucial role in ensuring end-user satisfaction. They are able to do this by examining the user's previous interactions with the system, which may have included search queries, scrolling behaviors, clicks, reviews, etc. This indicates that the system takes into account the items that a user chose or liked during a prior session in addition to the present session. Understanding consumer preferences or generating informed decisions are the goals. This is a somewhat novel method of recommendations because most systems look at the user's ratings that are now available without taking into account how the user has previously interacted with the system. This is commonly referred to as sequential group recommendation because it takes into consideration a series of rounds of user recommendations Stratigi, Pitoura, and Stefanidis 2023.

Besides sequential recommendations, group recommendations have recently attracted a lot of attention. Instead of concentrating on one individual, the goal, in this case, is to balance the interests of multiple individuals. For instance, a group of friends who wish to play a game online. The computer would suggest a game that each of them would enjoy. Group recommendations are more challenging than single recommendations since they take into account the preferences of the entire group, not just one individual. The algorithm for group recommendations needs to be able to generate a list that, in principle, is pertinent to everyone's interests Stratigi, Pitoura, and Stefanidis 2023. One of the most common methods for achieving group recommendations is to use a single recommendation system for each group member and obtain their corresponding recommendation lists. Then, using an aggregation method, the group recommendation system takes over and tries to aggregate these lists into a single group recommendation list Masthoff 2011.

The complexity of sequential and group recommendations increases when they are combined. As a result, sequential group recommendations constitute a new area of research. A sequential suggestion analyses the group's prior interactions to provide a more rich experience Stratigi, Pitoura, and Stefanidis 2023. Consider four siblings who want to watch a Netflix series together, for instance. Each sibling's previous viewing history of their respective series will be analyzed by the recommender, who will then make a list of series it believes all of the siblings will enjoy.

Group suggestions are possible in many ways, but it is not always obvious which aggregation method is appropriate for a given test scenario. To address this, the Sequential Group Recommendations through ReinforcEment Learning (SQUIRREL) framework, was created. There are three primary parts to it: the state, the actions, and the reward. The state defines the group's current situation, the actions are the various group recommendation techniques (aggregation methods) that can be used, and the reward is the system's main objective Stratigi, Pitoura, and Stefanidis 2023.

The system examined the group's state at each recommendation round, such as each member's satisfaction, and then it selected an appropriate action to take, i.e., it selected a group recommendation method to use. This made reinforcement learning an obvious choice for SQUIRREL. A group suggestion list is created based on this choice, given back to the group, and a reward is determined. The group's status is then changed to reflect the changes brought about by the action, i.e., how happy the group members are with the most recent round of recommendations Stratigi, Pitoura, and Stefanidis 2023.

The issue is that while efforts are made to make interesting recommendations to users based on their interests, it is frequently possible that recommendation systems are unable to find the best data items to recommend. There are several possible causes for this. One is the "cold start" issue, where the system lacks sufficient data about a user or group of users to make reliable predictions. Another cause can be the users' excessive specificity. This indicates that the user has previously expressed interest in a particular category, and the system assumes that the user adores it and repeatedly suggests that. Uncertain data on people and their preferences may cause the systems to be misdirected, which is a third potential explanation. Last but not least, as a system depends substantially on hyperparameters and thresholds, bad recommendations may be closely related to the configuration of the system Stratigi, A. Tzompanaki, and Stefanidis 2020.

In this thesis, we combine explanations with the SQUIRREL model. We introduce the notion of *why questions*, which are based on how many times a particular movie genre has been recommended to the group. For this, we have worked on a combination of two explanations, one that considers how many times a particular genre has been recommended to group members individually and to the group as a whole i.e., how many times a particular genre was recommended to the group in previous rounds and the other checks the action taken in the last iteration.

The aim was not only to provide explanations but to provide them in a way that's interactive and more understandable. For this, an interactive GUI has been created, which initially provides links to the plots that display the recommended items for each iteration along with the movies recommended. Additionally, the user is able to receive written and visual explanations of why a movie genre was recommended to the group.

In general, our work has contributed to the following:

- We introduce the why questions and explanations for the results of SQUIR-REL, a model (Stratigi, Pitoura, and Stefanidis 2023) based on reinforcement learning.
- We focus on three why questions and their explanations in particular scenarios.
- We create an interactive GUI and visualization plots to make it more understandable for the user.
- We evaluate the why questions and explanations using the 20M MovieLens dataset on three use cases.

2 Literature review

2.1 Group Recommendations

The group recommendation system, where users' interests must be balanced rather than focusing on just one user, has received more attention in recent years. For instance, consider a group of friends who want to watch a series on Netflix. The system would suggest a series that each of them would enjoy. Since each group member has different preferences and the group recommendation system needs to balance them in order to suggest a group recommendation list that is ideally pertinent to the interests of all members, recommending a group is much more challenging than recommending a single item Stratigi, Pitoura, and Stefanidis 2023.

A drawback of single-user recommendations is the cold start issue, which can be solved by group recommenders Kompan and Bielikova 2014. From a historical perspective, group recommendations are produced by combining user profiles, using personal recommendations from individuals, or just by treating the entire group as a single user and using single-user suggestions Berkovsky and Freyne 2010.

For group recommendations, there are primarily two methods Jameson and Smyth 2007. In the first method (e.g., Z. Yu et al. 2006), we create a common user profile by merging all user profiles and applying a standard recommendation approach. The second method, which is also the most common approach to achieving group recommendations (e.g., Masthoff 2011, Ntoutsi et al. 2012, Baltrunas, Makcinskas, and Ricci 2010), is to employ a single recommender system to get a single recommendation list for each group member. After this, the group recommendation system takes over and employs an aggregation mechanism to merge these lists into a single group recommendation list.

A group recommender system can examine a wide range of factors during the aggregation stage. Yuan, Cong, and C.-Y. Lin 2014 suggests a recommendation model that combines the preferences of the group members with varying weights. They state that those who are knowledgeable about subjects that are important to the group are typically more influential and therefore they will carry more weight during the aggregation phase. Furthermore, Cao et al. 2018 learns the aggregation technique from data, to address the issue of preference aggregation in group recommendations. Under the neural collaborative filtering (NCF) architecture, it makes use of an attention mechanism to modify the representation of a group and learn the relationships between groups and items. The model strengthens individual user recommendations in addition to group recommendations, particularly for cold-start users with no prior individual interactions. Yin et al. 2019 uses a Bipartite Graph

Embedding Model (BGEM), in addition to an attention mechanism, to determine how each participant influenced the group's decision. An attention mechanism is used to learn each user's social influence and adapt it to various groups. They also created a novel deep social influence learning framework to use and integrate users' knowledge of their local and global social network structures in order to further enhance the estimation of users' social influences. Salehi-Abari and Boutilier 2015 created a model called preference-oriented social networks. They imply that preferences (for things like goods, services, or political parties, for example) are likely to be correlated among people who directly interact in a social network. Therefore, their model represented these connections between individual preferences, where preferences were expressed as rankings among many possibilities.

By noticing the way the group members engage with one another, Vinh Tran et al. 2019 proposed a neural architecture, Medley of Sub-Attention Networks (MoSAN), which was developed on the insight that decision-making (in groups) is typically dynamic, i.e., a user's choice is greatly influenced by the other group members. Each sub-attention module in MoSAN model represents a single member, modeling a user's preference in relation to every other group member. The group then uses a Medley of Sub-Attention modules to decide on a course of action as a whole. In contrast to current approaches, which concentrate on small user groups, D. Qin et al. 2018 suggests a novel framework that divides big groups into subgroups based on interests, uses collaborative filtering to produce suggestions within each subgroup, and then combines the recommendations to produce a final group recommendation. By creating a recommendation set using collaborative filtering and deleting unnecessary items, Kim et al. 2010 suggests an enhanced group recommendation technique that improves both the efficacy of group suggestions and the satisfaction of individual group members.

In order to increase group members' happiness and reduce unfairness among them, X. Lin et al. 2017 analyzes the group recommendation problem from a fresh angle. Depending on how pertinent the suggested items are to each group member, a utility score is given to them. After that, it provides a list of group recommendations by balancing the utility of the group members. In Sacharidis 2019, the similarity between a user's individual and group recommendations serves to quantify their utility. When constructing the group recommendation list, their method takes sets of N-level Pareto optimum items into account. D. Qin et al. 2018 proposes the dynamic group aggregation system (DGAS) which is a different approach than the traditional aggregation methods. Traditional aggregation methods give every member of a group the same amount of weight, which does not accurately reflect their individual contributions. In contrast, DGAS determines subgroup weights which take subgroup contribution and interests into account and therefore overcomes the limitations of existing techniques.

2.2 Sequential Recommendations

In general, sequential recommenders are divided into three major categories based on how many prior interactions they take into account: *Last-N interactions-based recommendations*, *Session-based recommendations* and *Session-aware recommendations* Quadrana, Cremonesi, and Jannach 2018. The first method merely takes into account the most recent N user actions (Cheng et al. 2013, Lian, V. Zheng, and Xie 2013, Q. Liu et al. 2016). This is because for each user, the system records a substantial amount of historical data, yet a sizable chunk of this data is duplicated and offers no usable information. The system may encounter difficulties as a result of the overwhelming volume of data since it becomes challenging to sort through and derive useful insights from redundant and unnecessary data. Session-based recommendations do not take into account all previous interactions; rather, they just take into account the interactions that a user engages in during the current session. Personalized recommendations based on the user's current interests and preferences are frequently provided by using this strategy in news and advertising platforms (Garcin, Dimitrakakis, and Faltings 2013, Hidasi et al. 2016). The last group of recommenders includes data on the user's most recent engagement as well as their whole history. These recommenders can make more precise and customized recommendations to improve the user's experience by taking into account both the most recent interaction and the user's past interactions. They are commonly employed in e-commerce and app recommendation systems (Hariri, Mobasher, and Burke 2012, Jannach, Lerche, and Jugovac 2015, Quadrana, Karatzoglou, et al. 2017). In Hansen et al. 2020 a new session-aware music recommendation system is proposed. It proposes a neural network that models user preferences as a series of embeddings depending on prior consumption behavior (such as device usage and time) and the current situation. This can anticipate the songs a user will play.

Although recommender systems are crucial for web platforms, due to variations in ranking positions and attention levels, they sometimes treat goods unfairly. Borges and Stefanidis 2019 suggests employing Variational Autoencoders (VAEs) with extra randomness to improve fairness Pitoura, Stefanidis, and Koutrika 2021 in numerous rounds of recommendation, while reducing bias and encouraging diversity. (Borges and Stefanidis 2021, Borges and Stefanidis 2022, Borges and Stefanidis 2020) penalizes ratings awarded to items based on their previous level of popularity.

With the use of a bipartite graph representation, the framework proposed in J. Qin et al. 2020 uses cross-neighbor relation modeling to find collaborative information. Linkages between nodes in the graph represent the interactions between users and items. The framework takes into account 2-hop neighbors, which are referred to as high-order collaborative relations, in addition to only taking into account nodes that are directly connected. It intends to collect more detailed and nuanced information about collaboration between users and items by taking into account these high-order collaborative relations. In order to describe dynamic group representations, Wang et al. 2020 suggests a unique method termed GLS-GRL for sequential group recommendation, which combines sequential recommendation and group recommendation. To capture user-item interactions and item-item co-occurrence, GLS-GRL builds long-term and short-term graphs. Then it creates user representations based on graph representation learning. Finally, it employs a limited user-interacted attention mechanism to encode correlations between group members.

The work presented here is an extension of the SQUIRREL Model Stratigi, Pitoura, and Stefanidis 2023. The model utilizes sequential group recommendation techniques that were first put forth in (Stratigi, Nummenmaa, et al. 2020, Stratigi, Pitoura, Nummenmaa, et al. 2021). The SDAA (Sequential Dynamic Adaptation Aggregation method) takes the group into account as a whole and dynamically determines a weight based on how satisfied the group members are. Using this weight, the preference score of an item for the user who was least satisfied with the last round of recommendations is merged with the average preference score for the item for all group members. The Sequential Individual Adaptation Aggregation, or SIAA, method, on the other hand, is user-centric. Each member's satisfaction and disagreement scores are used to determine a weight for each person, which is then applied to the preference scores of the group members during the aggregation phase. Average+ takes into account the complete list of data items as opposed to the other algorithms, which examine each item separately. In order to benefit from the high satisfaction levels, it improves on the average technique while taking into account more than the specified k items. Items that provide the lowest disagreement ratings as compared to the remainder of the list are gradually added to the group suggestion list.

The empirical research of Masthoff 2004 and Piliponyte, Ricci, and Koschwitz 2013 examines several aggregation algorithms for suggesting a series of television episodes and music tracks, respectively, to groups of users under specific application instances. Masthoff 2004 has used strategies such as Average Strategy, Average Without Misery Strategy, and Least Misery Strategy when selecting sequences for groups to watch. Piliponyte, Ricci, and Koschwitz 2013 introduced a 'Balancing' technique to produce a series of music track recommendations that continuously balance users' satisfaction levels. The approach that has been proposed in Stratigi, Pitoura, and Stefanidis 2023 is the first to put a strong emphasis on choosing an aggregation method from a pool of aggregation methods for each round of group recommendations using reinforcement learning.

2.3 Reinforcement Learning in Recommendations

The recommendation problem has changed in recent years. It is now thought of as a sequential decision problem, represented as a Markov decision process (MDP), and resolved using reinforcement learning (RL) techniques. RL-based recommender systems (RLRSs) have emerged as a result of the ability of deep reinforcement learning (DRL) to apply RL to recommendation issues with huge state and action spaces Afsar, Crump, and Far 2022. One of the earliest works in this field is Taghipour, Kardan, and Ghidary 2007, which entails the system constantly engaging with users and learning from their behavior, the research provides a fresh machine-learning paradigm for online recommendation. Here the environment is the user's most recent N visited pages, actions are page recommendations, and rewards are a weighted sum of the user's time spent on the recommended page and its ranking. In order to solve the issues of dynamic news features, user preferences, and scant user feedback in previous models, G. Zheng et al. 2018 develops a Deep Reinforcement Learning framework for news recommendation. The framework uses Deep Q-Learning and is composed of two components: offline and online. In the offline stage, people and news are mined for four different kinds of features. The reward is forecasted using a multilayer Deep Q-Network (i.e., a combination of user-news click label and user activeness) from these four types of features. Utilizing offline user-news click data, this network is trained. Then, during the subsequent portion of the online training, recommendation agent G will communicate with users and update the network.

With regard to long-term recommendation accuracy, a brand-new top-N interactive recommender system built on deep reinforcement learning is proposed in Huang et al. 2021. This system performs noticeably better than earlier approaches. They focus on cold-start and warm-start as their two key areas. The model depended on interactions between users (agents) and the recommender system (environment). As a result, it could be used in settings with insufficient content knowledge. The X. Zhao et al. 2017 suggests a novel recommender system that uses Reinforcement Learning (RL) to continually learn better methods while interacting with users, characterizing the interactions as a Markov Decision Process (MDP). The authors present an online user-agent interacting environment simulator that enables pre-training and offline evaluation of model parameters prior to the online application. The research demonstrates the significance of list-wise suggestions during user-agent interactions and develops a unique method to include them in the framework for list-wise recommendations that is proposed, named LIRD. The study discusses the shortcomings of current recommender systems, which prioritize short-term gains at the expense of long-term gains. It uses RL to discover the best recommendation methods that take long-term benefits into account.

In order to track changes in user preferences over time, Yuyan, Xiayao, and Yong 2019 proposes a deep learning movie recommender model that makes use of reinforcement learning and prioritized experience replay. Reinforcement learning comes into play when the model uses agents to learn about user preferences and movie features. By employing prioritized experience replay, the model is able to accurately represent the changing interests of users and generate recommendations based on their choices. The deep learning methodology enables the model to record intricate links and patterns between users and movies, resulting in more precise and individualized suggestions and provides a fresh and practical approach to the problem of identifying user preferences and adjusting to their evolving preferences over time. Moling, Baltrunas, and Ricci 2012 examines a situation in which users can get content from many channels, and a personal recommender system deployed on the client side chooses which channel to suggest depending on the user's listening habits. When compared to a baseline system that does not use implicit feedback, the suggested system allows users to listen to streaming tracks for a greater percentage of the time since it incorporates reinforcement learning techniques to choose the next channel to play. The customers' preferred music channels make up the explicit input, while their requests for new songs create the implicit feedback. The usage of an MDP is also made in Shani, Brafman, and Heckerman 2015, where they suggest a commercial system that uses an ordered list of choices made by each user as the environment's current condition to forecast and suggest a new product. They also point out that their system is one of the few commercially available recommender systems and the first to disclose an experimental study carried out on a genuine commercial site, demonstrating the usefulness of their MDP-based methodology from a business perspective.

The existing research on reinforcement learning-based recommendation systems mainly focuses on particular recommendation domains. However, the SQUIRREL framework Stratigi, Pitoura, and Stefanidis 2023, on the other hand, strives to be more versatile in terms of the fields it can be used. It is made to combine several strategies in order to deal with the problems that individual recommendation methods have.

2.4 Transparency via Visualizations and Explanations

A lot of research has been done on how recommender systems provide explanations. Introducing the idea of why-not questions—explanations offered after recommendations, Stratigi, A. Tzompanaki, and Stefanidis 2020 broadens the notion of post-hoc, model-based explanations. It emphasizes the system designer as the recipient of these justifications and offers a thorough categorization of why-not inquiries based on absence, granularity, and dependence on already-recommended items. Additionally, it offers tailored explanations based on models that are aimed at system engineers.

In accordance with users' implicit or explicit feedback, CF explanations are given (for a review of explanations in recommenders, see Tintarev and Masthoff 2007). Giving the user the best option is conceivably the easiest approach to presenting a recommendation. The selection process for this item could then be part of the justification. Let's imagine a situation where a user of a music streaming service prefers rock music but dislikes hip-hop and pop genres. A recent rock record, perhaps one from a venerable rock band, can be suggested by the recommender system. This recommendation's generated explanation could be:

"We've seen from your recent listening history that you've been really into rock music. We believe you'll particularly enjoy this new album from one of the classic rock bands given your love for rock."

The recommender system in this example considers the user's music listening history and genre preferences to deliver a customized music recommendation and an explanation Tintarev and Masthoff 2007.

Kaffes, Sacharidis, and Giannopoulos 2021 offers counterfactual explanations, which are those minor adjustments to the user's interaction history that caused them to see the explanation-needed recommendation result. The suggested technique produces post-hoc explanations that are accurate, personal, defensible, and useful. The paper Bidoit-Tollu, Herschel, and K. Tzompanaki 2014 suggests the Ted method, which tackles the problem of developers receiving insufficient and inconsistent justifications for why their data modifications did not yield desired results in conjunctive queries. The technique makes sure that the same comprehensive query-based explanations are generated for reordered conjunctive query trees, enhancing usability and keeping the benefit of utilizing a declarative query language.

In Tao et al. 2019 regression trees are integrated with latent factor models for recommendation. The path of each component on regression trees offers an explanation for the suggestions that are produced as the regression tree grows and the latent factors are further refined. Vig, Sen, and Riedl 2009 uses community tags to provide explanations for the recommendations which they refer to as Tagsplanations. The two main components used were tag relevance, or how well a tag describes an object, and tag preference, or the user's opinion of a tag. Verma et al. 2022 uses users' preferences to explain recommendations and calls this approach as RecXplainer. In Morisawa and Yamana 2021 the explanations are generated by extracting features from the LIME (Local Interpretable Model-agnostic Explanations) which is the interpretation model. They select an ideal number of features in the interpretation model rather than using all the features so that the interpretation model becomes easy to learn. Chang, F. Harper, and Terveen 2016a created tailored natural language explanations, a process that combined crowdsourcing and computation. The way people explain word-of-mouth recommendations served as the inspiration for this technique.

Visualizations are frequently employed in recommender systems to present explanations due to their advantages (Nunes and Jannach 2017, Zhang and X. Chen 2020). Early studies on explainable recommendations proposed various methods of providing the user with explanations of recommendations based on charts (e.g., bar chart, pie chart, histogram, tag cloud), and they demonstrated how visual explanations can help to improve transparency and users' trust as well as a recommender system's acceptance (see e.g., (Bilgic and Mooney 2005, Gedikli, Jannach, and Ge 2014, Herlocker, Konstan, and Riedl 2000, Vig, Sen, and Riedl 2009)). For instance, Herlocker, Konstan, and Riedl 2000 compared 21 different explanatory visualizations based on charts that explained the recommendation in terms of the neighborhood of similar users in their seminal work of evaluating collaborative filtering-based recommender systems. Movie tags were used by Vig, Sen, and Riedl 2009 to create suggestions and explanations based on features. The technique employs bar charts to display the movie's features and explain to consumers why each element is important to them in order to explain the recommended movie.

Recent studies on visual explanation in recommender systems employed photos as the display format to offer explanations rather than charts. An excellent starting point for image-based, visually explicable recommender systems that exclusively use deep learning strategies is the survey by Zhang and X. Chen 2020. This line of research has attempted to use item images for explainable recommendations in order to make use of the intuitive power of visual imagery.

Human control and engagement can also help to boost the transparency of AI and decision-making systems, according to a number of research from several application fields, including human-centered AI (Shneiderman 2020, Shneiderman 2022), interactive machine learning (Amershi et al. 2014, Dudley and Kristensson 2018, Jiang, S. Liu, and C. Chen 2019), and visual analytics (S. Liu et al. 2017, Spinner et al. 2020]. Interactive, visual, and exploratory user interfaces, as noted in Shneiderman 2022, can help users progress progressively toward their objectives, improve their comprehension of how the system operates, and avoid the bewilderment and surprise that might otherwise necessitate the need for explanation. Users of interactive recommender systems can inspect the recommender process and manipulate the system to improve recommendations through visual and exploratory user interfaces He, Parra, and Verbert 2016. Users can influence the recommendations made by the algorithm in a variety of ways. Users have influence over the input (user profiles), process (algorithm parameters), and/or output (recommendations) of recommender systems (He, Parra, and Verbert 2016, Jannach, Naveed, and Jugovac 2017). Previous research demonstrates that interactive recommender systems can improve transparency (Masthoff 2011, C.-H. Tsai and Brusilovsky 2017], trust (Harambam et al. 2019, C. Tsai and Brusilovsky 2021), perceived effectiveness, and user happiness (Hijikata, Kai, and Nishida 2012, Jin, Tintarev, and Verbert 2018, Pu, L. Chen, and Hu 2012) as well as help users develop better mental models (Eiband et al. 2018, Ngo, Kunkel, and Ziegler 2020).

Human control, or exploration, as well as explanation, can help users create practical mental models in recommender systems, which can result in transparency. However, they support system transparency in a variety of ways (C. Tsai and Brusilovsky 2021). Users cannot be guaranteed to develop accurate or comprehensive mental models of how the system functions by using a visual exploratory user interface that allows users to manipulate the system's input, process, and/or output (Jannach, Naveed, and Jugovac 2017). It may be possible to increase user perception of transparency in some situations by letting users into the recommender system's "black box" by giving them reasons for system-generated recommendations (Gedikli, Jannach, and Ge 2014, R. Zhao, Benbasat, and Cavusoglu 2019]. This would allow users to gain a better grasp of how the system works. Additionally, users need insights into the system's logic, which may be obtained through explanation, in order to successfully provide feedback and exert control over the system's suggestions (Eiband et al. 2018, Ngo, Kunkel, and Ziegler 2020), C.-H. Tsai and Brusilovsky 2017]. We therefore go beyond interactive recommendation in this work and instead concentrate on pairing explanation with visualization methods to aid users in understanding and interacting with the recommendation process.

3 The SQUIRREL Framework

The SQUIRREL framework, SeQUentIal Group Recommendations through ReinforcEment Learning Stratigi, Pitoura, and Stefanidis 2023, is a model based on reinforcement learning (RL) that addresses the sequential nature of group recommendations. RL focuses on teaching computers how to make decisions in a given environment in a way that maximizes a cumulative reward, and that is why it is a natural choice for SQUIRREL. In the SQUIRREL, a user u's *satisfaction* with the items in the group recommendation list GL_G^j is determined by comparing them to the best items for the user u, i.e., $B_{u,k}^j$ list (that contains top k recommendations (items with the highest score) for user u at round j) and disagreement of a user u is determined as the difference between the u's satisfaction score and the maximum satisfaction score among the group members Stratigi, Pitoura, and Stefanidis 2023. SQUIRREL consists of three main components: state, actions, and reward.

- The state represents the current status of the group G (mainly how satisfied the current group is), taking into account factors such as historical data, feedback from users, and items recommended and selected by the group.
- The actions refer to the various group recommendation methods that can be used. These methods are selected based on the current state of the group and aim to produce the maximum reward (how applicable the suggested data items are to the entire group).
- The reward is what the system aims to accomplish as its main objective, which is typically the satisfaction of the group members with the recommended items. The relevance of the items in the group recommendation list for each group member is a key factor in determining this satisfaction.

In the SQUIRREL model, an agent interacts with an environment E in order to maximize the cumulative reward at the end of each recommendation round. This is similar to a Markov Decision Process (MDP). A tuple of (S, A, P, R) can be used to represent the Markov decision process, where:

- S represents the environmental state, specifically the group state expressed through the utility scores of the group members i.e., degree of satisfaction and disagreement for the entire group. Each group member has an individual state at each round, defined as S_u^j of user u at round j.
- A is a collection of distinct actions comprising several SQUIRREL model aggregation functions, ranging from straightforward ones like Average to more intricate ones like SDAA.

Table 3.1 *Summary of the notations used in the SQUIRREL framework Stratigi, Pitoura, and Stefanidis 2023.*

Notation	Definition	Notation	Definition
I	Set of items	GL^j_G	Group recommendation list at round j
U	Set of users	GR	Sequence of group recommendations
G	Group	μ	Number of rounds in the sequence
u	User	SQUIRREL Model	
d	Item	S	State of the environment
j	Round	A	Set of actions
B^j_u	Recommendation list for user u at round j	P_a(s,s')	Probability of transitioning from state s to state s' j
$B^j_{u,k}$	Top k recommendations for user u at round j	R_a(s,s')	Reward from transitioning from s to s under action a
p_j(u,d)	User u's relevance score for item d at round j	π	Policy of the model

- Pa(s, s') defines the probability of transitioning from state s to state s' during round j under action a.
- Ra(s, s') is the reward gained from transitioning from state s to state s', which describes the quality of recommendations given by the model. Based on group utility scores (which indicate the level of agreement and dissatisfaction for the entire group), two reward functions—general group satisfaction and member disagreement—are defined Stratigi, Pitoura, and Stefanidis 2023.

We outline the SQUIRREL model's recommendation round structure in Figure 3.1. The group is sent to a single-user recommender system at the start of round j, which creates a set of recommendation lists for each group member, B^j_u. These lists are then provided to the SQUIRREL model, in which the agent monitors the environment S_j and, specifically, the level of satisfaction of the current group. After that, it chooses a suitable action j(α_j) to combine the lists B^j_u. As a result, the model enters the subsequent state S_{j+1}, where we update both the calculated reward R_{j+1} and the overall satisfaction of the users. The model then sends the group its created group recommendation list, abbreviated GL_j. Table 3.1 lists all the notations used in this work Stratigi, Pitoura, and Stefanidis 2023).

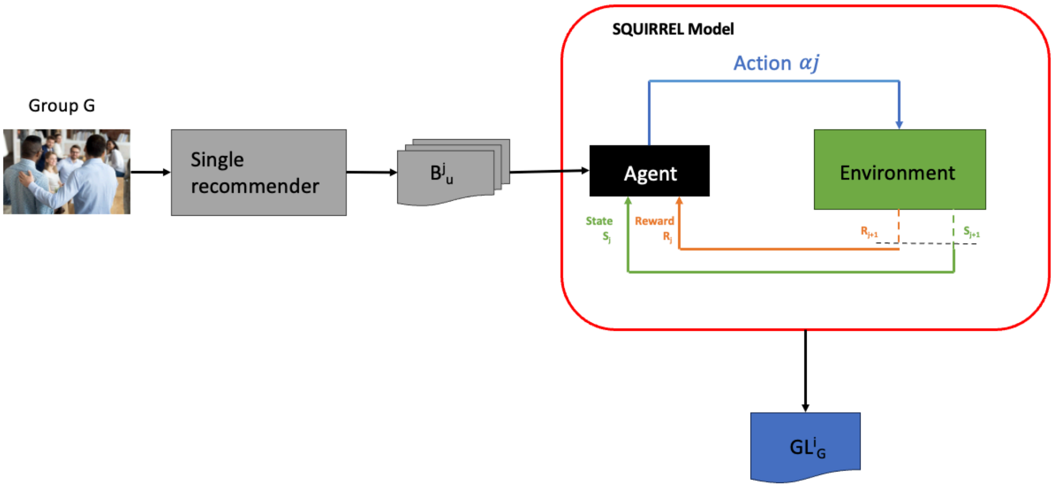

Figure 3.1 *The SQUIRREL Model Stratigi, Pitoura, and Stefanidis 2023*

3.1 The SQUIRREL Components

In this section, we briefly discuss the major components of the SQUIRREL model, namely state, action, and reward Stratigi, Pitoura, and Stefanidis 2023.

- The State of the model is defined using user utility scores that are defined as the satisfaction of each user individually with the group recommendations.
- There are two distinct reward functions—the reward function R_s, which is based on group-wide satisfaction, and the reward function R_{sd}, which is based on group-wide satisfaction and disagreement (the difference between the group's lowest and highest satisfaction scores). The reward mechanism, however, is quite adaptable and can be changed to support the particular goal for which the framework is utilized.
- The actions are the driving factor behind the model. Several recommendation lists, generated through user-based collaborative filtering (CF), are combined in a single group recommendation list using 6 actions which are Stratigi, Pitoura, and Stefanidis 2023:

  Average. The average of all predicted scores for a given item among all group members is the group predicted score for that item.

  RP80. This method combined group relevance (defined as the average prediction score produced from the single recommendation system across all the members of group G) and group disagreement scores (defined as Average Pairwise Disagreement between the predicted scores of the group members).

Par. This method balanced the variance in group members' satisfaction with an item with the average satisfaction that item generates for the group as a whole.

SDAA. This method determined a weight by taking into account the group members' historical satisfaction. This weight struck a balance between the group's average predicted score for a given item and the predicted score for that item given by the least satisfied member.

SIAA. This method also utilized a weight for aggregations but in contrast to SDAA focused on each group member individually. Weights are assigned according to the satisfaction level of the user and their satisfaction with previous recommendations.

Avg+. This method counters the drawback of the worst group disagreement scores in the classic average aggregation method by employing an average aggregation in the first phase and filling in the group recommendation list iteratively with items that produce the lowest potential group disagreement score in the second phase.

4 Visualizations and Explanations in Recommendation Systems

In this chapter, we go into details of why is there a need to introduce 'why questions' in recommendation systems and what are the different methods of representing their explanations.

4.1 Why Questions and Explanations in Recommendation Systems

Present-day recommender systems frequently give the impression of being a "black box" to their users by withholding crucial information about why items are recommended and how they relate to the users' interests. Although recommenders make an effort to accurately present users with interesting items based on their preferences, it frequently happens that they are unable to find the best data items to suggest. This may be the result of numerous factors. One explanation could be the cold start issue when the system lacks sufficient data about a user to make reliable predictions. The consumers' over-specification may also be a contributing factor. This indicates that a user has already indicated a preference for a particular category, making it unlikely for the system to suggest products that fall under a different category. Furthermore, confusing data on customers and their preferences frequently causes systems to be misdirected. Last but not least, as a system depends heavily on its hyper-parameters and thresholds, unlucky recommendations may be closely related to the configuration of the system Stratigi, A. Tzompanaki, and Stefanidis 2020.

The long-standing issue of explaining recommendations is most frequently resolved by including explanations with recommendations (C. Yu, Lakshmanan, and Amer-Yahia 2009, Chang, F. Harper, and Terveen 2016b). Increased interest in adding explanations to recommender systems has arisen as a result of the lack of transparency, with the aim of increasing the transparency of these systems and giving users knowledge that will help them construct a precise mental map of the system's behavior (Nunes and Jannach 2017, Masthoff 2011, Zhang and X. Chen 2020). This way, the user or the system's designer gets insights on why an item is suggested. The explanations can then vary on granularity or presentation format based on the final consumer, i.e., the final user of the recommender or the designer of the system Stratigi, A. Tzompanaki, and Stefanidis 2020. For example, if a user asks why a certain item was recommended, the system could use that information to refine its algorithms and provide better suggestions in the future. Let's say we have a movie recommendation system. The system mostly recommends the user movies of a certain genre e.g., Crime. The user can then ask a question *'Why have I been recommended crime most of the times?'*.

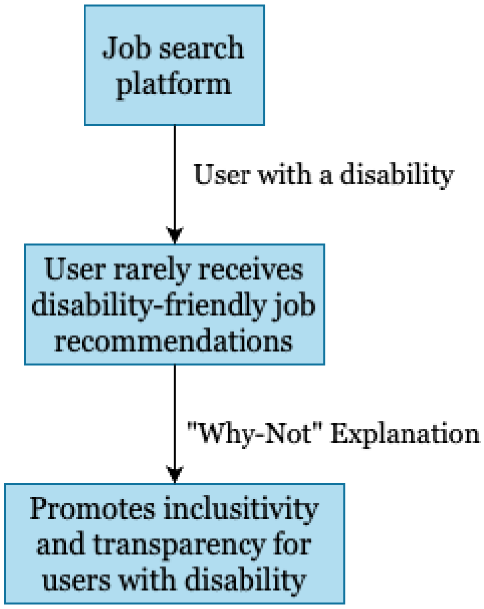

Figure 4.1 *Enhancing Inclusivity and Transparency in Job Recommendations for Users with Disabilities*

By utilizing the why-not inquiry notion, Stratigi, A. Tzompanaki, and Stefanidis 2020 elaborate on the idea of post-hoc, model-based explanations, i.e., explanations given after the recommendations have been generated and based on the knowledge of the system. These inquiries concern why some items weren't presented rather than why certain items were. Consider a service that streams music and builds playlists for users based on their listening habits and preferences. It might be a lost opportunity if the service's algorithm routinely excludes music from a specific genre or artist from being included in a user's playlist. The engineering team may look into the reasons behind these omissions and come up with solutions to improve user satisfaction and music discovery. This may help users, who enjoy a more varied and customized listening experience, as well as underappreciated artists or genres, who may get greater exposure through the playlists, leading to a more welcoming and enjoyable music streaming platform for everybody.

On the other hand, for a final user who is completely uninformed of the context or his or her preferences, asking a why-not question may not be such an easy assignment. It is still applicable in the case of an informed user who is aware of the context of the recommendations, though. Imagine a site for job searching (see Figure 4.1) where a user with a disability discovers that they seldom ever get suggestions for jobs that provide reasonable work accommodations or are more accommodating to people with disabilities. The platform could give this user a "why-not" explanation, assisting them in comprehending why these chances are not as prominently highlighted in order to foster trust in the system and ensure fairness. By doing this, the platform supports inclusivity for people with impairments as well as transparency and confidence in its recommendation system. This strategy makes certain that every user receives pertinent employment recommendations that take into consideration their unique requirements and circumstances. In this work, we separate explanations into two categories: general explanations based on the broad context of the issue and model-specific explanations based on the fundamental properties of the recommendation model, i.e., the SQUIRREL Model.

4.2 Dimensions of Explainable Recommendations

We can implement explanations for recommendations using a variety of different methods, strategies, and algorithms Nunes and Jannach 2017. However, Chatti, Guesmi, and Muslim 2023 divides explainable recommendation research into four categories (see Figure 4.2):

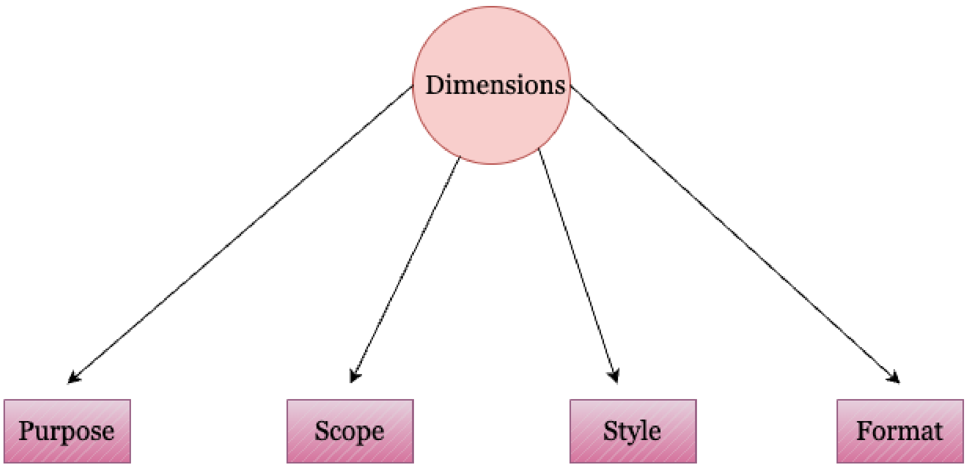

Figure 4.2 *Dimensions of explainable recommendations*

- **Explanation purpose**. Explanations can accomplish a variety of objectives in addition to assisting users in comprehending the output and logic of the recommender system. There are seven possible advantages that explanations may bring to a recommender system, according to Tintarev and Masthoff 2007 which are transparency, scrutability, trust, efficacy, persuasion, efficiency, and contentment.
- **Explanation scope**. Explainable recommendations can take the user model as their input, the algorithm as their recommendation process, and/or the product as their output into account. Those that concentrate on the recommendation process to learn how the algorithm functions are known as model-based explainable recommendations Zhang and X. Chen 2020. Explainability

of the recommendation output puts a lot of emphasis on the result. This strategy approaches the recommendation procedure as a black box, disregards its explainability, and instead creates unique methods to explain the recommendation outcomes that were produced by the black box. The term ”post-hoc explainable recommendation” refers to this strategy Zhang and X. Chen 2020. In the case of the user model, the method summarizes the system’s knowledge about the user’s preferences rather than explaining to the user why a specific item was recommended. The user is then free to examine this summary and directly adjust his or her user model Balog, Radlinski, and Arakelyan 2019.

- **Explanation style**. Another way to categorize explainable recommendation analysis is by explanation style, which is the model or method for coming up with explanations (Zhang and X. Chen 2020, Balog, Radlinski, and Arakelyan 2019). The four primary techniques for explanation can be broadly characterized as (1) Content-based, (2) Collaborative with its two subcategories of neighborhood-based and model-based, (3) social, and (4) hybrid techniques.
  - **Content-based methods**. These methods match products to users based on characteristics like genre, actor, director, and movie length. Users can immediately comprehend the item’s qualities and attributes, making it simple to explain why a product is suggested Zhang and X. Chen 2020.
  - **Collaborative-based methods**. These methods use the ”wisdom of crowds” to create recommendations based on ratings or usage patterns rather than depending just on content information (Zhang and X. Chen 2020, Balog, Radlinski, and Arakelyan 2019).
  - **Social explanation methods**. These methods use information about social friends to provide a user with the interests of his or her social friends as explanations for recommendations Zhang and X. Chen 2020.
  - **Hybrid explanation methods**. These methods offer explanations by combining two or more separate explanation techniques Kouki et al. 2019.
- **Explanation format**. We can display recommendation explanations using a wide range of display formats, such as related users or items, images, sentences, charts, or lists of logic (Zhang and X. Chen 2020, Masthoff 2011). Generally speaking, there are two types of recommendation explanation formats: textual explanations and visual explanations.
  - **Textual explanations** produce a portion of text information as a recommended explanation, and depending on how the textual explanations

are presented to users, they can be broadly categorized into sentence-level (give a complete, grammatically correct sentence that explains recommendation) and feature-level/aspect-level techniques (present product characteristics (such as color, quality, etc.) along with customer feedback and/or sentiment).

– **Visual explanations** Visual explanations give the user a visualization of the explanation. Especially in application situations involving social networks, the visualization can take the form of a graph, a chart, or an image (either the entire image or specific visual highlights in the image). Visual explanations can transmit more information than textual ones while needing less cognitive processing work.

In our work, we concentrate on the use of graphics to provide explanations for the SQUIRREL Model in addition to the textual explanation. We have used Python's Plotly library to create scatter and bar plots. We have used scatter plots to represent the movies recommended by SQUIRREL for the groups in each iteration. We have used bar plots to represent the text form of our explanations visually. The next two chapters go into depth about the visualizations and explanations used in SQUIRREL and the results from our use cases.

5 Visualizations and Explanations in SQUIRREL

In this section, we go into detail about the steps we adopted to create questions and their respective explanations along with their integration into the interactive GUI framework.

5.1 Dataset

The work in Stratigi, Pitoura, and Stefanidis 2023 was unable to access datasets that showed interactions between groups and a system, such as when a group rated an item as an entity. Therefore, they artificially made groups using data from three real datasets. For our work, we have used the MovieLens Dataset F. M. Harper and Konstan 2015. Between January 1995 and March 2015, 138,5K users rated 27,3K movies on MovieLens, contributing a total of 20M ratings.

The SQUIRREL Model was assessed for a series of recommendation rounds so a time flow had to be simulated to show that some time has passed between the rounds. As a result, the recommender does not start with all the data that is accessible; rather, the data is incrementally added after each round. To accomplish this, the dataset was ordered chronologically according to the time that each rating was given. The dataset was then divided into chunks, and a fresh chunk was added to the system after each round Stratigi, Pitoura, and Stefanidis 2023.

For the MovieLens dataset, 100 groups were created each with 5 members. The model was trained on 80 of those groups as a training set, while the other 20 groups served as the test set. Based on the recommended movies we got for each test set, we have developed our why questions and their relative explanations which we discuss in detail in section 5.3.

5.2 Group Formation

To evaluate how well the SQUIRREL Model performed, real-world scenarios were simulated. It was assumed that people with comparable interests will have similar preferences when rating the same data items in situations when we lack particular information, such as when recognizing user friendships or reviewing user likes. Depending on how similar the group members were to one another, two different group types were formed based on two scenarios. First, consider a situation in which a new team member joins an established group, and their tastes may differ from the rest of the team members or a new hire who joins a seasoned project team that is mostly concerned with software development. Second, consider a scenario in which a diverse group of people gathers for a single activity, such as taking a cooking class, where every single person has specific culinary tastes. However, they are all expected to make the same recipe in class.

The similarity was calculated using Pearson Correlation Resnick et al. 1994 similarity function. The groups that were considered are:

- **4 similar – 1 dissimilar (4+1):** The group's four members are similar to one another, but the fifth member is different from the others.
- **5 dissimilar (5 Diss):** The group's individuals are different from one another.

In our work, we have developed why questions and explanations based on the recommendations generated for the (4+1) group i.e., we have 20 groups, for each group we have 15 iterations, and for each iteration, we have 10 different recommendations as produced by the SQUIRREL Model.

5.3 Why Questions and Explanations

The SQUIRREL model recommends 10 new items in each iteration and it is assumed that the group is aware of items that were recommended in previous iterations. We have created a question that plays with the genres of movies and how many times have they been recommended to the group as a whole and also to the individual users of that group as per single-user recommendation lists. The format of our question is:

Why 'selected genre' occur 'selected frequency'?.

The 'selected genre' can be any genre from the list of 19 genres and the 'selected frequency' can be any frequency from a list of 3 frequencies as shown below:

selected genre = ['Action', 'Adventure', 'Animation', 'Children', 'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film-Noir', 'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller', 'War', 'Western', 'IMAX']

selected frequency = ['a few times', 'many times', 'not at all']

It depends on the group that they want to inquire about which particular genre. For example, if a group wants to ask a question related to the 'thriller' genre and why it appears 'many times', the question becomes:

Why 'Thriller' occur 'many times'?.

Similarly, if the gene is 'children' and the frequency is 'not at all' then the question becomes:

Why 'Children' occur 'not at all'?.

The explanation for the 'Why' question is the combination of a general explanation based on single-user recommendation lists for the users of the group and a model-based explanation that takes features of the model into account which in our case are the 6 aggregation methods used in the SQUIRREL Model. For the second part of the explanation, we have summarised the aggregation methods, earlier explained in section 3.1, as follows:

- **Average** considers the group members' preferences to be of equal importance.
- **RP80** aims to minimize disagreements among group members while considering their preferences.
- **Par** weighs the average satisfaction with an item among the group members against the variance of those satisfaction scores.
- **SDAA** balances the average predicted score of an item for the group with the predicted score of the least satisfied member.
- **SIAA** considers each group member's overall satisfaction along with their disagreement with the previous round of recommendations.
- **Average+** considers items that generate the minimum possible group disagreement scores.

We will discuss in more detail how we came up with the first part of the explanation which is based on single-user recommendation lists of the users in the next section.

5.4 Integrating Explanations into the SQUIRREL Framework

The first part of our explanation makes use of single-user recommendation lists of the group of users. The single-user recommender is considered a black box for the SQUIRREL Model. We took the following steps to reach our explanation:

1. We saved the recommendations that were being generated by the SQUIRREL Model in each iteration for each of the test groups in a CSV file as shown in Figure 5.2. The information here consists of Group, Iteration, Recommended Movies, and Satisfaction Score. It is to be noted that this file contained the information for all the 20 groups and for all their 15 iterations.

2. Since our questions and explanations are based on the 'genre' of the movies we incorporated movie names (which we have used in the hover text in the visualizations, chapter 6) and the relative genre of each movie. This leads us to the file as shown in Figure 5.3. This file also contains information for all the 20 groups and for all their 15 iterations.

3. The goal was to count how many times each genre has been recommended for each group in each iteration. Therefore, we removed the vertical bar '|' between genres and separated them with 'comma' as shown in Figure 5.4. Here also there are 20 groups each having 15 iterations.

4. Then we counted the genres that were recommended in each iteration for each group which led us to Figure 5.5. Here, only one group has been shown as an example.

5. In order to get just one value for each genre, we summed the genres from all the iterations so that now with every group we have one number under each genre which indicates the total number of times it was recommended in all the previous iterations, Figure 5.6. It is to be noted here that now we also have the 'Aggregation method' in the file. This aggregation method indicates the action that was taken to get the recommendations of the latest iteration which in our case is the 15^{th} iteration. This file shows the total number of genres for all the iterations for all 20 groups. This step and all the other steps above were done for the group recommendations that we got from the SQUIRREL Model.

6. We did similar steps for the single-user recommendation lists after we had filtered them according to only those movies that were recommended to the group i.e., incorporating movie names and genres, removing the vertical bar '|' from the genres, and separating them using a comma, counting the genres for each iteration and then summing all the genres from all the iterations. In Figure 5.7 we show what a file looks like in case the group ID is `131083_131094_131105_131399_11503`. The files for all the groups were saved in a folder so that they could be easily accessed later.

```
Group,Iteration,Recommended Movies,Satisfaction Score
131659_593_131850_132298_134155,0,"['2028', '50', '1617', '2571', '912', '1221', '1358', '1704', '1213', '1089']",0.9303900392391805
131659_593_131850_132298_134155,1,"['2762', '58', '1961', '1196', '1060', '1193', '1247', '1208', '1197', '2858']",0.9203175959159837
131659_593_131850_132298_134155,2,"['1234', '4226', '3578', '1242', '1610', '5060', '296', '800', '1222', '2959']",0.9188854047063714
131659_593_131850_132298_134155,3,"['4306', '3996', '1246', '111', '923', '293', '2268', '246', '1225', '1210']",0.9106161155831473
131659_593_131850_132298_134155,4,"['904', '4993', '5952', '1250', '2329', '1304', '2542', '1288', '3114', '2000']",0.9118733821299504
131659_593_131850_132298_134155,5,"['750', '1393', '1954', '1219', '1265', '1307', '1233', '1408', '1220', '1394']",0.9026424796650042
131659_593_131850_132298_134155,6,"['3897', '1080', '223', '1374', '1206', '32', '2716', '16', '4022', '924']",0.8958530237181787
131659_593_131850_132298_134155,7,"['1214', '6539', '7153', '920', '4886', '3147', '1784', '2997', '6377', '5349']",0.8972539602850214
131659_593_131850_132298_134155,8,"['2502', '5418', '4963', '3753', '2355', '2396', '1302', '5445', '318', '2797']",0.8906858854136077
131659_593_131850_132298_134155,9,"['778', '4973', '2791', '4034', '1721', '1923', '282', '1517', '2324', '2302']",0.8833819775615865
131659_593_131850_132298_134155,10,"['2692', '527', '1580', '110', '1198', '3793', '1136', '1259', '953', '356']",0.8744113953840064
131659_593_131850_132298_134155,11,"['858', '1203', '903', '1407', '1183', '908', '1917', '3408', '4014', '260']",0.8643788328419172
131659_593_131850_132298_134155,12,"['7361', '33794', '3481', '4027', '3052', '1485', '1148', '914', '593', '628']",0.8599816652901222
131659_593_131850_132298_134155,13,"['6874', '1968', '4995', '1201', '508', '2291', '6', '3623', '36', '2890']",0.8571207770173034
131659_593_131850_132298_134155,14,"['541', '4878', '7438', '30749', '48516', '589', '1090', '457', '608', '1240']",0.8521631993102824
```

Figure 5.1 *Recommendations from the SQUIRREL Model*

```
Group,Iteration,Recommended Movies,Satisfaction Score,Titles,Genres
131659_593_131850_132298_134155,0,"[2028, 50, 1617, 2571, 912, 1221, 1358, 1704, 1213, 1089]",0.9303900392391804,"['Saving Private Ryan (1998)', 'Usual Suspects, The (1995)', 'L.A. Confidential (1997)', 'Matrix, The (1999)', 'Casablanca (1942)', 'Godfather: Part II, The (1974)', 'Sling Blade (1996)', 'Good Will Hunting (1997)', 'Goodfellas (1990)', 'Reservoir Dogs (1992)']","['Action|Drama|War', 'Crime|Mystery|Thriller', 'Crime|Film-Noir|Mystery|Thriller', 'Action|Sci-Fi|Thriller', 'Drama|Romance', 'Crime|Drama', 'Drama', 'Drama|Romance', 'Crime|Drama', 'Crime|Mystery|Thriller']"
131659_593_131850_132298_134155,1,"[2762, 58, 1961, 1196, 1060, 1193, 1247, 1208, 1197, 2858]",0.9203175959159836,"['Sixth Sense, The (1999)', 'Postman, The (Postino, Il) (1994)', 'Rain Man (1988)', 'Star Wars: Episode V - The Empire Strikes Back (1980)', 'Swingers (1996)', ""One Flew Over the Cuckoo's Nest (1975)"", 'Graduate, The (1967)', 'Apocalypse Now (1979)', 'Princess Bride, The (1987)', 'American Beauty (1999)']","['Drama|Horror|Mystery', 'Comedy|Drama|Romance', 'Drama', 'Action|Adventure|Sci-Fi', 'Comedy|Drama', 'Drama', 'Comedy|Drama|Romance', 'Action|Drama|War', 'Action|Adventure|Comedy|Fantasy|Romance', 'Comedy|Drama']"
131659_593_131850_132298_134155,2,"[1234, 4226, 3578, 1242, 1610, 5060, 296, 800, 1222, 2959]",0.9188854047063714,"['Sting, The (1973)', 'Memento (2000)', 'Gladiator (2000)', 'Glory (1989)', 'Hunt for Red October, The (1990)', 'M*A*S*H (a.k.a. MASH) (1970)', 'Pulp Fiction (1994)', 'Lone Star (1996)', 'Full Metal Jacket (1987)', 'Fight Club (1999)']","['Comedy|Crime', 'Mystery|Thriller', 'Action|Adventure|Drama', 'Drama|War', 'Action|Adventure|Thriller', 'Comedy|Drama|War', 'Comedy|Crime|Drama|Thriller', 'Drama|Mystery|Western', 'Drama|War', 'Action|Crime|Drama|Thriller']"
131659_593_131850_132298_134155,3,"[4306, 3996, 1246, 111, 923, 293, 2268, 246, 1225, 1210]",0.9106161155831471,"['Shrek (2001)', 'Crouching Tiger, Hidden Dragon (Wo hu cang long) (2000)', 'Dead Poets Society (1989)', 'Taxi Driver (1976)', 'Citizen Kane (1941)', 'Léon: The Professional (a.k.a. The Professional) (Léon) (1994)', 'Few Good Men, A (1992)', 'Hoop Dreams (1994)', 'Amadeus (1984)', 'Star Wars: Episode VI - Return of the Jedi (1983)']","['Adventure|Animation|Children|Comedy|Fantasy|Romance', 'Action|Drama|Romance', 'Drama', 'Crime|Drama|Thriller', 'Drama|Mystery', 'Action|Crime|Drama|Thriller', 'Crime|Drama|Thriller', 'Documentary', 'Drama', 'Action|Adventure|Sci-Fi']"
131659_593_131850_132298_134155,4,"[904, 4993, 5952, 1250, 2329, 1304, 2542, 1288, 3114, 2000]",0.9118733821299504,"['Rear Window (1954)', 'Lord of the Rings: The Fellowship of the Ring, The (2001)', 'Lord of the Rings: The Two Towers, The (2002)', 'Bridge on the River Kwai, The (1957)', 'American History X (1998)', 'Butch Cassidy and the Sundance Kid (1969)', 'Lock, Stock & Two Smoking Barrels (1998)', 'This Is Spinal Tap (1984)', 'Toy Story 2 (1999)', 'Lethal Weapon (1987)']","['Mystery|Thriller', 'Adventure|Fantasy', 'Adventure|Fantasy', 'Adventure|Drama|War', 'Crime|Drama', 'Action|Western', 'Comedy|Crime|Thriller', 'Comedy', 'Adventure|Animation|Children|Comedy|Fantasy', 'Action|Comedy|Crime|Drama']"
```

Figure 5.2 *Recommendations including movie names and their genres*

```
Group,Iteration,Recommended Movies,Satisfaction Score,Titles,Genres
131659_593_131850_132298_134155,0,"[2028, 50, 1617, 2571, 912, 1221, 1358, 1704, 1213, 1089]",0.9303900392391804,"['Saving Private Ryan (1998)', 'Usual Suspects, The (1995)', 'L.A. Confidential (1997)', 'Matrix, The (1999)', 'Casablanca (1942)', 'Godfather: Part II, The (1974)', 'Sling Blade (1996)', 'Good Will Hunting (1997)', 'Goodfellas (1990)', 'Reservoir Dogs (1992)']","Action,Drama,War,Crime,Mystery,Thriller,Crime,Film-Noir,Mystery,Thriller,Action,Sci-Fi,Thriller,Drama,Romance,Crime,Drama,Drama,Drama,Romance,Crime,Drama,Crime,Mystery,Thriller"
131659_593_131850_132298_134155,1,"[2762, 58, 1961, 1196, 1060, 1193, 1247, 1208, 1197, 2858]",0.9203175959159836,"['Sixth Sense, The (1999)', 'Postman, The (Postino, Il) (1994)', 'Rain Man (1988)', 'Star Wars: Episode V - The Empire Strikes Back (1980)', 'Swingers (1996)', ""One Flew Over the Cuckoo's Nest (1975)"", 'Graduate, The (1967)', 'Apocalypse Now (1979)', 'Princess Bride, The (1987)', 'American Beauty (1999)']","Drama,Horror,Mystery,Comedy,Drama,Romance,Drama,Action,Adventure,Sci-Fi,Comedy,Drama,Drama,Comedy,Drama,Romance,Action,Drama,War,Action,Adventure,Comedy,Fantasy,Romance,Comedy,Drama"
131659_593_131850_132298_134155,2,"[1234, 4226, 3578, 1242, 1610, 5060, 296, 800, 1222, 2959]",0.9188854047063714,"['Sting, The (1973)', 'Memento (2000)', 'Gladiator (2000)', 'Glory (1989)', 'Hunt for Red October, The (1990)', 'M*A*S*H (a.k.a. MASH) (1970)', 'Pulp Fiction (1994)', 'Lone Star (1996)', 'Full Metal Jacket (1987)', 'Fight Club (1999)']","Comedy,Crime,Mystery,Thriller,Action,Adventure,Drama,Drama,War,Action,Adventure,Thriller,Comedy,Drama,War,Comedy,Crime,Drama,Thriller,Drama,Mystery,Western,Drama,War,Action,Crime,Drama,Thriller"
131659_593_131850_132298_134155,3,"[4306, 3996, 1246, 111, 923, 293, 2268, 246, 1225, 1210]",0.9106161155831471,"['Shrek (2001)', 'Crouching Tiger, Hidden Dragon (Wo hu cang long) (2000)', 'Dead Poets Society (1989)', 'Taxi Driver (1976)', 'Citizen Kane (1941)', 'Léon: The Professional (a.k.a. The Professional) (Léon) (1994)', 'Few Good Men, A (1992)', 'Hoop Dreams (1994)', 'Amadeus (1984)', 'Star Wars: Episode VI - Return of the Jedi (1983)']","Adventure,Animation,Children,Comedy,Fantasy,Romance,Action,Drama,Romance,Drama,Crime,Drama,Thriller,Drama,Mystery,Action,Crime,Drama,Thriller,Crime,Drama,Thriller,Documentary,Drama,Action,Adventure,Sci-Fi"
131659_593_131850_132298_134155,4,"[904, 4993, 5952, 1250, 2329, 1304, 2542, 1288, 3114, 2000]",0.9118733821299504,"['Rear Window (1954)', 'Lord of the Rings: The Fellowship of the Ring, The (2001)', 'Lord of the Rings: The Two Towers, The (2002)', 'Bridge on the River Kwai, The (1957)', 'American History X (1998)', 'Butch Cassidy and the Sundance Kid (1969)', 'Lock, Stock & Two Smoking Barrels (1998)', 'This Is Spinal Tap (1984)', 'Toy Story 2 (1999)', 'Lethal Weapon (1987)']","Mystery,Thriller,Adventure,Fantasy,Adventure,Fantasy,Adventure,Drama,War,Crime,Drama,Action,Western,Comedy,Crime,Thriller,Comedy,Adventure,Animation,Children,Comedy,Fantasy,Action,Comedy,Crime,Drama"
```

Figure 5.3 *Genres separated with a comma*

```
Group,Iteration,Action,Adventure,Animation,Children,Comedy,Crime,Documentary,Drama,Fantasy,Film-
Noir,Horror,Musical,Mystery,Romance,Sci-Fi,Thriller,War,Western,IMAX
131659_593_131850_132298_134155,0,2,0,0,0,0,5,0,6,0,1,0,0,3,2,1,4,1,0,0
131659_593_131850_132298_134155,1,3,2,0,0,5,0,0,8,1,0,1,0,1,3,1,0,1,0,0
131659_593_131850_132298_134155,2,3,2,0,0,3,3,0,7,0,0,0,0,2,0,0,4,3,1,0
131659_593_131850_132298_134155,3,3,2,1,1,1,3,1,7,1,0,0,0,1,2,1,3,0,0,0
131659_593_131850_132298_134155,4,2,4,1,1,4,3,0,3,3,0,0,0,1,0,0,2,1,1,0
131659_593_131850_132298_134155,5,3,0,0,0,5,1,0,3,1,0,1,1,0,4,0,0,3,1,0
131659_593_131850_132298_134155,6,2,2,0,0,3,2,0,5,0,0,0,0,1,0,5,3,0,0,0
131659_593_131850_132298_134155,7,3,5,2,2,5,1,0,5,4,0,1,0,0,2,2,1,1,0,0
131659_593_131850_132298_134155,8,3,1,1,2,4,4,0,5,2,0,0,0,2,2,1,3,1,0,0
131659_593_131850_132298_134155,9,1,1,0,0,7,2,0,5,0,0,0,0,0,4,0,1,1,0,0
131659_593_131850_132298_134155,10,5,4,0,0,3,1,0,5,2,0,0,0,0,2,2,0,3,0,0
131659_593_131850_132298_134155,11,3,2,0,0,1,1,0,6,0,0,1,0,3,5,2,4,1,0,0
131659_593_131850_132298_134155,12,1,2,1,1,6,5,0,4,1,0,1,1,1,3,1,2,0,0,1
131659_593_131850_132298_134155,13,5,3,0,0,2,3,0,6,1,0,0,0,0,2,0,3,1,1,0
131659_593_131850_132298_134155,14,4,0,0,0,1,2,0,6,0,0,0,0,1,0,4,7,2,0,0
```

Figure 5.4 *Total number of genres for each group in each iteration*

```
Group,Action,Adventure,Animation,Children,Comedy,Crime,Documentary,Drama,Fantasy,Film-Noir,Horror,Musical,Mystery,Romance,Sci-
Fi,Thriller,War,Western,IMAX,Aggregation method
131083_131094_131105_131399_11503,47,42,9,13,52,29,1,64,20,3,6,6,12,24,29,34,13,2,1,SDAA
131141_131145_100_365_18680,46,48,17,19,51,32,0,64,23,4,7,9,13,30,17,35,9,3,4,SDAA
131145_131150_100_287_1583,42,39,10,14,60,30,0,68,19,2,8,5,13,34,15,30,12,2,2,SDAA
131155_87_229_334_17878,43,41,12,15,66,22,0,60,21,3,8,5,17,35,23,26,13,2,2,SDAA
131226_158_131243_131246_4727,48,41,11,13,48,34,0,66,22,3,11,5,14,24,26,33,13,3,5,SDAA
131243_131246_131274_131802_134127,39,31,6,8,50,32,1,78,15,2,8,5,11,32,15,34,14,4,1,SDAA
131253_188_202_816_135484,61,46,12,15,47,33,0,67,21,2,8,5,10,24,35,32,12,2,4,SDAA
131278_131282_399_131538_3350,42,47,6,13,53,30,0,69,20,2,6,5,12,33,24,33,14,3,2,SDAA
131279_224_131445_1717_19084,44,41,12,16,53,30,1,68,17,1,8,5,16,27,30,35,11,0,6,SDAA
131299_131316_131357_961_138159,49,39,10,16,58,33,0,59,21,1,8,7,11,28,20,44,9,5,3,SDAA
131330_261_131354_334_136862,45,42,13,16,59,26,0,65,24,2,3,5,13,36,27,32,8,2,7,SDAA
131405_394_131578_624_132161,42,38,12,14,58,35,0,63,19,2,10,5,17,34,21,41,13,2,1,SDAA
131410_354_131482_576_16961,53,45,11,15,56,28,1,64,23,1,4,5,8,36,24,43,8,4,2,SDAA
131423_354_426_132680_8482,46,39,12,16,56,25,1,69,20,2,4,7,12,32,19,36,8,5,3,SDAA
131659_593_131850_132298_134155,43,30,6,7,50,36,1,81,16,1,5,2,16,31,20,37,19,4,1,SDAA
183_131283_542_1237_132521,41,30,8,10,47,33,1,74,17,3,5,5,13,39,15,40,16,3,2,SDAA
231_131305_310_722_133072,42,40,7,11,53,34,0,66,16,2,5,7,12,28,16,38,11,4,2,SDAA
331_333_131437_509_134241,54,47,11,15,47,28,0,67,18,1,7,5,13,32,26,34,14,4,1,SDAA
352_131436_414_131544_6667,44,40,10,12,62,31,0,52,20,4,10,5,14,37,26,39,7,2,3,SDAA
93_131178_329_1294_18686,41,36,3,7,48,27,1,75,16,3,4,4,18,40,21,39,16,3,0,SDAA
```

Figure 5.5 *Total number of genres for each group along with the aggregation method used in the latest iteration i.e.,* 15^{th}

```
User ID,Action,Adventure,Animation,Children,Comedy,Crime,Documentary,Drama,Fantasy,Film-Noir,Horror,Musical,Mystery,Romance,Sci-
Fi,Thriller,War,Western,IMAX
11503,2110,1867,540,741,3362,1150,32,3071,877,76,534,462,488,1555,1136,1712,438,254,91
131083,1653,1444,385,598,2736,872,27,2802,806,62,235,314,426,1593,1053,1242,397,133,89
131094,2240,1859,354,548,2660,1019,26,2928,951,80,501,297,558,1359,1428,1769,446,172,97
131105,1501,1149,253,304,2051,1047,48,2553,518,155,322,132,462,964,893,1327,372,138,42
131399,1526,1149,224,318,1601,711,24,1634,550,43,246,230,305,877,999,1047,319,77,49
```

Figure 5.6 *Total number of genres for* 131083_131094_131105_131399_11503

```
Iteration 0
1160[2288:4.39,1148:4.34,1266:4.34,2248:4.31,260:4.27,3114:4.27,2571:4.22,1198:4.19,4993:4.17,3147:4.16,527:4.15,1200:4.15,778:4.1
2,1196:4.09,1201:4.09,6:4.08,4361:4.08,741:4.07,1291:4.07,2762:4.07,2987:4.07,1625:4.04,2858:4.04,593:4.03,903:4.02,1748:4.02,1136
:4.0,1356:4.0,920:3.98,1358:3.98,1233:3.96,5952:3.96,1204:3.95,924:3.94,1197:3.94,2324:3.94,50:3.93,1077:3.93,1193:3.93,1242:3.92,
1270:3.92,2395:3.92,3578:3.92,3996:3.92,1210:3.91,1240:3.91,858:3.9,1374:3.9,1234:3.89,1394:3.89,2115:3.89,4995:3.89,923:3.88,1199
:3.88,2501:3.88,910:3.87,1225:3.87,3897:3.87,1089:3.86,1214:3.86,1288:3.86,1221:3.85,1258:3.85,1387:3.85,1704:3.85,2804:3.85,1682:
3.84,2692:3.84,2716:3.84,919:3.83,2396:3.83,551:3.82,750:3.82,3408:3.82,3499:3.81,1527:3.8,969:3.79,1304:3.79,2028:3.78,1287:3.77,
1219:3.76,3809:3.76,196:3.75,541:3.75,2116:3.75,1:3.74,1206:3.74,904:3.73,1610:3.73,2997:3.73,805:3.72,1617:3.72,4226:3.72,2918:3.
71,4306:3.7,253:3.69,1275:3.69,2706:3.68,32:3.67,2000:3.67,1307:3.66,1965:3.66,1391:3.65,1584:3.65,1222:3.64,3510:3.64,112:3.63,12
59:3.63,3527:3.63,1923:3.62,2529:3.62,3072:3.62,5060:3.62,1265:3.6,3793:3.6,837:3.59,2353:3.59,1376:3.58,5349:3.58,2599:3.56,1784:
3.55,3160:3.55,111:3.54,1079:3.54,2194:3.54,2791:3.54,1208:3.53,2011:3.53,2959:3.53,4022:3.53,1393:3.52,852:3.5,1278:3.5,3753:3.5,
832:3.49,1127:3.48,671:3.47,1372:3.47,3471:3.47,2336:3.46,2355:3.46,17:3.45,223:3.45,1320:3.44,2947:3.44,104:3.42,1267:3.42,1645:3
.42,1721:3.42,1961:3.42,3175:3.42,1073:3.41,1968:3.41,4011:3.41,4896:3.41,1183:3.4,1653:3.4,1215:3.39,2968:3.39,235:3.38,367:3.38,
1580:3.37,1396:3.36,1917:3.35,1220:3.34,1690:3.34,2541:3.31,3418:3.31,5:3.3,2167:3.29,912:3.28,2671:3.28,1375:3.27,3354:3.27,1641:
3.26,1129:3.25,1876:3.25,1047:3.24,1573:3.24,1080:3.23,1663:3.23,329:3.21,1676:3.21,2617:3.2,3396:3.2,1097:3.19,1213:3.17,2012:3.1
7,653:3.16,4963:3.16,2406:3.15,2105:3.14,39:3.13,708:3.13,788:3.13,748:3.12,1544:3.12,783:3.11,2502:3.11,2628:3.11,1028:3.1,2640:3
.09,2700:3.08,1285:3.06,1644:3.06,2916:3.06,3977:3.05,2683:3.04,2657:3.03,673:3.01,1371:3.0,21:2.99,1909:2.96,1590:2.95,3448:2.95,
1517:2.94,1500:2.92,2985:2.92,849:2.9,2724:2.89,2321:2.87,2428:2.86,3697:2.85,2094:2.84,3176:2.84,2433:2.77,2966:2.77,3986:2.75,20
54:2.71,785:2.7,2291:2.69,2699:2.67,3033:2.66,2712:2.65,1485:2.64,2710:2.64,1407:2.62,2722:2.61,379:2.53,784:2.5,198:2.45,317:2.45
,2694:2.43,3623:2.1,2701:2.09,880:1.94]
```

Figure 5.7 *Single-user recommendation list with relevance score*

In the next chapter, we explain how we have used these files to create visualizations for the recommendations which include scatter plots for the recommendations generated by SQUIRREL and bar plots to visualize the explanations generated for our why questions.

6 Visualizations for Recommendations

In order for the group of users to understand recommendations easily, we have created visualizations at each step. We have used Python's Plotly library to create interactive plots. These plots let users zoom, pan, hover over data points for more details, and turn on and off different data series. Specifically, we have created 3 plots:

1. Group recommendations with a satisfaction score
2. Group recommendations with disagreement score
3. Single user recommendations for all the users of a group

For the first plot, we wanted to have movie names along with their production year as well as their respective relevance scores in the hover text. We already had all this information in the file in Figure 5.4 except for the relevance score. The relevance score was present in the single-user recommendation lists that were generated initially for the SQUIRREL Model as shown in Figure 5.8. This is just for one user and one iteration. However, we had these files for all the users and all the iterations. In this file, *1160* is the user ID and as we go into the square brackets we first have the item ID which is *2288*, and after the colons ':' is the relevance score which is *4.39* in the case of first reading and so on. Then we filtered the item IDs based on items that were recommended to the group the user belonged to. Since there are 5 users in each group we had 5 relevance scores for a particular item ID in each iteration. Therefore, to get just one relevance score for each iteration we calculated their average.

Figure 5.9 shows how a file looks like for a particular group, in this case `352_131436_414_131544_6667`, with the relevance score included for each iteration. In total we had 20 files, all having the same format as Figure 5.9, and were saved in a folder. These files were then combined with the file in Figure 5.4 which gave us the file as shown in Figure 5.10. Using this file we created our first plot *Group recommendations with satisfaction score* Figure 5.11.

```
Iteration,User_ID,Item_ID,Summed_Relevance_Score
0,352_131436_414_131544_6667,"[1148, 2762, 1193, 2028, 750, 858, 2858, 733, 1393, 141]","[3.614, 3.654, 3.626, 3.588,
3.5539999999999994, 3.618, 3.63, 3.872, 4.022, 3.56]"
1,352_131436_414_131544_6667,"[47, 17, 39, 36, 104, 786, 62, 805, 52, 708]","[2.7060000000000004, 3.218, 2.9240000000000004,
2.6239999999999997, 3.492, 3.4659999999999997, 2.38, 2.3920000000000003, 2.364, 2.972]"
2,352_131436_414_131544_6667,"[913, 912, 2918, 2571, 4993, 923, 1198, 1704, 1196, 112]","[3.506, 3.5160000000000005, 3.55, 3.524,
3.662, 3.5159999999999996, 3.5420000000000003, 3.5379999999999994, 3.498, 3.5119999999999996]"
3,352_131436_414_131544_6667,"[6, 628, 661, 380, 1407, 736, 3, 376, 788, 7]","[3.1079999999999997, 3.97, 2.9139999999999997,
2.886, 2.876, 2.626, 3.308, 3.274, 3.2660000000000005, 2.758]"
4,352_131436_414_131544_6667,"[1206, 4226, 2692, 5952, 2324, 4306, 919, 1136, 1199, 3996]","[3.444, 3.476, 3.44, 4.454,
3.5439999999999996, 3.4739999999999993, 3.508, 3.4579999999999997, 3.4580000000000006, 4.252000000000001]"
5,352_131436_414_131544_6667,"[50, 4886, 7153, 25, 1061, 2012, 1485, 95, 595, 349]","[2.756, 4.232, 4.531999999999999, 2.388,
2.45, 3.0740000000000003, 2.698, 2.6079999999999997, 2.298, 2.246]"
6,352_131436_414_131544_6667,"[4973, 904, 908, 1221, 2997, 920, 1214, 924, 1617, 1197]","[3.5159999999999996, 3.531999999999999,
3.502, 3.386, 3.4339999999999997, 3.4199999999999995, 3.3880000000000003, 3.3619999999999997, 3.408, 3.396]"
7,352_131436_414_131544_6667,"[1234, 1250, 1219, 293, 2959, 1213, 10, 1917, 58, 5]","[3.386, 3.414, 3.3920000000000003, 3.382,
3.351999999999999, 3.34, 2.16, 2.478, 1.714, 1.998]"
8,352_131436_414_131544_6667,"[903, 1210, 3578, 6539, 6377, 1222, 1240, 1208, 3114, 1252]","[3.53, 3.3280000000000003,
3.2840000000000003, 3.3520000000000003, 4.088, 3.276, 3.2960000000000003, 3.29, 3.264, 3.2840000000000003]"
9,352_131436_414_131544_6667,"[1258, 1089, 1036, 1200, 1278, 4995, 2396, 1961, 6711, 6874]","[3.418, 3.2840000000000003, 3.258,
3.2700000000000005, 3.2739999999999996, 3.276, 3.26, 3.2619999999999996, 3.276, 3.2620000000000005]"
10,352_131436_414_131544_6667,"[1358, 1097, 1230, 1291, 1307, 597, 1784, 2699, 593, 14]","[3.398000000000001, 3.276, 3.248, 3.218,
3.25, 2.79, 3.2260000000000004, 3.196, 0.8619999999999999, 1.362]"
11,352_131436_414_131544_6667,"[2797, 2502, 7361, 1220, 5989, 1304, 4963, 1270, 3408, 33794]","[3.222, 3.1759999999999997, 3.4,
3.17, 3.192, 3.182, 3.2059999999999995, 3.1740000000000004, 3.158, 3.254]"
12,352_131436_414_131544_6667,"[1265, 3948, 1500, 1610, 1080, 1732, 1584, 145, 342, 724]","[3.162, 3.162, 3.136,
3.1740000000000004, 3.1719999999999997, 3.142, 3.1420000000000003, 1.566, 0.782, 1.2739999999999998]"
13,352_131436_414_131544_6667,"[5349, 1247, 3481, 3751, 2916, 2355, 2599, 1127, 1527, 5378]","[3.074, 3.086, 3.082, 3.114,
3.1079999999999997, 3.096, 3.07, 3.0700000000000003, 3.064, 3.605999999999999]"
14,352_131436_414_131544_6667,"[1282, 2791, 1748, 1376, 2011, 1079, 1387, 2716, 1580, 4027]","[3.028, 2.996, 3.158,
3.0279999999999996, 3.014, 2.994, 3.008, 3.018, 2.986, 2.984]"
```

Figure 6.1 *Relevance score for 352_131436_414_131544_6667*

```
Group,Iteration,Satisfaction Score,Recommended Movies,Relevance score,Titles,Genres
131659_593_131850_132298_134155,0,0.9303900392391804,"[2028, 2571, 50, 1221, 1089, 912, 1213, 1358, 1617, 1704]","
[4.7299999999999995, 4.522, 4.65, 4.508, 4.4719999999999995, 4.514, 4.4799999999999995, 4.504, 4.526000000000001, 4.488]","
['Saving Private Ryan (1998)', 'Matrix, The (1999)', 'Usual Suspects, The (1995)', 'Godfather: Part II, The (1974)', 'Reservoir
Dogs (1992)', 'Casablanca (1942)', 'Goodfellas (1990)', 'Sling Blade (1996)', 'L.A. Confidential (1997)', 'Good Will Hunting
(1997)']","['Action|Drama|War', 'Action|Sci-Fi|Thriller', 'Crime|Mystery|Thriller', 'Crime|Drama', 'Crime|Mystery|Thriller',
'Drama|Romance', 'Crime|Drama', 'Drama', 'Crime|Film-Noir|Mystery|Thriller', 'Drama|Romance']"
131659_593_131850_132298_134155,1,0.9203175959159836,"[58, 1961, 1196, 2762, 1247, 1060, 1197, 1193, 1208, 2858]","[4.496,
4.465999999999999, 4.418, 4.43, 4.324, 4.444000000000001, 4.286, 4.384, 4.366, 4.39]","['Postman, The (Postino, Il) (1994)', 'Rain
Man (1988)', 'Star Wars: Episode V - The Empire Strikes Back (1980)', 'Sixth Sense, The (1999)', 'Graduate, The (1967)', 'Swingers
(1996)', 'Princess Bride, The (1987)', ""One Flew Over the Cuckoo's Nest (1975)"", 'Apocalypse Now (1979)', 'American Beauty
(1999)']","['Comedy|Drama|Romance', 'Drama', 'Action|Adventure|Sci-Fi', 'Drama|Horror|Mystery', 'Comedy|Drama|Romance',
'Comedy|Drama', 'Action|Adventure|Comedy|Fantasy|Romance', 'Drama', 'Action|Drama|War', 'Comedy|Drama']"
131659_593_131850_132298_134155,2,0.9188854047063714,"[4226, 1234, 1610, 3578, 1242, 800, 2959, 296, 5060, 1222]","[4.7, 4.654,
4.328, 4.554, 4.351999999999999, 4.298, 4.29, 4.336, 4.31, 4.287999999999999]","['Memento (2000)', 'Sting, The (1973)', 'Hunt for
Red October, The (1990)', 'Gladiator (2000)', 'Glory (1989)', 'Lone Star (1996)', 'Fight Club (1999)', 'Pulp Fiction (1994)',
'M*A*S*H (a.k.a. MASH) (1970)', 'Full Metal Jacket (1987)']","['Mystery|Thriller', 'Comedy|Crime', 'Action|Adventure|Thriller',
'Action|Adventure|Drama', 'Drama|War', 'Drama|Mystery|Western', 'Action|Crime|Drama|Thriller', 'Comedy|Crime|Drama|Thriller',
'Comedy|Drama|War', 'Drama|War']"
```

Figure 6.2 *Recommended movies including the satisfaction and relevance score*

This plot showed all the movies that were recommended in each iteration for all the test groups. Here, the x-axis is the *Iteration* and the y-axis is the *Satisfaction score*. From the drop-down button, the group can choose a particular group ID for which they want to see the recommendations. To make the plot visually attractive, we have used different colors for different groups. Also, as we hover over any point the hover text shows the movie with the production year and the relevance score. Figure 5.11 in our case shows the movies recommended in the 13^{th} iteration for the group 131083_131094_131105_131399_11503.

In a similar way, we created the plot 'Group recommendations with disagreement score' (Figure 5.12) by replacing the satisfaction score with the disagreement score that we got from the SQUIRREL Model for each iteration of each group (Figure 5.13).

Figure 6.3 *Plot for Group recommendations with satisfaction score*

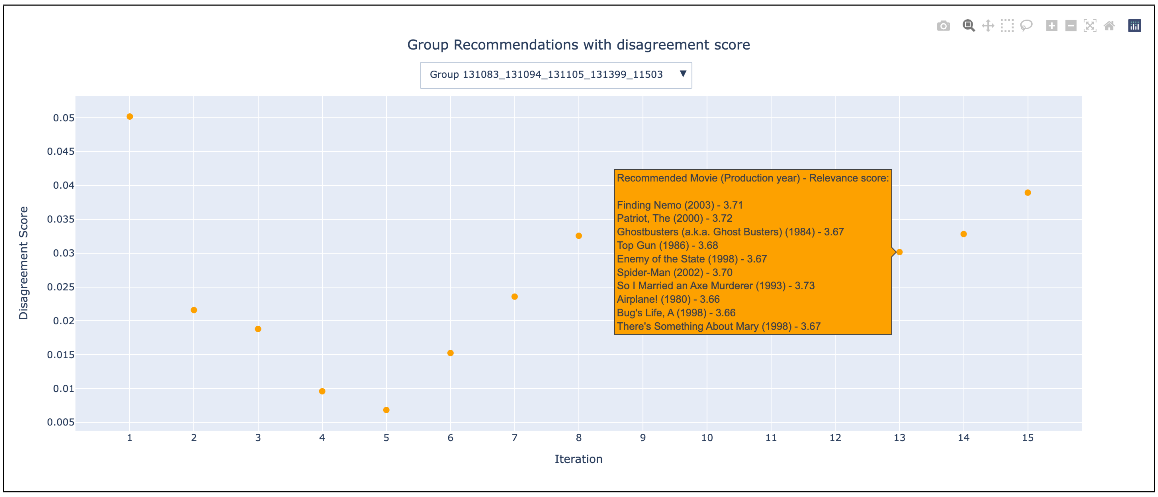

Figure 6.4 *Plot for Group recommendations with disagreement score*

The last plot (Figure 5.14) is created when the user asks a question and gets an explanation based on single-user recommendation lists of the group. The plot is actually a visualization of Figure 5.7. Since every group has 5 members each, 5 subplots are created which are actually bar plots and as you hover over a bar it shows the genre name along with its count for that user. This will help the group understand the explanation in a better way. In the next chapter, we go into detail about the setup of our experiment, analyzing the results and integrating them with the GUI framework.

```
Group ID,Iteration,Disagreement Score,Recommended Movies,Titles,Genres
131659_593_131850_132298_134155,0,0.05021870295674569,"[2028, 50, 1617, 2571, 912, 1221, 1358, 1704, 1213, 1089]","['Saving Private Ryan (1998)', 'Usual Suspects, The (1995)', 'L.A. Confidential (1997)', 'Matrix, The (1999)', 'Casablanca (1942)', 'Godfather: Part II, The (1974)', 'Sling Blade (1996)', 'Good Will Hunting (1997)', 'Goodfellas (1990)', 'Reservoir Dogs (1992)']","['Action|Drama|War', 'Crime|Mystery|Thriller', 'Crime|Film-Noir|Mystery|Thriller', 'Action|Sci-Fi|Thriller', 'Drama|Romance', 'Crime|Drama', 'Drama', 'Drama|Romance', 'Crime|Drama', 'Crime|Mystery|Thriller']"
131659_593_131850_132298_134155,1,0.05319312882490146,"[2762, 58, 1961, 1196, 1060, 1193, 1247, 1208, 1197, 2858]","['Sixth Sense, The (1999)', 'Postman, The (Postino, Il) (1994)', 'Rain Man (1988)', 'Star Wars: Episode V - The Empire Strikes Back (1980)', 'Swingers (1996)', ""One Flew Over the Cuckoo's Nest (1975)"", 'Graduate, The (1967)', 'Apocalypse Now (1979)', 'Princess Bride, The (1987)', 'American Beauty (1999)']","['Drama|Horror|Mystery', 'Comedy|Drama|Romance', 'Drama', 'Action|Adventure|Sci-Fi', 'Comedy|Drama', 'Drama', 'Comedy|Drama|Romance', 'Action|Drama|War', 'Action|Adventure|Comedy|Fantasy|Romance', 'Comedy|Drama']"
131659_593_131850_132298_134155,2,0.047798451674357034,"[1234, 4226, 3578, 1242, 1610, 5060, 296, 800, 1222, 2959]","['Sting, The (1973)', 'Memento (2000)', 'Gladiator (2000)', 'Glory (1989)', 'Hunt for Red October, The (1990)', 'M*A*S*H (a.k.a. MASH) (1970)', 'Pulp Fiction (1994)', 'Lone Star (1996)', 'Full Metal Jacket (1987)', 'Fight Club (1999)']","['Comedy|Crime', 'Mystery|Thriller', 'Action|Adventure|Drama', 'Drama|War', 'Action|Adventure|Thriller', 'Comedy|Drama|War', 'Comedy|Crime|Drama|Thriller', 'Drama|Mystery|Western', 'Drama|War', 'Action|Crime|Drama|Thriller']"
```

Figure 6.5 *Recommended movies including the disagreement and relevance score*

Figure 6.6 *Genre count for each user in the group*

7 Movies Explanations in SQUIRREL

7.1 Setup

Figure 7.1 shows how an explanation is formed in the case of 'few times', 'many times', and 'not at all'. We have chosen *40* as our threshold in case of group recommendations because 18 of the 20 groups have been recommended 4 or more genres around 40 times. Therefore, if the genre count is less than 40 it comes under the category of 'a few times' and if it is greater or equal to 40, it comes under the category of 'many times'. For the category 'not at all', the genre count should be in the range of 0 to 10 with 0 and 10 included. The reason for choosing this range is that we have a huge dataset with 138,5K users. A genre count of 10 or less than 10 almost means it is a zero. We can't go with just zero because there are iterations where the group has been recommended a particular genre at least 1 time and out of all 15 iterations if it has just been recommended once then it's equivalent to zero.

In the case of single-user recommendations, we have considered *2,000* as a threshold to come up with the explanations for 'a few times' and 'many times' respectively. Due to the fact that almost all movies are multi-genre, we have such a high number i.e., *2,000*. Consequently, calculating a sum for a particular genre produces large numbers due to the multiple recommendations. The reason behind choosing 2000 as a threshold is because in the cases where a genre has been recommended 40 or more times then there are at least 3 users out of 5 who have been recommended that genre more than 2000 times. Therefore, if there are 3 or more users in a group that have been recommended a particular genre less than or equal to 2,000 times but greater than 1,000 times, then that becomes the reason why the group sees that genre, less frequently. On the other hand, if 3 or more users have been recommended a genre more than 2,000 times then that becomes the case the group sees that genre more frequently.

For the case of 'not at all', we have considered that if a genre count is less than or equal to 1,000 for 3 or more users of the group then that becomes the reason why a group hasn't been recommended that genre at all. All three of these explanations can be explained by considering any genre and the three frequencies one at a time. Consider the genre is *Adventure*. We will have 3 cases:

- **Few times**

  Question: *Why Adventure occurs a few times?*
  Answer: *The genre Adventure is less likely to be enjoyed by 4 members of this group, therefore it occurs less frequently.*

More details: This explanation uses the SDAA action that balances the average predicted score of an item for the group with the predicted score of the least satisfied member.

- **Many times**

  Question: *Why Adventure occurs many times?*
  Answer: *The genre Adventure is more likely to be enjoyed by 4 members of this group, therefore it occurs less frequently.*
  More details: This explanation uses the SDAA action that balances the average predicted score of an item for the group with the predicted score of the least satisfied member.

- **Not at all**

  Question: *Why Adventure occurs a few times?*
  Answer: *The genre Adventure is not likely to be enjoyed at all by 4 members of this group.*
  More details: This explanation uses the SDAA action that balances the average predicted score of an item for the group with the predicted score of the least satisfied member.

In the examples above we have assumed 4 members in all three scenarios however they can be 3, 4, or 5 depending on the situation. It can also be seen that we have added the second part of the explanation as explained in section 5.3 based on the action that was chosen in the last iteration which in our case is SDAA. Also, it is to be noted that the threshold values can change if the data changes. In the next section, we have discussed how we incorporated the visualizations and the explanations in the GUI (Graphical User Interface) framework.

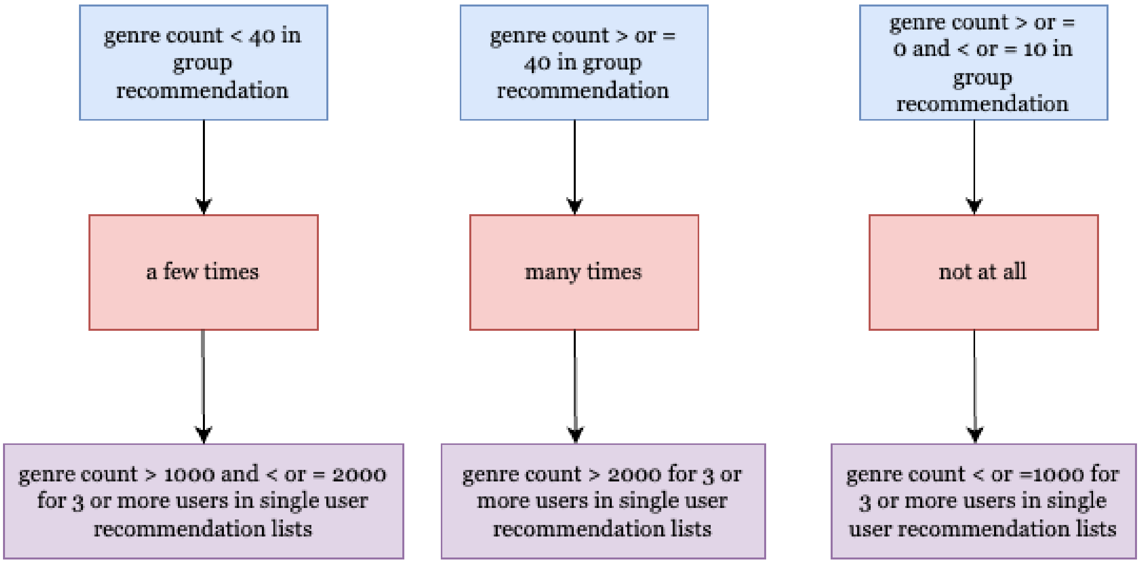

Figure 7.1 *Logic behind the formation of an explanation*

7.2 Creating an INTERACTIVE GUI

We have incorporated all the visualizations that we discussed in chapter 6 along with the questions and explanations as discussed in sections 5.3 and 5.4 in the GUI framework. The group first gets a dialog box titled 'Group Recommendations' with two clickable links and the question 'Are you satisfied with the recommendations?' along with two push buttons as 'Yes' and 'I need clarification'. The first link 'Click here to see the recommendations with satisfaction' opens Figure 6.3 in a browser window so that the group can see the movies that were recommended to them in all the iterations along with the relevance score. Likewise, the second link 'Click here to see the recommendations with disagreement' allows the group to get to know about the disagreement scores between the users of that group in each iteration Figure 6.4.

These visualizations help users better understand the recommendations and based on that they answer the question 'Are you satisfied with the recommendations?'. In case the group is satisfied they press 'Yes' and a dialog box with the message 'Yayy! That's great!' appears (Figure 7.3) with a push button 'Yes'. When 'Yes' is pressed this dialog box along with the Group Recommendations (Figure 7.2) closes. This indicates that the group has no further questions to ask. In case the group presses 'I need clarification', then they get a dialog box titled 'Need some clarifications' (Figure 7.4) where they input their group ID and then select a genre and frequency from the lists as mentioned in section 5.3. After that, they click on 'Submit' which extends this dialog box and now includes the 'Why question' with the explanation and a link to the plot which opens Figure 6.6 in a browser window. The group would be able to see visually what all members of the group were recommended in the single-user recommendation lists on which the explanation is based.

Here, the figure will be according to the group.

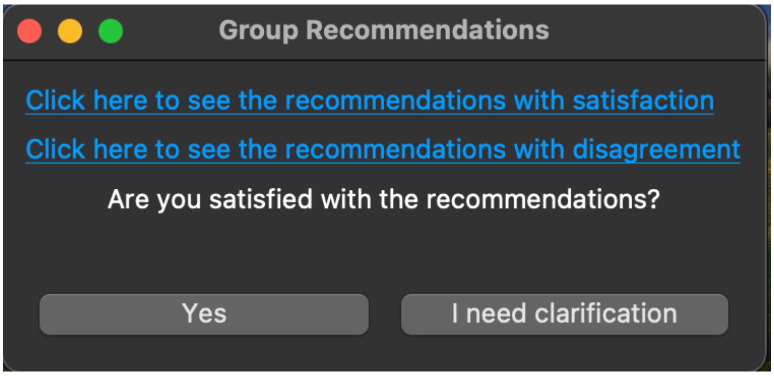

Figure 7.2 *Dialog box to display group recommendations*

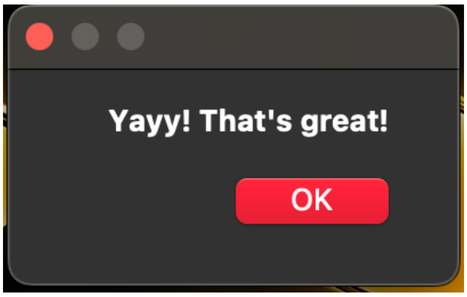

Figure 7.3 *Group is satisfied with the recommendations*

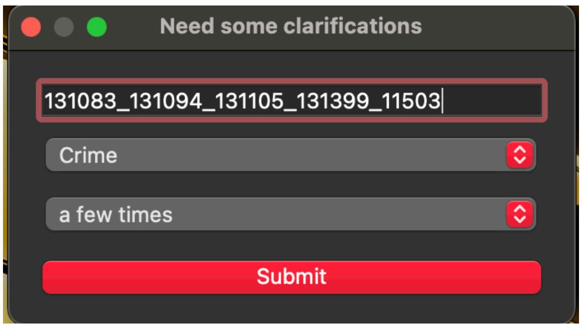

Figure 7.4 *Group is not satisfied with the recommendations*

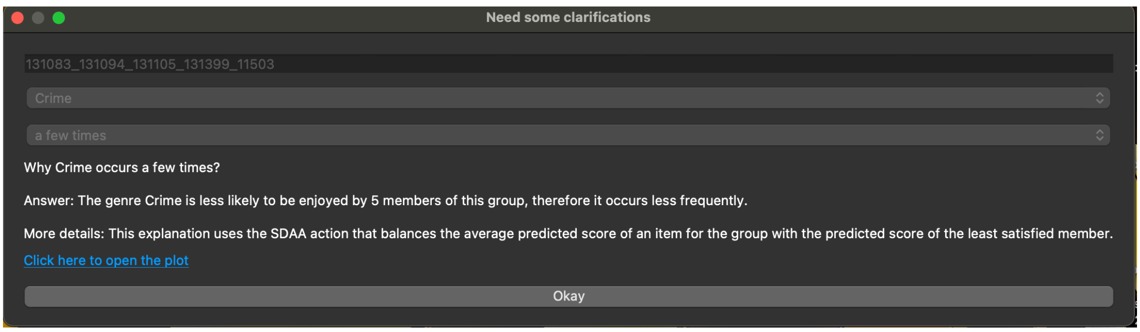

Figure 7.5 *Explanation for 'a few times' for 131083_131094_131105_131399_11503*

In the next section, we share the results from the 3 use cases by taking one group into account and analyzing the explanations we get for all the 3 options of frequency.

7.3 Results

Consider the group `131083_131094_131105_131399_11503` and the genre as *Crime*. Figure 5.6 shows the number of times each genre was recommended to all the groups in the test set. In the case of our group, *Crime* has been recommended 29 times which is less than 40 which makes it fall under the category of 'a few times'. Next, we see how many times *Crime* was recommended to all the users of the group individually which is shown in Figure 5.7. As per our threshold defined in section 7.1, here all 5 members have been recommended *Crime* less than 2,000 times. Therefore, we get the question and explanation as shown in Figure 7.5.

To analyze the result for the category 'many times', consider the genre *Comedy* for the same group Figure 7.6. As seen in Figure 5.6, *Comedy* has been recommended 52 times for this group. Now if we look at Figure 5.7, we see that 4 out of 5 members have been recommended this genre more than 2,000 times. Therefore, we get the question and explanation as shown in Figure 7.7.

Lastly for the category 'not at all', consider the genre *Animation* Figure 7.8. It has been recommended 9 times as seen in Figure 5.6. Now if we look at Figure 5.7, we see that all 5 members of the group have been recommended this genre less than 1,000 times. Therefore, we get the question and explanation as shown in Figure 7.9.

For all three cases, the action is 'SDAA' as seen in Figure 5.6 therefore the second part of the explanation remains constant for all the cases. This analysis was for a particular group. However, it works in the same way for all the groups.

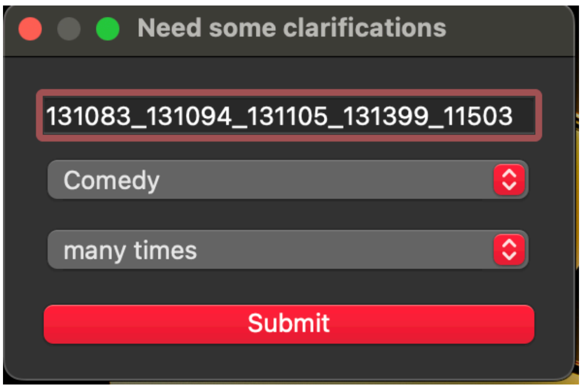

Figure 7.6 *Why 'Comedy' occur many times?*

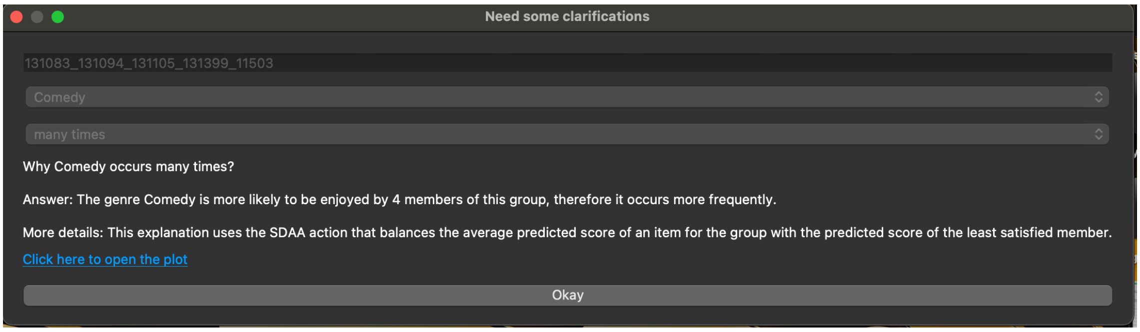

Figure 7.7 *Explanation for 'many times' for 131083_131094_131105_131399_11503*

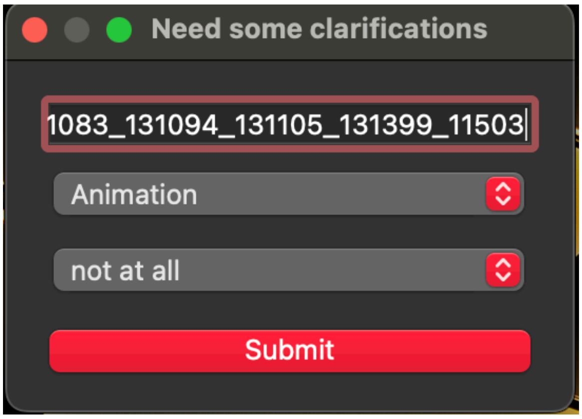

Figure 7.8 *Why 'Animation' occur not at all?*

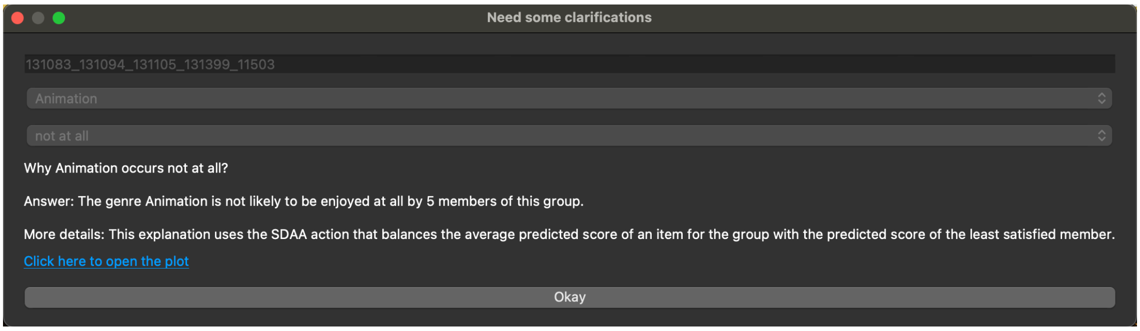

Figure 7.9 *Explanation for 'not at all' for 131083_131094_131105_131399_11503*

8 Conclusion

In this work, we have proposed 'Why' questions and explanations for sequential group recommendations via reinforcement learning. This work is an extension of the SQUIRREL framework. We used a real-world dataset, 20M MovieLens. We have created visualizations for the recommendations generated by SQUIRREL in the form of scatter plots. The scatter plots are between iterations and satisfaction and disagreement scores respectively. The hover text indicates the 10 movies recommended in each iteration along with their relevance scores. We developed 3 questions on the 'genre' of the movies and explored the single-user and group recommendation lists to come up with the explanations.

We used 3 cases to explore our questions and their related explanations. Additionally, we used bar plots to display the explanation in a visual form for better understanding. These visualizations and explanations were incorporated into the GUI framework where a group can ask questions multiple times.

In our future work, we want to explore 'Why' questions related to other features of the dataset such as movie names, ratings, or year of production. Also, in the second part of the explanation, a more specific description can be given such as defining actual values for the average predicted score in place of the text 'average predicted score' and the user ID in place of the text 'least satisfied user'. This will require getting all the values from the SQUIRREL framework and incorporating them with the explanations.

References

Afsar, Mehdi, Trafford Crump, and Behrouz Far (June 2022). "Reinforcement Learning based Recommender Systems: A Survey". In: *ACM Computing Surveys* 55. DOI: 10.1145/3543846.

Amershi, Saleema et al. (Dec. 2014). "Power to the People: The Role of Humans in Interactive Machine Learning". In: *Ai Magazine* 35, pp. 105–120. DOI: 10.1609/aimag.v35i4.2513.

Balog, Krisztian, Filip Radlinski, and Shushan Arakelyan (July 2019). "Transparent, Scrutable and Explainable User Models for Personalized Recommendation". In: pp. 265–274. ISBN: 978-1-4503-6172-9. DOI: 10.1145/3331184.3331211.

Baltrunas, Linas, Tadas Makcinskas, and Francesco Ricci (Sept. 2010). "Group recommendations with rank aggregation and collaborative filtering". In: pp. 119–126. DOI: 10.1145/1864708.1864733.

Berkovsky, Shlomo and Jill Freyne (2010). "Group-Based Recipe Recommendations: Analysis of Data Aggregation Strategies". In: *Proceedings of the Fourth ACM Conference on Recommender Systems*. RecSys '10. Barcelona, Spain: Association for Computing Machinery, pp. 111–118. ISBN: 9781605589060. DOI: 10.1145/1864708.1864732. URL: https://doi.org/10.1145/1864708.1864732.

Bidoit-Tollu, Nicole, Melanie Herschel, and Katerina Tzompanaki (June 2014). "Immutably Answering Why-Not Questions for Equivalent Conjunctive Queries". In: vol. 20. DOI: 10.3166/isi.20.5.27-52.

Bilgic, Mustafa and Raymond Mooney (Jan. 2005). "Explaining Recommendations: Satisfaction vs. Promotion". In.

Borges, Rodrigo and Kostas Stefanidis (Nov. 2019). "Enhancing Long Term Fairness in Recommendations with Variational Autoencoders". In: pp. 95–102. DOI: 10.1145/3297662.3365798.

— (2020). "On Measuring Popularity Bias in Collaborative Filtering Data". English. In: *Proceedings of the Workshops of the EDBT/ICDT 2020 Joint Conference*. Vol. 2578. CEUR Workshop Proceedings. jufoid=53269; EDBT/ICDT Workshops ; Conference date: 01-01-2020. CEUR-WS.org.

— (Mar. 2021). "On mitigating popularity bias in recommendations via variational autoencoders". In: pp. 1383–1389. DOI: 10.1145/3412841.3442123.

— (2022). "F2VAE: A Framework for Mitigating User Unfairness in Recommendation Systems". In: *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*. SAC '22. Virtual Event: Association for Computing Machinery, pp. 1391–1398. ISBN: 9781450387132. DOI: 10.1145/3477314.3507152. URL: https://doi.org/10.1145/3477314.3507152.

Cao, Da et al. (2018). “Attentive Group Recommendation”. In: *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. SIGIR ’18. Ann Arbor, MI, USA: ACM, pp. 645–654. ISBN: 978-1-4503-5657-2. DOI: 10.1145/3209978.3209998. URL: http://doi.acm.org/10.1145/3209978.3209998.

Chang, Shuo, Franklin Harper, and Loren Terveen (Sept. 2016a). “Crowd-Based Personalized Natural Language Explanations for Recommendations”. In: pp. 175–182. DOI: 10.1145/2959100.2959153.

— (Sept. 2016b). “Crowd-Based Personalized Natural Language Explanations for Recommendations”. In: pp. 175–182. DOI: 10.1145/2959100.2959153.

Chatti, Mohamed Amine, Mouadh Guesmi, and Arham Muslim (2023). *Visualization for Recommendation Explainability: A Survey and New Perspectives*. arXiv: 2305.11755 [cs.IR].

Cheng, Chen et al. (2013). “Where You like to Go next: Successive Point-of-Interest Recommendation”. In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*. IJCAI ’13. Beijing, China: AAAI Press, pp. 2605–2611. ISBN: 9781577356332.

Dudley, John J. and Per Ola Kristensson (June 2018). “A Review of User Interface Design for Interactive Machine Learning”. In: *ACM Trans. Interact. Intell. Syst.* 8.2. ISSN: 2160-6455. DOI: 10.1145/3185517. URL: https://doi.org/10.1145/3185517.

Eiband, Malin et al. (Mar. 2018). “Bringing Transparency Design into Practice”. In: pp. 211–223. DOI: 10.1145/3172944.3172961.

Garcin, Florent, Christos Dimitrakakis, and Boi Faltings (Mar. 2013). “Personalized News Recommendation with Context Trees”. In: *RecSys 2013 - Proceedings of the 7th ACM Conference on Recommender Systems*. DOI: 10.1145/2507157.2507166.

Gedikli, Fatih, Dietmar Jannach, and Mouzhi Ge (Apr. 2014). “How Should I Explain? A Comparison of Different Explanation Types for Recommender Systems”. In: *Int. J. Hum.-Comput. Stud.* 72.4, pp. 367–382. ISSN: 1071-5819. DOI: 10.1016/j.ijhcs.2013.12.007. URL: https://doi.org/10.1016/j.ijhcs.2013.12.007.

Hansen, Casper et al. (Sept. 2020). “Contextual and Sequential User Embeddings for Large-Scale Music Recommendation”. In: pp. 53–62. DOI: 10.1145/3383313.3412248.

Harambam, Jaron et al. (2019). “Designing for the Better by Taking Users into Account: A Qualitative Evaluation of User Control Mechanisms in (News) Recommender Systems”. In: *Proceedings of the 13th ACM Conference on Recommender Systems*. RecSys ’19. Copenhagen, Denmark: Association for Computing

Machinery, pp. 69–77. ISBN: 9781450362436. DOI: 10.1145/3298689.3347014. URL: https://doi.org/10.1145/3298689.3347014.

Hariri, Negar, Bamshad Mobasher, and Robin Burke (2012). “Context-Aware Music Recommendation Based on Latenttopic Sequential Patterns”. In: *Proceedings of the Sixth ACM Conference on Recommender Systems*. RecSys ’12. Dublin, Ireland: Association for Computing Machinery, pp. 131–138. ISBN: 9781450312707. DOI: 10.1145/2365952.2365979. URL: https://doi.org/10.1145/2365952.2365979.

Harper, F. Maxwell and Joseph A. Konstan (Dec. 2015). “The MovieLens Datasets: History and Context”. In: *ACM Trans. Interact. Intell. Syst.* 5.4. ISSN: 2160-6455. DOI: 10.1145/2827872. URL: https://doi.org/10.1145/2827872.

He, Chen, Denis Parra, and Katrien Verbert (Mar. 2016). “Interactive recommender systems: A survey of the state of the art and future research challenges and opportunities”. In: *Expert Systems with Applications* 56. DOI: 10.1016/j.eswa.2016.02.013.

Herlocker, Jonathan L., Joseph A. Konstan, and John Riedl (2000). “Explaining Collaborative Filtering Recommendations”. In: *Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work*. CSCW ’00. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, pp. 241–250. ISBN: 1581132220. DOI: 10.1145/358916.358995. URL: https://doi.org/10.1145/358916.358995.

Hidasi, Balázs et al. (2016). “Parallel Recurrent Neural Network Architectures for Feature-Rich Session-Based Recommendations”. In: *Proceedings of the 10th ACM Conference on Recommender Systems*. RecSys ’16. Boston, Massachusetts, USA: Association for Computing Machinery, pp. 241–248. ISBN: 9781450340359. DOI: 10.1145/2959100.2959167. URL: https://doi.org/10.1145/2959100.2959167.

Hijikata, Yoshinori, Yuki Kai, and Shogo Nishida (2012). “The Relation between User Intervention and User Satisfaction for Information Recommendation”. In: *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. SAC ’12. Trento, Italy: Association for Computing Machinery, pp. 2002–2007. ISBN: 9781450308571. DOI: 10.1145/2245276.2232109. URL: https://doi.org/10.1145/2245276.2232109.

Huang, Liwei et al. (Feb. 2021). “A deep reinforcement learning based long-term recommender system”. In: *Knowledge-Based Systems* 213, p. 106706. DOI: 10.1016/j.knosys.2020.106706.

Jameson, Anthony and Barry Smyth (Jan. 2007). “Recommendation to Groups”. In: vol. 4321, pp. 596–627. ISBN: 978-3-540-72078-2. DOI: 10.1007/978-3-540-72079-9_20.

Jannach, Dietmar, Lukas Lerche, and Michael Jugovac (2015). "Adaptation and Evaluation of Recommendations for Short-Term Shopping Goals". In: *Proceedings of the 9th ACM Conference on Recommender Systems*. RecSys '15. Vienna, Austria: Association for Computing Machinery, pp. 211–218. ISBN: 9781450336925. DOI: 10.1145/2792838.2800176. URL: https://doi.org/10.1145/2792838.2800176.

Jannach, Dietmar, Sidra Naveed, and Michael Jugovac (Feb. 2017). "User Control in Recommender Systems: Overview and Interaction Challenges". In: vol. 278, pp. 21–33. ISBN: 978-3-319-53675-0. DOI: 10.1007/978-3-319-53676-7_2.

Jiang, Liu, Shixia Liu, and Changjian Chen (Apr. 2019). "Recent Research Advances on Interactive Machine Learning". In: *J. Vis.* 22.2, pp. 401–417. ISSN: 1343-8875. DOI: 10.1007/s12650-018-0531-1. URL: https://doi.org/10.1007/s12650-018-0531-1.

Jin, Yucheng, Nava Tintarev, and Katrien Verbert (Sept. 2018). "Effects of personal characteristics on music recommender systems with different levels of controllability". In: pp. 13–21. DOI: 10.1145/3240323.3240358.

Kaffes, Vassilis, Dimitris Sacharidis, and Giorgos Giannopoulos (2021). "Model-Agnostic Counterfactual Explanations of Recommendations". In: *Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. UMAP '21. Utrecht, Netherlands: Association for Computing Machinery, pp. 280–285. ISBN: 9781450383660. DOI: 10.1145/3450613.3456846. URL: https://doi.org/10.1145/3450613.3456846.

Kim, Jae Kyeong et al. (June 2010). "A group recommendation system for online communities". English. In: *International Journal of Information Management* 30.3. Funding Information: This research is supported by the Ubiquitous Computing and Network (UCN) Project, Knowledge and Economy Frontier RD Program of the Ministry of Knowledge Economy (MKE) in Korea, and a result of subproject UCN 09C1-T2-10M., pp. 212–219. ISSN: 0268-4012. DOI: 10.1016/j.ijinfomgt.2009.09.006.

Kompan, Michal and Maria Bielikova (Jan. 2014). "Group Recommendations: Survey and Perspectives". In: *Computing and Informatics* 33, pp. 1–31.

Kouki, Pigi et al. (Mar. 2019). "Personalized explanations for hybrid recommender systems". In: pp. 379–390. DOI: 10.1145/3301275.3302306.

Lian, Defu, Vincent Zheng, and Xing Xie (May 2013). "Collaborative filtering meets next check-in location prediction". In: pp. 231–232. DOI: 10.1145/2487788.2487907.

Lin, Xiao et al. (Aug. 2017). "Fairness-aware group recommendation with pareto-efficiency". English (US). In: *RecSys 2017 - Proceedings of the 11th ACM Conference on Recommender Systems*. RecSys 2017 - Proceedings of the 11th ACM

Conference on Recommender Systems. Funding Information: This work is supported by Natural Science Foundation of China (Grant No. 61532011, 61672311) and National Key Basic Research Program (2015CB358700). Part of the work is sponsored by the Center for Intelligent Information Retrieval. Meanwhile it is supported in part by the Hong Kong Scholars Program. Publisher Copyright: © 2017 ACM.; 11th ACM Conference on Recommender Systems, RecSys 2017 ; Conference date: 27-08-2017 Through 31-08-2017. Association for Computing Machinery, Inc, pp. 107–115. DOI: 10.1145/3109859.3109887.

Liu, Qiang et al. (2016). "Predicting the next Location: A Recurrent Model with Spatial and Temporal Contexts". In: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*. AAAI'16. Phoenix, Arizona: AAAI Press, pp. 194–200.

Liu, Shixia et al. (Feb. 2017). "Towards Better Analysis of Machine Learning Models: A Visual Analytics Perspective". In: *Visual Informatics* 1. DOI: 10.1016/j.visinf.2017.01.006.

Masthoff, Judith (Feb. 2004). "Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers". In: *User Model. User-Adapt. Interact.* 14, pp. 37–85. DOI: 10.1023/B:USER.0000010138.79319.fd.

— (Jan. 2011). "Group Recommender Systems: Combining Individual Models". In: pp. 677–702. ISBN: 978-0-387-85819-7. DOI: 10.1007/978-0-387-85820-3_21.

Moling, Omar, Linas Baltrunas, and Francesco Ricci (2012). "Optimal Radio Channel Recommendations with Explicit and Implicit Feedback". In: *Proceedings of the Sixth ACM Conference on Recommender Systems*. RecSys '12. Dublin, Ireland: Association for Computing Machinery, pp. 75–82. ISBN: 9781450312707. DOI: 10.1145/2365952.2365971. URL: https://doi.org/10.1145/2365952.2365971.

Morisawa, Shun and Hayato Yamana (Nov. 2021). "Faithful Post-hoc Explanation of Recommendation Using Optimally Selected Features". In: pp. 159–173. ISBN: 978-3-030-89384-2. DOI: 10.1007/978-3-030-89385-9_10.

Ngo, Thao, Johannes Kunkel, and Jürgen Ziegler (July 2020). "Exploring Mental Models for Transparent and Controllable Recommender Systems: A Qualitative Study". In: pp. 183–191. DOI: 10.1145/3340631.3394841.

Ntoutsi, Eirini et al. (Oct. 2012). "Fast Group Recommendations by Applying User Clustering". In: ISBN: 978-3-642-34001-7. DOI: 10.1007/978-3-642-34002-4_10.

Nunes, Ingrid and Dietmar Jannach (Dec. 2017). "A systematic review and taxonomy of explanations in decision support and recommender systems". In: *User Modeling and User-Adapted Interaction* 27. DOI: 10.1007/s11257-017-9195-0.

Piliponyte, A, Francesco Ricci, and Julian Paul Jakob Koschwitz (2013). "Sequential Music Recommendations for Groups by Balancing User Satisfaction". eng. In: vol. 997. CEUR Workshop Proceedings. CEUR-WS.

Pitoura, Evaggelia, Kostas Stefanidis, and Georgia Koutrika (Oct. 2021). "Fairness in Rankings and Recommendations: An Overview". In: *The VLDB Journal* 31.3, pp. 431–458. ISSN: 1066-8888. DOI: 10.1007/s00778-021-00697-y. URL: https://doi.org/10.1007/s00778-021-00697-y.

Pu, Pearl, Li Chen, and Rong Hu (Oct. 2012). "Evaluating recommender systems from the user's perspective: Survey of the state of the art". In: *User Modeling and User-Adapted Interaction* 22. DOI: 10.1007/s11257-011-9115-7.

Qin, Dong et al. (Nov. 2018). "Dynamic Connection-Based Social Group Recommendation". In: *IEEE Transactions on Knowledge and Data Engineering* PP, pp. 1–1. DOI: 10.1109/TKDE.2018.2879658.

Qin, Jiarui et al. (Jan. 2020). "Sequential Recommendation with Dual Side Neighbor-based Collaborative Relation Modeling". In: pp. 465–473. DOI: 10.1145/3336191.3371842.

Quadrana, Massimo, Paolo Cremonesi, and Dietmar Jannach (July 2018). "Sequence-Aware Recommender Systems". In: *ACM Comput. Surv.* 51.4. ISSN: 0360-0300. DOI: 10.1145/3190616. URL: https://doi.org/10.1145/3190616.

Quadrana, Massimo, Alexandros Karatzoglou, et al. (2017). "Personalizing Session-Based Recommendations with Hierarchical Recurrent Neural Networks". In: *Proceedings of the Eleventh ACM Conference on Recommender Systems*. RecSys '17. Como, Italy: Association for Computing Machinery, pp. 130–137. ISBN: 9781450346528. DOI: 10.1145/3109859.3109896. URL: https://doi.org/10.1145/3109859.3109896.

Resnick, Paul et al. (1994). "GroupLens: An Open Architecture for Collaborative Filtering of Netnews". In: *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*. CSCW '94. Chapel Hill, North Carolina, USA: Association for Computing Machinery, pp. 175–186. ISBN: 0897916891. DOI: 10.1145/192844.192905. URL: https://doi.org/10.1145/192844.192905.

Sacharidis, Dimitris (2019). "Top-N Group Recommendations with Fairness". In: *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*. SAC '19. Limassol, Cyprus: Association for Computing Machinery, pp. 1663–1670. ISBN: 9781450359337. DOI: 10.1145/3297280.3297442. URL: https://doi.org/10.1145/3297280.3297442.

Salehi-Abari, Amirali and Craig Boutilier (2015). "Preference-Oriented Social Networks: Group Recommendation and Inference". In: *Proceedings of the 9th ACM Conference on Recommender Systems*. RecSys '15. Vienna, Austria: Association for Computing Machinery, pp. 35–42. ISBN: 9781450336925. DOI: 10.1145/2792838.2800190. URL: https://doi.org/10.1145/2792838.2800190.

Shani, Guy, Ronen I. Brafman, and David Heckerman (2015). *An MDP-based Recommender System*. arXiv: 1301.0600 [cs.LG].

Shneiderman, Ben (Oct. 2020). "Bridging the Gap Between Ethics and Practice: Guidelines for Reliable, Safe, and Trustworthy Human-Centered AI Systems". In: *ACM Trans. Interact. Intell. Syst.* 10.4. ISSN: 2160-6455. DOI: 10.1145/3419764. URL: https://doi.org/10.1145/3419764.

— (2022). "Human-Centered AI: Ensuring Human Control While Increasing Automation". In: *Proceedings of the 5th Workshop on Human Factors in Hypertext*. HUMAN '22. Barcelona, Spain: Association for Computing Machinery. ISBN: 9781450394017. DOI: 10.1145/3538882.3542790. URL: https://doi.org/10.1145/3538882.3542790.

Spinner, Thilo et al. (2020). "explAIner: A Visual Analytics Framework for Interactive and Explainable Machine Learning". In: *IEEE Transactions on Visualization and Computer Graphics* 26.1, pp. 1064–1074. DOI: 10.1109/TVCG.2019.2934629.

Stratigi, Maria, Jyrki Nummenmaa, et al. (Mar. 2020). "Fair sequential group recommendations". In: pp. 1443–1452. DOI: 10.1145/3341105.3375766.

Stratigi, Maria, Evaggelia Pitoura, Jyrki Nummenmaa, et al. (2021). "Sequential group recommendations based on satisfaction and disagreement scores". In: *Journal of Intelligent Information Systems* 58, pp. 227–254. URL: https://api.semanticscholar.org/CorpusID:237012585.

Stratigi, Maria, Evaggelia Pitoura, and Kostas Stefanidis (Feb. 2023). "SQUIRREL: A Framework for Sequential Group Recommendations through Reinforcement Learning". In: *Inf. Syst.* 112.C. ISSN: 0306-4379. DOI: 10.1016/j.is.2022.102128. URL: https://doi.org/10.1016/j.is.2022.102128.

Stratigi, Maria, Aikaterini Tzompanaki, and Kostas Stefanidis (Oct. 2020). "Why-Not Questions & Explanations for Collaborative Filtering". In: *WISE*. Amsterdam, Netherlands. DOI: 10.1007/978-3-030-62008-0_21. URL: https://hal.science/hal-03172873.

Taghipour, Nima, Ahmad Kardan, and Saeed Ghidary (Oct. 2007). "Usage-based web recommendations: a reinforcement learning approach". In: pp. 113–120. DOI: 10.1145/1297231.1297250.

Tao, Yiyi et al. (2019). "The FacT: Taming Latent Factor Models for Explainability with Factorization Trees". In: *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR'19. Paris, France: Association for Computing Machinery, pp. 295–304. ISBN:

9781450361729. DOI: 10.1145/3331184.3331244. URL: https://doi.org/10.1145/3331184.3331244.

Tintarev, Nava and Judith Masthoff (May 2007). "A Survey of Explanations in Recommender Systems". In: pp. 801–810. ISBN: 978-1-4244-0832-0. DOI: 10.1109/ICDEW.2007.4401070.

Tsai, Chun-Hua and Peter Brusilovsky (2017). "Providing Control and Transparency in a Social Recommender System for Academic Conferences". In: *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. UMAP '17. Bratislava, Slovakia: Association for Computing Machinery, pp. 313–317. ISBN: 9781450346351. DOI: 10.1145/3079628.3079701. URL: https://doi.org/10.1145/3079628.3079701.

Tsai, Chunhua and Peter Brusilovsky (July 2021). "The effects of controllability and explainability in a social recommender system". In: *User Modeling and User-Adapted Interaction* 31, pp. 1–37. DOI: 10.1007/s11257-020-09281-5.

Verma, Sahil et al. (2022). "RecXplainer: Post-Hoc Attribute-Based Explanations for Recommender Systems". In: *Progress and Challenges in Building Trustworthy Embodied AI*. URL: https://openreview.net/forum?id=1-rCv9xeK2U.

Vig, Jesse, Shilad Sen, and John Riedl (2009). "Tagsplanations: Explaining Recommendations Using Tags". In: *Proceedings of the 14th International Conference on Intelligent User Interfaces*. IUI '09. Sanibel Island, Florida, USA: Association for Computing Machinery, pp. 47–56. ISBN: 9781605581682. DOI: 10.1145/1502650.1502661. URL: https://doi.org/10.1145/1502650.1502661.

Vinh Tran, Lucas et al. (July 2019). "Interact and Decide: Medley of Sub-Attention Networks for Effective Group Recommendation". In: pp. 255–264. ISBN: 978-1-4503-6172-9. DOI: 10.1145/3331184.3331251.

Wang, Wen et al. (July 2020). "Group-Aware Long- and Short-Term Graph Representation Learning for Sequential Group Recommendation". In: pp. 1449–1458. DOI: 10.1145/3397271.3401136.

Yin, Hongzhi et al. (2019). "Social Influence-Based Group Representation Learning for Group Recommendation". In: *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 566–577. DOI: 10.1109/ICDE.2019.00057.

Yu, Cong, Laks Lakshmanan, and Sihem Amer-Yahia (Mar. 2009). "Recommendation Diversification Using Explanations". In: pp. 1299–1302. DOI: 10.1109/ICDE.2009.225.

Yu, Zhiwen et al. (Mar. 2006). "TV Program Recommendation for Multiple Viewers Based on user Profile Merging". In: *User Model. User-Adapt. Interact.* 16, pp. 63–82. DOI: 10.1007/s11257-006-9005-6.

Yuan, Quan, Gao Cong, and Chin-Yew Lin (2014). "COM: A Generative Model for Group Recommendation". In: *Proceedings of the 20th ACM SIGKDD Inter-*

national Conference on Knowledge Discovery and Data Mining. KDD '14. New York, New York, USA: Association for Computing Machinery, pp. 163–172. ISBN: 9781450329569. DOI: 10.1145/2623330.2623616. URL: https://doi.org/10.1145/2623330.2623616.

Yuyan, Zhang, Su Xiayao, and Liu Yong (Oct. 2019). "A Novel Movie Recommendation System Based on Deep Reinforcement Learning with Prioritized Experience Replay". In: pp. 1496–1500. DOI: 10.1109/ICCT46805.2019.8947012.

Zhang, Yongfeng and Xu Chen (2020).

Zhao, Ruijing, Izak Benbasat, and Hasan Cavusoglu (2019). "Do Users Always Want to Know More? Investigating the Relationship between System Transparency and Users' Trust in Advice-Giving Systems". In: *European Conference on Information Systems*. URL: https://api.semanticscholar.org/CorpusID:174803722.

Zhao, Xiangyu et al. (2017). *Deep Reinforcement Learning for List-wise Recommendations*. arXiv: 1801.00209 [cs.LG].

Zheng, Guanjie et al. (Apr. 2018). "DRN: A Deep Reinforcement Learning Framework for News Recommendation". In: pp. 167–176. ISBN: 978-1-4503-5639-8. DOI: 10.1145/3178876.3185994.