Project Work

The project work may be completed in pairs. Each pair submits one only output, and both students in the group are expected to understand, be able to explain, and be able to modify the implementation. Send your groups to konstantinos.stefanidis@tuni.fi before 5.00pm, **November 6, 2024**.

Note: For each 1-week delay in your submissions, you lose 10% of your score.
Any programming language for your assignment is acceptable. Please explain any assumptions you made.

Part I

Due: **November 9, 2024**
Points: **25/100**
Submit: A file upload at Moodle

The goal of the first part of the project is to implement a user-based collaborative filtering approach and then exploit it for producing group recommendations.

(a) Download the **MovieLens 100K** rating dataset from https://grouplens.org/datasets/movielens/ (the *small* dataset *recommended for education and development*). Read the dataset, display the first few rows to understand it, and display the count of ratings (rows) in the dataset to be sure that you download it correctly.

(b) Implement the user-based collaborative filtering approach, using the Pearson correlation function for computing similarities between users (**4 points**), and (c) the prediction function presented in class for predicting movies scores (**4 points**).

(d) Design and implement a new similarity function for computing similarities between users. Explain why this similarity function is useful for the collaborative filtering approach. Hint: Exploiting ideas from related works are highly encouraged. **4 points**

(e) Use the user-based collaborative filtering approach to produce group recommendations. Specifically, first compute the movies recommendations for each user in the group, and then aggregate the lists of the individual users, to produce a single list of movies for the group. You will implement two well established aggregation methods for producing the group recommendations.

The first aggregation approach is the *average* method. The main idea behind this approach is that all members are considered equals. So, the rating of an item for a group of users will be given be averaging the scores of an item across all group members. **4 points**

The second aggregation method is the *least misery* method, where one member can act as a veto for the rest of the group. In this case, the rating of an item for a group of users is computed as the minimum score assigned to that item in all group members recommendations. **3 points**

Use the **MovieLens 100K** rating dataset for checking the correctness of your solutions.

(f) Define a way for counting the disagreements between the users in a group and propose a method that takes disagreements into account when computing suggestions for the group. Implement your method and explain why it is useful when producing group recommendations. Prepare also a short presentation (about 5 slides) to show how your method works. **6 points**

Submit your files at Moodle the latest on **NOVEMBER 9, 2024 (before 11.00pm)**. Some instructions on how to run your codes are necessary.

Part II

Due: **November 16, 2024**
Points: **25/100**
Submit: Upload files at Moodle

Motivated by the sequential methods we discussed in class, our goal here is to design (**7 points**) and implement (**7 points**) a new method for producing sequential group recommendations. Also, provide detailed explanations and clarifications about why the method you propose works well for the case of sequential group recommendations (**6 points**). Prepare also a short presentation (about 5 slides) to show how your method works (**5 points**).

Hint: There is no need to design a method from scratch. For the needs of this assignment, you can suggest simple modifications of the existing approach, e.g., by proposing and using alternatives for group aggregation that ensure good results for the group.

Use the **MovieLens 100K** rating dataset for checking the correctness of your solutions.

Submit your files at Moodle the latest on **November 16, 2024 (before 11.00pm)**. Some instructions on how to run your codes are necessary.

Part III

Due: **November 23, 2024**
Points: **25/100**
Submit: Upload files at Moodle

Diversity in recommendations ensures that users are exposed to a wide range of content. Moreover, diversity in sequential group recommendation systems is particularly important since it helps prevent the reinforcement of biases that may arise from repeated exposure to similar types of content. For example, in a system where recommendations are based solely on past behaviour, there is a risk that users will only be exposed to content that aligns with their previous interactions, thus limiting their exposure to new movies.

In Part III of the project, we would like to promote diversity in recommender systems to offer a wide range of content and ensure that this content is relevant and engaging for all members of the group. Motivated by the diversification methods we discussed in class, design (**7 points**) and implement (**7 points**) a new method for producing diverse sequential group recommendations. Explain why this method works well for the case of sequential group recommendations (**6 points**). Prepare also a short presentation (about 5 slides) to show how your method works (**5 points**).

Submit your files at Moodle the latest on **November 23, 2024 (before 11.00pm)**. Some instructions on how to run your codes are necessary.

Part IV

Due: **November 30, 2024**
Points: **25/100**
Submit: Upload files at Moodle

Counterfactual explanations systematically remove some items from the user's previous interactions and then call on the recommendation system again to check if the item they want to provide an explanation for is removed from the user's suggestions. If the item is successfully removed, then the set of items responsible for that alteration is the explanation provided to the user. This would result in an explanation of the form: "If you had not liked item A, then item B would not have been suggested," where A is a set of items removed from the user's feedback, and B is the item the user wanted an explanation for.

In this part of the project, the goal is to produce counterfactual explanations for a group of users. Specifically, design (**10 points**) and implement (**10 points**) a method that generates counterfactual explanations that adhere to some characteristics to be more in tune with group recommendations. For example, an explanation that only consists of items interacted by a single user is undesirable since it would single out that user to the rest of the group. Ideally, we would like explanations that consist of items most users have interacted with to make the explanation fairer. In this context, it means that no user should have changed their preferences for the group, but the group as a whole is responsible for the suggestions provided by the system. Prepare also a short presentation (about 5 slides) to show how your method works (**5 points**).

Submit your files at Moodle the latest on **November 30, 2024 (before 11.00pm)**. Some instructions on how to run your codes are necessary.