Recommender Systems DATA.ML.360-2025-2026

Sequential Group Recommendations with Reinforcement Learning

Maria Stratigi

Kostas Stefanidis

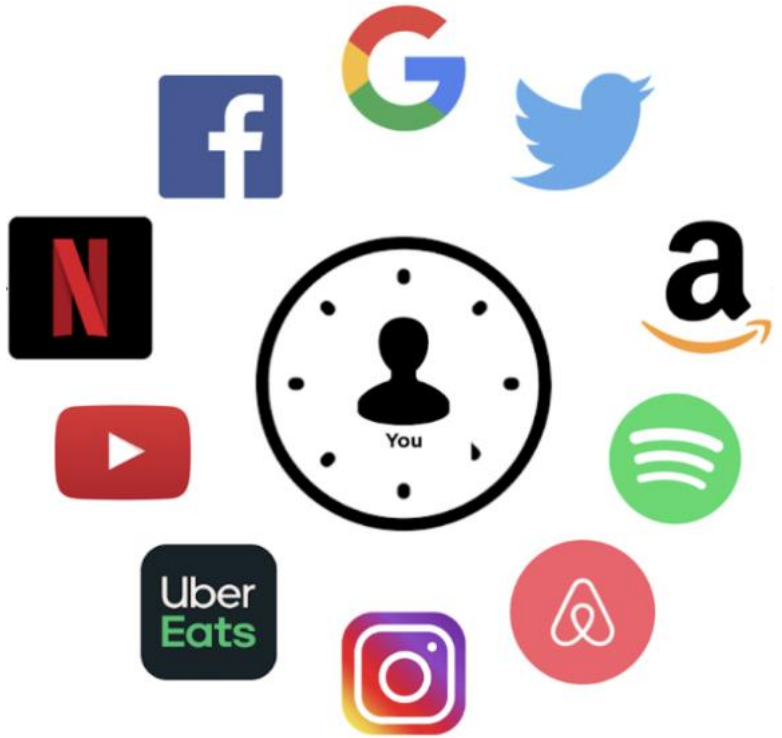
Recommendation Systems

Why use Recommendation Systems?

Recommendations have been integrated in many of the services available to users in recent times

They provide the users with a list of items that are most relevant to them.

Recommenders are employed to make the user experience better and smoother: listening to music, health information, queries, jobs, etc.



How to Use Recommendation Models

Most of the recommendation models can be divided into two categories.

Memory-based

- They use a user-ratings matrix that contains the ratings each user has given to items to find similar users to a target user, by applying a similarity function. The suggestions are made by examining the ratings of similar users.

Model-based

- First construct a model to represent the behavior of the users and, therefore, to predict their ratings.

Group Recommendation

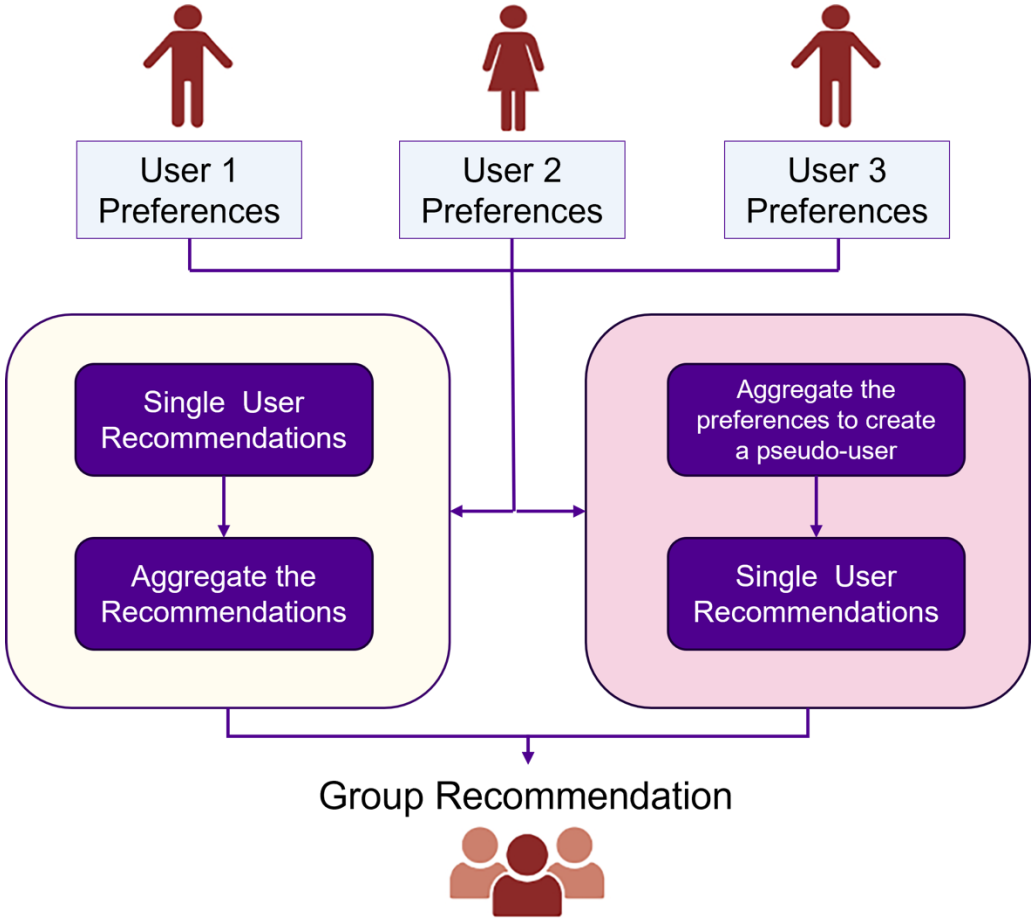
Especially due to the expansion of social media: **group recommendations**

- Instead of a single user requesting recommendations, it is more likely that a group will make a query

E.g.: Friends want to see a movie / Each friend has his/her own likes and dislikes / The system needs to balance them and offer to the group items that are somehow relevant to all members

Two main ways for doing so:

- Create a pseudo-user by combing the preferences of each group member, and apply a standard recommendation method
- Apply a recommendation method to each member individually, and aggregate the separate lists into one for the group



Aggregation in Recommendations

Recommend the item that has the best relevance to the group

Fairness: A important criterion that one can consider during the aggregation stage

Average

All group members are considered equals.

Calculate the average score across all the group members preference score for an item

Drawback: Easy to ignore the minority.

Assume a group of 3 - 2 of them are quite similar to each other, while the 3rd is not. By using the average method the opinion of the last user is lost.

Minimum-Least Misery

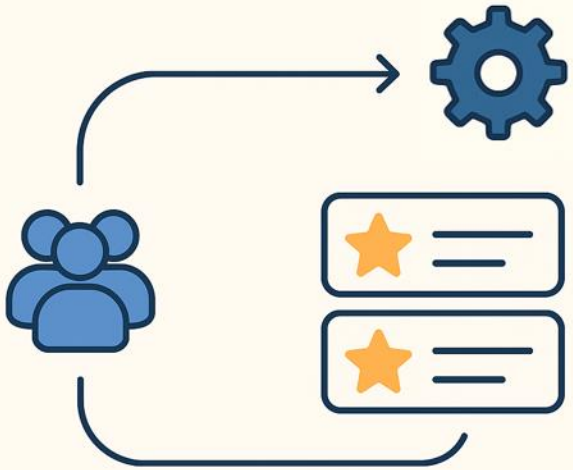
The user with the minimum preference score will act as veto to the rest of the group.

Drawback: Consider the opinion of one group member - “ignore” the others.

The system in most cases will recommend an item that is acceptable for all members, but it will be an item with a low preference score

We need to combine the equality of the average method and the inclusivity of the least misery method

Multiple Iterations

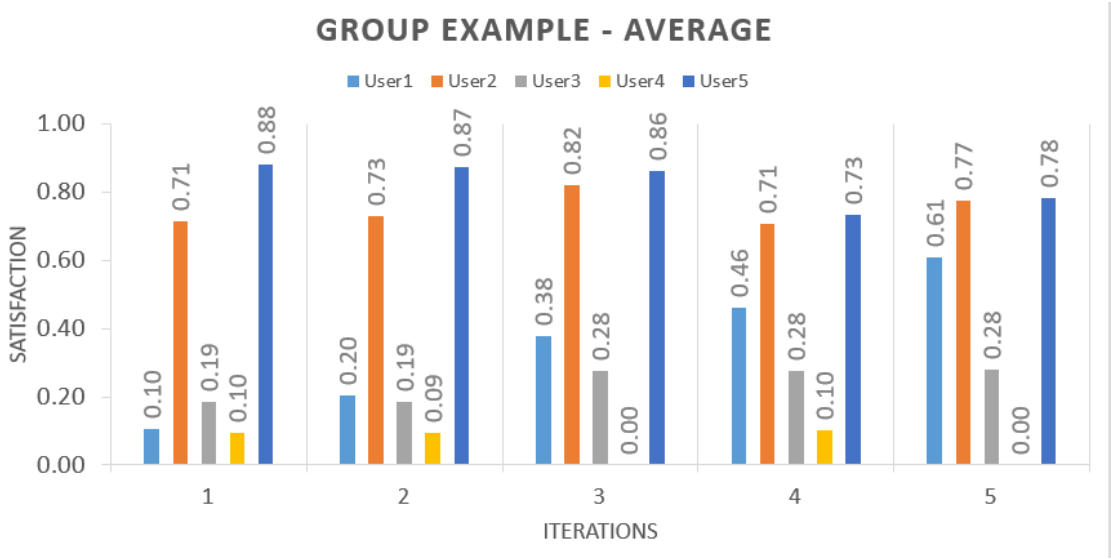


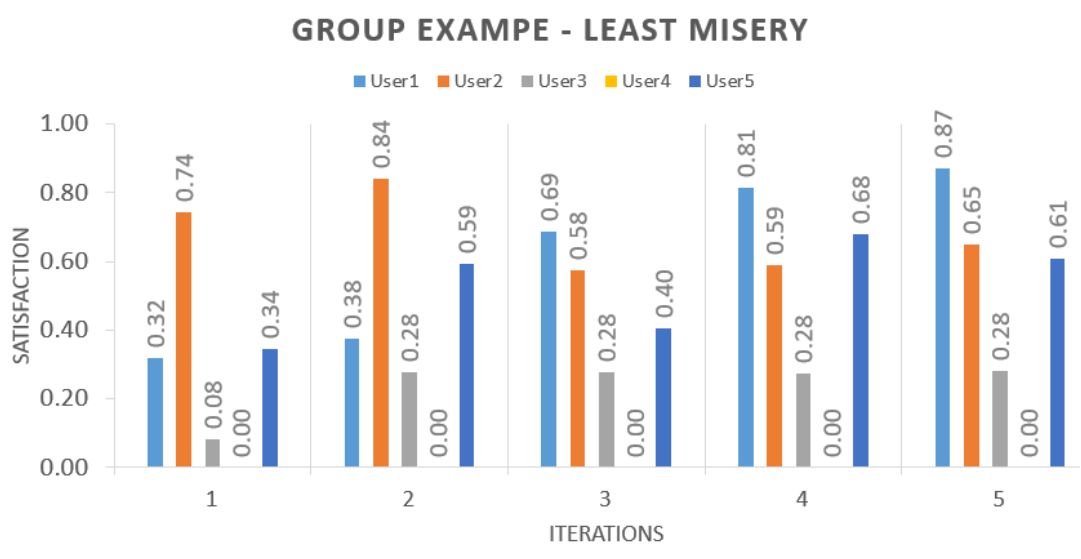
We imply that each time a group is using the system is distinct from the previous ones - this is not a realistic scenario!

- A group interacts with the system multiple times. The system should recommend different items at each iteration and retain knowledge from past interactions to keep the output diverse

We want a system that has a memory and can adjust its recommendations accordingly

Why standard recommendation techniques do not work?





Both the average and least misery approaches fail.

Average: the out-flier user is never satisfied.
Least misery: the system recommends movies not highly interested by anyone in the group

Goal

Exploit here the idea of multiple iterations of recommendations: If a user is not satisfied in a round, then he/she was either satisfied in a previous or will be satisfied in the next round

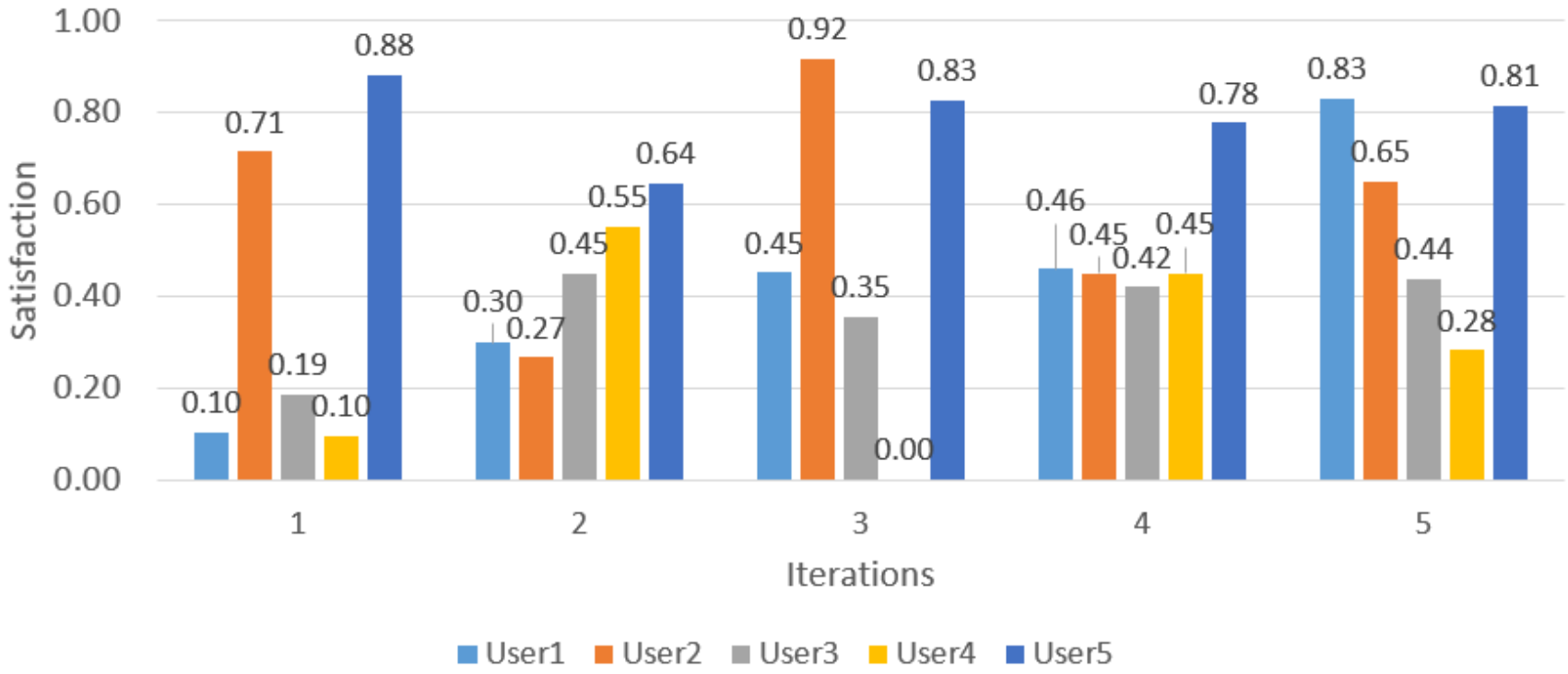
- Satisfaction: estimate the happiness of each group member after each iteration of the system
- Disagreement: estimate the disagreement among the group members.

Apply these scores during the aggregation phase as weights on the individual preference scores of the group members

- The users not satisfied in the previous round will have more weight in the next
- A member is not continuously biased against (as in average), since the calculation of the satisfaction scores is done dynamically at each iteration
- We consider the opinions of the entire group (something that least misery was lacking)

Sequential Group Recommendation Example

Same example using SDAA



Users Satisfaction / Disagreement

User Satisfaction

$$sat(u, Gr_j) = \frac{\sum_{i \in Gr_j} p_j(u,i)}{\sum_{i \in A_u^j} p_j(u,i)}$$

Sum of u's rating in the group recommendation list

Sum of u's ratings in their preference list

User Disagreement

$$userDis(u, G, Gr_j) = max_{\forall u' \in G} sat(u', Gr_j) - sat(u, Gr_j)$$

To examine the effectiveness of our group recommender through a series of iterations we introduce the notion of satisfaction per iteration.

Our goal is to directly compare the user's satisfaction from the group recommendation list with the ideal case for that user

Note: These metrics are still static. Calculated just for one iteration.

Dynamic Group Recommendation Strategy

Across multiple recommendation rounds, user satisfaction and group disagreement scores change.

☺ When **users' satisfaction is high,** use the **Average** method – treat all users equally.

☹ When **disagreement is high** use the **Least Misery** – focus on the least satisfied member.

We need a **dynamic model** that adapts the recommendation strategy **based on satisfaction and disagreement scores**.

Machine Learning in Recommendation Systems

A small overview

Roles of Machine Learning in RS

Data Analysis

- Analysis of user data user behavior, preferences, historical interactions, demographic information.
- Helps to understand user preferences.

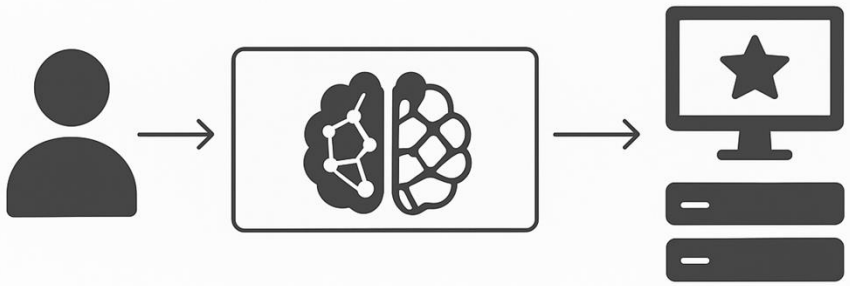
Pattern Recognition

- Identification of patterns and relationships in user-item interactions.
- Example: recognize that users who liked similar movies tend to have similar preferences.

User Profiling

- Creation of user profiles by identifying and categorizing user preferences.
- Profiles help in understanding individual user behavior and what they are likely to be interested in.

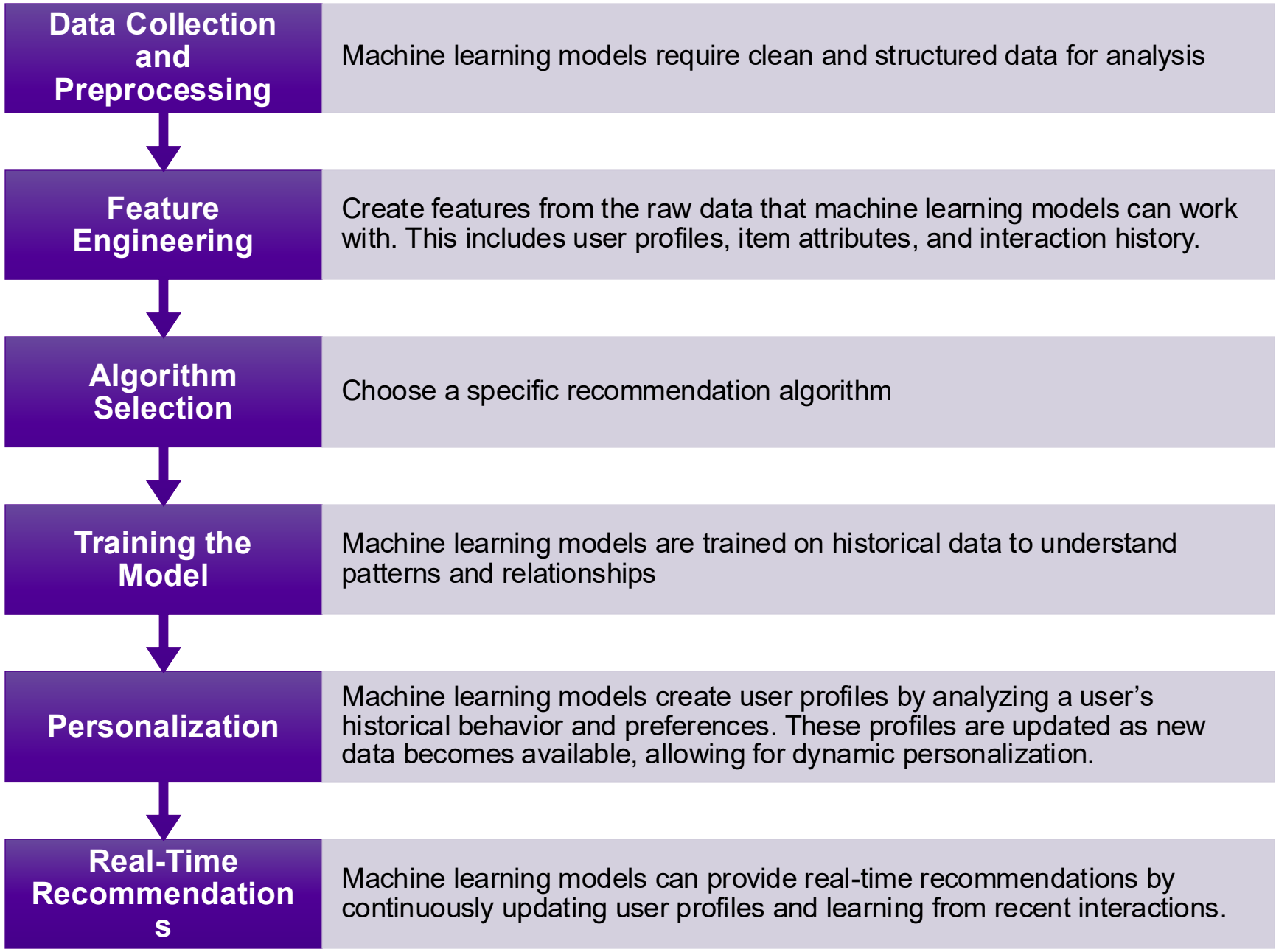
Machine Learning in RS



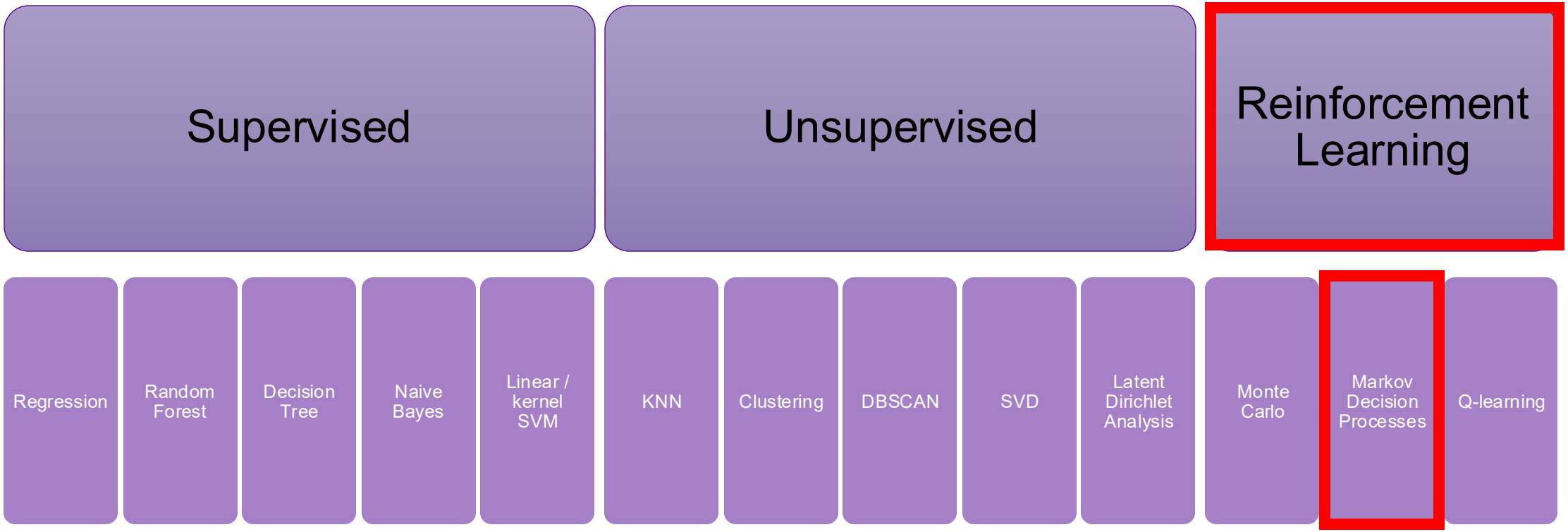
Content-Based Filtering: Machine learning is used to analyze item attributes and content, such as keywords, genres, or product descriptions.

Collaborative Filtering: Machine learning models learn from historical user-item interactions to identify user-user or item-item similarities, helping to make recommendations based on the behavior of similar users.

How to use ML in RS

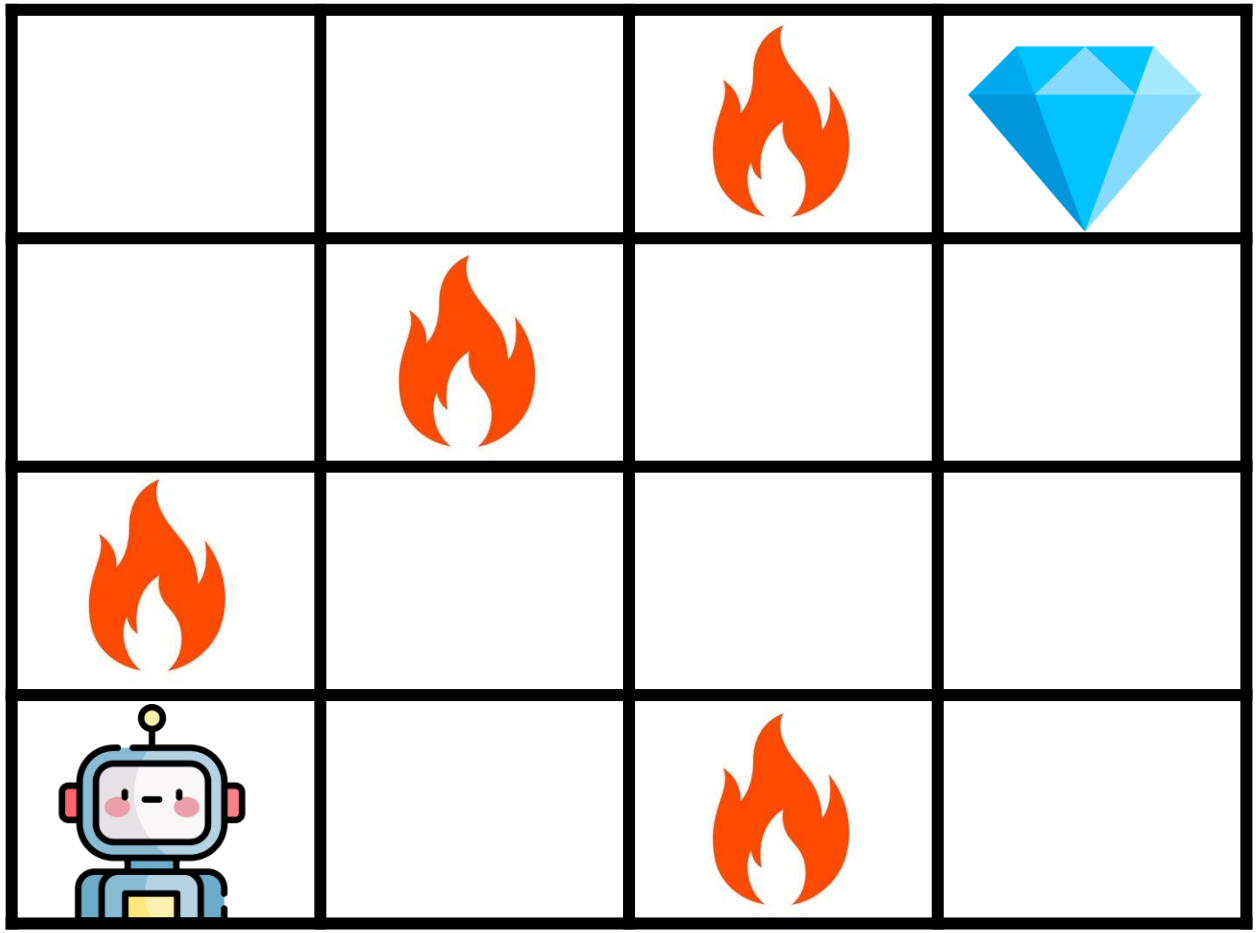


ML Algorithms



Reinforcement Learning Key Components

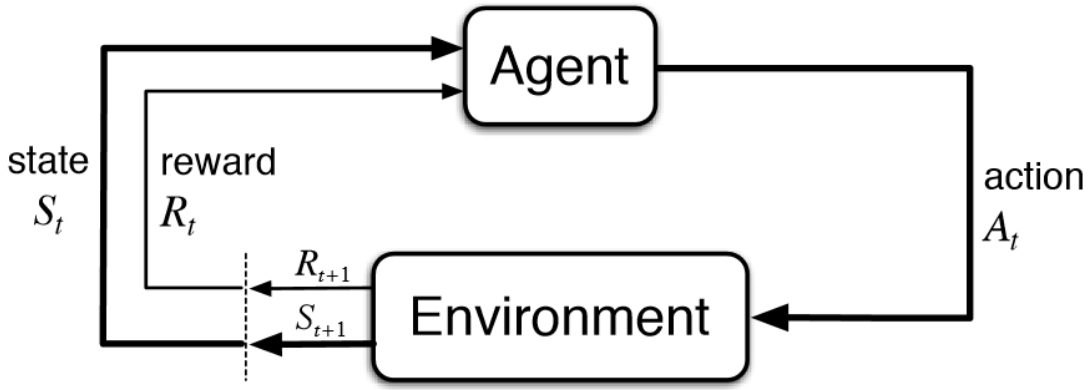
1. **Agent:** The learner or decision-maker.
2. **Environment:** Everything the agent interacts with.
3. **State:** A specific situation in which the agent finds itself.
4. **Action:** All possible actions the agent can make.
5. **Reward:** Feedback from the environment based on the action taken.



How RL Works

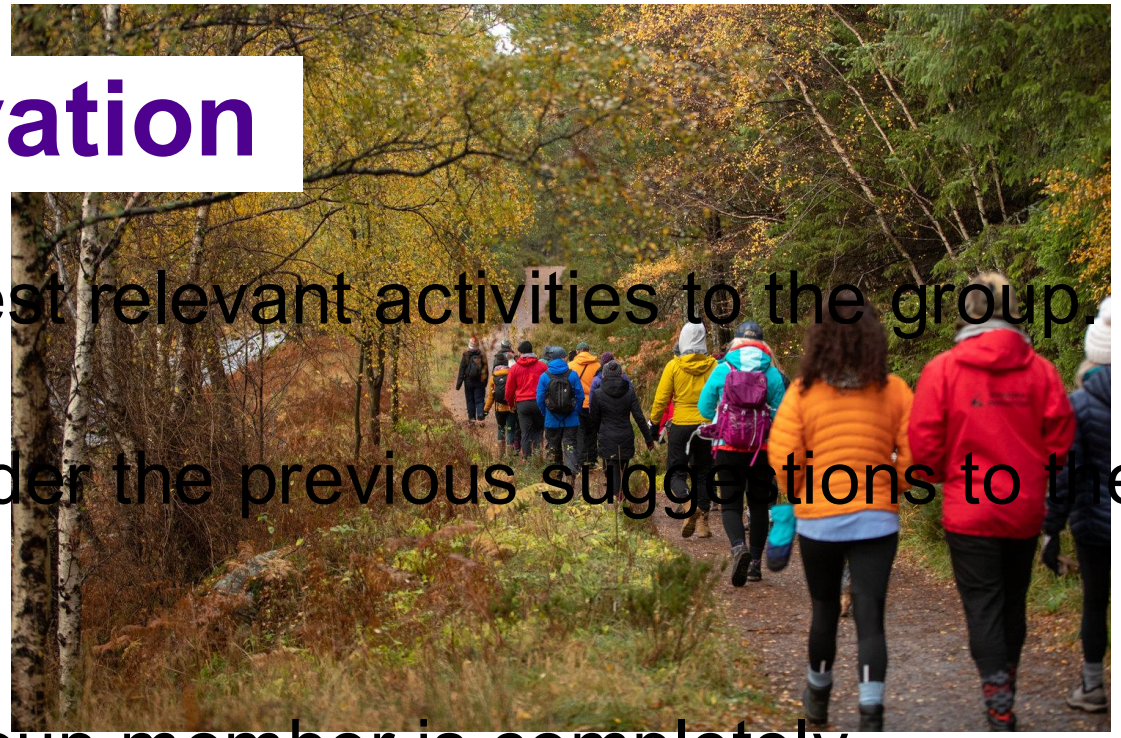
RL operates on the principle of learning optimal behavior through trial and error.
The agent takes actions within the environment, receives rewards or penalties, and adjusts its behavior to maximize the cumulative reward.

- **Policy**: A strategy used by the agent to determine the next action based on the current state.
- **Reward Function**: A function that provides a scalar feedback signal based on the state and action.
- **Value Function**: A function that estimates the expected cumulative reward from a given state.
- **Model of the Environment**: A representation of the environment that helps in planning by predicting future states and rewards.



SQUIRREL FRAMEWORK

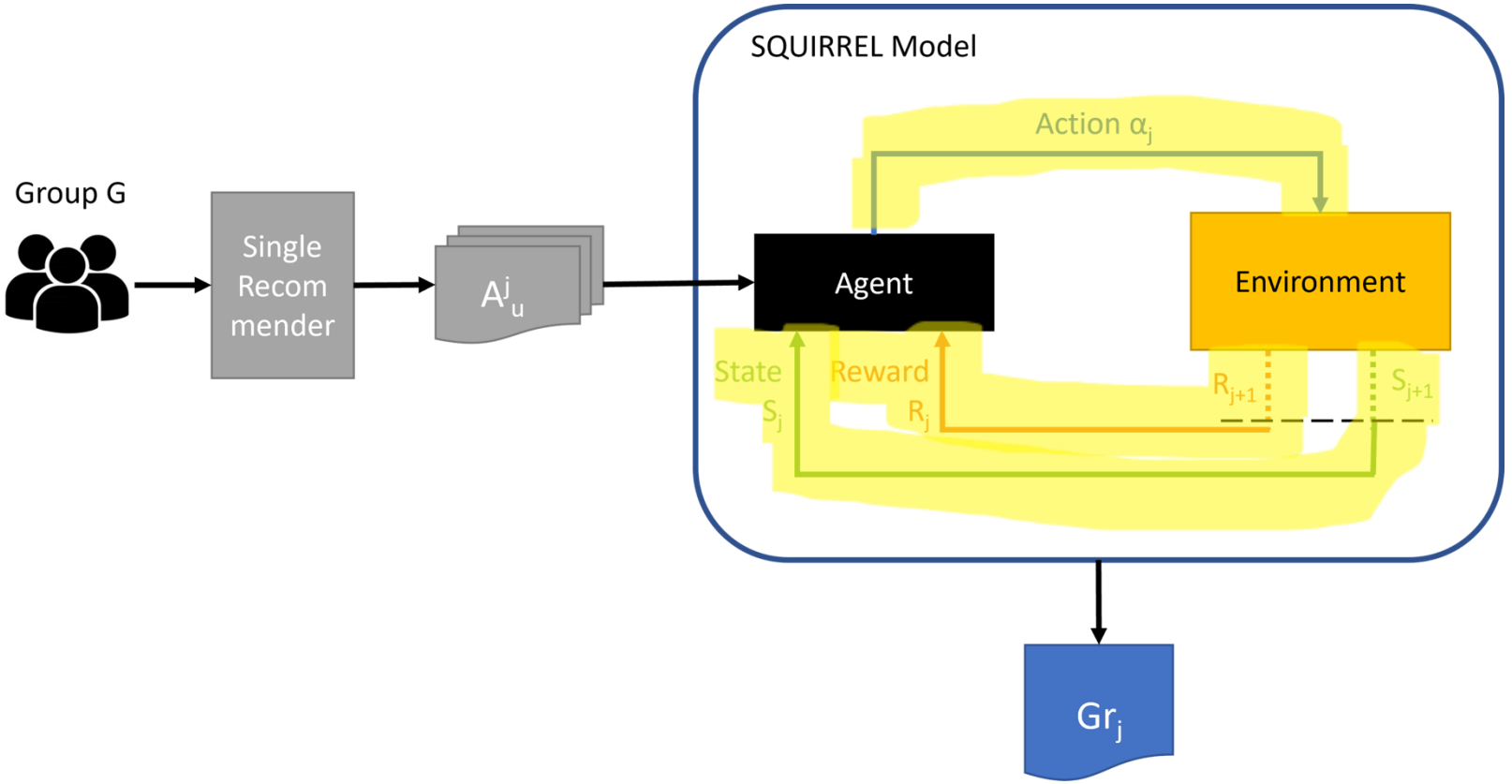
Motivation



- Suggest relevant activities to the group.
- Consider the previous suggestions to the group.
- No group member is completely dissatisfied at the end of the vacations.

Fairness

SQUIRREL Model



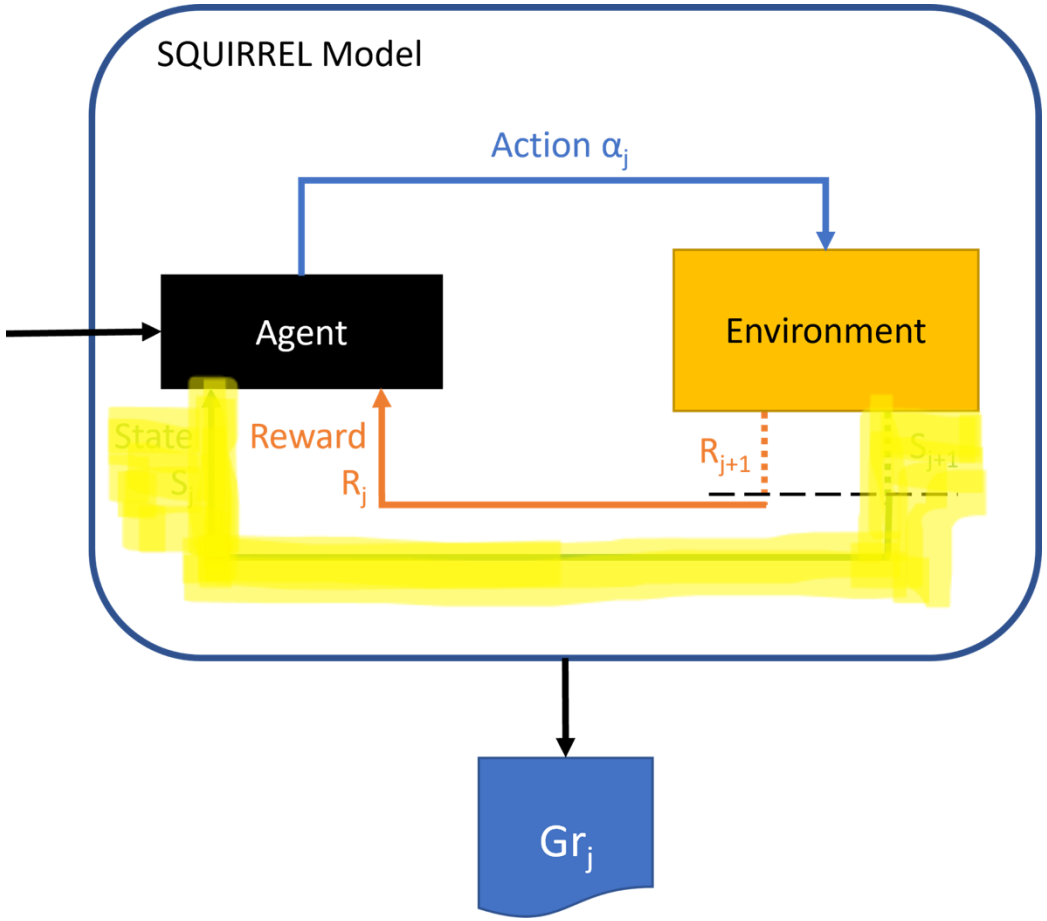
State	Action	Reward
The overall satisfaction of each group member	Six group recommendation methods	Satisfaction R_s Satisfaction + disagreement R_{sd}

SQUIRREL Model - State

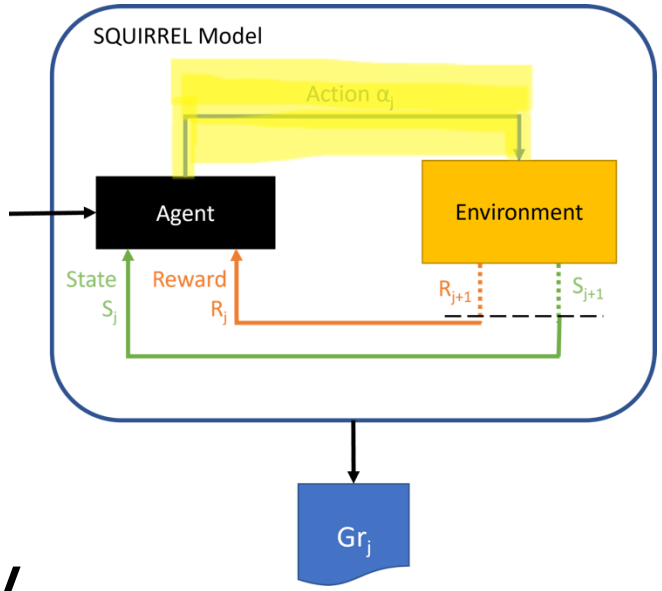
We keep an individual state for each group member *u* at each recommendation round *j*.

$$satO(u, \mathrm{RS}) = \frac{\sum_{j=1}^{\mu} sat(u, Gr_j)}{\mu}$$

The overall satisfaction of user *u* with respect to a sequence *RS* of μ rounds of recommendations is the average of the satisfaction scores after each round.



SQUIRREL Model – Actions SDAA



- Utilizes satisfaction scores.

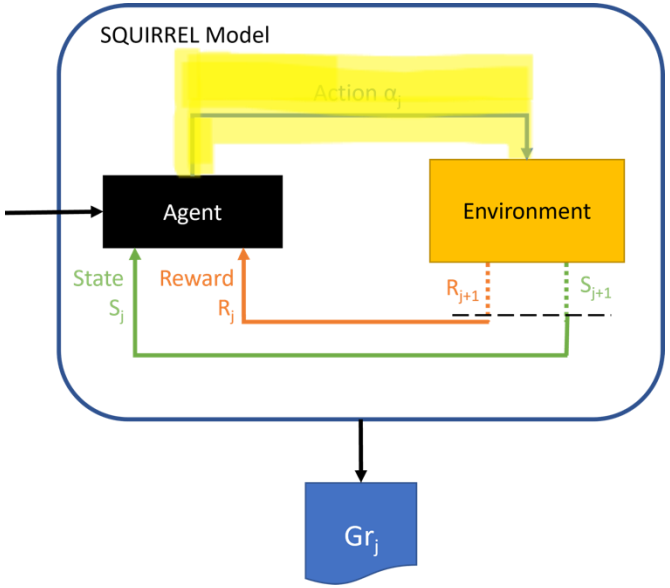
- Weighted Sum of Average and Least Misery.

$$score(G,i,j) = (1-\alpha_j) * avgG(G,i,j) + \alpha_j * leastG(G,i,j)$$

$$\alpha_j = max_{u \in G} sat(u, Gr_{j-1}) - min_{u \in G} sat(u, Gr_{j-1})$$

SQUIRREL Model - Actions SIAA

- Utilizes both satisfaction and disagreement in a weight assigned to each group member.

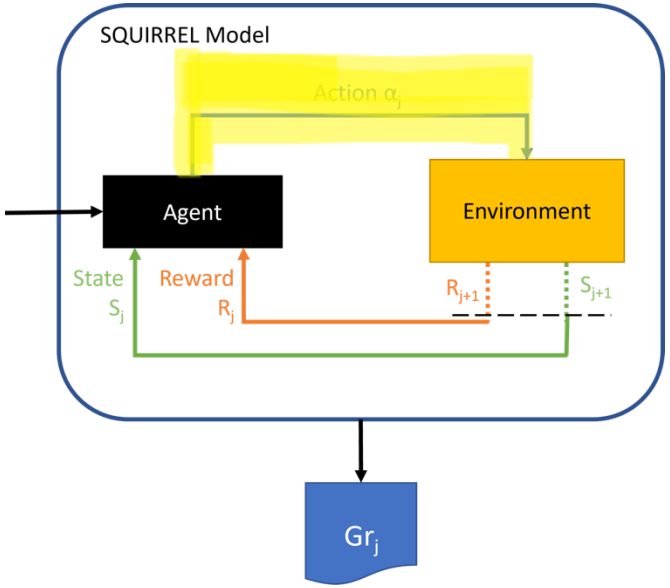


$$w_{u,j} = (1-b) * \left(1 - satO\left(u, \mathrm{RS}_{j-1}\right)\right) + b * userDis(u, G, Gr_{j-1})$$

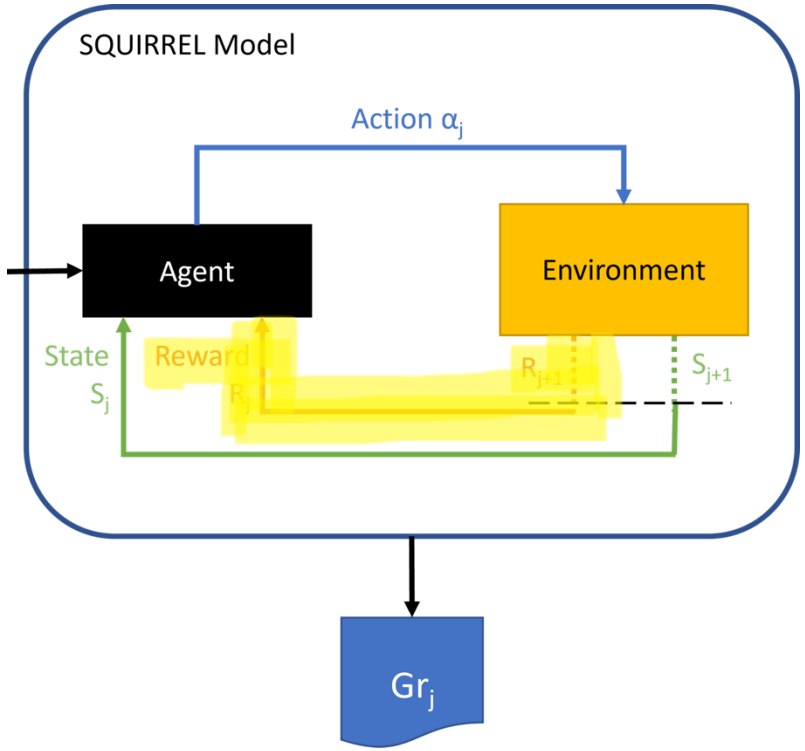
$$score(G, i, j) = \sum_{u \in G} w_{u,j} * p_j(u, i)$$

SQUIRREL Model - Actions Avg+

- Phase 1: Retain the *5k* items produced from the Average aggregation method.
  - Where *k* is the number of items recommended to the group.
- Phase 2: Recursively add items in the group recommendation list that produce the minimum disagreement.



SQUIRREL Model – Reward Satisfaction



Focuses on maximizing the satisfaction of the users

We define the overall group satisfaction of a group *G* for a recommendation sequence *RS* of *μ* group recommendations as:

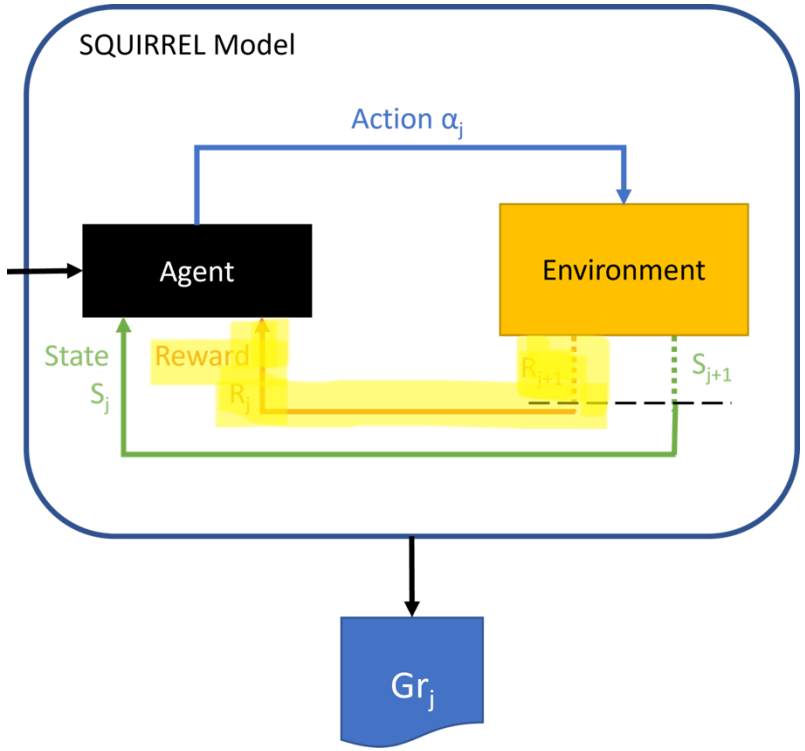
$$groupSatO(RS) = \frac{\sum_{u \in G} satO(u, RS)}{|G|}$$

This overall group satisfaction can be used for expressing the reward achieved at recommendation round j by action a.

$$R_s(RS^j) = groupSatO\left(RS^j\right)$$

Where RS^j refers to all the rounds up to the j^{th} one.

SQUIRREL Model – Reward Satisfaction - Disagreement



Balances the satisfaction and disagreement of the users

We define the overall group disagreement of G for the entire recommendation sequence as:

$$groupDisO(RS) = \max_{u \in G} satO(u, RS) - \min_{u \in G} satO(u, RS)$$

We utilize the harmonic mean of *groupSatO* and *groupDisO*, namely their F-score.

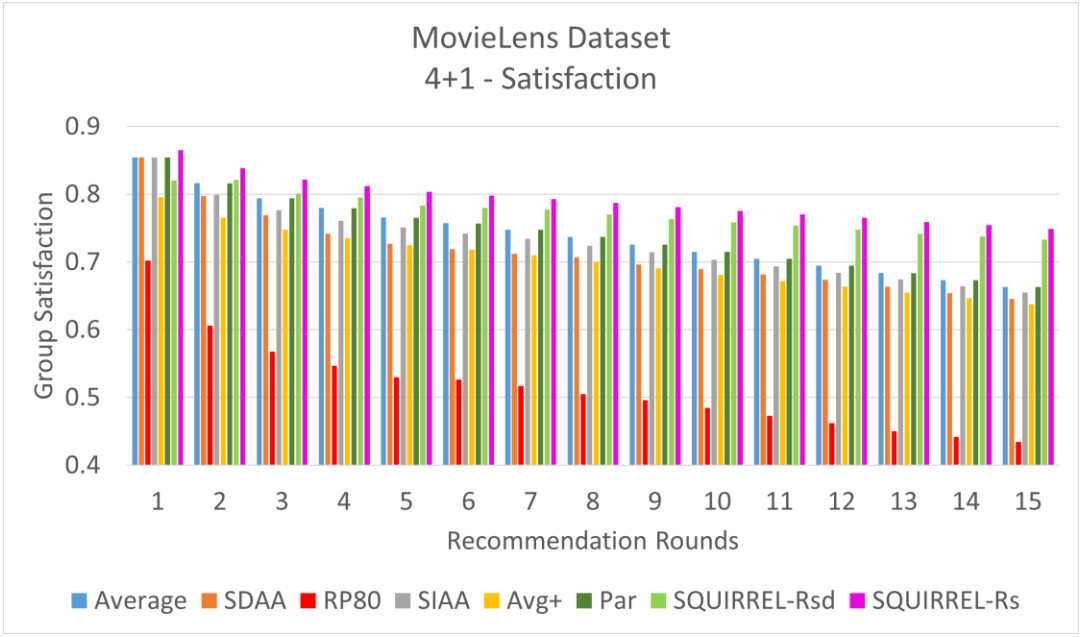
$$R_{sd}(RS^j) = 2\frac{groupSatO\left(RS^j\right) * (1 - groupDisO(RS^j))}{groupSatO(RS^j) + (1 - groupDisO(RS^j))}$$

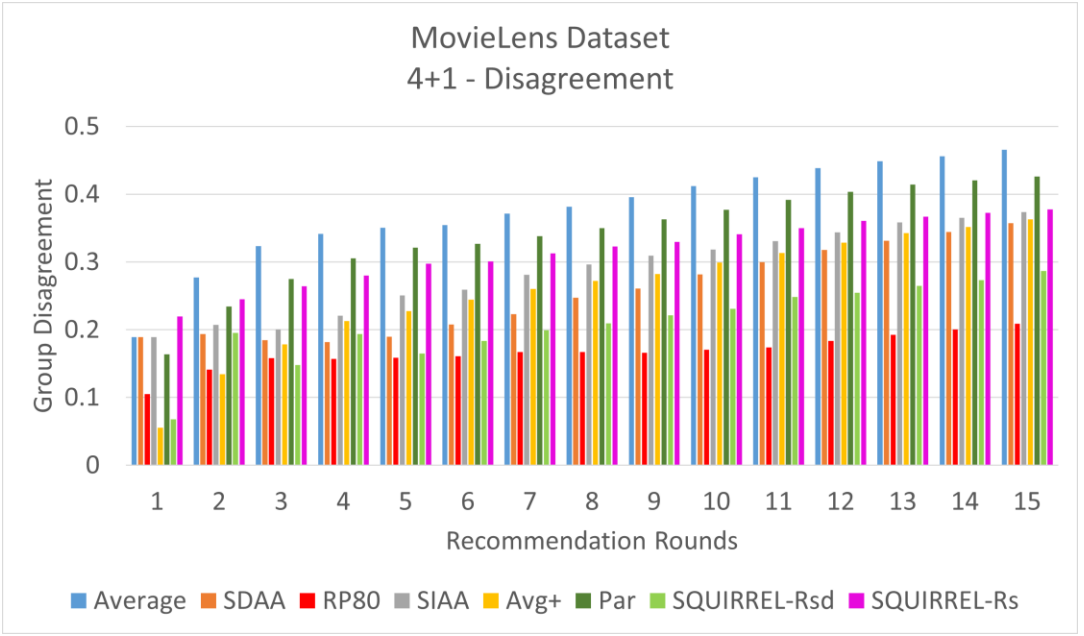
Tampere University
Tampere University of Applied Sciences

Evaluations

- Three Datasets, Two group types. Three different group sizes.

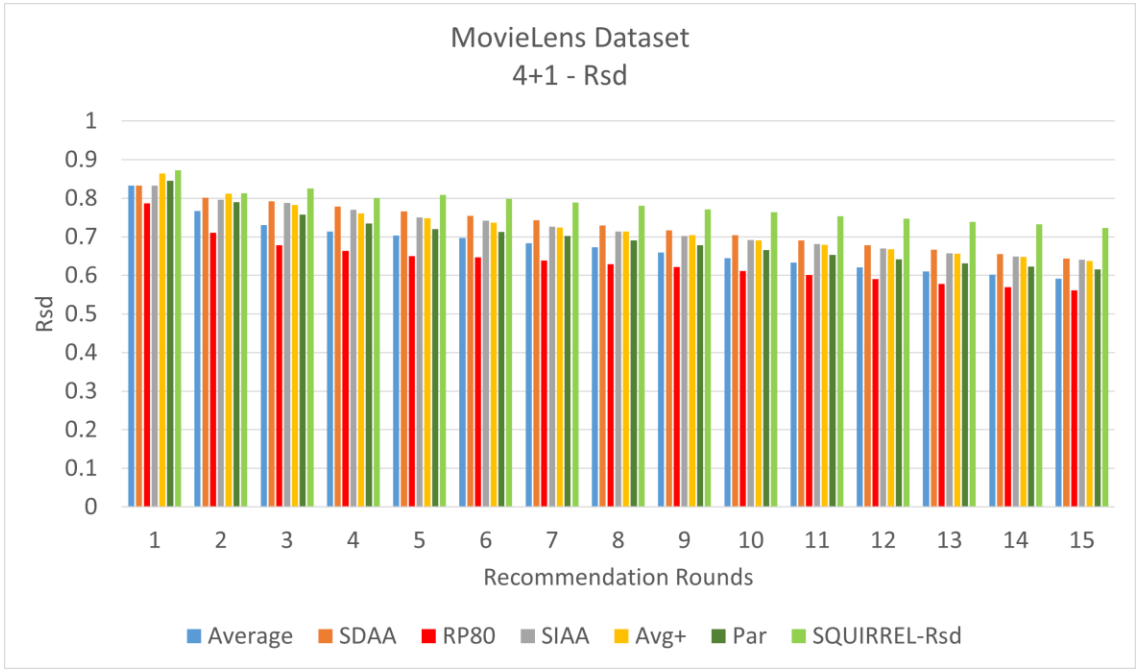
- The SQUIRREL model is the best performing one regardless of which reward was used.
- Reward function R_{sd} offers better disagreement scores.
- Reward function R_s offers the best satisfaction scores.





Evaluations

- Reward function R_{sd} is the best performing one when combining both satisfaction and disagreement scores.
- NDCG values for the SQUIRREL model indicate that the quality of the recommendations remain high while also improving satisfaction/disagreement.
- Similar results are observed when the group size is increased.



NDCG - MovieLens								
	Avg	SDAA	SIAA	Avg+	Par	RP80	R_{sd}	R_s
4+1	0.038	0.041	0.038	0.034	0.042	0.021	**0.054**	0.052
5 Diss	0.043	0.044	0.043	0.036	0.043	0.017	0.054	**0.060**
All Groups	0.042	0.044	0.041	0.034	0.042	0.019	**0.058**	0.057

Thank you!

Questions?