# Reinforcement Learning (RL) in Movie Recommendation Systems

## 1. Introduction to Reinforcement Learning (RL) in Recommendations

Reinforcement learning (RL) is an advanced approach to use machine learning in which an agent takes an action within an environment. The agent acts in order to optimise accumulated reward across time horizon. Here in this project, we have implemented RL in a sense of Q-learning algorithm to design the Personalized Movie Recommendation Systems. The system adjusts the solutions based on the user preferences and rating of movies hence, the effectiveness of the system improves in handling user interests more as times goes on.

## 2. Why RL Was Chosen for This Project

Existing recommendation matrices usually do not take into account the feedback of users or improve their models dynamically. Solutions, such as collaborative filtering or content-based filtering are inclined to rely on the users’ similarities or the content characteristics of the items, but are not flexible. Because of RL, the recommendation system can modify all the recommendation corresponding to the user and learn which movie genres are more adequate to him. This flexibility makes RL highly applicable for recommendation systems where constant feedback is provided to update the recommendation algorithms corresponding to the level of accuracy.

## 3. Where RL is Used in the Project

In this project, RL is implemented to pick movies given some user given genres and ratings. We employed Q learning as a way to implement RL to learn our best policy for recommending movies to users to achieve the highest probability of positive user feedback. The project’s key RL components are:

**Agent**: The movies that the recommendation system itself will recommend will be the agents.

**Environment**: How each rating or preference feedback influences the learning process the user’s interaction with the recommendations.

**State**: Each genre is one state and maps to one user’s current preference in the 'movie dataset'

**Action**: The genre chosen is taken to be a space of alternative actions, which we denote the title of each movie in that genre to be an action, that is, a recommendation.

**Reward**: The rating of the movie is used to simulate the user’s feedback and assigned as a reward to the agent. The higher the ratings, the more positive the rewards, the lower the ratings or negative feedback, the less or negative rewards.

## 4. How RL and Q-learning Were Implemented

Our implementation of RL is built around the core of Q learning algorithm. A model free RL algorithm, Q learning takes the (state,action) pairs, and updates the Q values, whose values denote the cumulative expected reward of each of those (state, action) pairs, and helps an agent learn the best action (movie) for each state (genre).

**Project RL Application in Step-by-step Process:**

1. **Data Preprocessing and State-Action Definition:**

* We feed the dataset into preprocessing and delete duplicates and transform missing values. Scores are converted to numeric types and the the title of each movie is standardized.
* Genres are treated as states, and movies in each genre are treated as actions, the foundation of the Q learning model.

1. **Initializing the Q-table:**

* It initializes a Q table to zero values, which for each row corresponds to a genre (state) and each column corresponds to a movie title (action).
* The Q-table contains Q-values, the agent’s belief about the reward potential for each movie recommendation in each genre.

**3. Reward Function:**

* The reward function simulates user feedback, on a movie rating basis. If ratings are 7 or higher, the rewards is +1; if ratings are between 4 and 6.9, the rewards is 0; otherwise, it is -1.
* This functions helps the agent give priority to movies of higher ratings which will ‘learn’ which movies that has higher chances of getting Users attention.

1. **Q-learning Algorithm for Training:**

An agent exploring movies from each genre run 500 episodes using Q learning.

- Exploration-Exploitation Trade-Off: For example, the agent is using an epsilon greedy strategy in which it explores new movies randomly or grabs on known high value recommendations.

- Reward Collection: The agent is given a reward of based on a user assumed rating for each movie it’s deemed to have recommended (more reward indicates the user is considered more likely to enjoy the movie).

- Q-value Update: Each (state, action) pair gets modified by the Q learning formula to update its Q value.

Q(s,a) ←Q(s,a) +α(R+γmaxQ(s’,a’)-Q(s,a) where the parameters of Q(s,a) are α, discount γ, R – reward, and Q(s’, a’) – maximal expected future reward in new state.

## Recommendation Function:

## To avoid providing low quality recommendations, the recommendation function filters movies in the user selected genre to only get those with rating between 6 and 10.

## It then grabs the top 5 unique recommendations based on Q value and these are trained to maximize the Q value.

## It has the function of ensuring that duplicate recommendations are removed and then orders based on the rating in descending order for a neat user experience.

## User Interaction and Recommendations:

## Current state is to represent, the User has to select a genre((a select) having options of available genres) from a list of available options.

## This genre recommends to the movie that has the highest Q value, and then shows this top rated recommended movies.

## 5. Benefits and Observations

Q learning is used in this project so that the recommendation system will learn what is the best to recommend in the future from very little data of user preferences. With Q values, the model tracks the quality of recommendations per genre, becoming capable of adapting in order to simulate a positive user feedback. This RL driven approach makes sure that the system keeps changing and keeps evolving with generation of personalized recommendations.

## 6. Conclusion

Through the Q learning algorithm, RL gives a structured, adaptive way to movie recommendations. Compared to static recommendation algorithms, we use user feedback as rewards to train a RL model that learns a good recommendation strategy for each genre. This adaptability positions RL as an excellent solution for recommendations which evolve over time with user interactions to become increasingly personalized and satisfying.