# Reinforcement Learning in Recommendation Systems

Presented by Tirth Patel, Priyank Patel, Neel Patel, Nachiket Prajapati

# Introduction to Reinforcement Learning (RL)

•Reinforcement Learning is a subset of Machine Learning focused on decision-making. Key components: Agent, Environment, Actions, Rewards. Applications span various fields.

# Types of Reinforcement Learning

•Outlines Q-Learning, SARSA, and others. Q-Learning suits recommendation systems for personalized content.

01.

- Value-Based: Focuses on finding the optimal policy by estimating the value of actions in each state. Q-learning is a popular example, where the agent learns the value of action-state pairs.

02.

Policy-Based: Directly learns the optimal policy by optimizing actions to maximize reward, often used in high-dimensional or continuous action spaces.

03.

Model-Based: Builds a model of the environment to predict future states and rewards, allowing the agent to plan actions based on this model.

04.

Actor-Critic: Combines value-based and policy-based methods, where the "actor" makes decisions, and the "critic" evaluates them, improving convergence and stability.

# Why Recommendation Systems?

•Purpose: To deliver personalized content effectively using RL. Advantages over traditional methods.

# Q-Learning in Recommendation Systems

## 01

Standard Q-Learning: The traditional form of Q-learning, where an agent iteratively updates Q-values for state-action pairs to learn optimal actions by balancing exploration and exploitation.

## 02

Deep Q-Learning (DQN): Uses deep neural networks to approximate Q-values in high-dimensional state spaces, allowing it to handle complex environments where standard Q-learning may struggle.

## 03

Double Q-Learning: Addresses the overestimation bias of Q-values in standard Q-learning by using two separate Q-value estimators, improving stability and accuracy

## 04

Dueling Q-Learning: Separates the Q-value function into two streams (value and advantage), enabling more efficient learning by focusing on the relative advantage of actions over baseline state values.

## 05

Multi-Agent Q-Learning: Extends Q-learning to multiple agents, where each agent learns in a shared environment, often dealing with challenges like cooperation, competition, or shared rewards.

# Mathematical Foundation

- Core Q-Learning equation: $Q(s, a) = Q(s, a) + \alpha * [R + \gamma * \max Q(s', a') - Q(s, a)]$.

# Practical Demonstration Setup
# Success cases

•Dataset (movie or user-item interactions). Features include time of day, population density, weather.

# Implementing Q-Learning in Jupyter Notebook

•Steps: Initialize Q-values, update based on rewards. Discuss challenges and solutions.

# Results and Evaluation

•Evaluation: Precision, recall, and cumulative reward showing recommendation improvements.

# Group Contributions

•Detail each member's role:
Tirth Patel:- research,
Tirth Patel:- coding,
Priyank Patel:- documentation,
Nachiket Prajapati:- presentation.

# Thank you very much!

www.reallygreatsite.com