CIA-1

**Problem Statement**

Given a large set of items (products, articles or movies) and a user base with diverse preferences, the goal is to recommend items to each user in such a way that maximizes the overall satisfaction or engagement .We use K-armed bandit approach to iteratively learn and improve the recommendation strategy based on feedback (clicks or ratings) from users.

* Each arm represents a unique item or item category.
* Pulling an arm corresponds to recommending an item to a user.
* The reward is the user’s interaction with the action (recommending an item).

**Objective**:

Maximize cumulative rewards by recommending items that are more likely to be accepted positively by users.

**Description:**

Multi-armed bandits (MABs) are a framework for sequential decision making under uncertainty. MABs solve problems in online advertising, information retrieval, and media recommendation.

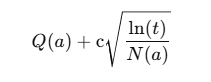
Actions are recommendations of different items (each item represents an arm). Based on the interaction of the user with the action we reward each action, positive reward for a user click (add to cart, add to wish list , watch later or saved to playlist ) , negative reward if the user ignores the recommendation .The per round cumulative reward represents click-through-rate

The challenge here is to balance exploration and exploitation, we need to balance between exploring new items to understand user new preferences and exploiting known preferences to maximize engagement.

As the recommendation system frequently encounters new items Upper Confidence Bound Multi arm bandit is preferred ,

The UCB algorithm chooses an action based on a combination of the estimated reward of the action and the uncertainty in that estimate.

Each action a is selected based on the formula:



* Q(a): Estimated reward of action a so far.
* t: Current timestep (total number of trials so far).
* N (a): Number of times action a has been selected.
* c: Exploration parameter, controlling the degree of exploration.

Q(a) allows exploitation as higher estimated rewards are preferred, whereas c encourages actions with fewer trials. As N(a) increases, this bonus term decreases, eventually reducing exploration for that action.

Actions with high estimated rewards will tend to have higher values. Actions with fewer trials have higher exploration bonuses, encouraging the algorithm to try them more often initially, over time actions with lower rewards will have fewer trials and gradually be explored less.

UCB works by testing new recommendations more often at first. If a recommendation hasn’t been shown to many users yet, UCB will give it a chance, letting it appear more often until there's enough data to judge how good it is. This helps spot popular content quickly.

Once UCB has enough information about a recommendation, it focuses on showing the best ones more frequently. So, as time goes on, UCB gets better at giving users the recommendations that are most likely to be liked, but it also adapts when new recommendations are added by giving them a fair chance.