**ONLINE CLICK RATE IMPROVEMENT**

**Table of Contents**

**1. Project Overview**

**2. Business and Problem Understanding**

**3. Implementation of Strategies**

**- Random Selection**

**- Upper Confidence Bound (UCB)**

**- Thompson Sampling**

**- Greedy and Epsilon-Greedy**

**4. Results and Conclusion**

**Project Overview**

This project aims to improve the Click-Through Rate (CTR) of online advertisements by applying Reinforcement Learning, specifically using Multi-Armed Bandit (MAB) algorithms. The goal is to determine the most effective strategy for serving ads to users to maximize clicks and, therefore, revenue.

**Business and Problem Understanding**

Online advertising is a crucial revenue source for many internet-based companies. The primary objective is to entice users to click on ads. However, it's challenging to know in advance which type of ad a particular user will respond to. This is where MAB algorithms come into play.

We are tasked with simulating an ad-serving scenario where we have data on how users have responded to different ads in the past. Our goal is to use this data to train and evaluate four different MAB strategies: Greedy, Epsilon-Greedy, UCB1, and Thompson Sampling. We will compare their performance based on two key metrics:

**Cumulative Regret:** The total loss incurred by not choosing the optimal ad. A lower regret indicates a more effective strategy.

**Reactivity:** How quickly an algorithm learns to choose the optimal ad.

**Implementation of Strategies**

We implemented and compared four different MAB strategies:

**Random Selection:** This is our baseline strategy. It involves randomly selecting an ad to show to the user in each interaction. This approach is purely exploratory and does not leverage any past information to make decisions.

**Explanation:** In this step, we simulate the process of showing ads randomly. We iterate through each ad impression (representing a user interaction) and select an ad to display completely at random from the 10 available ads. We then check if the user clicked on this randomly selected ad based on our historical dataset. The total number of clicks received from this random strategy is accumulated.

**End Result:** The total reward (clicks) obtained by the random selection strategy after all impressions is calculated. This gives us a baseline to compare the performance of the other, more intelligent MAB strategies against.

**Upper Confidence Bound (UCB):**

UCB1 is a more sophisticated strategy that balances exploration and exploitation. It not only considers the average reward of each ad but also incorporates a "curiosity bonus" for ads that have been tried less frequently. This encourages the algorithm to explore less-known options while still favoring those with a good track record.

**Explanation:**

The UCB1 algorithm works by calculating an "upper confidence bound" for each ad. This bound is a combination of the ad's average reward so far and an exploration term that decreases as the ad is shown more times. In each iteration, the algorithm chooses the ad with the highest upper confidence bound. If an ad hasn't been shown yet, it's given a very high initial upper bound to encourage exploration. As ads are shown and rewards are observed, the average reward and the exploration term for each ad are updated.

**End Result:**

After simulating all ad impressions using UCB1, we get the total reward obtained by this strategy and how many times each ad was selected. We expect UCB1 to perform better than random selection by learning to favour ads that have shown higher rewards.

**Thompson Sampling:**

Thompson Sampling is a Bayesian approach that models the probability of each ad being the best one. It uses the Beta distribution to represent the uncertainty about the true click-through rate of each ad. In each step, it samples from the posterior distribution of each ad and chooses the one with the highest sampled value. This allows for a more "intelligent" exploration that is guided by the data.

**Explanation:**

Thompson Sampling maintains a Beta distribution for each ad. Initially, these distributions are uninformed. When an ad is shown and a reward (click) is received, the "successes" parameter for that ad's Beta distribution is incremented. If no reward is received, the "failures" parameter is incremented. In each iteration, the algorithm samples a value from the Beta distribution of each ad and chooses the ad with the highest sampled value. This sampling process naturally balances exploration and exploitation; ads with higher success rates or less observed outcomes will tend to have higher sampled values more often.

**End Result:**

After simulating all ad impressions using Thompson Sampling, we get the total reward and the allocation of impressions to each ad. We anticipate that Thompson Sampling will perform very well due to its probabilistic approach to balancing exploration and exploitation.

**Greedy and Epsilon-Greedy:**

**Greedy:** This strategy always chooses the ad with the highest known average reward. It's a purely exploitative approach.

**Epsilon-Greedy:** This is a variation of the Greedy strategy that introduces a small probability (epsilon) of choosing a random ad. This allows for some exploration, preventing the algorithm from getting stuck on a suboptimal choice.

**Explanation:**

The Greedy strategy is the simplest. In each step, it calculates the average reward obtained so far for each ad and simply chooses the ad with the highest average reward. This approach quickly latches onto the best-performing ad it has encountered but risks getting stuck on a locally optimal ad if it doesn't explore other options.

The Epsilon-Greedy strategy improves upon the Greedy approach by adding a small chance (`epsilon`) of choosing a random ad instead of the one with the highest average reward. This controlled exploration helps the algorithm discover potentially better ads that it might have missed otherwise. With probability `1 - epsilon`, it behaves like the Greedy strategy.

**End Result:**

After simulating with both Greedy and Epsilon-Greedy, we get the total rewards and ad allocations for each. We expect Greedy to perform poorly in terms of regret due to limited exploration, while Epsilon-Greedy should perform better than pure Greedy by incorporating some exploration.

**Results and Conclusion**

After running simulations for each of the four strategies, we analyzed their performance based on cumulative regret and arm allocation.

Cumulative Regret Comparison

The cumulative regret graph clearly shows that Thompson Sampling and Epsilon-Greedy are the most effective strategies, with Thompson Sampling having a slight edge. The Greedy strategy performed the worst due to its lack of exploration, while UCB1 had a higher regret than Thompson Sampling and Epsilon-Greedy because of its more aggressive exploration.

Arm Allocation

The arm allocation pie charts further support our findings. Both Thompson Sampling and Epsilon-Greedy allocated the vast majority of their plays to the optimal ad (Arm 8), demonstrating their ability to quickly identify and exploit the best option. The Greedy strategy, on the other hand, had a more scattered allocation, indicating its failure to consistently choose the best ad. UCB1 also allocated a significantportion of its plays to the optimal ad but with more exploration compared to Thompson Sampling and Epsilon-Greedy.

**Conclusion:**

Based on our analysis, we can conclude that Thompson Sampling is the most effective strategy for this particular ad-serving problem. It achieves the lowest cumulative regret and quickly converges to the optimal ad, thereby maximizing the click-through rate and overall revenue. The Epsilon-Greedy strategy is also a strong contender, offering a good balance between exploration and exploitation.