Table of Contents

1. INTRODUCTION

1.1 PURPOSE

The purpose of this document is to build a project using reinforcement learning that can be used to find the optimal ad that can be placed on a website that can attract the maximum number of clicks.

1.2 INTENDED AUDIENCE

This document is intended to be read by the team that builds the project and the admin of the target website that employs the project.

1.3 PROJECT SCOPE

We hope to provide a system by which the website admin can post multiple ads on a website and find out which one the audience responds the most positively to. This is done using the least number of ad postings as each posting costs money. This will minimize the total cost of posting the ads one by one.

1.4 REFERENCES

https://github.com/deltonmyalil/ReinfLearningSEproject

https://www.analyticsvidhya.com/blog/2018/09/reinforcement-multi-armed-bandit-scratch-python/

2. OVERALL DESCRIPTION

2.1 PRODUCT PERSPECTIVE

The project stores the following information:

1. The user click details of the ad version for future use
2. The ad version which seems most promising at an instant of time
3. Cost of placing an ad
4. Reward to be given when the user clicks on an ad

The project is intended to be used in tandem with an already up and running website. The project collects the above information and then uses the Reinforcement Learning algorithms to find the best ad to be placed permanently on the website.

2.2 PRODUCT FEATURES

The additional features of the intended project, apart from the basic functions are the following:

1. It may be able to work as a daemon in the backend of the website.
2. It may not slow down or affect the performance of the website in any way.
3. It may have provision to be able to employ additional Reinforcement Learning algorithms in the future to do the same.
4. It may store the ad click data in a database for analysis of the data in the future.

2.3 USER CLASS AND CHARACTERISTICS

2.3.1. Website Administrator:

The admin(s) of the website which employs the project is the primary intended user of the project. The admin is responsible for finding the costs of posting an ad in the website. He is also responsible for deciding the reward to be given to the clicked ad. These measures are decided by the admin himself or the ad agency/service which the website uses.

2.3.2. Data Analyst or Machine Learning Engineer:

The user click data collected by the website can further be used for data mining to find interesting trends in the ads posted. The Analyst may find the trends and report it to the admin.

2.4 OPERATING ENVIRONMENT

The project may work as a background process to a website in the backend. It is to be designed to be able to work in the remote web server irrespective of its tech stack – whether it be Spring-Boot, django or Flask.

2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

1. Upper Confidence Bound algorithm should be the primary algorithm that does the learning, but provision to use additional learning algorithms like Thompson Sampling along with UCB should be left for future expansion of the project.

2. Should never impact the performance of the webserver where the project is deployed.
3. Provision to change the costs and rewards should be integrated to the admin panel of the website.

3. SYSTEM FEATURES

3.1 FUNCTIONAL REQUIREMENTS

1. The learning algorithm should give the optimal ad out of all ads posted in the least number of trials.
2. The learning algorithm should give the optimal ad out of all ads posted till the current iteration (user visit) during each iteration and should be displayed in realtime.
3. If the optimal ad stays the same for a fixed number of iterations, it should be selected as the most promising ad and the admin must be notified.
4. The learning algorithm should adapt to changing costs on the fly.
5. The input to the project form the admin will only be the cost and reward for the algorithm to learn. The ad clicks may be taken as input implicitly each time a user visits the website.

4. EXTERNAL INTERFACE REQUIREMENTS

4.1 USER INTERFACE

Inputs: The input from the admin – ie the cost and the reward may be taken from the admin from the admin panel of the website. Hence the user interface may be integrated into the admin panel. Inputs may be taken as float values for each of the textfields and may be submitted using a submit button.

Outputs: The output will be displayed in realtime on the admin panel as the ID of the ad that shows the most promise till the current iteration. Once the algorithm converges on a solution, the admin must be alerted with a message box displaying the ID of the ad that is the best ad from the lot.

5. NONFUNCTIONAL REQUIREMENTS

5.1 PERFOMANCE REQUIREMENTS

The project may never compromise the performance of the web app system.

5.2 SAFETY REQUIREMENTS

The project may never interfere in the data of the website and may never possess write permission in any other database other than the one used to store the user click history.

5.3 QUALITY ATTRIBUTES

Availability: The project should run in parallel with the webapp and should be available for the admin to see whenever the webserver is up and running.

Correctness: The ad chosen as the most promising ad after the algorithm converges must generate sufficient clicks from the users of the website.

Maintainability: The cost/reward of the algorithm may be easily altered. The algorithm must be implemented in such a way that the maintenance of the project will not impact the functioning of the website.

Upgradability: The project may be upgradeable in the future by using more advanced reinforcement learning algorithms alongside Upper Confidence Bound