

DATA INTELLIGENCE APPLICATIONS

Project: Pricing and Matching

**a.y. 2020/2021**

Xiong Yuhang - student code 10800838

David Leonardo Gomez - student code 10787907

# Contents

[**Contents**](#_heading=h.slthlgawf074) **1**

[Scenario](#_heading=h.gjdgxs) **2**

[**Step 1. Mathematical formula**](#_heading=h.8u3jexa4b4lo) **4**

[**Step 2. Uncertainty mathematical formula**](#_heading=h.40atwplq13mb) **6**

[**Step 3. Base Simulation Case**](#_heading=h.gjmpj5jqhpty) **7**

[**Step 4. Learning in Online mode (Unknown Conversion Rates)**](#_heading=h.9edby2pe3dz7) **9**

[**Step 5. Promo Optimal Assignment (Promotion Distribution)**](#_heading=h.odb24h5297cu) **11**

[**Step 6. Pricing and Matching (Price 2 not fixed)**](#_heading=h.y5l2f1t5xyq4) **13**

[**Step 7. Sliding-Window**](#_heading=h.xk7ttlovrw3h) **15**

[**Step 8. Change Detection**](#_heading=h.n2nfwv3xfwg1) **17**

Consider the scenario in which a shop has several promo codes to encourage the customers that buy an item to buy another, different item. The customers can belong to different classes and the promo codes can provide different discounts.

The essence of this problem is a multi-arm slot machine problem. We need to use algorithms to determine the price and discount distribution that will get the most profit. Among them, in terms of pricing, arm refers to the price of the product, and in terms of matching, arms are the best promotion task for each customer category.

# Scenario

Consider a scenario where a retailer of videogames and related products sells two products. The first product is a game, denoted as product 1. The second product is a DLC bundle of the game, denoted as product 2. To increase sales of the DLC bundle, the retailer offers a promotional discount to customers who purchase the game. This means that all customers who purchase the game will receive one of the different promotional offers for the DLC bundle.

The number of promotional discounts is defined by the retailer’s business department and provided to customers based on the customer category to which they belong, as the promotional discounts are limited.

There are a total of four types of customers, and each customer is divided into one of the following four categories:

1. Casual Player

Casual players are interested in buying the game, but they are not willing to spend too much for games, also if they receive a discount for the complete DLC bundle after the purchase, they will probably buy it.

2. Family member

Family members buy games for gifts to a member of the family, typically their sons. They have a good budget and are willing to pay for the game if there is a special date or holiday, but unless there is a good discount on the DLC bundle, they might not buy it.

3. Hardcore gamer

They are fans of the game, they are willing to pay for the game and the DLC bundle even with no discounts, but they often have tight budgets that make them. unable to buy the products, with better discounts, they will probably make the purchase.

4. Collector

Collectors have a big budget and if they are collecting this game, they are willing to pay higher prices for getting the game and the DLC bundle even if the discounts are not attractive.

The game price is 30$ per unit. The conversion rate is related to the category of customers who purchased the product, the price paid, and promotional discounts, so we constructed the conversion rate into a functional relationship between these items, which has been given by code. The following table shows the price and profit of product 1.

| Game Price | 40 | 45 | 50 | 55 |
| --- | --- | --- | --- | --- |
| Profit | 10 | 15 | 20 | 25 |

For retailers, the cost of the DLC Bundle is $10 each. The following table lists the price and profit of each original price and promotion. Consider a customer buying the game, and then giving the customer a promotion. The promotion includes four different discount levels on the DLC Bundle.

|  | D$ | Price1 | Pr. | Price2 | Pr. | Price3 | Pr. | Price4 | Pr. |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| P0 | -0$ | 20 | 10 | 24 | 14 | 28 | 18 | 32 | 22 |
| P1 | -3$ | 17 | 7 | 21 | 11 | 25 | 15 | 29 | 19 |
| P2 | -6$ | 14 | 4 | 18 | 8 | 22 | 12 | 26 | 16 |
| P3 | -9$ | 11 | 1 | 15 | 5 | 19 | 9 | 23 | 13 |

Pr. – Profit, D$ - Discount

The following assumptions are made in this scenario:

• On average, 100 customers arrive at the store every day

• The average number of customers in each category are:

Distribution of customers categories 1, 2, 3 and 4 fornstatic scenario: [40, 30, 20, 10]

• Promotional distribution for static scenarios: [0.3, 0.3, 0.15, 0.25] are respectively for promotion levels P0, P1, P2 and P3.

# 

# Step 1. Mathematical formula

First, match the appropriate promotion method for each type of customer:

Parameters indicate:

，Cost matrix, customer type i allocates the cost of promotion j;



，Binary allocation matrix, if and only if the promotion j is allocated to customer i



**After determining the matching method, calculate the profit in the corresponding situation:**

Parameters indicate:

：It is the average number of customers of this category who arrive at the store every day.



：Is the price selected by the store for product 1 at time t.



:Is the price selected by the store for product 2 at time t.



:It refers to the profit when the store sells a product 1 at price P.



：It refers to the profit when the store sells a product 2 at price Q and promotion F.



：It is the conversion rate of type i of the first product at time t, assuming the price is P.



:It is the conversion rate of category i of the second product at time T, taking into account price Q and promotion F.



:Is the score of promotion f provided to category i at time t.



For the first product, the total revenue of the store in a day will be equal to the price profit it chooses multiplied by the number of customers who bought it, which is equivalent to the average number of customers in each category multiplied by the conversion rate of the category at the design price with. This means that the total profit can be expressed as:



The number of customers of each type who can purchase the second product only corresponds to the part of the first product, and then the total number of customers who purchase the second product is:



We need to aggregate all promotions to get the total amount. If we add up all categories and include the profit of the second product, we will get the second part of the total daily profit of the store, corresponding to:



In a one-year time interval, by adding the two total profit items, we get the total revenue of the store in one year:



This is the amount the store is trying to maximize, and the variables they need to modify are the price and quantity of the two products and the price and quantity of promotions offered for each category. Then the store needs to solve the following optimization problems:



The only restriction we need to add is that the sum score of the promotions offered is equal to 1.

# 

# Step 2. Uncertainty mathematical formula

We have 3 random-defined variables.

* First, the number and types of customers who arrive at the store every day.
* Then how many of these customers will buy the first product?
* How many of the customers who buy the first product will buy the second product?

The number of customers arriving each day can be modeled by supporting any distribution of natural numbers. We will represent the distribution of the number of customers in Ci, t（θi，t）, where θi, tare weights are the corresponding distribution parameters.

The number of customers who bought the first product can be modeled as the result of the Bernoulli experiment, where the number of experiments is the number of customers that arrive, and the probability of success is the category conversion rate calculated at the corresponding price. Then, the number of buyers for the first product can be modeled as a binomial distribution, with the number of trials equal to the number of customers, and the probability of success equal to the conversion rate.

Finally, the sales volume of the second product can also be modeled as a Bernoulli test, but in this case, the number of trials corresponds to the number of buyers of the first product, and the probability of success is the conversion rate of the second product. In addition, we need to consider where the customer’s promotion is provided.

Then, the deterministic optimization problem in Part 1 can be modified to the following stochastic optimization problem:



In the formula, Ci, t（θ） is any probability distribution that supports natural numbers and parameters （θ）), and B (c, p) is a binomial distribution without the number of trials and the probability of success. In this case, ci，t，χ1i，t and χ2i, t is a random variable, modeling the number of customers who arrive at the store in one day, the number of people who bought the first product, and the number of people who bought the second product.

Since this optimization problem is not deterministic, we cannot use the same method as in the previous section, we need to use online learning methods. For this type of problem, we can use the multi-arm bandit method, where each arm is a combination of the prices of two products and the effective distribution of promotional activities, and the parameters are updated every day.

# Step 3. Base Simulation Case

The goal is to learn the optimal price of the Product 1 (Game), by using a fixed price for the product 2 (DLC Bundle). Assuming that the number of customers, their distribution per class is known and the conversion rate associated with the items. It adopted both an upper-confidence bound approach and a Thompson-sampling approach and compared their performance.

**Process:**

The aim is using both the upper limit of confidence method (UCB1) and Thompson sampling method to learn the optimal price of sale for the first item and compare its performance in terms of revenue. The maximum number of customers arriving at the store every day is set to 100, distributed {45,30,20,5}. The conversion rate of product 2 is fixed.

The simulation has run in an offline environment, two learners are used to simulate the learning process for 365 days a day. Every day, the arms of the two independent learners were pulled, the reward of each selected arm was calculated, based on the maximum possible value, and the learners were updated.

**Learning Results:**

* Thompson learner converges to price 55$ for product 1
* UCB learner converges to price 50$ for product 1

**Profit**

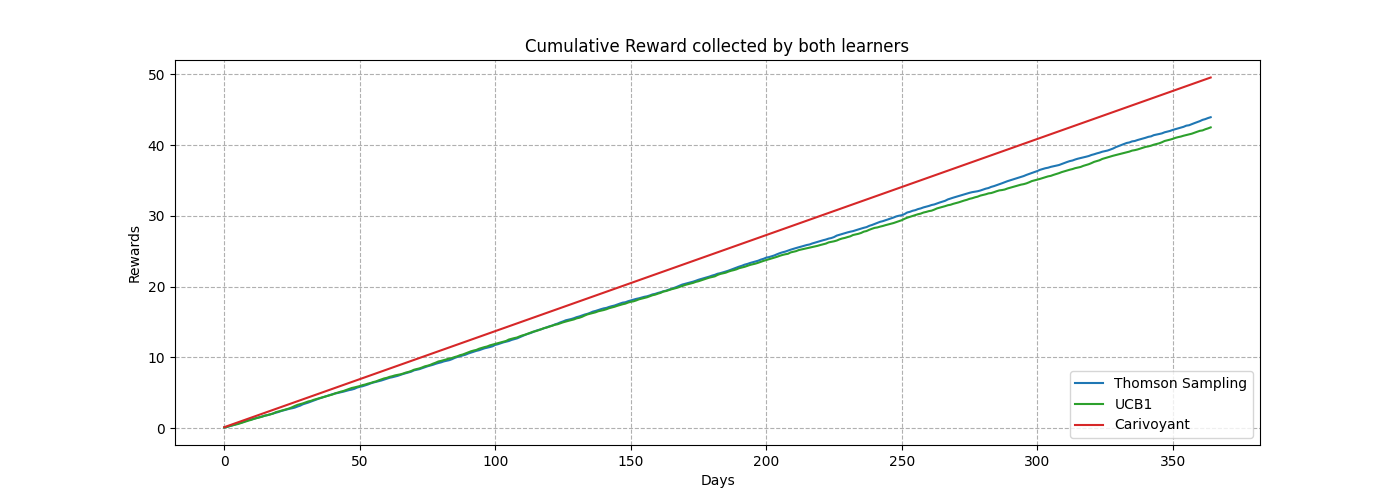
* Thompson Sampling: 43.942857142857136 $
* UCB: 42.50714285714282 $

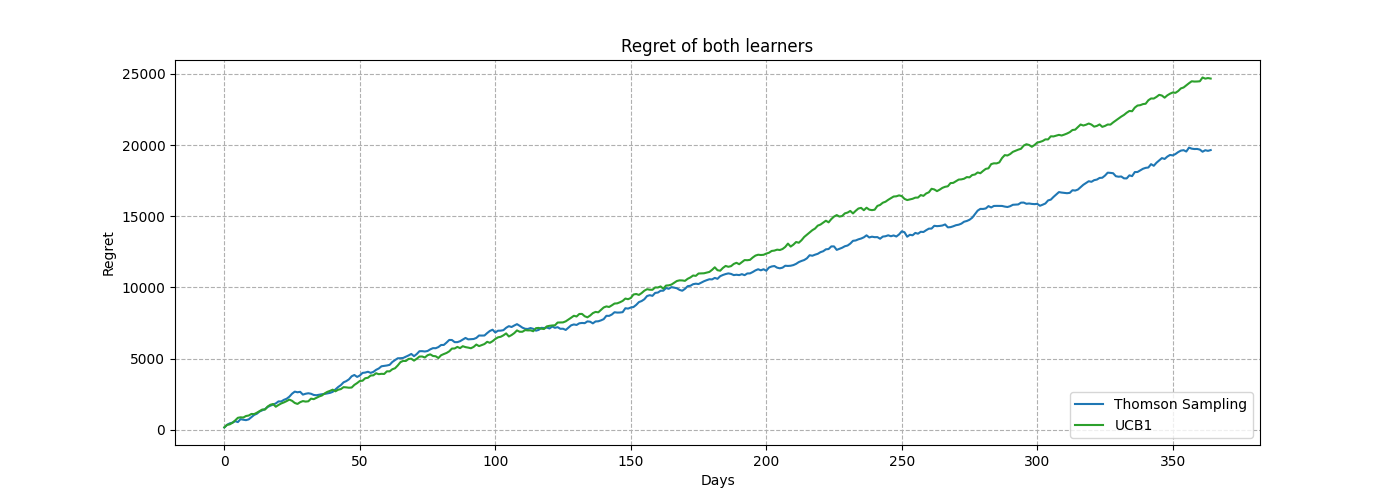
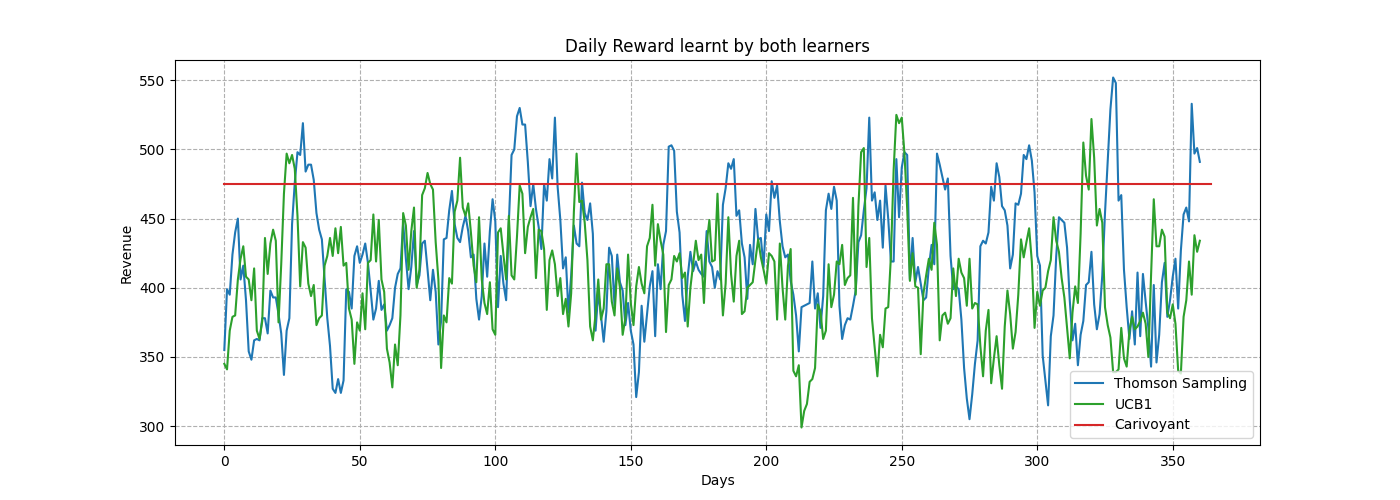
**Regret**

* Thompson Sampling: 55.19999999999999 $
* UCB: -34.80000000000001 $

**Learning Process Charts:**

The following charts show the cumulative rewards, regrets, and the daily advancement of the two algorithms. The cumulative reward collected by the Thompson sampling algorithm is slightly higher than the one collected by UCB algorithm.





The clairvoyant is the optimal result for the process algorithm is the optimal solution of the problem calculated when all the parameters in the problem are known from the start. According to the data, the solution produced by Thompson's sampling algorithm is closer to the optimal solution.

# 

# Step 4. Learning in Online mode (Unknown Conversion Rates)

The goal is to provide solutions to the online pricing problems of the previous step, this means that the customers from each class are randomly selected with a horizon of 1 year (365 days) and the conversion rate of the product 2 is not known.

**Process:**

Like the previous step, the simulation was made with both the upper limit of confidence method (UCB1) and Thompson sampling method to learn the optimal price of sale for the first item and compare its performance.The conversion rate of product 2 is randomly generated.

The learning process is the same as described for Step 3.

**Learning Results:**

* Thompson Learner converges to price 55$ for product 1
* UCB learner converges to price 50$ for product 1

**Profit**

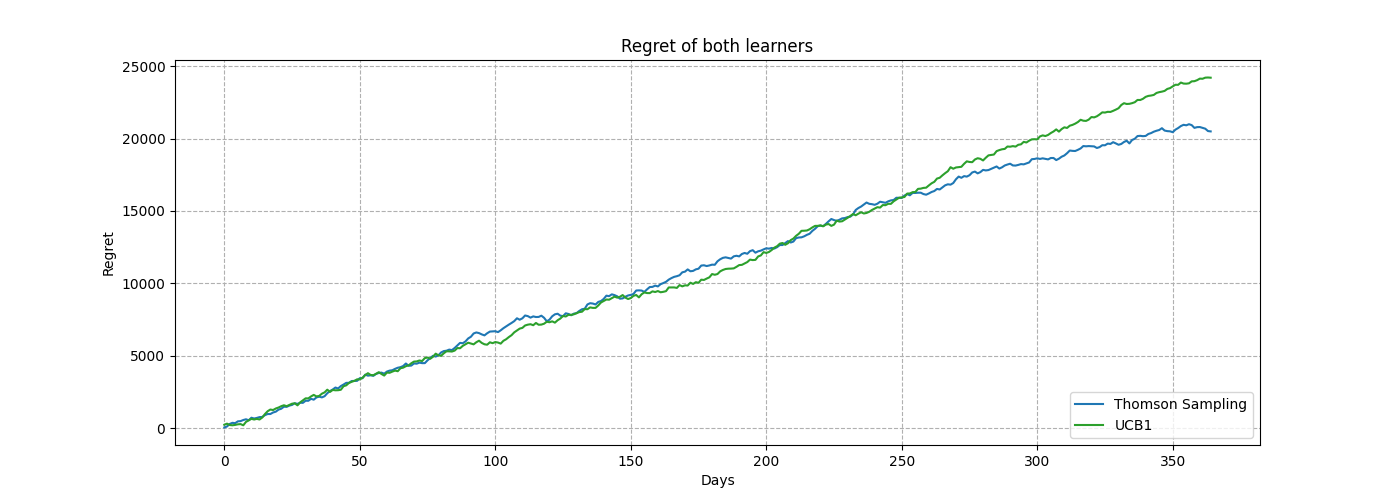
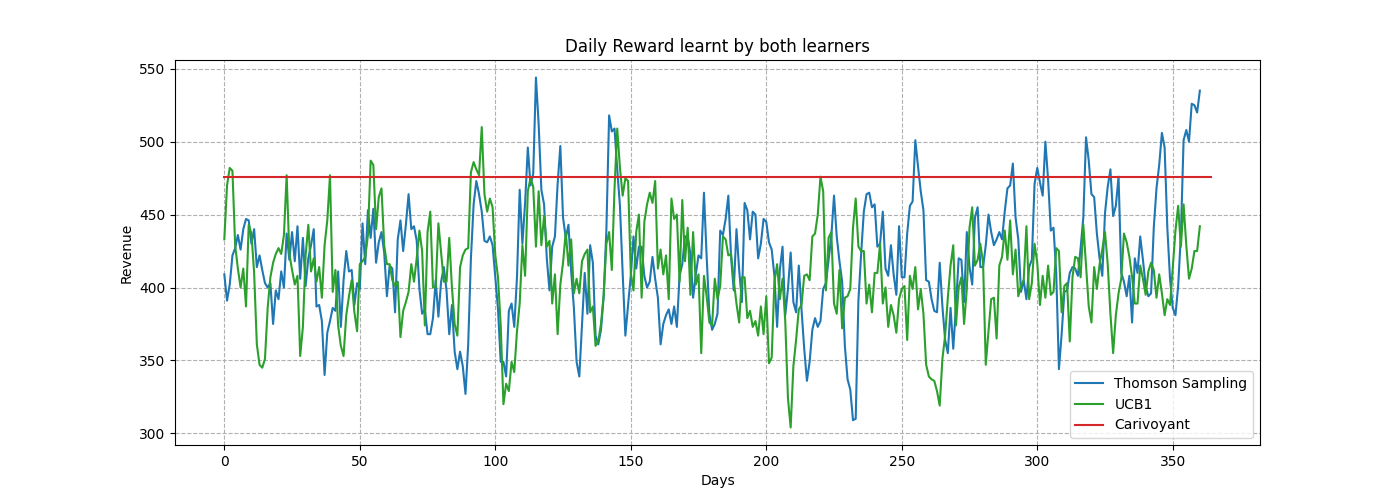
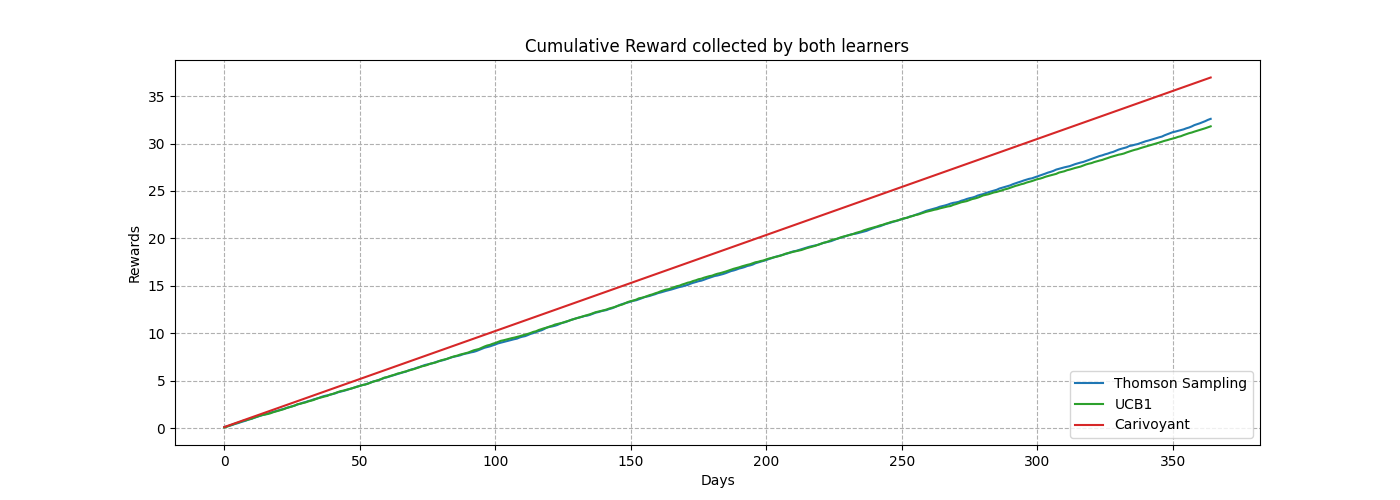
* Thompson Sampling: 32.60425531914895 $
* UCB: 31.814893617021287 $

**Regret**

* Thompson Sampling: -24.024999999999977 $
* UCB: -14.024999999999977 $

**Learning Process Charts:**

The following charts show the cumulative rewards, regrets, and the daily advancement of the two algorithms. The cumulative reward collected by the Thompson sampling algorithm is still slightly higher than the one collected by UCB algorithm.



The clairvoyant is the optimal result for the process algorithm is the optimal solution of the problem calculated when all the parameters in the problem are known from the start. According to the data, the solution produced by Thompson's sampling algorithm is closer to the optimal solution.

# 

# Step 5. Promo Optimal Assignment (Promotion Distribution)

The goal is to find the best distribution of promotions for the customer categories, the non-specified parameters are the same as the ones used for Step 4 (conversion rates must be learnt).

**Process:**

The method is similar to the previous step, the difference is that the promotions distribution is being calculated using a hungarian algorithm to get the best distribution of promotions. For each day, the number of promotions are based on the number of customers, then the matching is solved using the Hungarian algorithm, and then assigned until no more promotions of type can be assigned, or all the customers of the class have a promotion assigned, repeating the sorting until there is a evident assignation of the promotions (for example assign the last 10 promotions to last 10 users of a class). Based on the assignment, learners will be updated with the observed rewards.

**Learning Results**

One of the asignements given by the matching:

**Predictions:**

* Thompson Learner converges to price 55$ for product 1
* UCB learner converges to price 50$ for product 1

**Profit:**

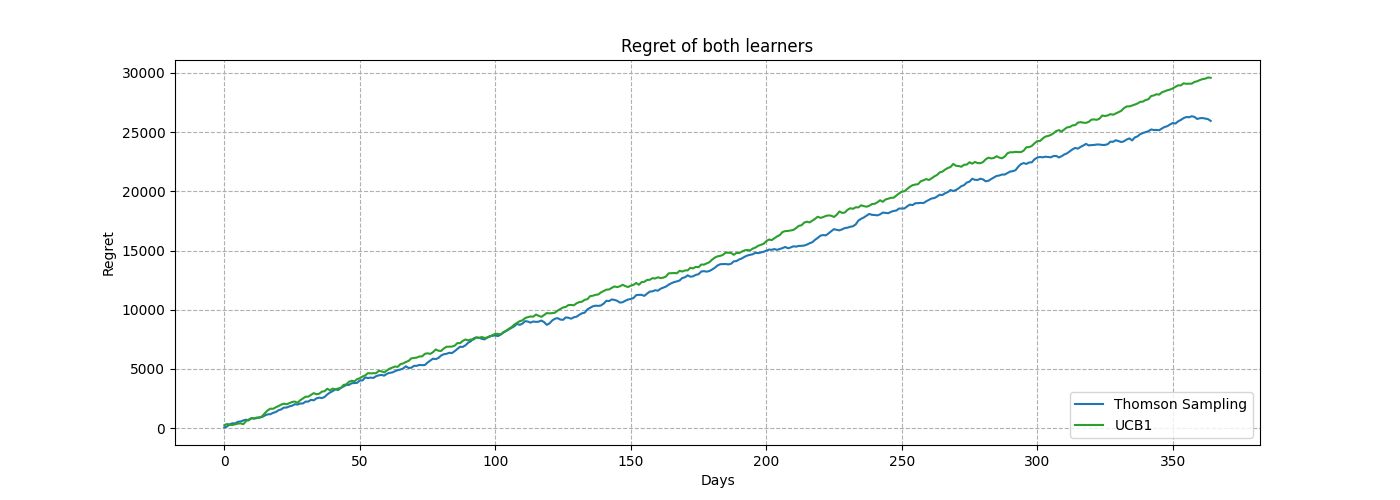
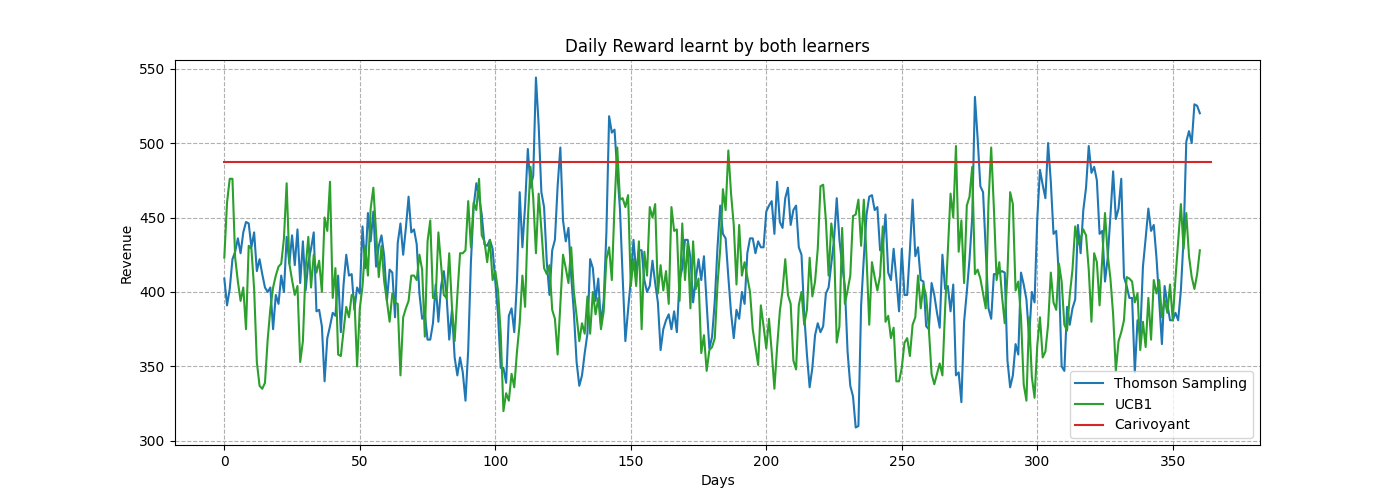
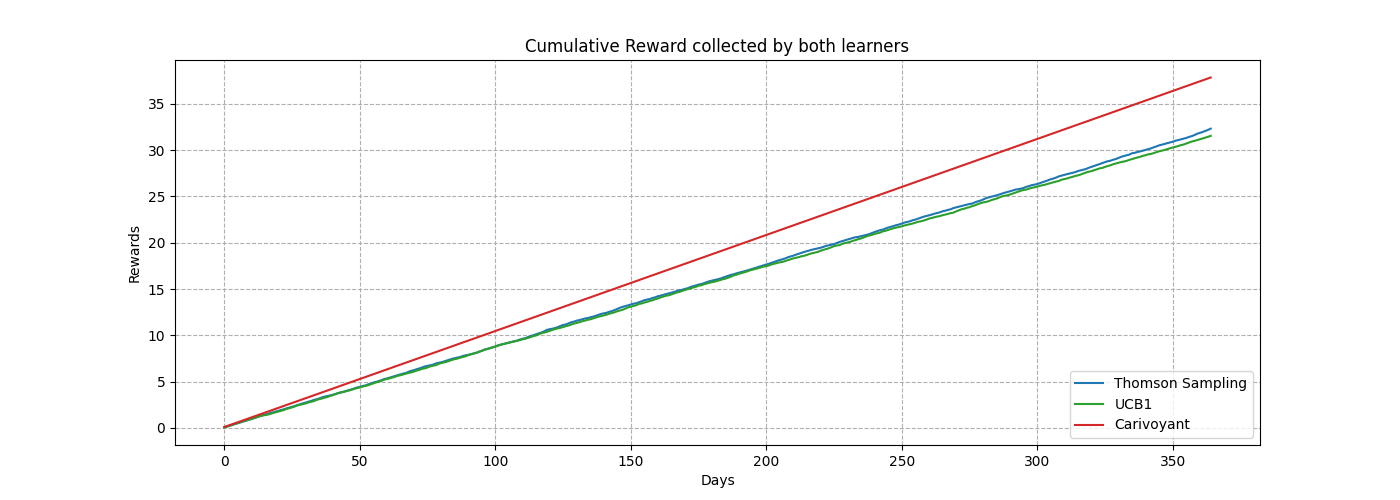
* Thompson Sampling: 32.314893617021255 $
* UCB: 31.539361702127657 $

**Regret:**

* Thompson Sampling: -152.79299999999995 $
* UCB: -22.79299999999995 $

**Learning Process Charts:**

The following charts show the cumulative rewards, regrets, and the daily advancement of the two algorithms. The cumulative reward collected by the Thompson sampling algorithm is still slightly higher than the one collected by UCB algorithm.



# 

# Step 6. Pricing and Matching (Price 2 not fixed)

The goal is the same as the previous step, but the products prices are no longer fixed, the problem now is pricing and matching and the left parameters are learnt in the same way as the Step 5.

**Process:**

The general process for learning is similar to step 5, but now the learners have to take into account another set of possible prices with an extra pair of arm learners for this purpose, making a couple of learners for each day, one for item 1 buys, and another one for item 2 buys.

**Learning Results**

**Predictions:**

* Best prices learnt by Thompson Sampling:
  + Item 1: 55 $
  + Item 2: 28 $
* Best prices learnt by UCB1:
  + Item 1: 55 $
  + Item 2: 32 $

**Profit**

* Thompson Sampling: 34.02340425531918 $
* UCB: 33.83510638297872 $

**Regret**

* Thompson Sampling: -26.817999999999984 $
* UCB: 12.182000000000016 $

**Learning Process Charts:**

The following charts show the cumulative rewards, regrets, and the daily advancement of the two sets of algorithms, mixed into the same set of rewards. The cumulative reward collected by the Thompson sampling algorithms is almost the same as the one collected by UCB algorithms.

# 

# Step 7. Sliding-Window

The goal is to make the simulation in a non-stationary environment, which means that the parameters will not remain constant over time and have seasonality. In the simulation, we placed 4 seasons along the year all for the same duration (¼ year), the rest of the parameters are the same from Step 6.

**Process:**

As the parameters change over time with some degree of sudden change, the learners will notice that their rewards may not be optimal as the year continues, and they must quickly adapt. We added the Sliding Window variation of the Thompson sampling algorithm along with the seasons over the year implementation.

**Learning Results**

**Predictions:**

* Best Global Price learnt by Thompson Sampling:
  + Item 1: 55 $
  + Item 2: 32 $
* Best Global price learnt by UCB1:
  + Item 1: 55 $
  + Item 2: 32 $
* Best Global price learnt by Sliding Window:
  + Item 1: 55 $
  + Item 2: 32 $

**Profit**

* Thompson Sampling: 44.71063829787236 $
* UCB: 48.54042553191484 $
* Sliding Window: 55.48723404255319 $

**Regret**

* Thompson Sampling: 638.6819999999999 $
* UCB: 840.682 $
* Sliding Window: 406.6819999999998 $

**Learning Process Charts:**

The following charts show the cumulative rewards, regrets, and the daily advancement of the 3 sets of algorithms, mixed into the same set of rewards. The cumulative reward collected by the Sliding Window Thompson Sampling converges and gets better results than the other 2 algorithms, not designed for the task.

# 

# Step 8. Change Detection

The goal and setup of this step is the same as on step 7, but using a change detection algorithm instead of Sliding Window.

**Process:**

This step requires the use of an algorithm to detect changes, so CUSUM (Cumulative Sum) algorithm is being used. The basic idea of the CUSUM algorithm is to use the function of the observed sample as the step size of the random walk. This random walk is designed to have a positive average drift after a change point, and a negative average drift if there is no change. Therefore, if the random walk exceeds a certain positive threshold, CUSUM will send a change signal.

**Learning Results**

**Predictions:**

* Best Global Price learnt by Thompson Sampling:
  + Item 1: 55 $
  + Item 2: 32 $
* Best Global price learnt by UCB1:
  + Item 1: 55 $
  + Item 2: 32 $
* Best Global price learnt by Cumulative Sum:
  + Item 1: 40 $
  + Item 2: 20 $

Profit:

Profit:

**Profit**

* Thompson Sampling: 53.14851063829788 $
* UCB: 46.62468085106384 $
* Change Det. 25.29340425531913 $

**Regret**:

* Thompson Sampling: 354.88200000000006 $
* UCB: 368.8820000000003 $
* Change Det. 1099.882 $

**Learning Process Charts:**

The following charts show the cumulative rewards, regrets, and the daily advancement of the 3 sets of algorithms, mixed into the same set of rewards. The cumulative reward collected by the Thompson Sampling, and UCB1 algorithms converges slower, but gets better results than the change detection algorithm, which got static in the bottom of the carts before each season change algorithms, not designed for the task.

