Pricing and Advertising

Project for the Data Intelligence Application Class
PoliMi 2019-2020

Mattia Bacarella - 928345
Francesco Corda - 920212
Davide Damato - 920616
Luciano Franchin - 921093
Luca Massini - 945027

September 30, 2020

Objective of the Project

The goal is to model a scenario in which a seller exploits advertising tools to attract more and more users to its website, thus increasing the number of possible buyers. The seller needs to learn simultaneously the conversion rate and the number of users the advertising tools can attract.

Project Structure

- Repository:
  https://github.com/francescocorda/dia-project
- Structure:

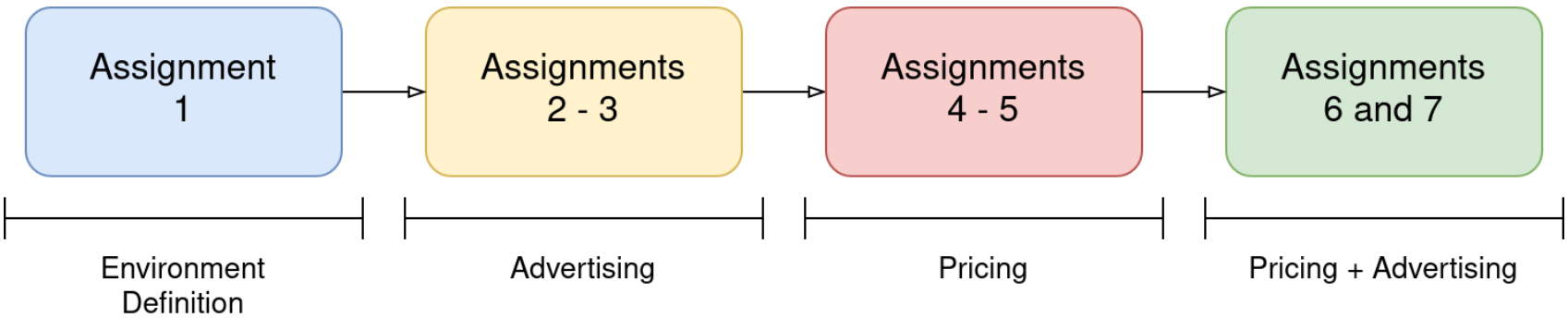
```
dia_project
├── assignment_2
├── assignment_3
├── assignment_4
├── assignment_5
├── assignment_6
├── assignment_7
├── data
└── utils
```

- The project is organized in modules, one for each assignment. To execute the code and reproduce our results, run the file assignment_x.py in each module.

Project Roadmap

After the definition of the working environment, the project can be divided in three main sections:

- a first part dedicated to advertising the product, with the goal of maximizing the traffic (expressed as number of clicks) directed to our web page
- a second part dedicated to the pricing, with the goal of finding the optimal selling point to obtain the highest revenue
- a third part in which the actions are combined, the algorithm learns simultaneously to advertise and sell



Assignment 1

Imagine one product to sell:

- *three classes of users, where, for every user, we can observe the values of two binary features (feel free to choose the features and their domains);*
- *the conversion rate curve of each class of users;*
- *three subcampaigns, each with a different ad, to advertise the product, and each targeting a different class of users;*
- *there are three abrupt phases;*
- *for every abrupt phase and for every subcampaign, the probability distribution over the daily number of clicks for every value of budget allocated to that subcampaign.*

Assignment 1 - The Product

We decided to analyze the pricing and the advertising of a **gym membership**, a product that presents the following properties:

- it's a well-known and established item, used by a vast and diversified population of users
- it's a highly customizable product, that allows for different advertising campaigns and price offers for different classes of users
- it's strongly affected by seasonality during the year

Assignment 1 - Users

The users are characterized with a set of 2 binary features, namely:

- Age = {Under 40, Over 40}
- Fitness Level = {Novice, Experienced}

We decided to focus on the following 3 classes generated by the features:

- (Under 40, Novice)
- (Under 40, Experienced)
- (Over 40, Experienced)

We excluded the class of (Over 40, Novice) users because of the online nature of our advertising campaign: an adult person which is not already interested in fitness is less likely to look on the internet for arguments related to our campaign.

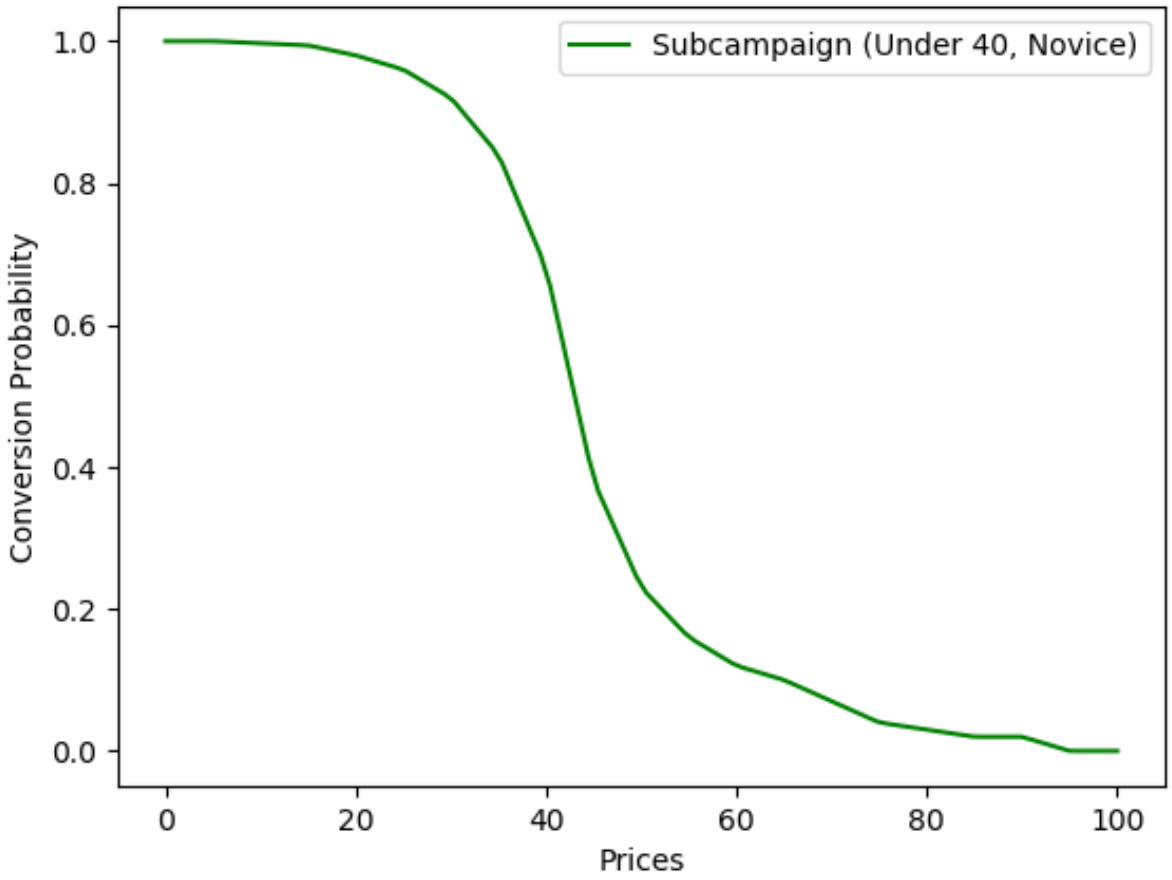
Assignment 1 - Conversion Rate Curves

Given a domain for the prices of a gym subscription that ranges from € 0 to € 100, we identified 3 curves with the following characteristics:

- all curves present a monotonically decreasing trend, starting from a conversion rate of 1 and ending towards 0
- to add realism to the curves, the conversion rates maintain values close to 1 until a minimum price threshold is reached

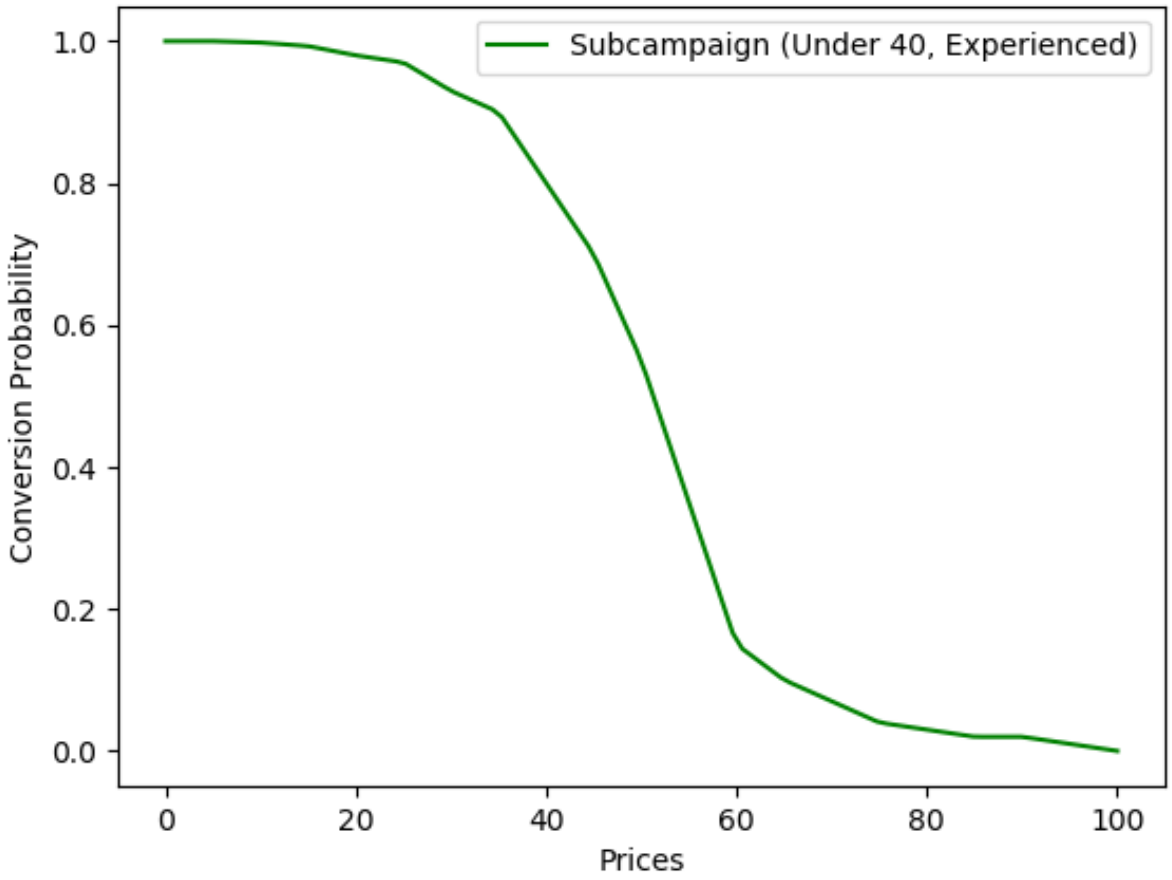
The classes differ in their sensitivity to price increases, according to principles that a novice is in general less interested in an expensive item than an expert and that an adult has on average an higher purchasing power than a young person.

Assignment 1 - (Under 40, Novice) class



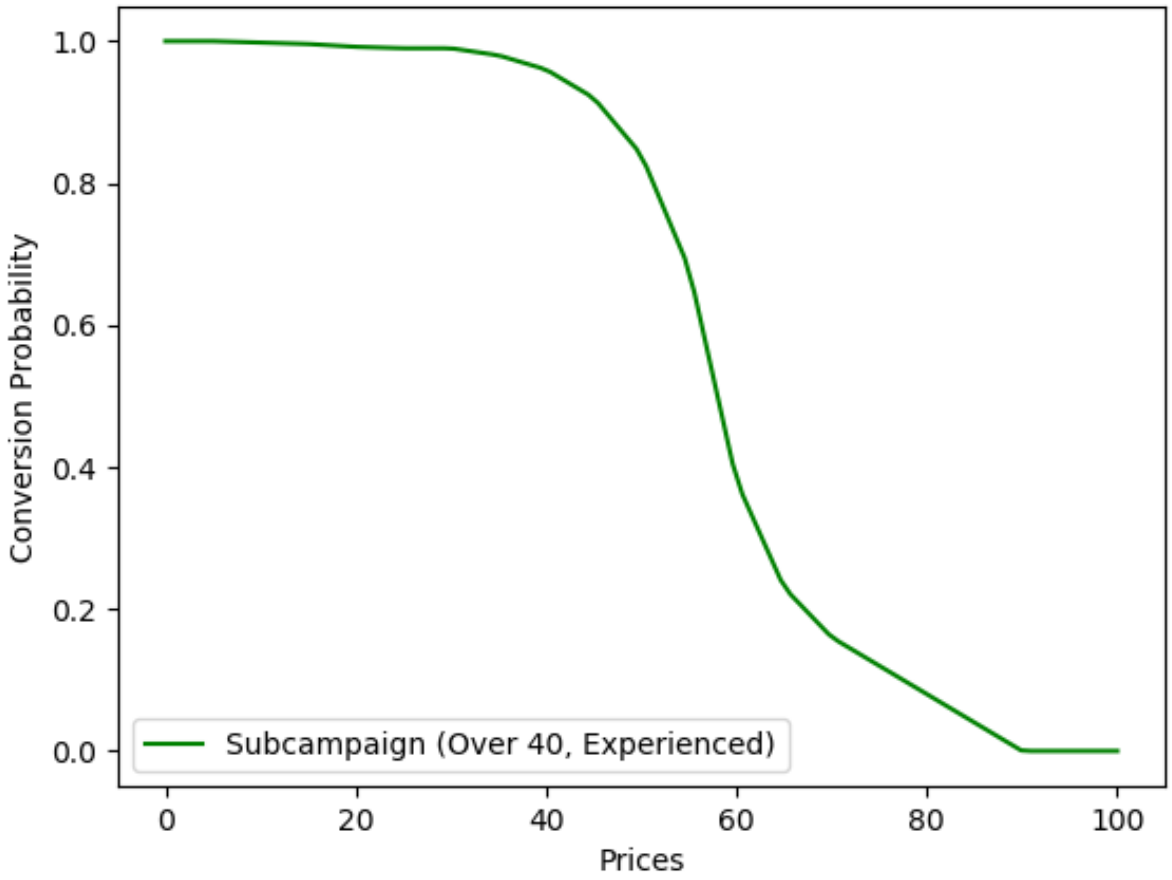
Fast decrease.

Assignment 1 - (Under 40, Experienced) class



Intermediate decrease.

Assignment 1 - (Over 40, Experienced) class

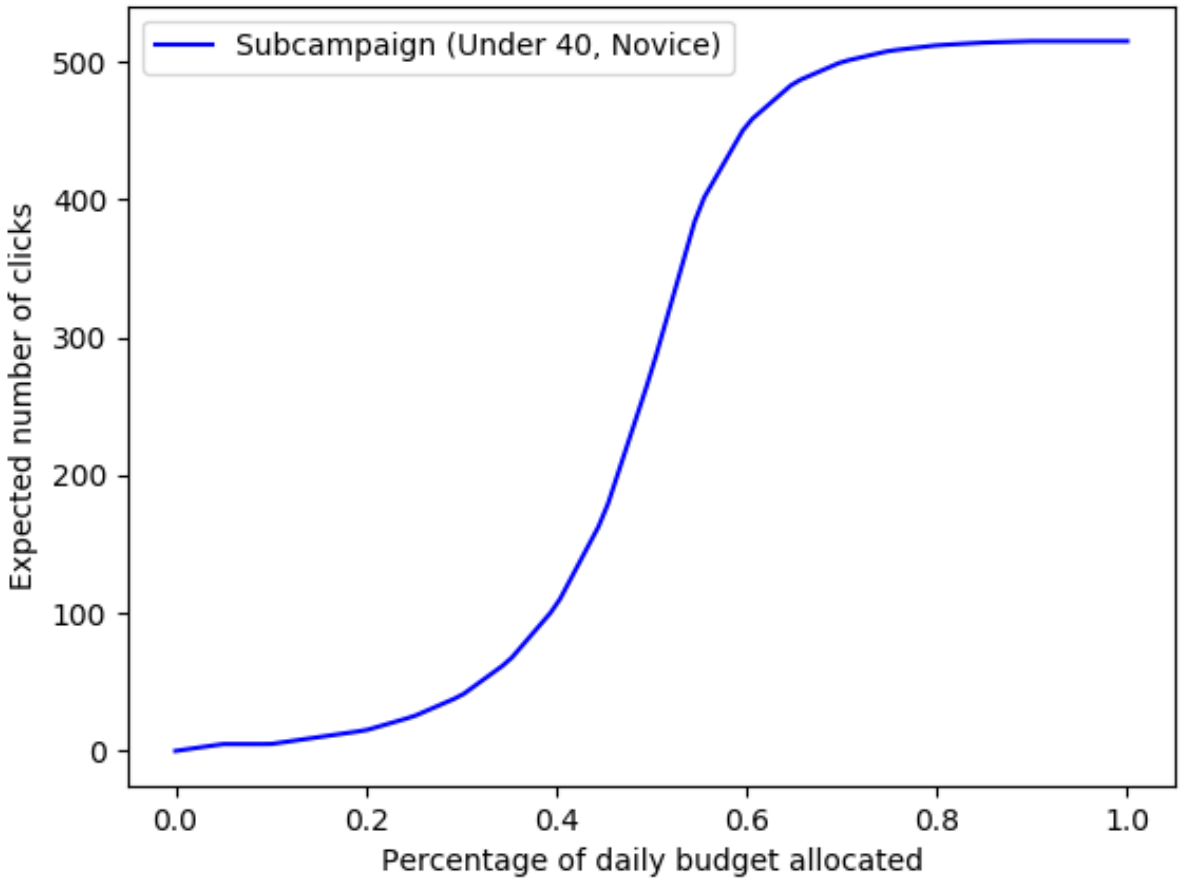


Slow decrease.

Assignment 1 - Subcampaigns

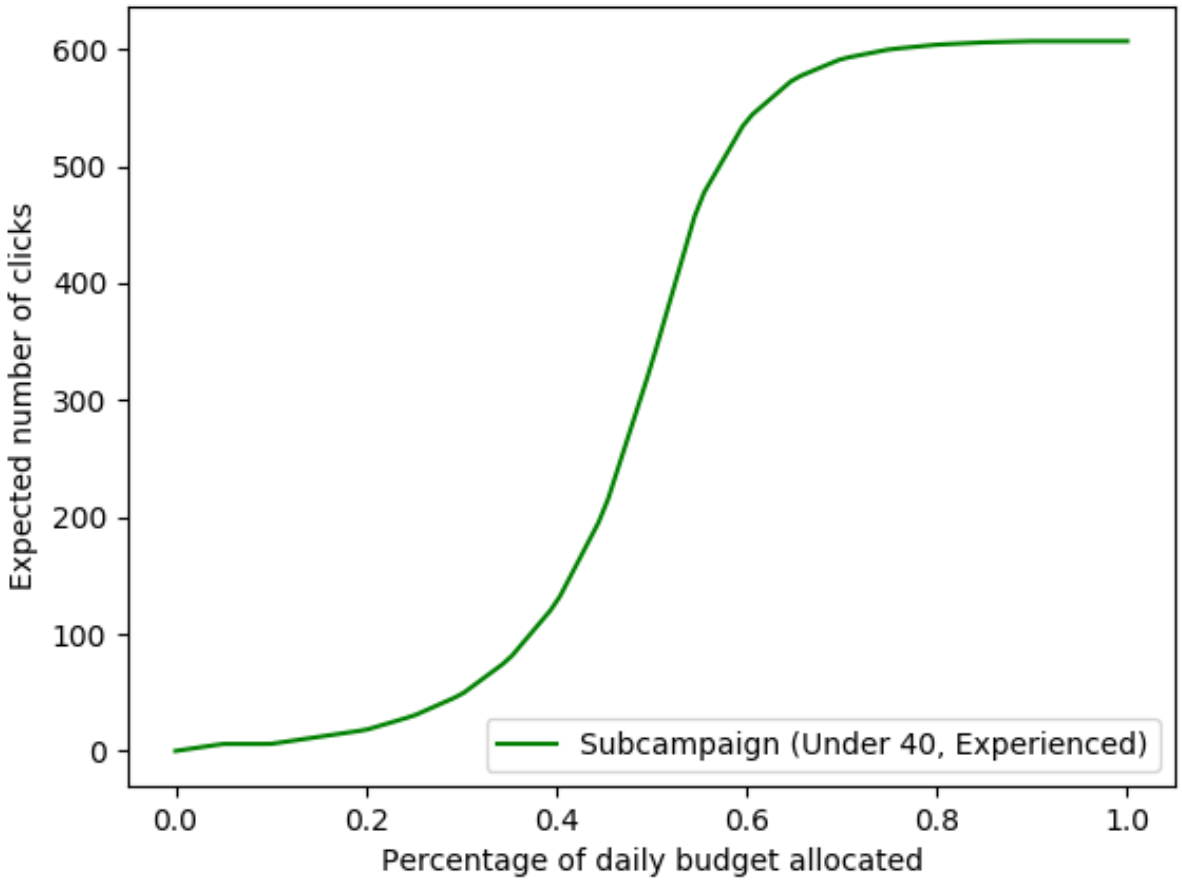
Each class is the object of interest of a web marketing subcampaign. One goal of the project is the allocation of the advertising daily budget to maximize the total number of clicks.
To do this, we identified 3 curves that describe this number as a function of the portion of the budget allocated to the campaign (from 0% to 100%).
All curves increase monotonically with the budget spent, until they reach a plateau in which allocating more money to the campaign does not produce results.
The campaigns differ for the maximum number of clicks reached and the sensitivity to the allocated budget.

Assignment 1 - Subcampaign 1



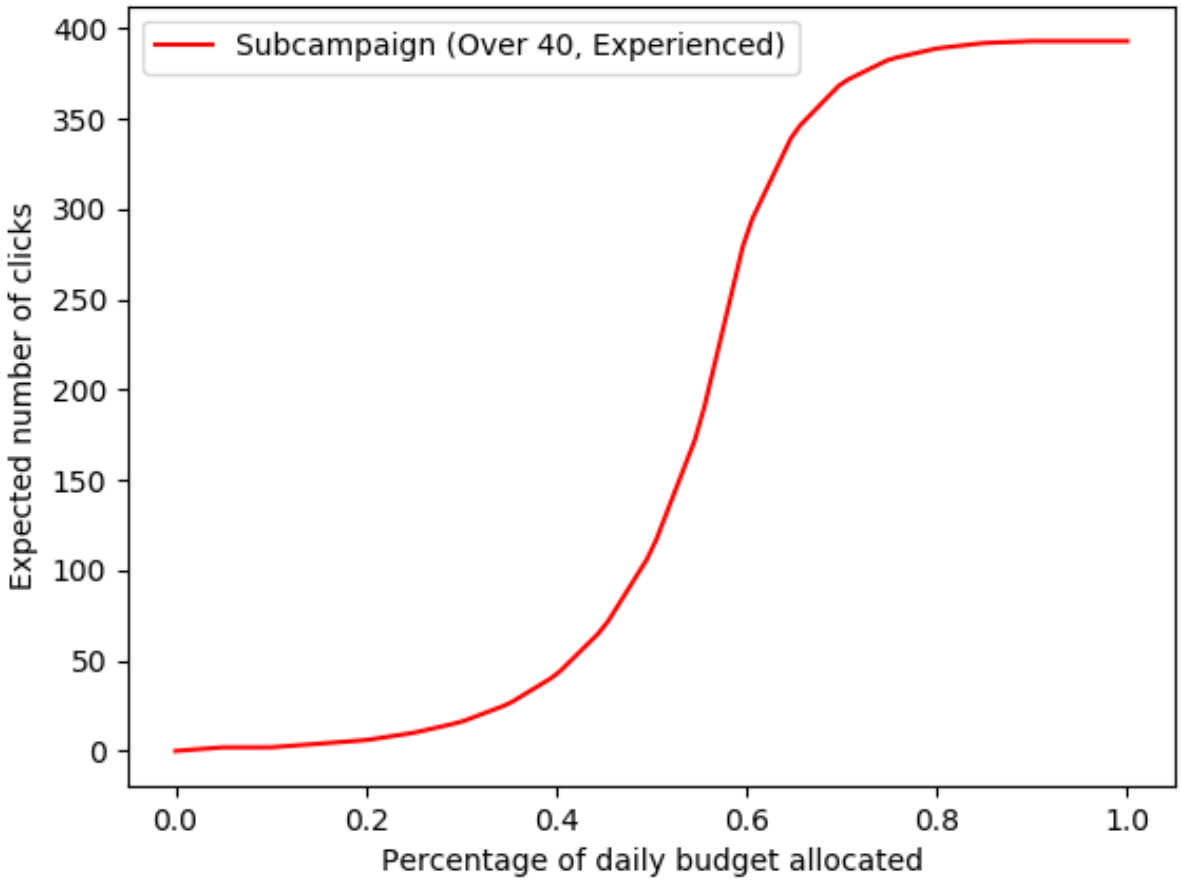
Fast increase, medium value plateau.

Assignment 1 - Subcampaign 2



Fast increase, high value plateau.

Assignment 1 - Subcampaign 3



Slow increase, small value plateau.

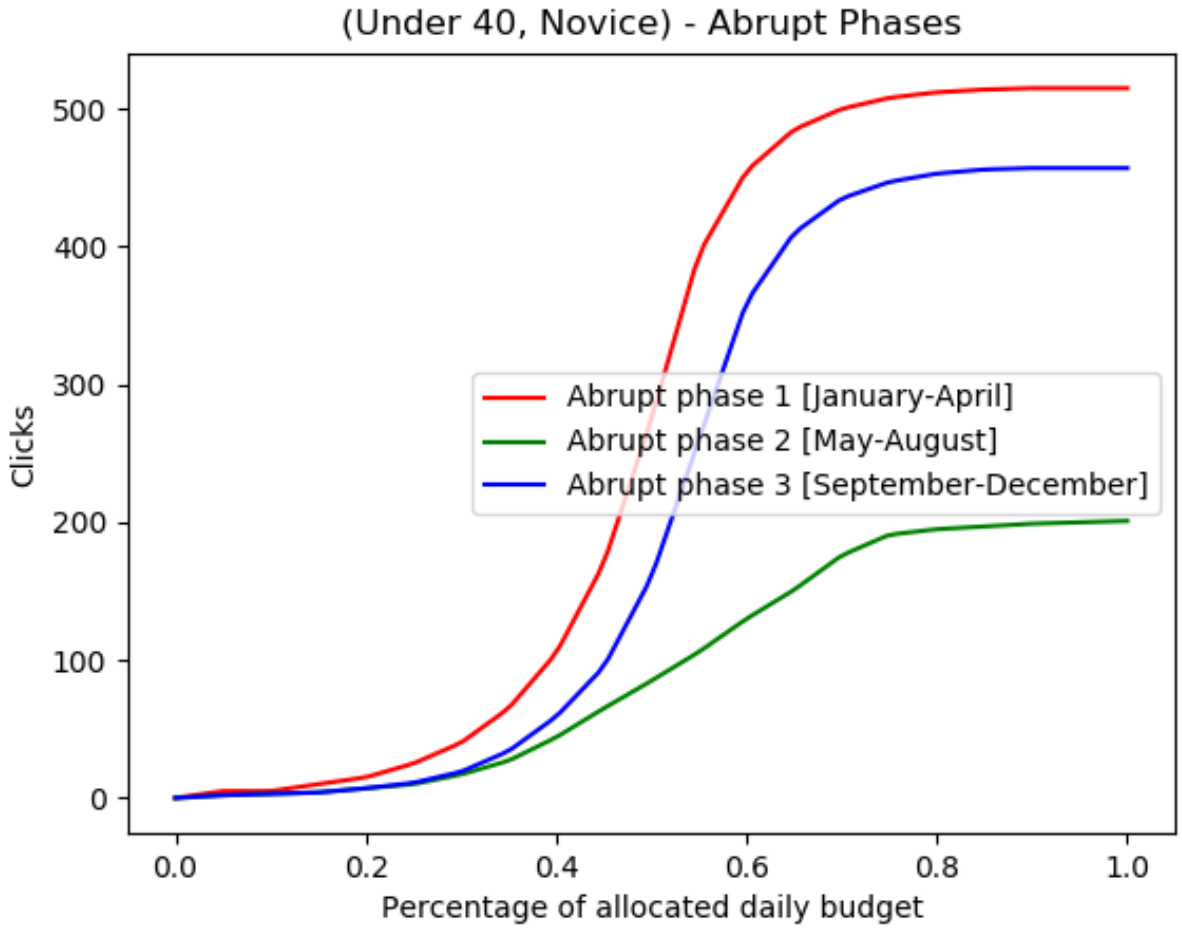
Assignment 1 - Abrupt Phases

We identified 3 abrupt phases influenced by the seasonality that characterizes gym subscriptions during a year:

1. [January-April], the "New Year's Resolutions" phase, it's the period with the highest interest
2. [May-August], the "Swimsuit Season" phase, in which the least number of users is attracted
3. [September-December], the "Back to Normal" phase, with an intermediate level of activity

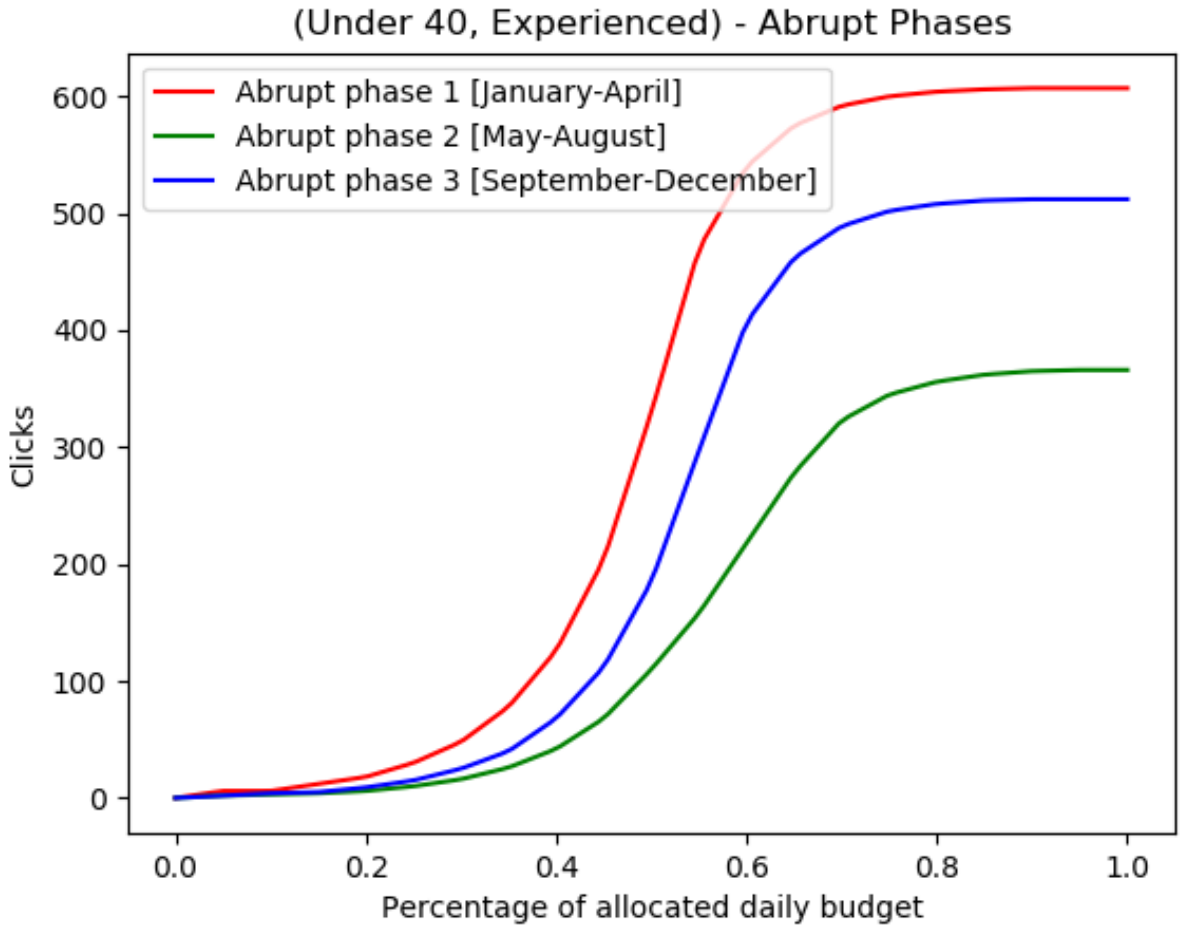
Each period has the same length of 4 months and is denoted by almost-stationary behaviors of the users for its duration. The classes are influenced by the abrupt phases in different measures.

Assignment 1 - Subcampaign 1, Abrupt Phases



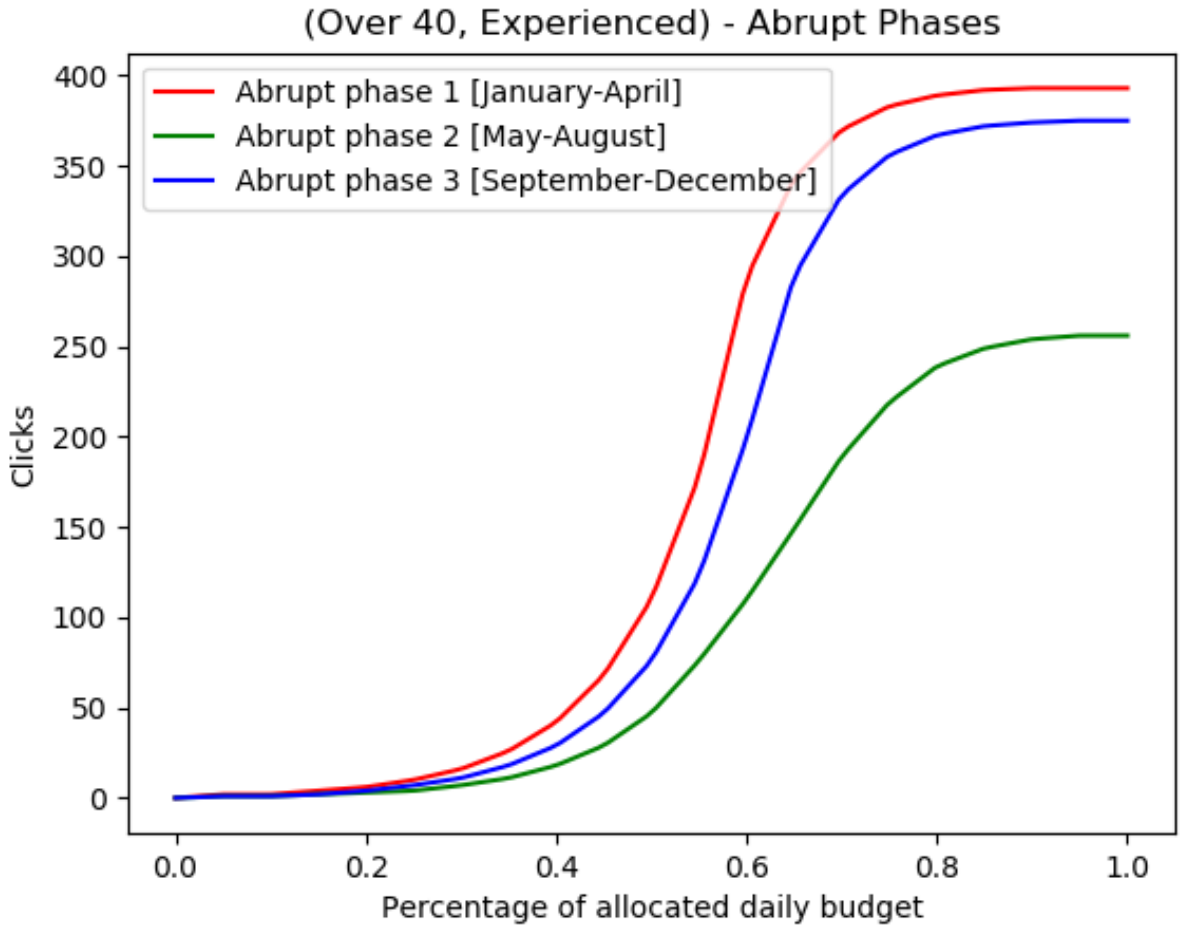
The (Under 40, Novice) class has a medium interest in the product during the year, with a significant drop during Summer.

Assignment 1 - Subcampaign 2, Abrupt Phases



The (Under 40, Experienced) class has the highest interest in the product in all phases.

Assignment 1 - Subcampaign 3, Abrupt Phases



The (Over 40, Experienced) class has the lowest interest in the product, but is the least affected by seasonality.

Assignment 2

Design a combinatorial bandit algorithm to optimize the budget allocation over the three subcampaigns to maximize the total number of clicks when, for simplicity, there is only one phase. Plot the cumulative regret.

Assignment 2 - Pseudocode

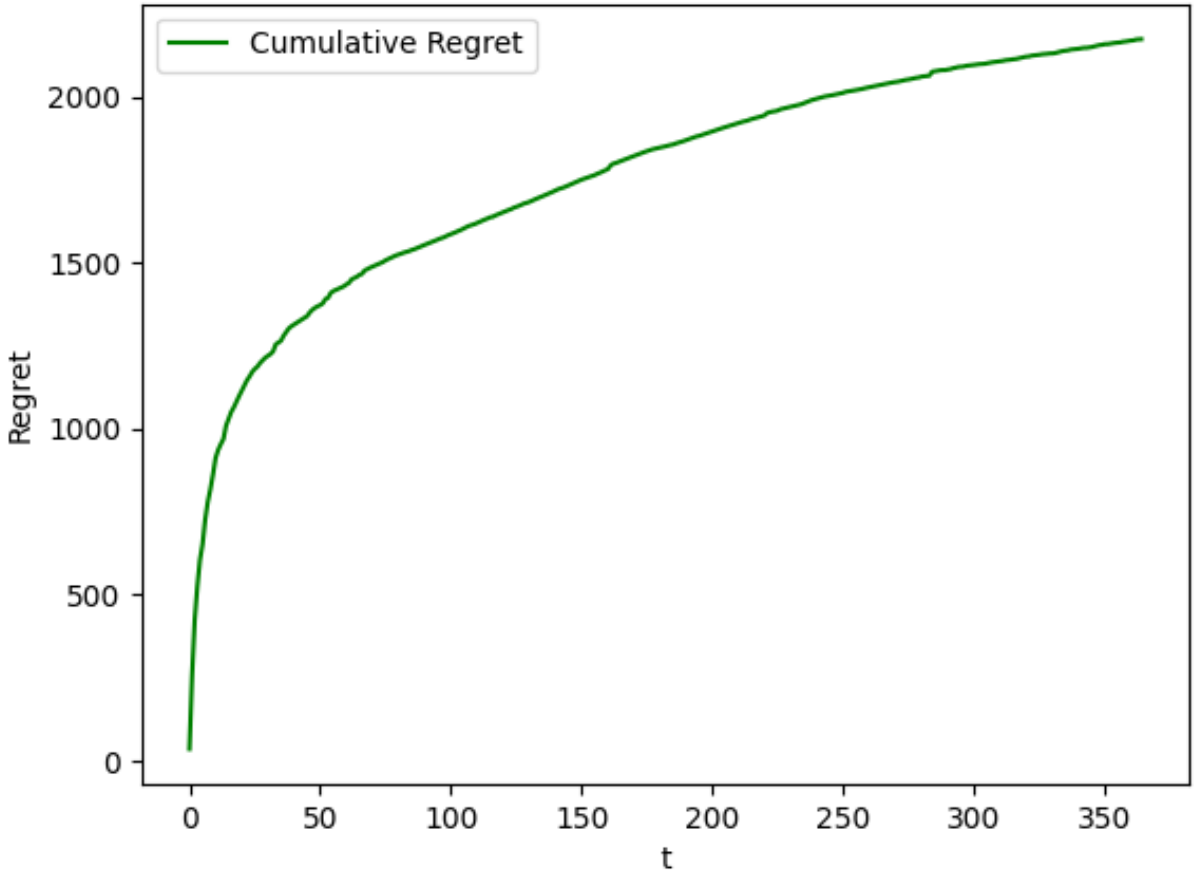
```
initialize click envs
for exp in n_experiments:
  initialize gp-ts learners
  for t in time_horizon:
    draw samples for each subcampaign
    find super-arm by solving knapsack problem
    pull arms given by super-arm
    update gp-ts learners
calculate clairvoyant value
plot rewards and regrets
```

Assignment 2 - Experiments

Setting parameters:

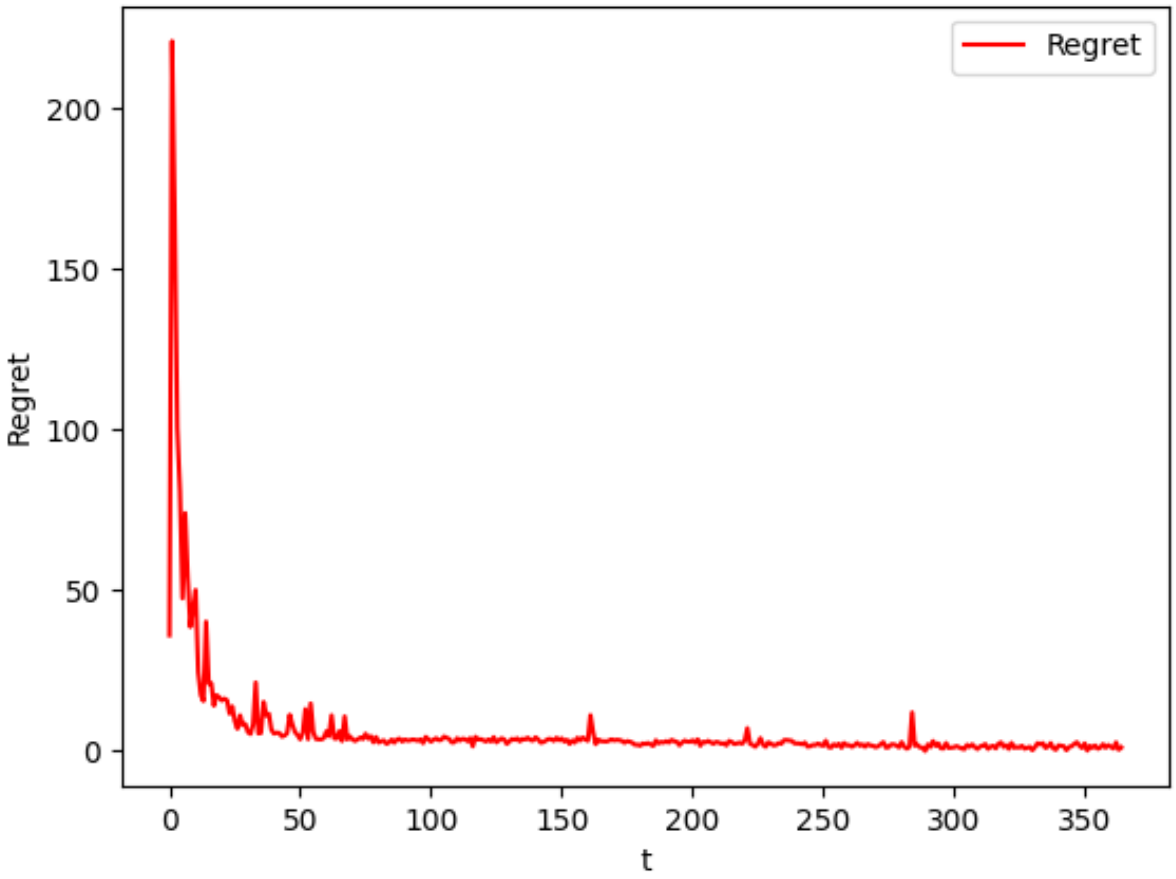
- T $= 365$
- Number of arms $= 21$
- Number of experiments $= 50$

Assignment 2 - Cumulative Regret



The cumulative regret shows a fast increase in the beginning, corresponding to a predominant exploration, and then slows down favoring the exploitation of the results.

Assignment 2 - Instantaneous Regret



The regret goes to 0 in approximately 50 steps.

Assignment 3

Design a sliding-window combinatorial bandit algorithm for the case, instead, in which there are the three phases aforementioned. Plot the cumulative regret and compare it with the cumulative regret that a non-sliding-window algorithm would obtain.

Assignment 3 - Pseudocode

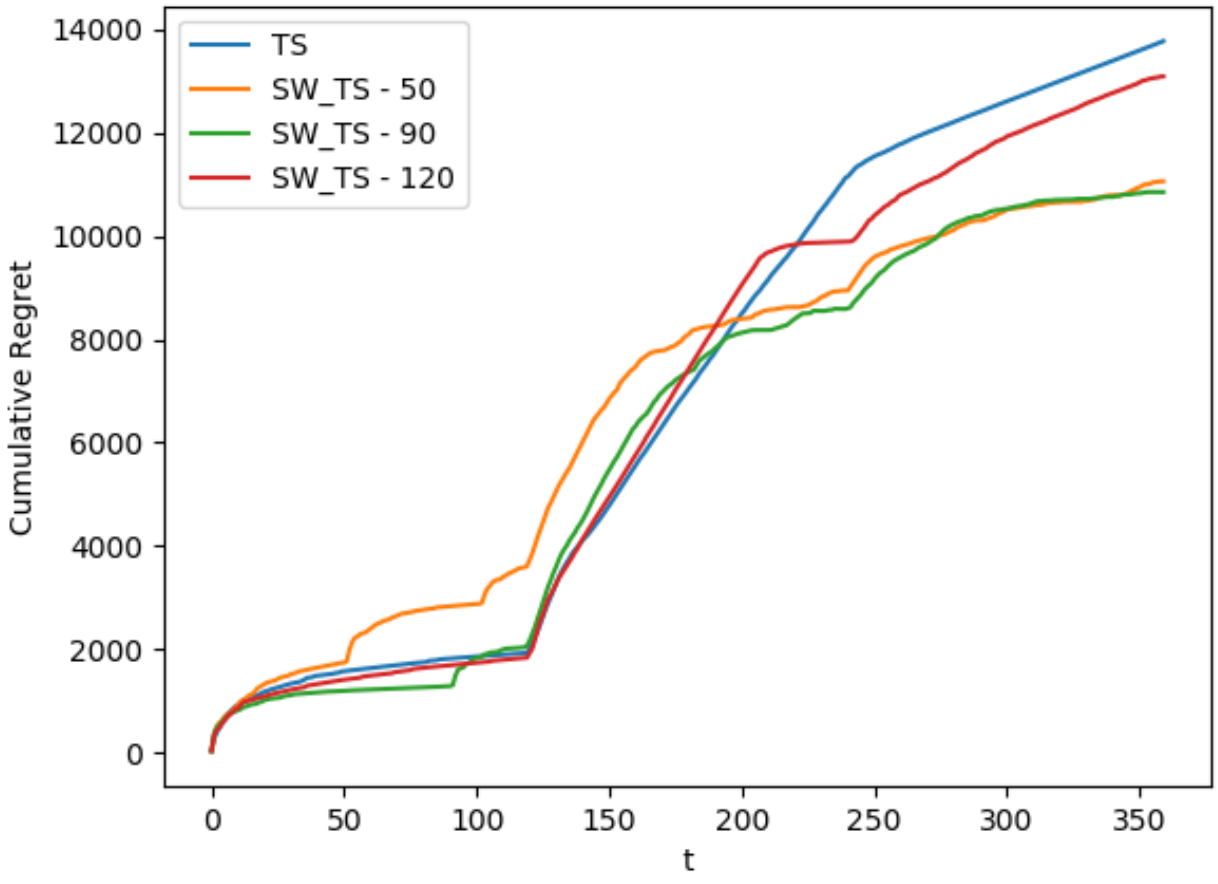
```
initialize click envs
for exp in n_experiments:
  initialize gp-ts and gp-ts-sw learners
  for t in time_horizon:
    if abprupt_phase is changed:
      reset memory of the learners
    draw samples for each subcampaign
    find super-arm by solving knapsack problem
    pull arms given by super-arm
    update gp-ts and gp-ts-sw learners
calculate clairvoyant value
plot rewards and regrets for the 2 learners
```

Assignment 3 - Experiments

Setting #1, #2, #3 parameters:

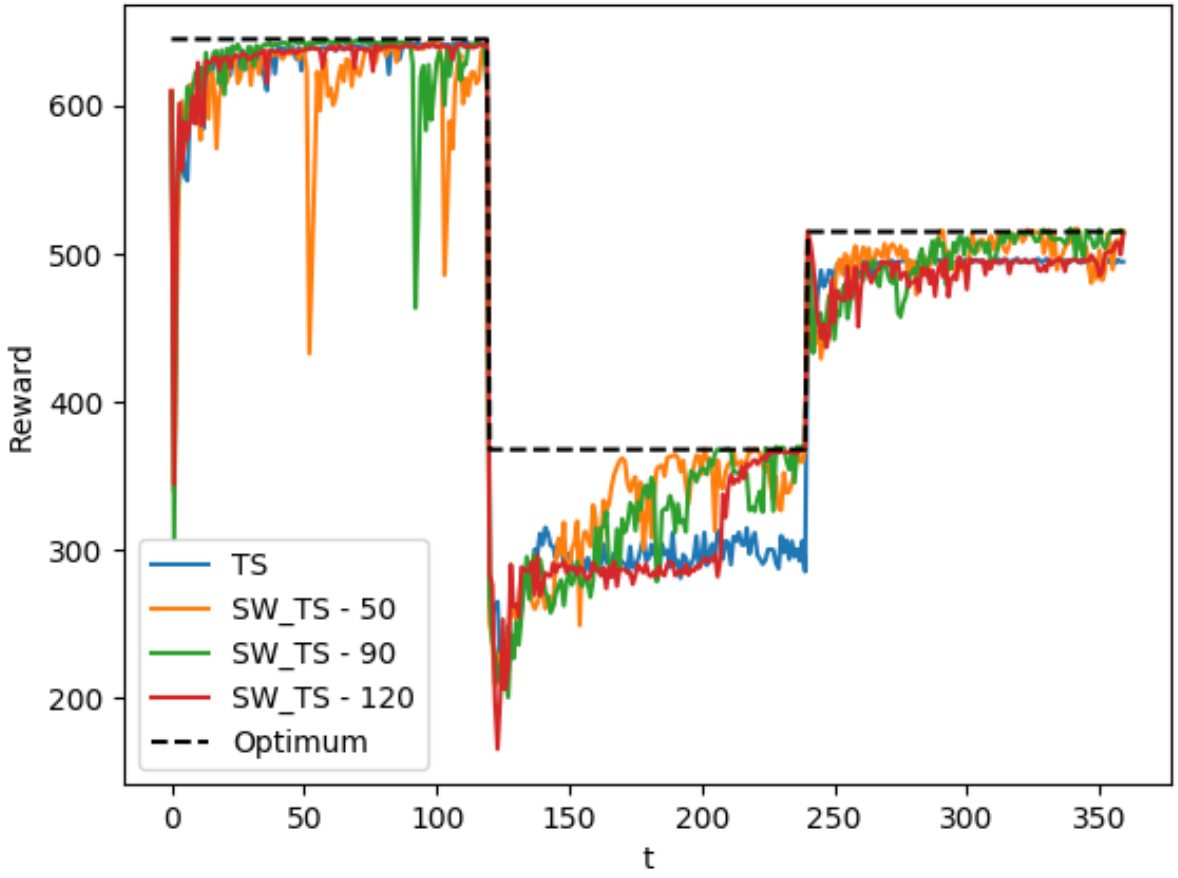
- T = 360
- Number of arms = 21
- Number of experiments = 25
- Sliding window = 50, 90, 120

Assignment 3 - Cumulative Regret



All TS-SW settings perform better than the pure TS approach. Among them, the sliding window $= 90$ has the overall best performance.

Assignment 3 - Rewards



This plot shows that the TS approach is unable to reach the optimal values in the second and third phases, while the sliding window $= 90$ always produces good results.

Assignment 4

Design a learning algorithm for pricing when the users that will buy the product are those that have clicked on the ads. Assume that the allocation of the budget over the three subcampaigns is fixed and there is only one phase (make this assumption also in the next steps). Plot the cumulative regret.

Assignment 4 - Pseudocode

```
initialize pricing envs
for exp in n_experiments:
  initialize ts and greedy learners
  for t in time_horizon:
    for subcampaign in campaigns:
      draw samples from Beta distribution (TS-learner)
      select best arm according to the TS-learner
      play arm and collect reward (TS-learner)
      update TS-learner
      select best arm according to the Greedy-learner
      play arm and collect reward (Greedy-learner)
      update Greedy-learner
calculate clairvoyant value
plot rewards and regrets for the 2 learners
```

Assignment 4 - Experiments

General parameters:

- Number of experiments $= 50$
- User class probabilities $= [\frac{1}{5}, \frac{2}{5}, \frac{2}{5}]$

Setting #1, parameters:

- T $= 365$
- Number of arms $= \lceil (T \log(T))^{\frac{1}{4}} \rceil = 8$

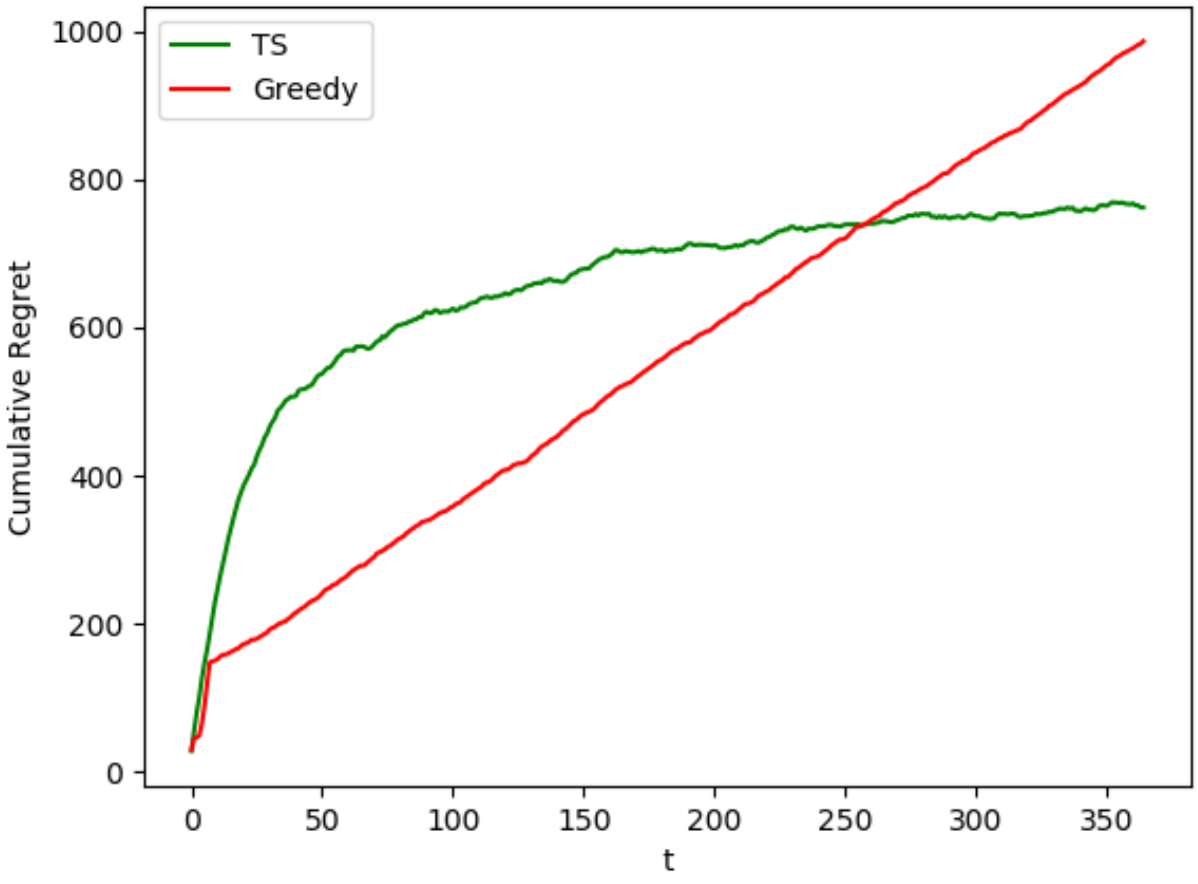
Setting #2, parameters:

- T $= 365$
- Number of arms $= 4 \times \lceil (T \log(T))^{\frac{1}{4}} \rceil = 32$

Setting #3, parameters:

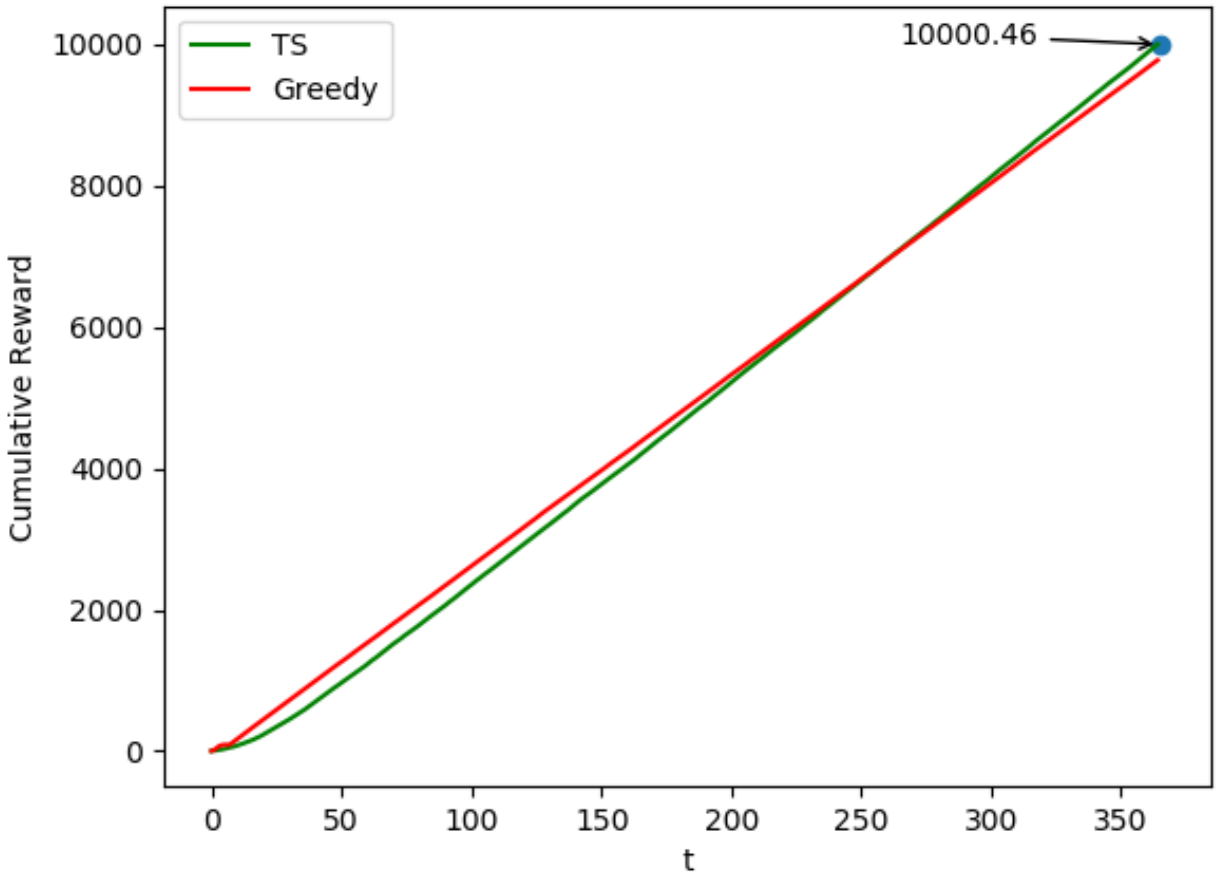
- T $= 18250$
- Number of arms $= \lceil (T \log(T))^{\frac{1}{4}} \rceil = 23$

Assignment 4 - Setting #1, Regret



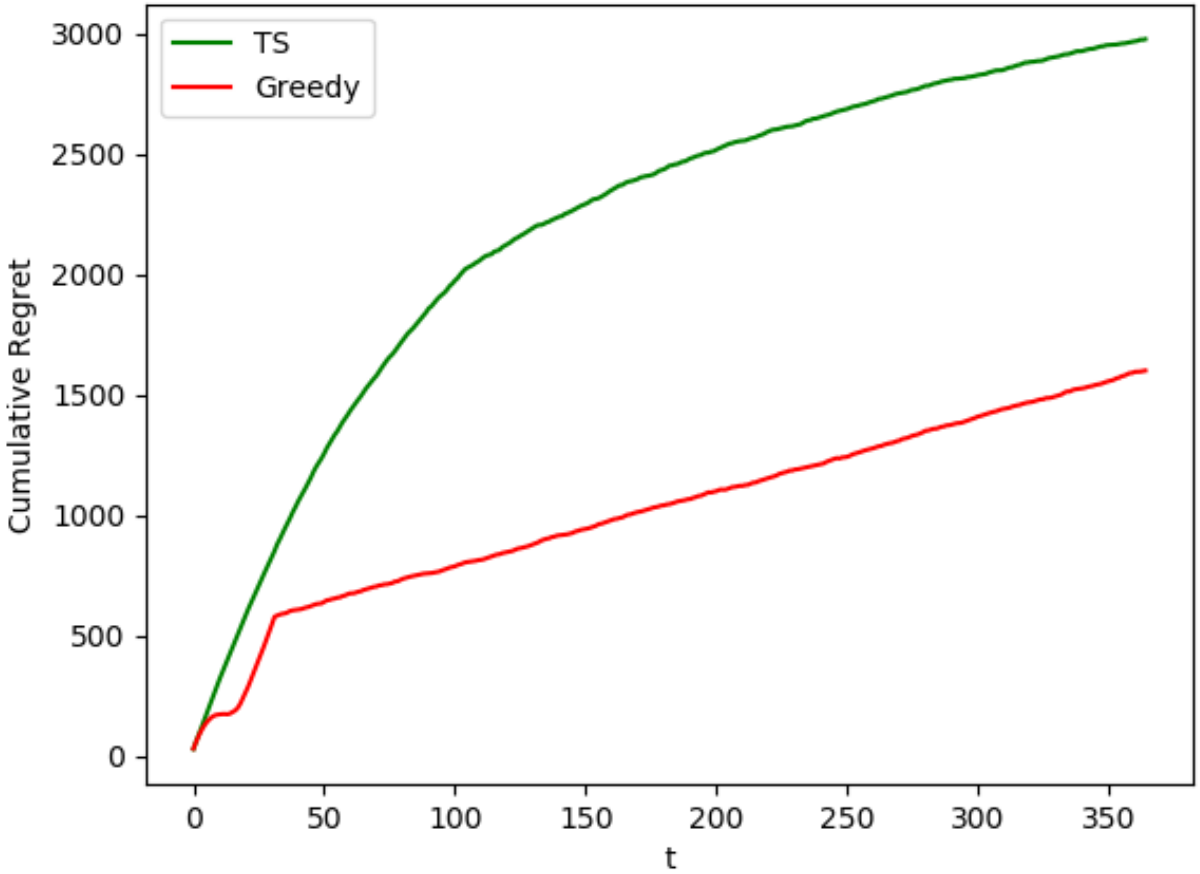
Comparison of the cumulative regrets of a TS and a Greedy approach.
The Greedy algorithm performs well at the beginning but loses in the end.

Assignment 4 - Setting #1, Reward



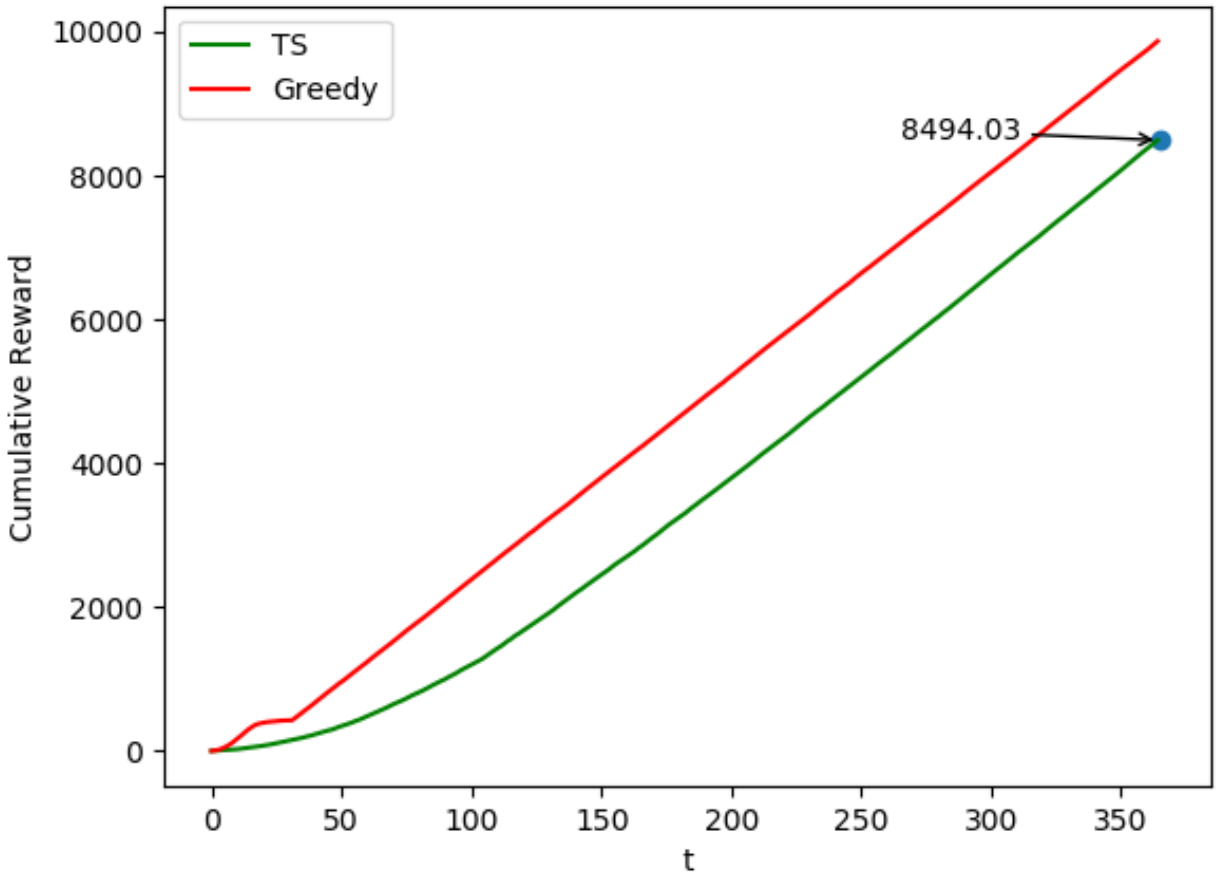
Cumulative reward collected during the learning.

Assignment 4 - Setting #2, Regret



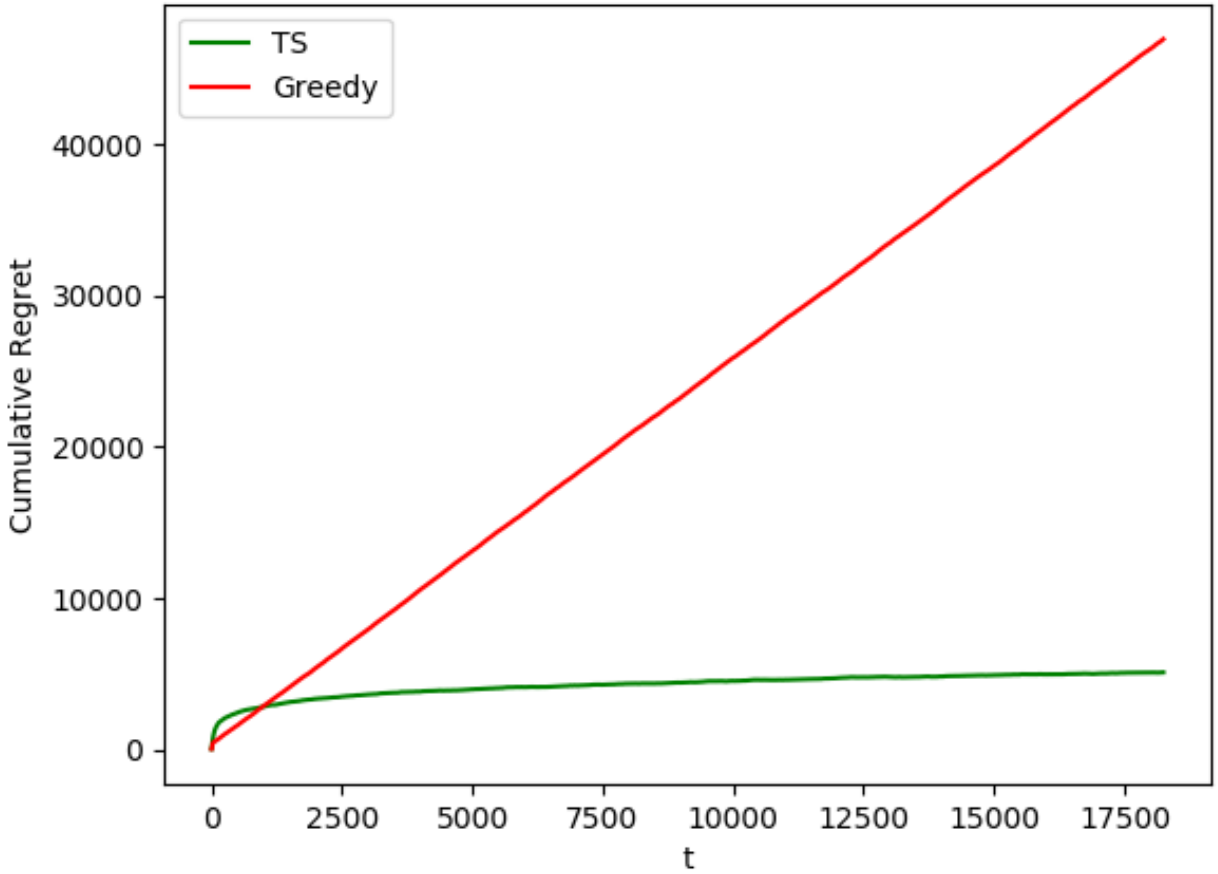
This experiment shows that selecting a large number of candidates (4x than before) doesn't guarantee better results. The regret is more than tripled and the TS approach is performing worse than the Greedy one.

Assignment 4 - Setting #2, Reward



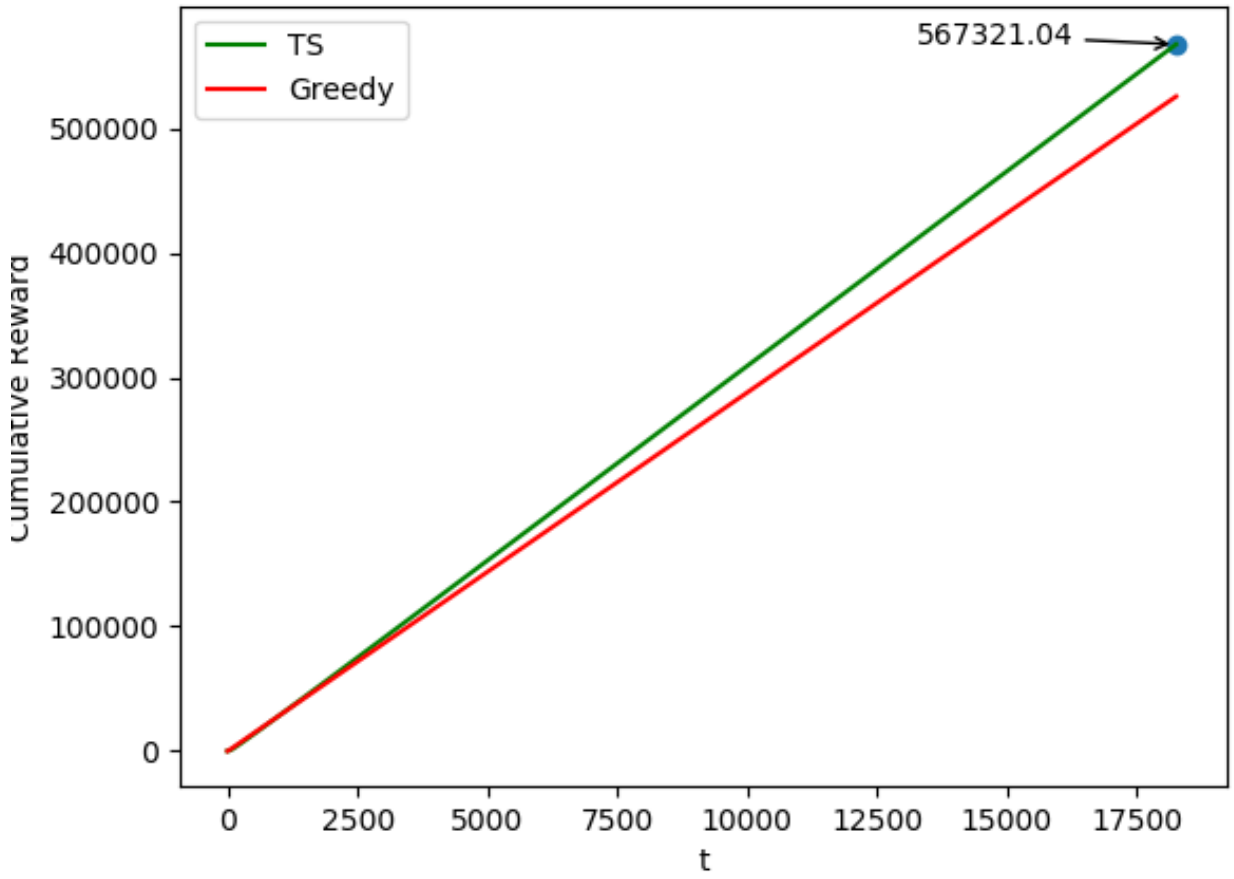
Cumulative reward collected during the learning. The TS algorithm is significantly under the Greedy algorithm.

Assignment 4 - Setting #3, Regret



This experiment aims to produce a reference for the results obtained with context generation in the next assignment. The algorithm is launched for a number of steps = 50 (updates each day) $\times$ 365 (days) = 18250.

Assignment 4 - Setting #3, Reward



Cumulative reward collected during the learning.

Assignment 5

Design and run a context generation algorithm for the pricing when the budget allocated to each single subcampaign is fixed. At the end of every week, use the collected data to generate contexts and then use these contexts for the following week. Plot the cumulative regret as time increases. In the next steps, do not use the generated contexts, but use all the data together.

Assignment 5 - Pseudocode

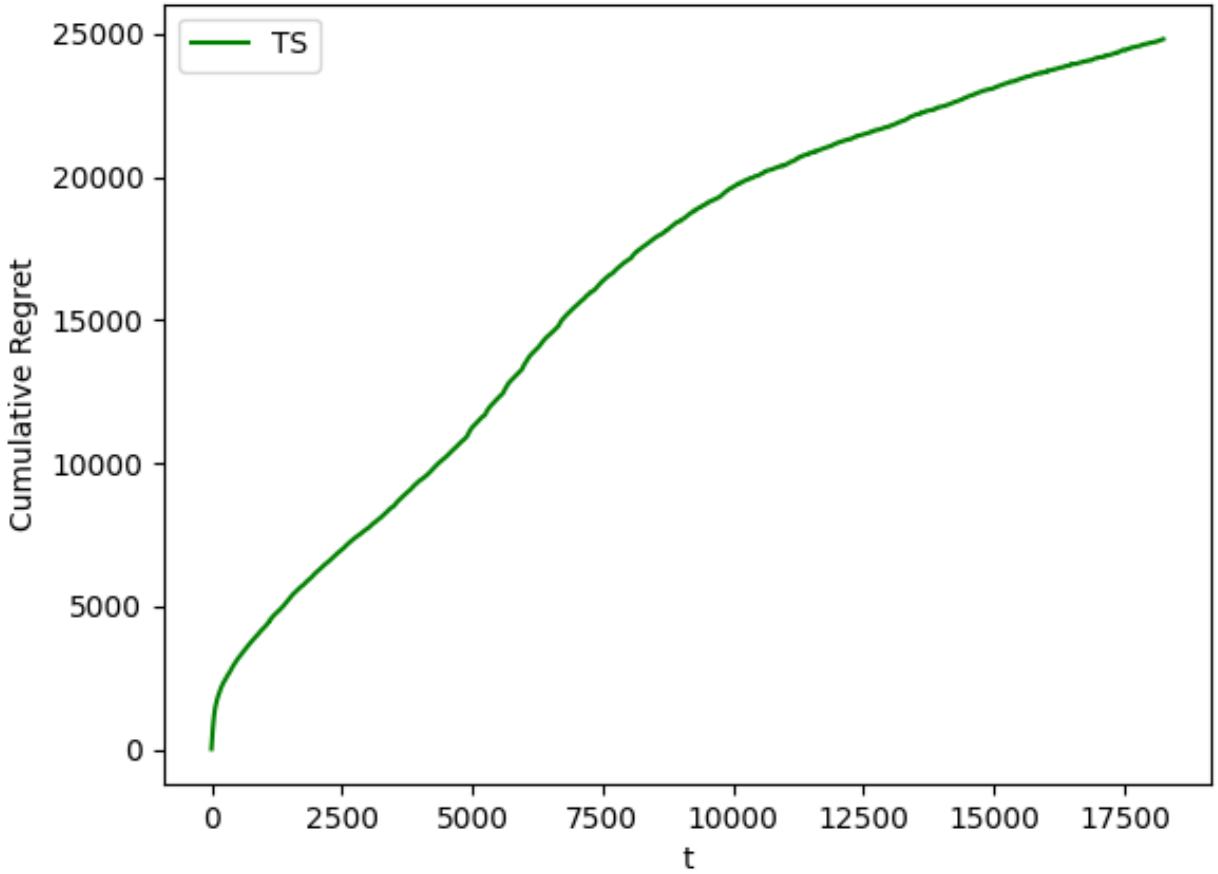
```
initialize pricing envs
for exp in n_experiments:
  initialize context generator
  for t in time_horizon:
    if week is ended:
      generate new contexts (greedy approach)
    for context in n_contexts:
      run ts algorithm
calculate clairvoyant value
plot rewards and regrets
```

Assignment 5 - Experiments

Setting #1, parameters:

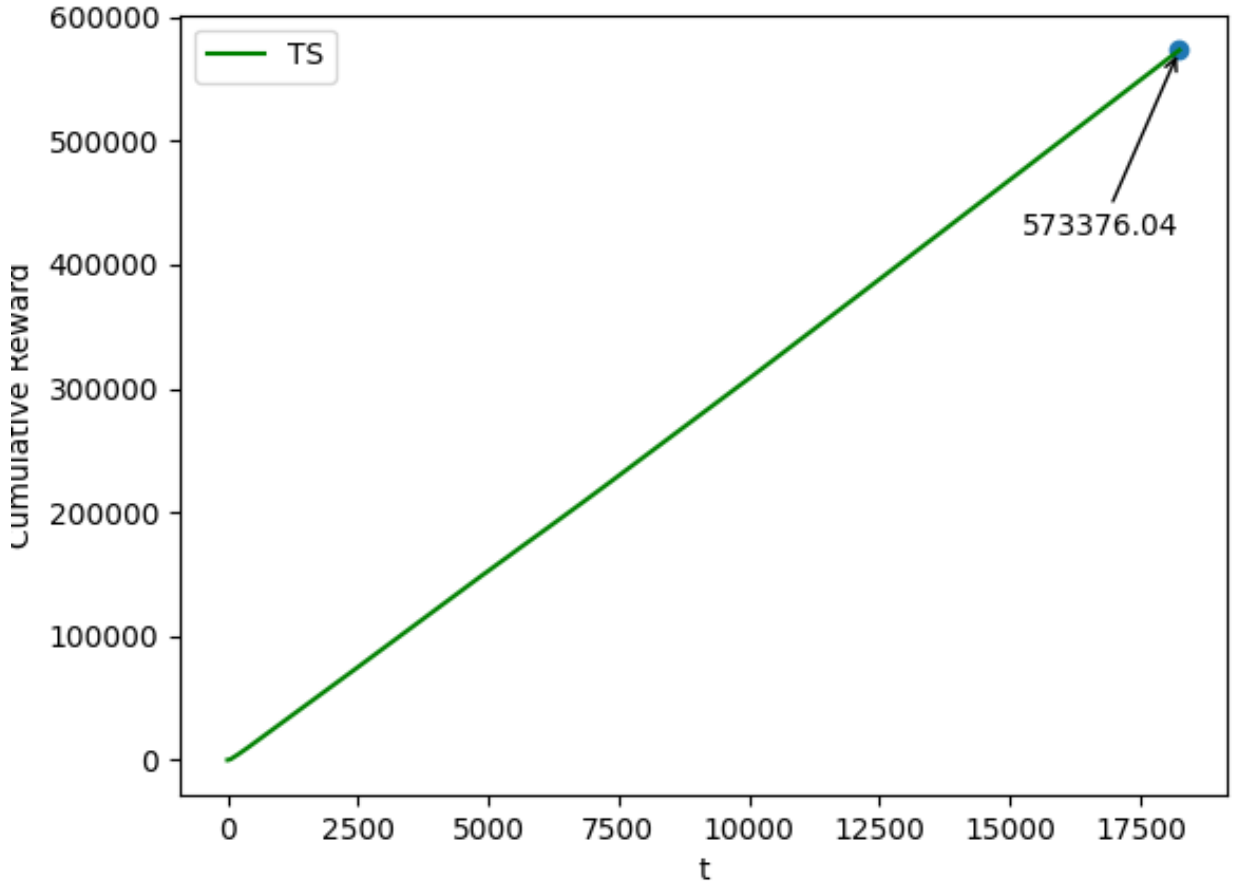
- T $= 18250$
- Number of arms $= \lceil (T \log(T))^{\frac{1}{4}} \rceil = 23$
- User class probabilities $= [\frac{1}{5}, \frac{2}{5}, \frac{2}{5}]$
- Number of experiments $= 50$

Assignment 5 - Cumulative Regret



If we compare this result with the one obtained in step 4, we can see that we obtain a worse outcome in terms of cumulative regret, but this is due to the method that we use to compute the optimum in the two cases.

Assignment 5 - Cumulative Reward



If we look instead to the cumulative reward, we can see that we get a better result compared to assignment 4 (573k vs 567k).

Assignment 6

Design an optimization algorithm combining the allocation of budget and the pricing when the seller a priori knows that every subcampaign is associated with a different context and charges a different price for every context. Suggestion: the value per click to use in the knapsack-like problem depends on the pricing, that depends on the number of users of a specific class interested in buying the product. Notice that the two problems, namely, pricing and advertising, can be decomposed since each subcampaign targets a single class of users, thus allowing the computation of the value per click of a campaign only on the basis of the number of clicks generated by that subcampaign. Plot the cumulative regret when the algorithm learns both the conversion rate curves and the performance of the advertising subcampaigns.

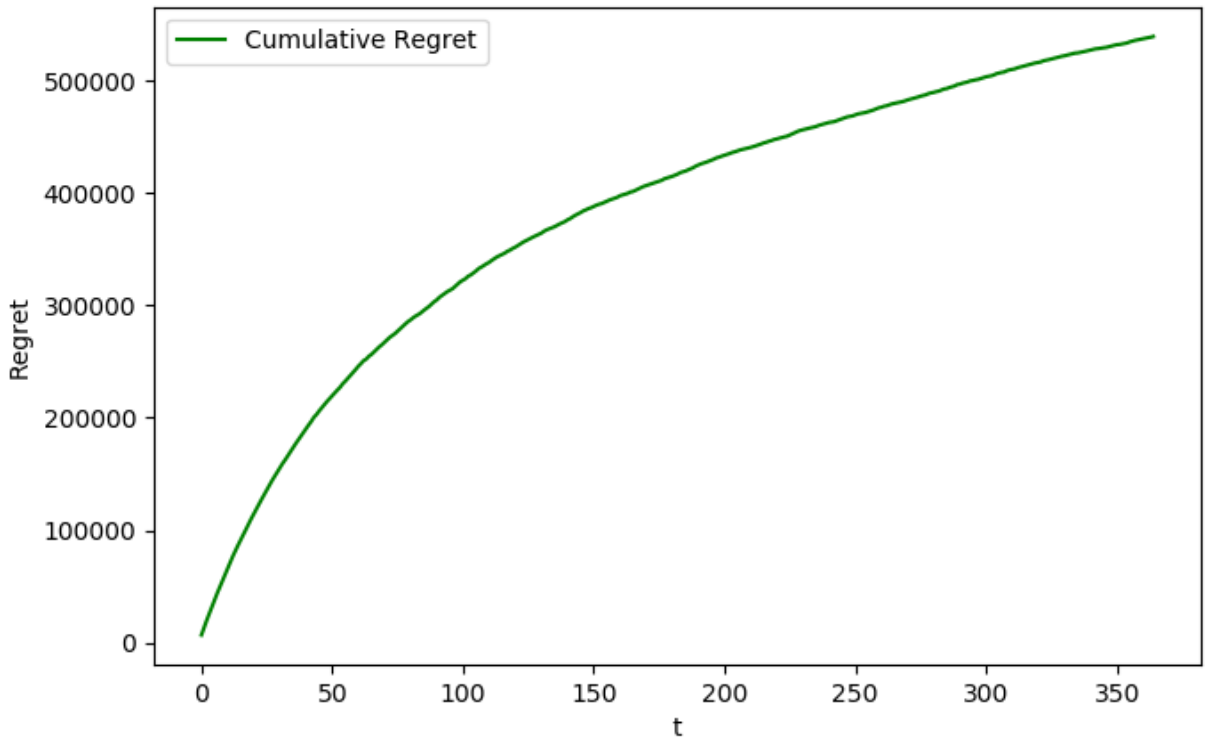
Assignment 6 - Experiment

- T $= 365$
- Number of experiments $= 50$
- Class probabilities $= \left[\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right]$
- Number of arms pricing $= \lceil (T \log(T))^{\frac{1}{4}} \rceil = 8$
- Number of arms advertising $= 21$

Assignment 6 - Pseudocode

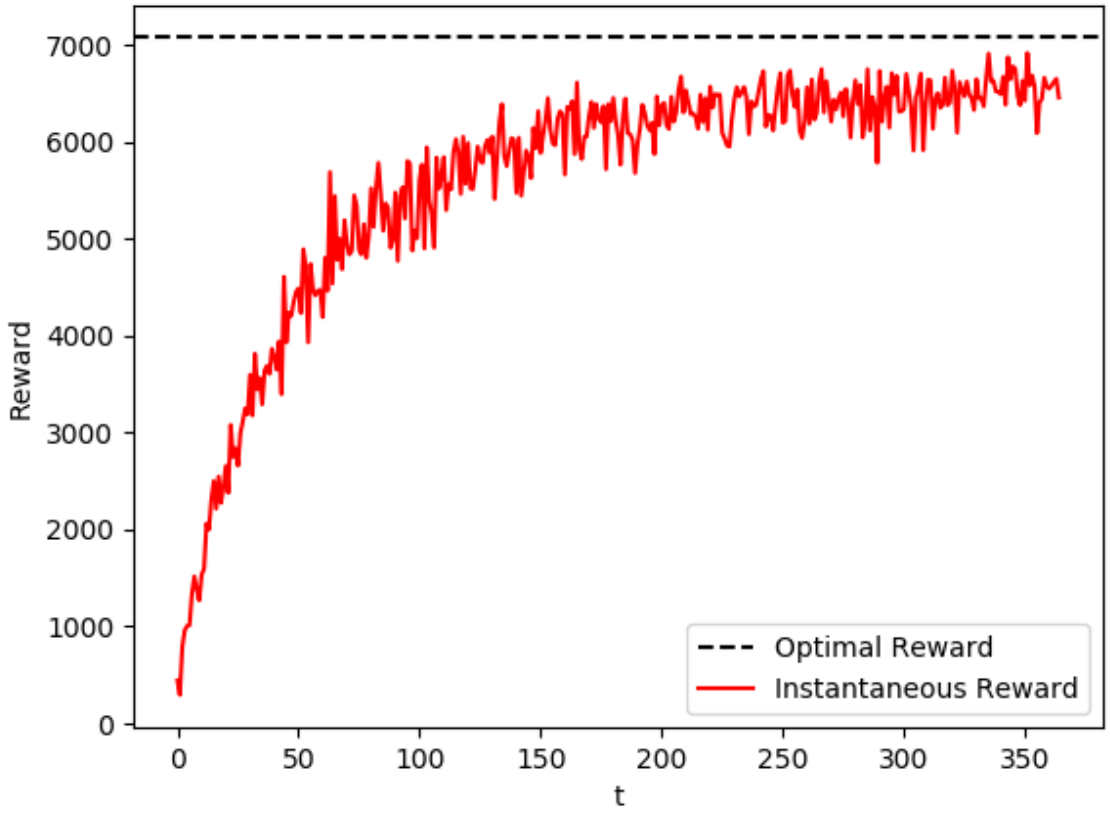
```
1
2  initialize pricing and click envs
3  for exp in n_experiments:
4    initialize ts and gp learners
5    for t in time_horizon:
6      for s in subcampaign:
7        pull-arm (price, conversion_rate) with ts learner
8        retrieve result from the pricing env
9        update ts learner
10       pull-arm (clicks)with gp learner
11       weight the clicks with price and conversion_rate
         previously selected by the ts learner
12     find super-arm by solving knapsack problem
13     collected-revenues = clicks * price * conversion_rate
14     update gp-ts learners
15 calculate clairvoyant value
16 plot rewards and regrets
```

Assignment 6 - Cumulative Regret



Analyzing the cumulative regret of the algorithm we can observe that the curve has almost an asymptotic progress meaning that the algorithm is almost near to the optimal solution.

Assignment 6 - Cumulative Reward



If instead we observe the instantaneous reward, we can see that the algorithm is almost near to reach the optimal value.

Assignment 7

Do the same of Step 6 under the constraint that the seller charges a unique price to all the classes of users. Suggestion: for every possible price, fix this price and repeat the algorithm used in Step 6. Plot the cumulative regret when the algorithm learns both the conversion rate curves and the performance of the advertising subcampaigns.

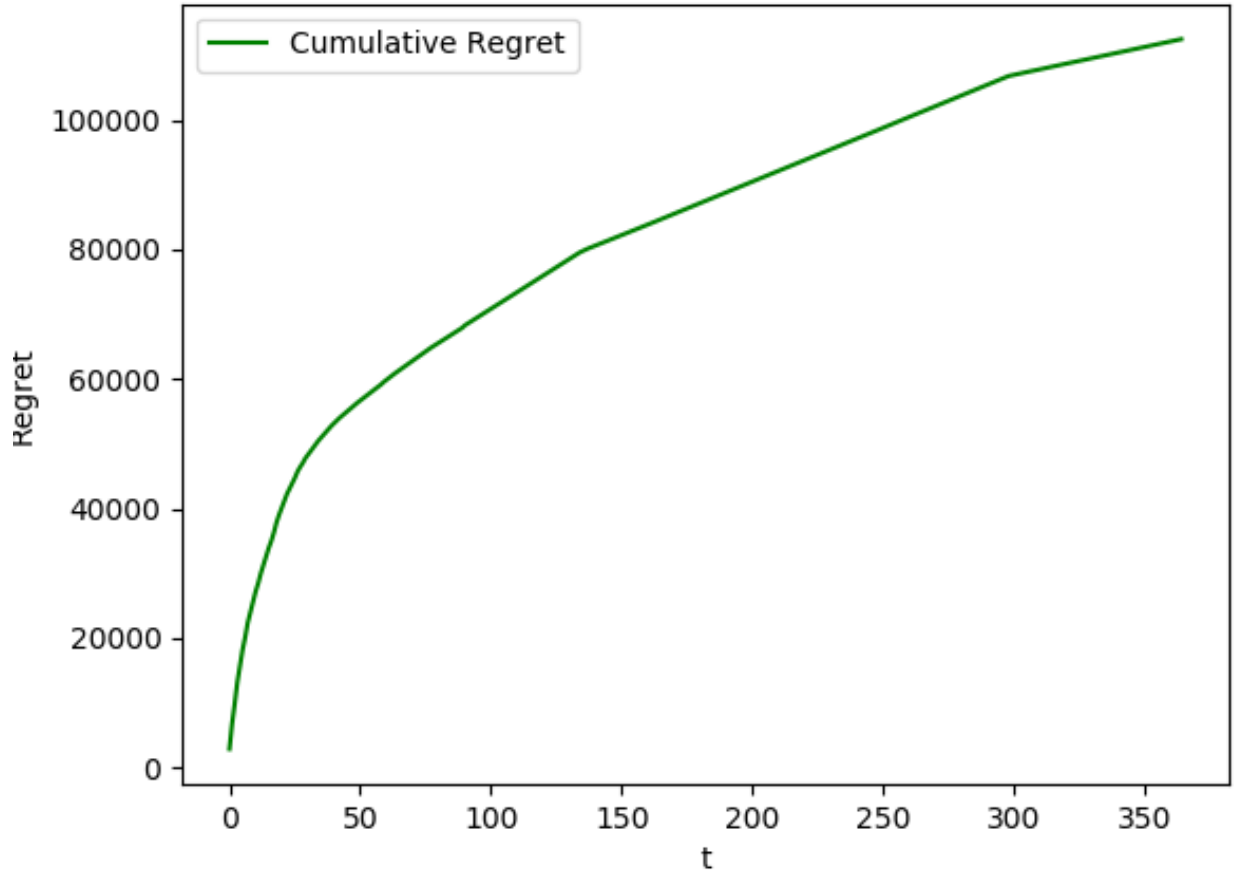
Assignment 7 - Experiment

- T $= 365$
- Number of experiments $= 15$
- Class probabilities $= \left[\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\right]$
- Number of arms pricing $= \lceil (T \log(T))^{\frac{1}{4}} \rceil = 8$
- Number of arms advertising $= 21$

Assignment 7 - Pseudocode

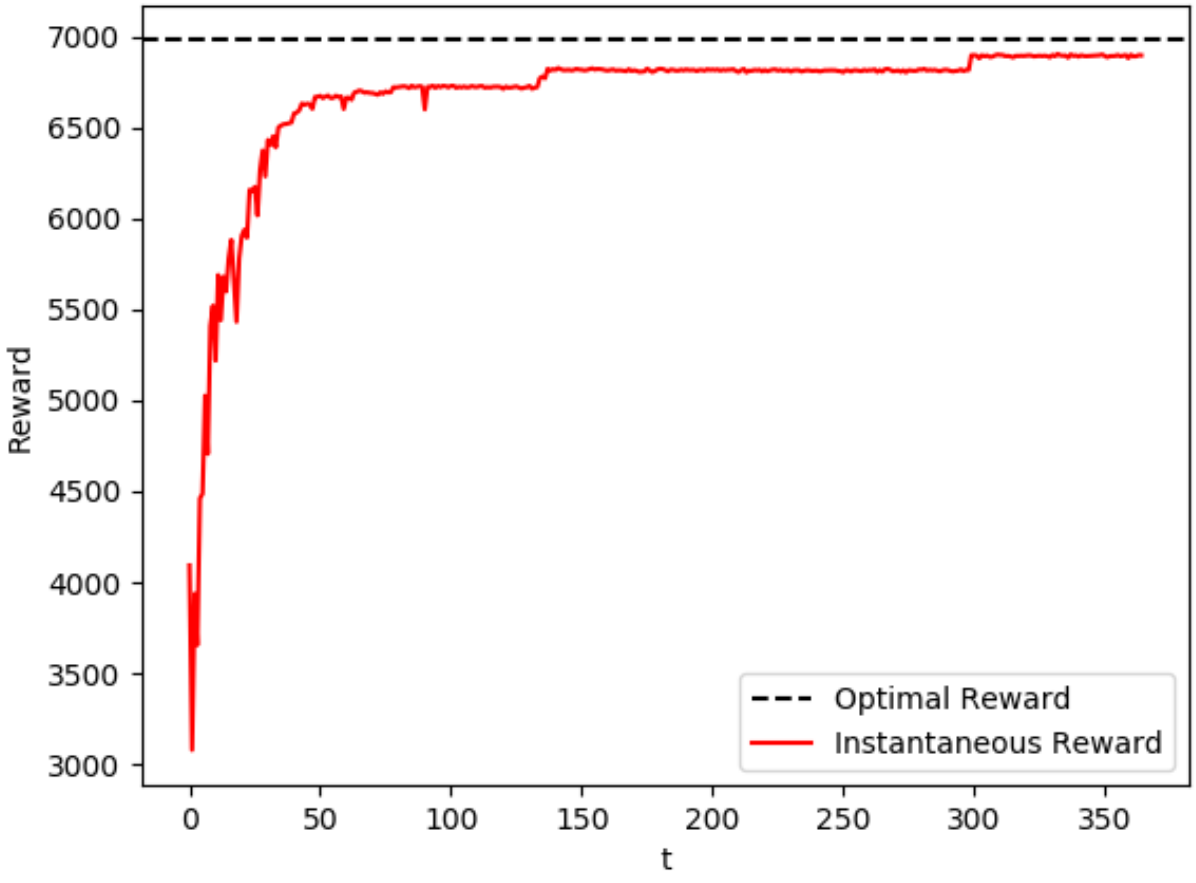
```
initialize pricing and click envs
for exp in n_experiments:
  for price in n_prices:
    initialize ts and gp learners
    for t in time_horizon:
      for s in subcampaign:
        select current price and conversion_rate
        retrieve result from the pricing env, given
         the price
        update ts learner
        draw click samples with gp-ts learner
        weight the clicks with price and
         conversion_rate previously selected
      find super-arm  by solving knapsack problem
      collected-revenues = clicks * price *
                           conversion_rate
      update gp-ts learners
calculate clairvoyant value
plot rewards and regrets
```

Assignment 7 - Regrets of optimal arm



The slope of the curve shows an initial phase of exploration and then followed by a phase of suboptimal solution in which the regret increases linearly.

Assignment 7 - Rewards of optimal arm



Here we can notice that the algorithm reaches a stable point near the optimum but cannot manage to increase its value.

Assignment 6 & 7 Comparison

- Optimal value:
  - Assignment 6: 7074
  - Assignment 7: 6977

  The difference between the two is due to the imposed price of the assignment 7.
- Assignment 6 shows a slower increase of instant reward while assignment 7 reaches high values quite immediately. Responsibility lies with the exploration phase of assignment 6 that tries a wider range of prices while assignment 7 has only one imposed price.

Assignment 6 & 7 Comparison

- Assignment 6 shows a bigger cumulative regret than assignment 7. This can be justified by the initial exploration phase in which a lot of regret is accumulated.
- Assignment 6 has a continuously decreasing slope of cumulative regret while assignment 7 shows a quasi-linear slope, this underlines that assignment 6 is improving while assignment 7 has settled in a sub-optimal solution.

Thank you!