Data Intelligence Application 2018/19 Project



Advertising

Antoniazzi Matteo (895712)

Bonali Luca (896641)

Chittò Pietro (899045)

Lamparelli Andrea (894005)

Ravelli Leonardo (894222)

Index

[1. Introduction 3](#_Toc18922086)

[1.1 Product description 3](#_Toc18922087)

[2. Classes and environment description 3](#_Toc18922088)

[2.1 Features Selection 3](#_Toc18922089)

[2.2 Class descriptions 4](#_Toc18922090)

[2.3 Sub-Campaign definition 5](#_Toc18922091)

[2.4 Average daily budget/clicks curves 6](#_Toc18922092)

[3 Combinatorial GP-TS 16](#_Toc18922093)

[3.1 Combinatorial GP-TS considering all sub-campaign 16](#_Toc18922094)

[3.2 Combinatorial GP-TS without strict assignment rule 18](#_Toc18922095)

[3.3 CGPTS vs CGTS 19](#_Toc18922096)

[4. Average regression error 20](#_Toc18922097)

[4.1 Average regression error of the GP considering all sub-campaign 20](#_Toc18922098)

[4.2 Average regression error of the GP without strict assignment rule 23](#_Toc18922099)

[5. Disaggregation and context identification 25](#_Toc18922100)

# 1. Introduction

The product we decided to use for this project is the AMAZON ECHO PLUS. We’re basing our project on this version of the product with respect to other ones (like Echo Dot, Echo Sub etc.) because, even if the Echo Dot is the more sold product for its small price, it lacks some features like the hub for connecting more smart devices to make a more advanced “smart home”.

## 1.1 Product description

The second generation of Amazon Echo Plus was released in September 2018.

It’s a smart speaker developed by Amazon. Echo devices connect to the voice-controlled intelligent personal assistant service Alexa, which responds to the names "Alexa", "Echo", or "Computer". The features of this device include: voice interaction, music playback, making to-do lists, setting alarms, streaming podcasts, in addition to providing weather, traffic and other real-time information. It can also control several smart devices, acting as a home automation hub like, for example, Smart TV, specific appliances, lamps, windows, doors, temperature control etc. It shares design similarities with the first-generation Echo, but also doubles as a smart home hub, connecting to most common wireless protocols to control connected smart devices within a home.

We consider it more customizable than its direct competitor “Google Home” and Alexa's encourages faster and broader development and support from third-parties of its skills market.

# 2. Classes and environment description

## 2.1 Features Selection

We describe our possible customers by means of 3 main features, with the following values:

* Age: Young, Adults, Retires
* Home status: Living alone, Living with family
* Welfare: Normal, Richer

We decided to divide with respect to the age because younger people are more inclined to accept new technologies to make their lives simpler than older ones. We also thought that a person who lives alone is more worn to have something that can help and fasten the way he approaches some of his daily tasks, especially if young. Of course, we consider that this product is a commodity and so, even if the price is not too large, some class of customer may not be interested in buying our product.

## 2.2 Class descriptions

In the following tables we show how, using the previous explained features, we’ve created our main class of customers.

For readability, we split the 3D features tensor into 2 tables according to the feature home status. In each cell of the table we reported the probability of a user to belong to that specific class.

Each colour represents one class.

|  |  |  |  |
| --- | --- | --- | --- |
| **FAMILY 0.40** | Young 0.50 | Workers 0.40 | Retires 0.1 |
| Richer 0.55 | 0,11 | 0,088 | 0,022 |
| Normal 0.45 | 0,09 | 0,072 | 0,018 |

|  |  |  |  |
| --- | --- | --- | --- |
| **ALONE 0.60** | Young 0.50 | Workers 0.40 | Retires 0.1 |
| Richer 0.55 | 0,165 | 0,132 | 0,033 |
| Normal 0.45 | 0,135 | 0,108 | 0,027 |

* **Class 1:** this class is characterized by people who live alone, like young student or worker, who have a good economic wellness and are more willing to spend money on this commodity. Young families with economic possibilities are in this class too. We assume that this is the class that we expect to click more on our advertised product.

* **Class 2:** in the second class we have other kinds of families, from younger to older, which are less incline to spend money on a product like that, so they will less probably click on our ad. Considering age , we can include in this class also older families, or retired couples, who have money to spend, and are curious about these new technologies.
* **Class 3:** this class is composed by all those people who, for different reasons, are not so interested in our product, can be for economic reason for example. But advertising the product on those people, may bring some of them to click on the product anyway.

## 2.3 Sub-Campaign definition

We have identified 5 different sub-campaigns:

* Search advertising:
  + ***Google***
  + ***Bing***
* Display Advertising:
  + ***YouTube***
* Social advertising:
  + ***Facebook***
  + ***Instagram***

***Google*** is the most used search engine, so it seemed mandatory to have our focus on an advertising sub-campaign on it. We assume that all the three classes use Google.

***Bing*** is the default search engine when dealing with Microsoft OS, and we are assuming that people belonging to the 2nd and 3rd class most probably have a windows based computer, since we assume it to be more user friendly, and that are not interested in changing the default web search engine.

For search advertising we are considering keywords like: “assistente vocale”, “smart speaker”, “home speaker” exc. Searching this kind of keywords will display ads and banners of our product, as form of slot for Google and ads for Bing, that will bring the user to the Amazon link to buy our product.

***YouTube*** is used for watching videos, and ads are displayed during the video as banners and under the video. Sometimes the ads are related to the video the user is watching, and sometimes it depends on the collected info of that user, like cookies. By displaying our product in some videos as banner, for example during tech videos, which are related to our assistant, we can assume that some user will click on the ad.

***Facebook*** and ***Instagram*** are two of the most used social networks. By displaying sponsored ads in the people feed we can increase people’s awareness in our product and make them interested in it.

We have chosen YouTube and Instagram, which are more probably used by the first two classes, and Facebook, which is used also by people belonging to the 3rd class.

## 2.4 Average daily budget/clicks curves

- In this chapter we will show the assumption we have made to construct the curves which represent the functions daily budget/clicks.

- X axis: daily budget

- Y axis: expected number of clicks

We show all the three different curves, for each class of user, and for each sub-campaign, then the aggregated curve for each sub-campaign. The actual numbers on the graphs are our assumptions with respect to the actual behaviour of the users.

We have three levels of clicks limits:

* High = 90
* Medium = 60
* Low = 40

We expect that when a curve has its maximum arrives at one of these three levels, also because being the mechanism to decide which product is placed on the banner, or slot, or ad, an auction ,we have that increasing the budget we increase the probability that our product will be displayed in the best position. But once the product is displayed in the best position the expected number of clicks will remain the same even increasing the budget. (we are implicitly assuming that the bidding is performed automatically by the advertising platform, according to the budget)

The aggregation is achieved by averaging the curves with respect to the probability of belonging to one class (0.65 1st , 0.305 2nd , 0.045 3rd).

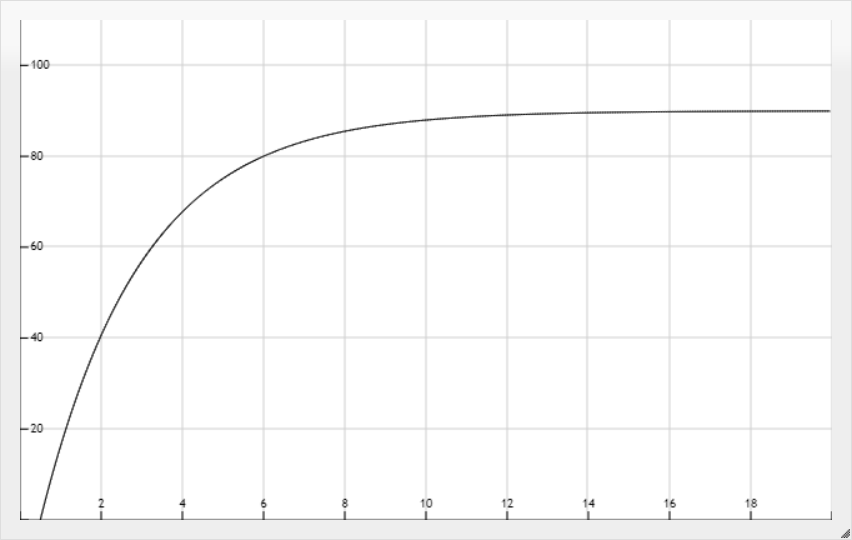
Google c1:

Figure 1

* This represent the most prolific curve, in which we have the most interested users where we expect to have the most clicks, google. The curve grows fast starting with low values of daily budget, and arrives early near to its maximum value, which is the highest level of click we have assumed for the platform, once the product is displayed in the first slot (best position).

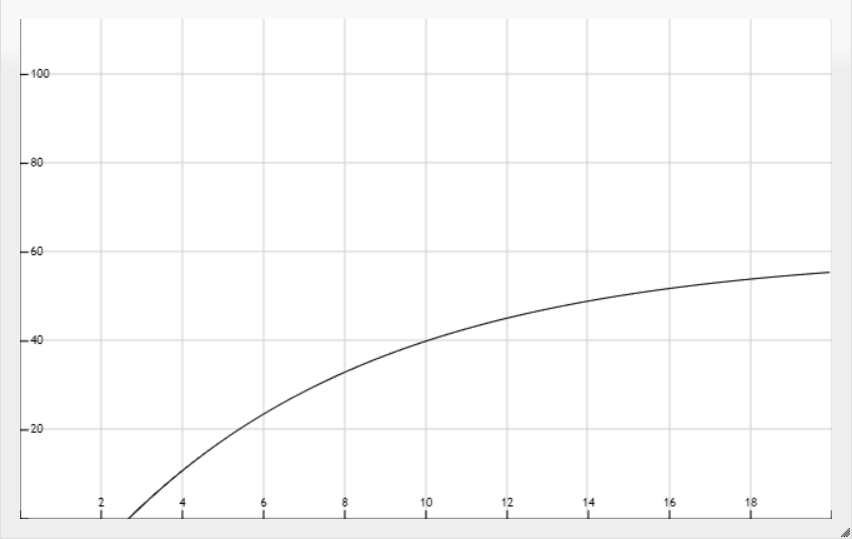
Google c2:

Figure 2

-Second class of the google users, we have a maximum value of clicks which is our Medium Value, because we expect this class to be less interested in our advertisement. The curve also has a less increasing behaviour.

Google c3:

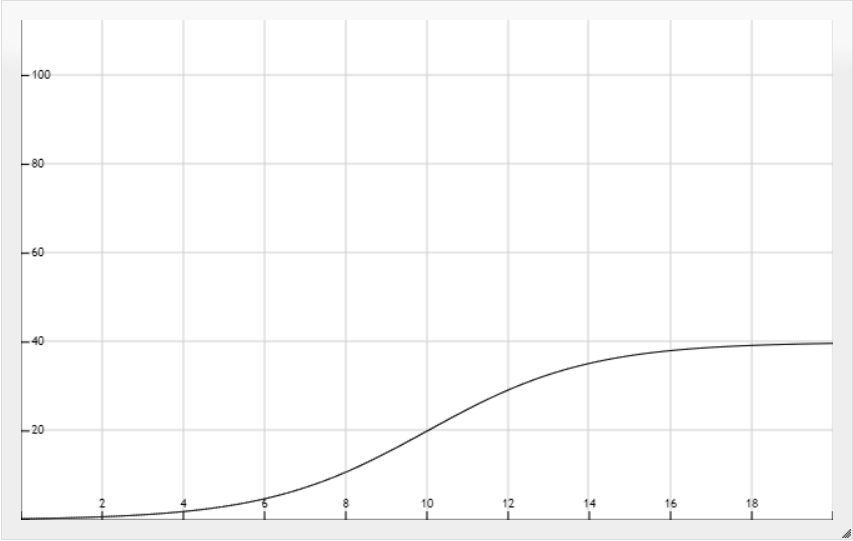


Figure 3

* The third class curve has a slower increasing behaviour, since this is the class which we assume not so interested in our product. So the limit is the lower level of our definition.

Google aggregated:

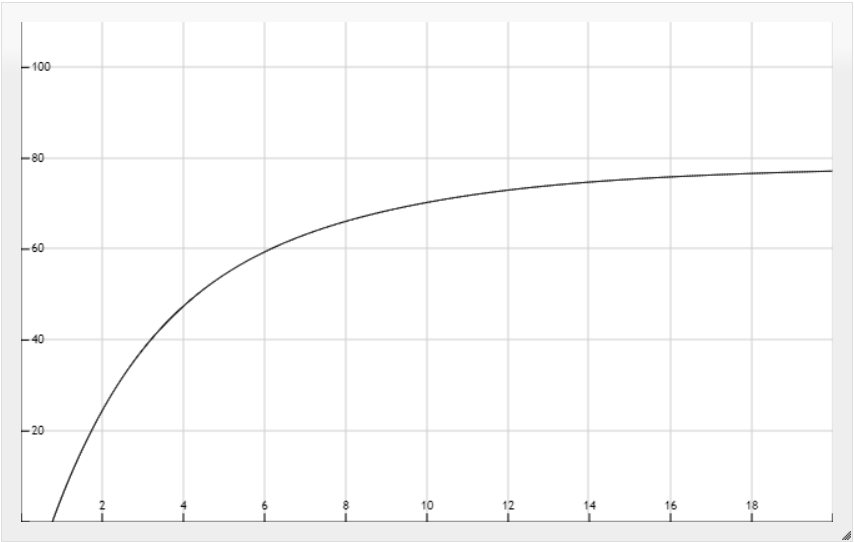
* The aggregated curve has the maximum value near 80 and grows quite fast, since the first curve has a bigger influence on the aggregation.

Figure 4

Facebook c1

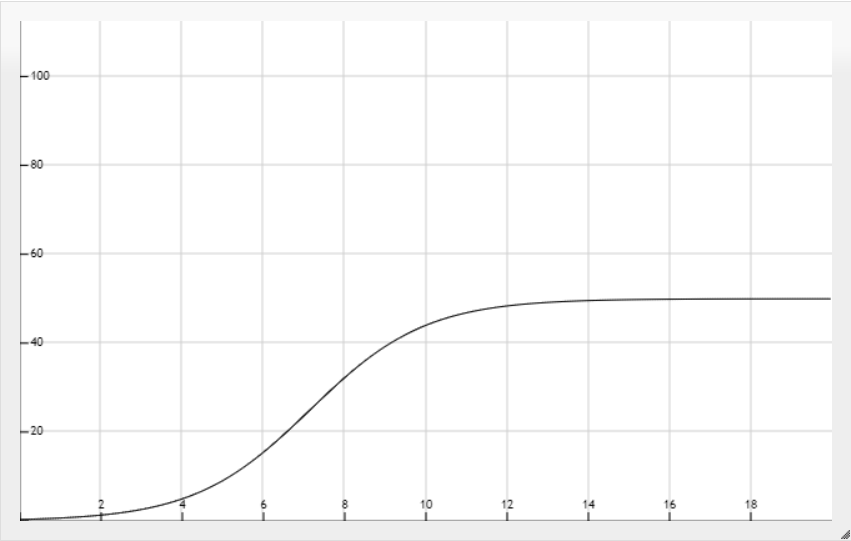
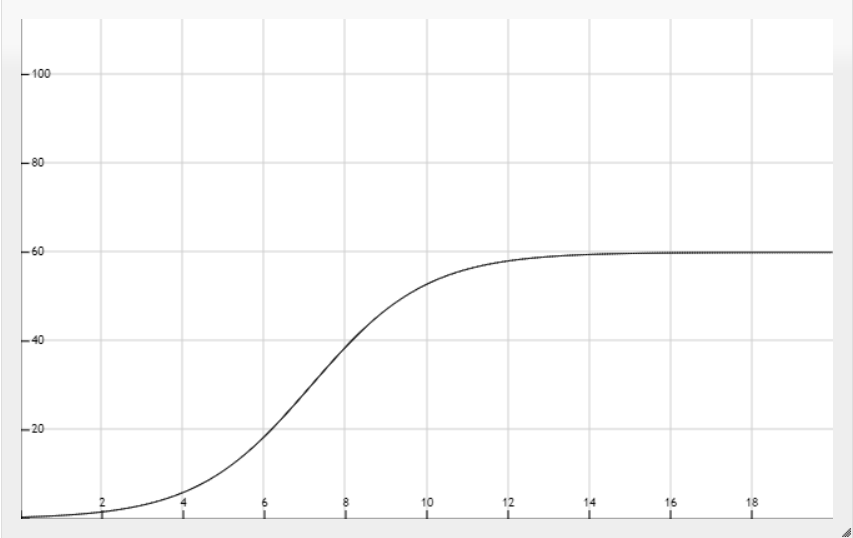
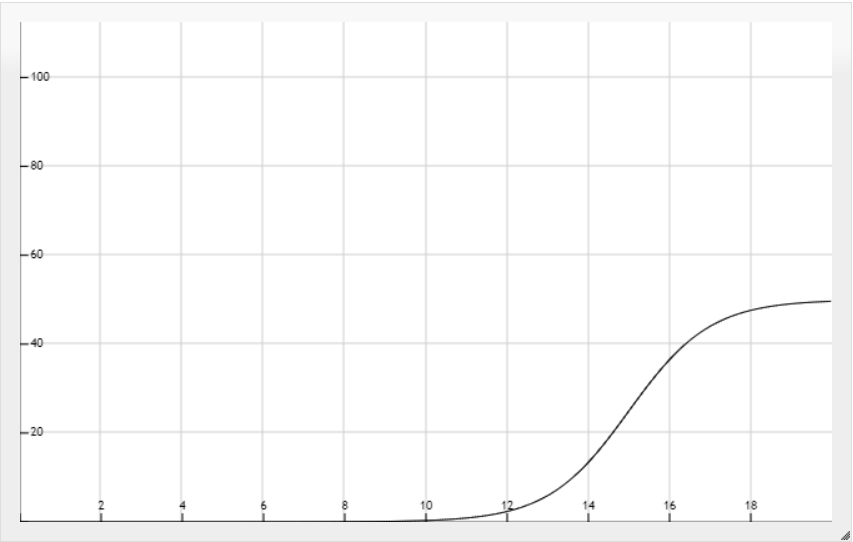
* First class curve of the second sub-campaign, we are assuming that the maximum is lower with respect to the first sub-campaign, and also grows slower.

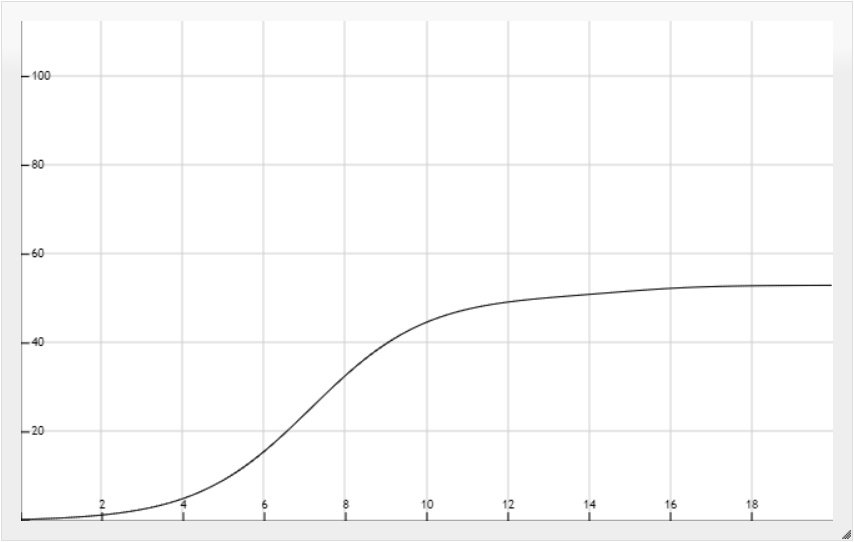
Figure 5

Facebook c2

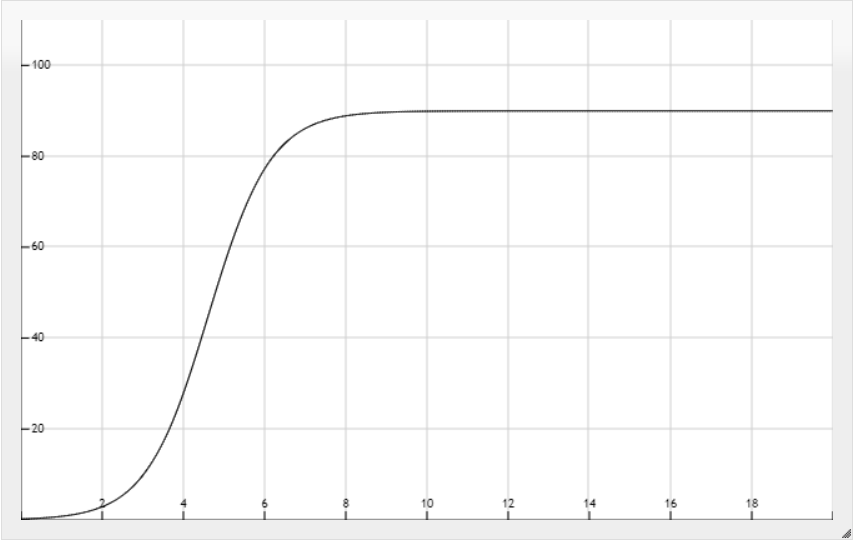
* This curve has an higher maximum value than the first class one, basing on our assumption that the 2 class of users are more active on Facebook, and they are more inclined to click on our ad on this platform.

Facebook c3

* Third class of user on Facebook are not so interested in our product, and so less likely they will click on our ad.

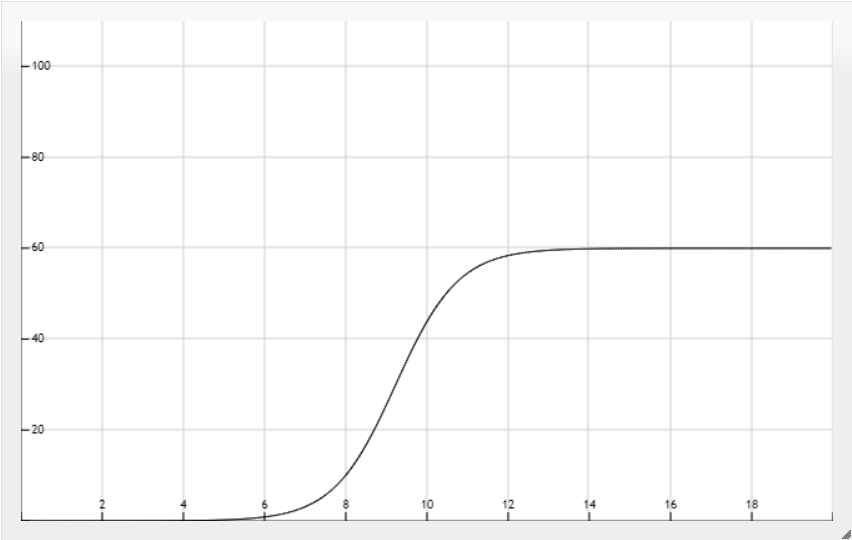
Facebook aggregation

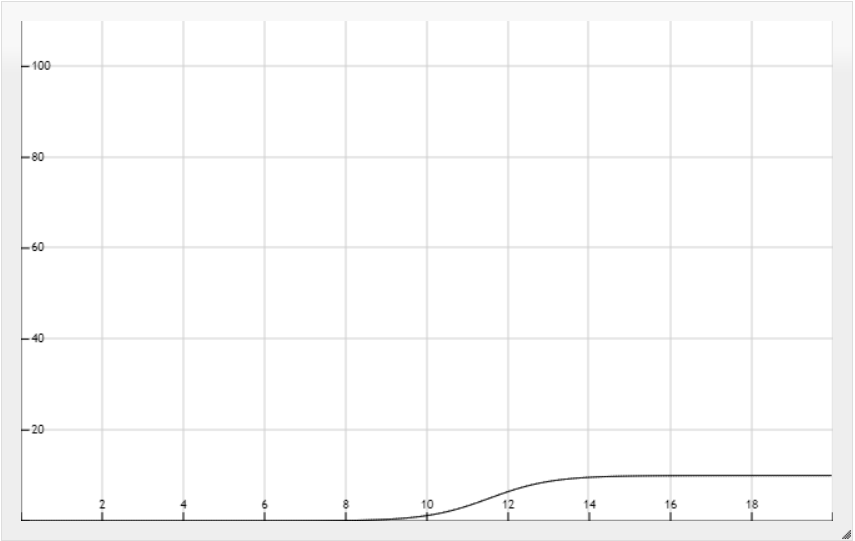
* Aggregating the three Facebook curves is less interesting with respect to the Google aggregation, where we expect to have an higher number of clicks. It may be not interesting in general because the growth is quite slow.

Instagram c1

* Instagram first class has a very interesting behaviour, it grows fast with respect to the budget, and its maximum value is high. Also a lot of Instagram users belong to our identified first class, and they are more inclined to click on our product.

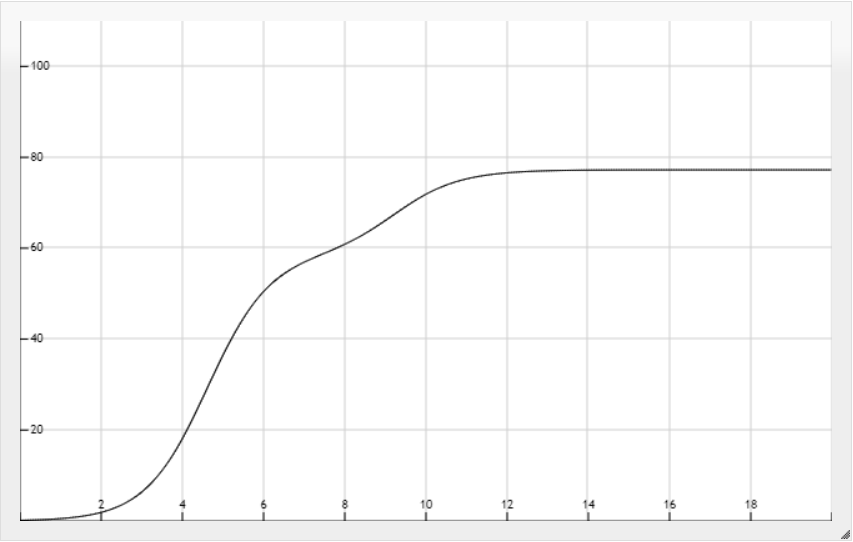
Instagram c2

* Second class of users on Instagram has a slower growth, regarding budget, but has an interesting behaviour if the ad wins more auction, and it is displayed more times.

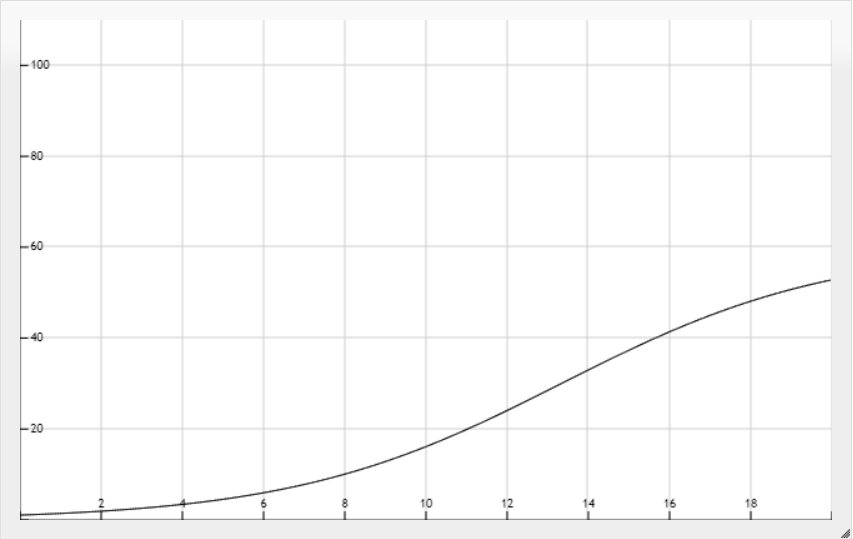
Instagram c3

* Instagram is not so used by people belonging to our third class, and the curve is very low, but if aggregated can bring to some advantages.

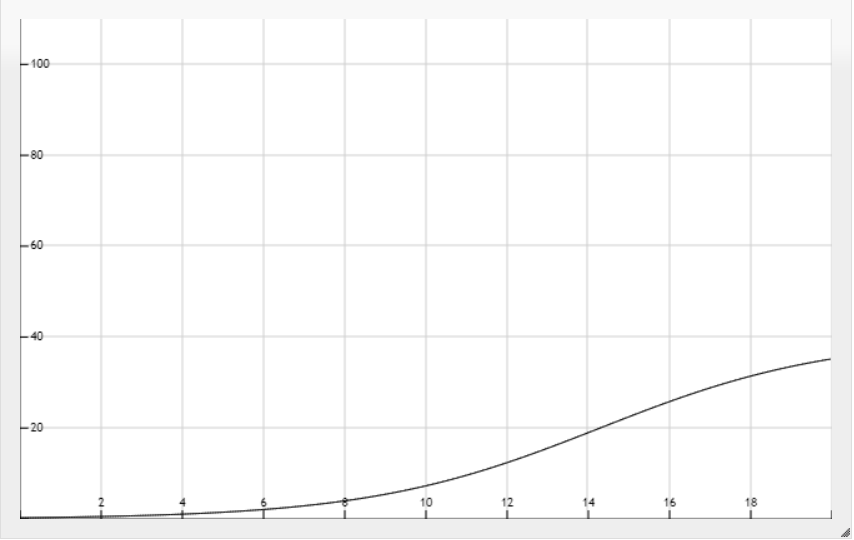
Instagram aggregation

* Aggregating the Instagram curves leads to a interesting result, grows quickly, and has a high maximum value. Also considering the third class of user, advertise our product to a broader audience, which is an advantage.

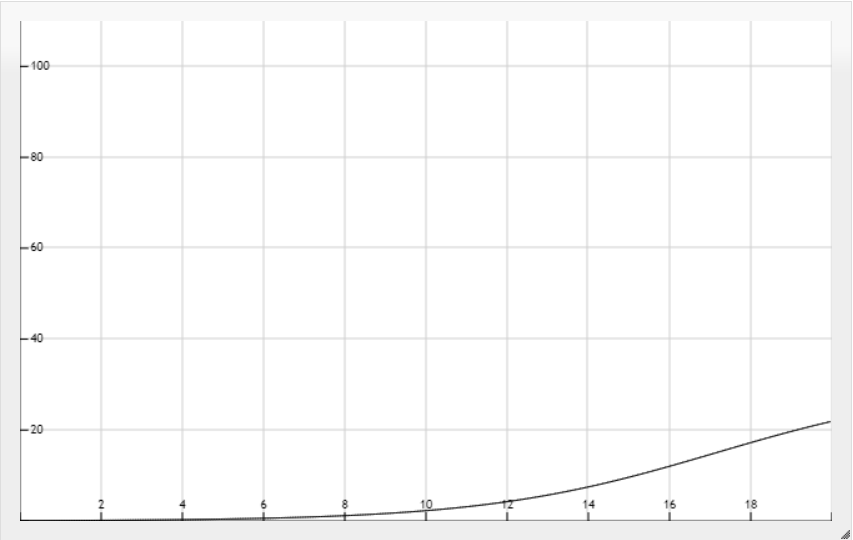
YouTube c1

* Display advertising is seen as a disturb many times, so we have assumed that on YouTube, some of the user which are already interested in our product, and knows about it, will click on the ad when it is displayed. But this number will grow very slowly, and it may not be useful regarding the budget cost.

YouTube c2

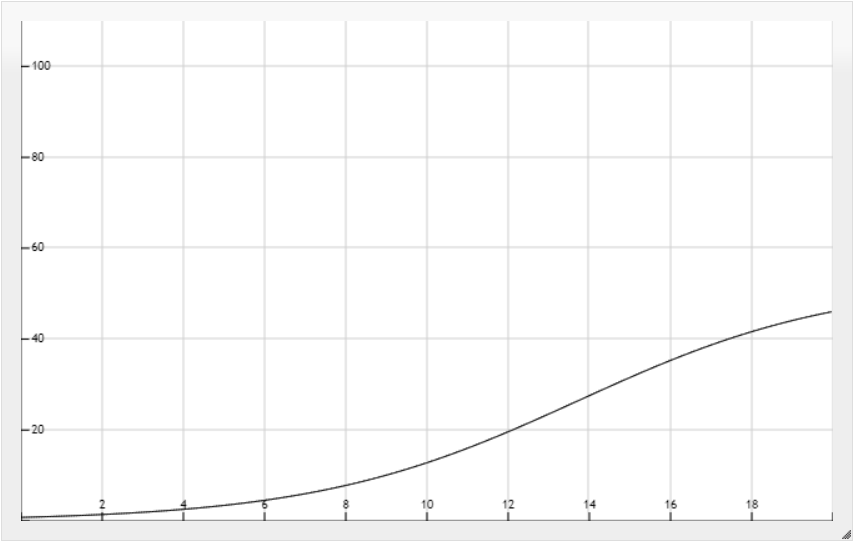


YouTube c3

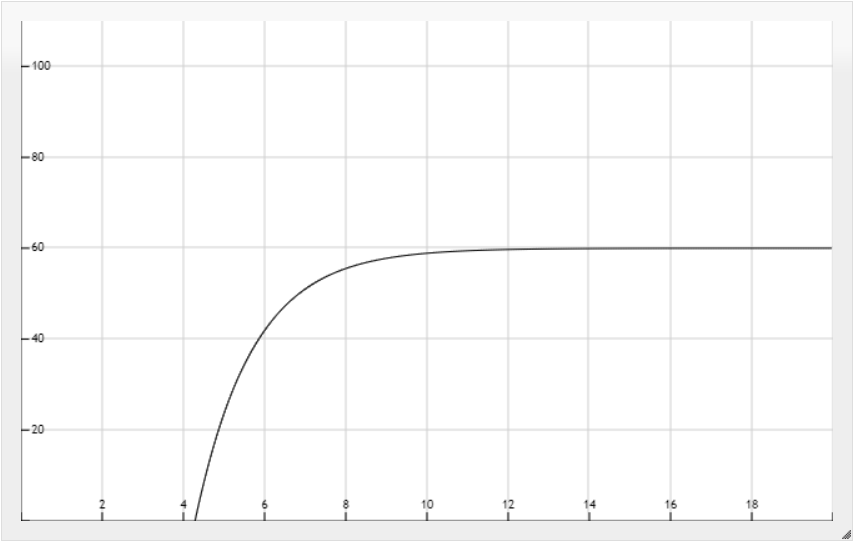


* With these two curves, we are showing that displaing ad on YouTube may not be so effective, considering these two classes of users.

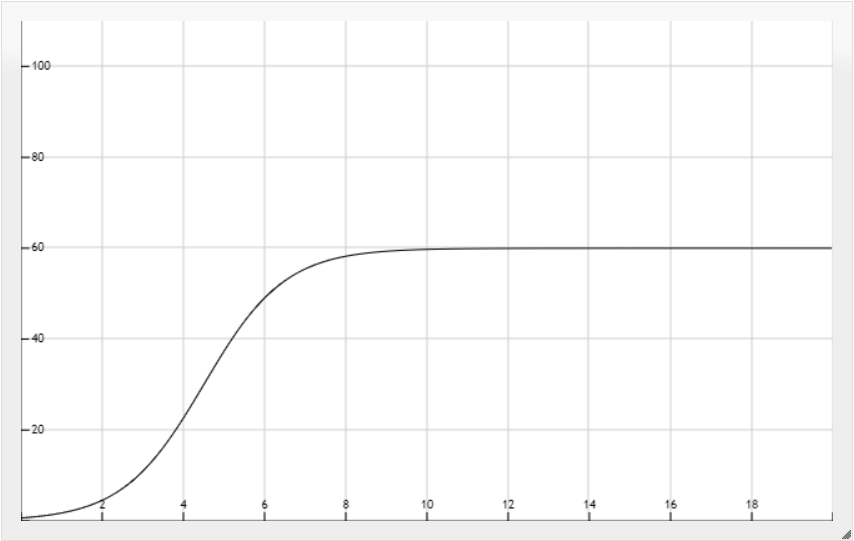
YouTube aggregation

* Using the aggregated curve lead to a poor result, we have medium low level clicks with high cost of budget.

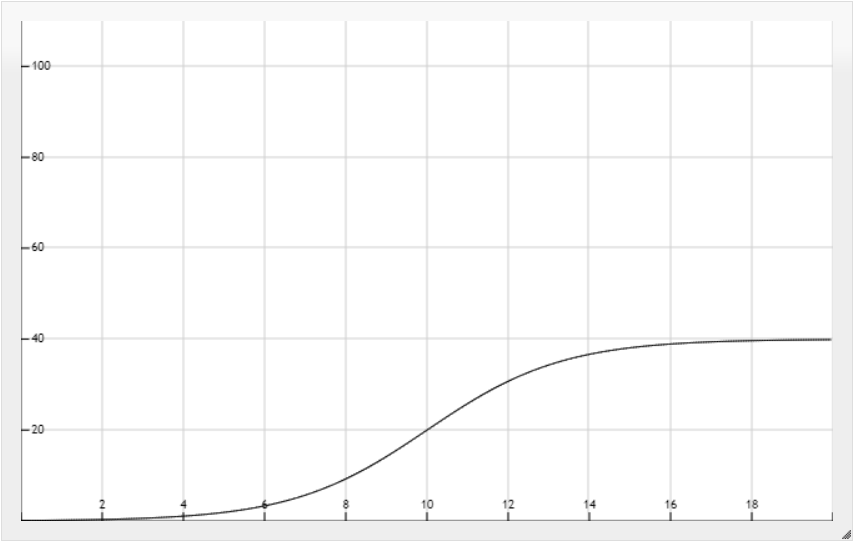
Bing c1

* Bing is used mostly because it is the default engine in windows devices. The nature of a search engine brings the curve to be more effective in advertisement, since probably the users go visiting the website to buy some product. If the auction is lost for too much low budget, the product is not displayed, so the clicks will be 0. Once the auction is won the expected number of clicks grows fast, making a sub- campaign on Bing a interesting choice.

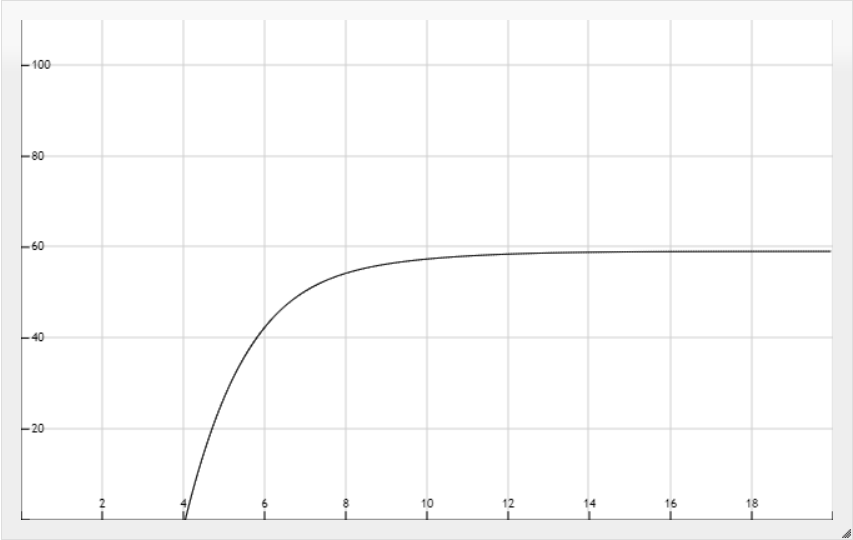
Bing c2

* For the second class we have a quick increase in the function, since more user of the second class will click on the product on this platform, and the maximum number is the same as the first one.

Bing c3

* In the third there is less interest in the product, and the curve grows slowly.

Bing aggregation

* Once aggregating the curves we have that in order to get a good number of clicks, we would need to spend budget on this sub- campaign, since the curve grows very quickly and with a considerable maximum amount of expected number of clicks.

# 3 Combinatorial GP-TS

## 3.1 Combinatorial GP-TS considering all sub-campaign

(We are forcing the algorithm to place at least 1 € per sub-campaign)

(We have chosen small numerical values for the algorithm to simplify the computation)

- T time horizon 100 (number of days)

- Number of experiments 100

- Optimum value reached by the clairvoyant algorithm 136 €

- Daily budget partition (given a daily cumulative budget of 20 €, and a linear discretization of 20 values) found by the algorithm

- Google 4 €

- Facebook 1 €

- Instagram 7 €

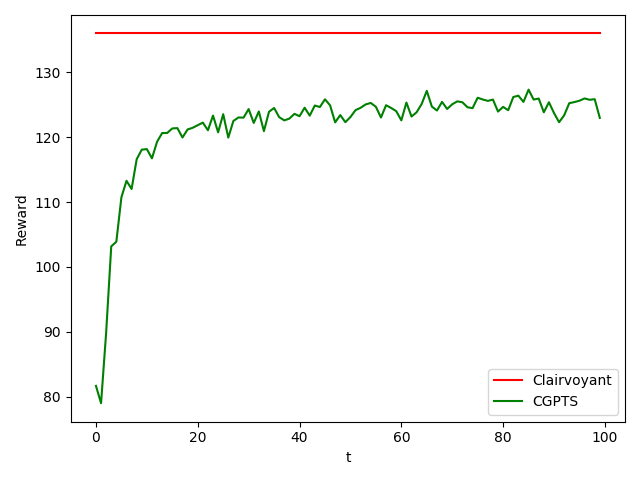
- YouTube 1 €

- Bing 7 €

Gaussian process TS works in a similar way than standard TS, but instead of updating a Beta distribution for the pulled arms, they update a Gaussian Process. This allows them to have an estimation of the curve, and also to have the uncertainty (by means of confidence interval) on the estimation. Updating the estimation of an arm, affects also the estimation of the near arms, by changing the form of the estimated curve.

A combinatorial GP-TS launches in parallel n GP-TS algorithms, and at each round a sample is drawn from all arms from all n GP-TS, than a combinatorial problem is solved given some constraints (like the one that forces to have that the cumulative budget of all GP-TS does not exceed the daily budget), and the algorithm select the budget partition, so the budget to assign to each sub-campaign, and once selected, the correspondent GP is updated with the sampled reward value. At the end, we obtain the estimated budget partition to assign to each sub-campaign.

Figure 6 CGPTS vs. Clairvoyant reward graph (forcing 1-budget)



* Red line represent the reward achieved by the clairvoyant algorithm.
* Green line is the reward achieved by the CGPTS algorithm.

Increasing the number of experiment regularize the shape of the reward curve, more experiments means less uncertainty and spikes in the curve.

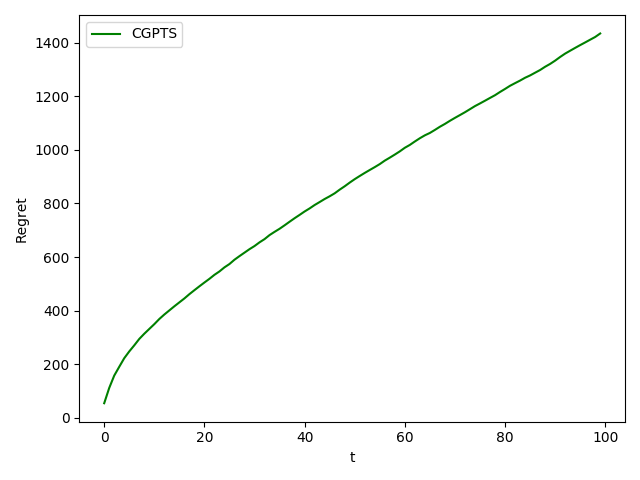


Figure 7 CGPTS cumulative regret in time (forcing 1-budget)

-Regret increases almost linearly with time. The algorithm has a very fast learning rate, in few days it is capable of reaching its avg maximum value of reward. After that initial period, it stabilizes and its maximum value, with some uncertainty.

## 3.2 Combinatorial GP-TS without strict assignment rule

Given that the CGPTS has found that 2 sub-campaign may be not useful, we have considered to launch the algorithm without any rule that forces the assignment to 1.

These are the result, with these settings:

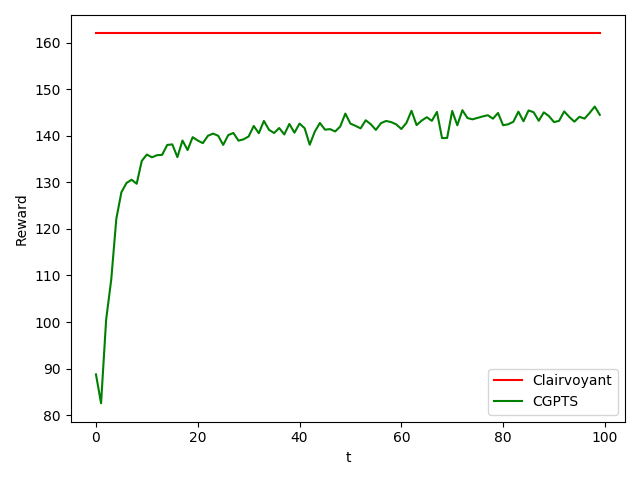
**Combinatorial Gaussian Process Thompson Sampling**

allowing to have 0-budget for some sub-campaigns

T = 100

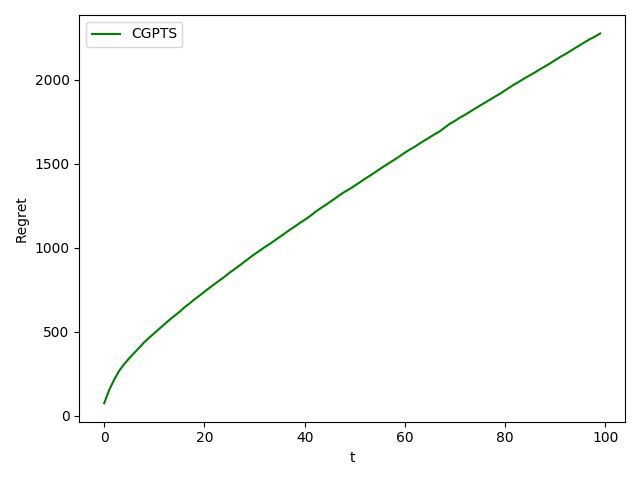
Exp = 100

optimum = 162

best budgets = 5,0,7,0,7

The result have shown that allowing 0-budget sub-campaigns increase fairly de optimal and the obtained reward, with respect to the one obtained by forcing to at least one the budget for all sub-campaign

Figure Reward of CGPTS



The regret of this algorithm is significantly higher than the other one, since the difference between the optimum clairvoyant and our algorithm is higher. But the overall result regarding the reward is much higher with this algorithm, so it can be considered a better option than the first one.

## 3.3 CGPTS vs CGTS

This chapter is a comparison between the Combinatorial Gaussian process Thompson Sampling and the Combinatorial Gaussian Thompson Sampling.

GTS has the peculiarity of having Gaussian priori distribution for each arm, instead of the classical Beta distribution of the regular Thompson sampling.

So, each time that each GTS pulls an arm and samples a reward, it updates a Gaussian distribution associated to that arm.

Combinatorial algorithm works the same as the CGPTS.

Here are some results with the same settings as before:

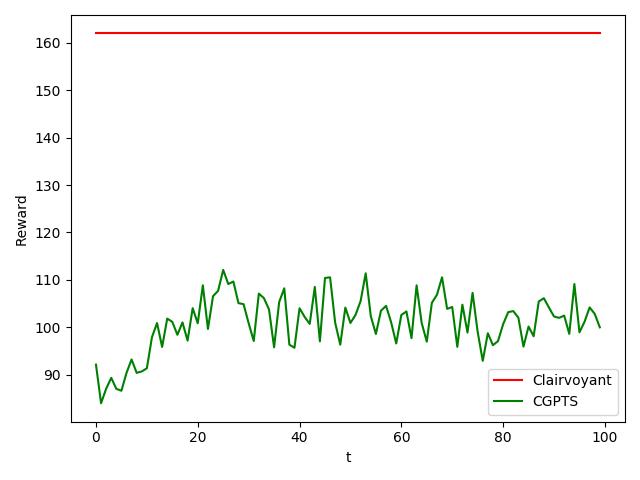
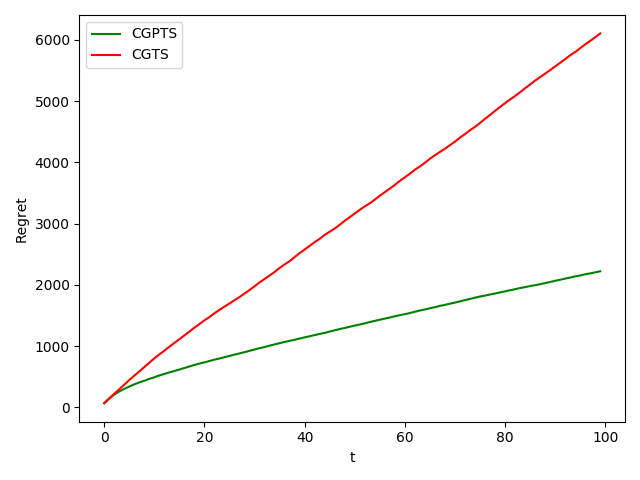
This graph is the reward obtained by our CGTS with the same settings of the one obtained with CGPTS in figure 8.

Figure Reward of CGTS

The results are very poor and unstable, as we can see also from the regret comparison.

Figure 10 Regret comparison CGPTS vs CGTS

The regret of the CGPTS (green) is much lower than the one obtained using CGTS (red).

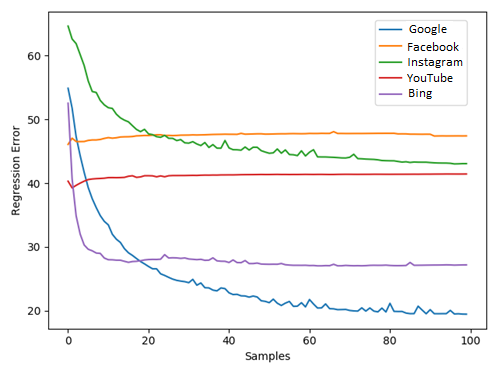
This means that CGTS is not a usable algorithm for our problem.

This is because in GPTS we have that pulling an arm of the process we update not just the value of the sampled arm, but also the neighbour arms and returns the uncertainty on the estimation on the budget/clicks curve.

# 4. Average regression error

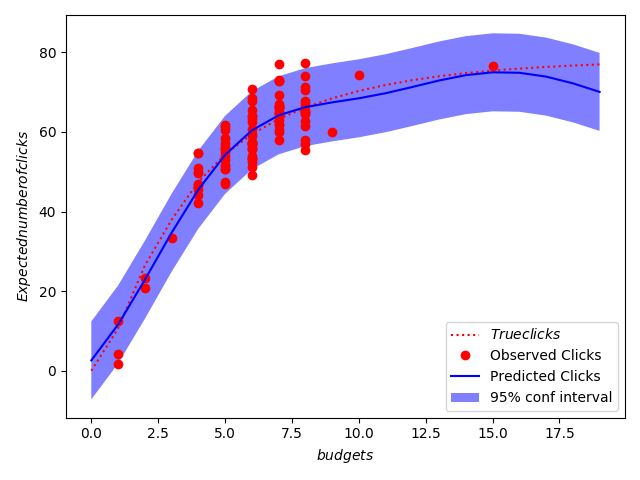
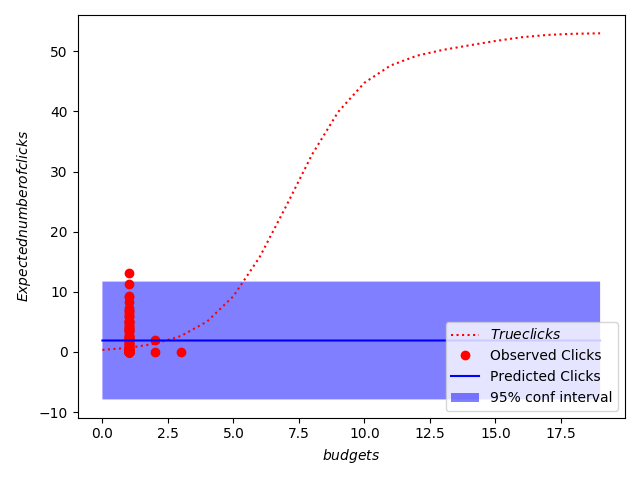
## 4.1 Average regression error of the GP considering all sub-campaign

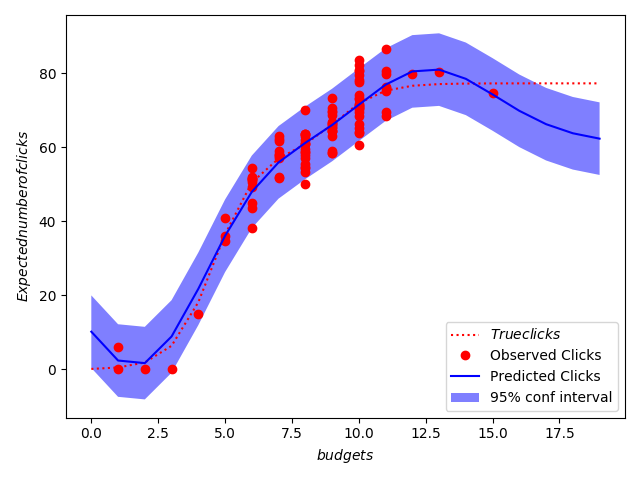
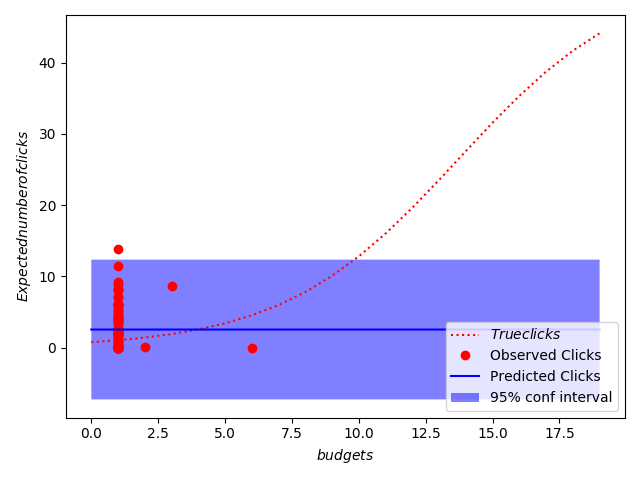
These are the plot, for every sub-campaign, of the average regression error (the maximum error among all the possible arms). As we can see from the plots, we have that the regression error is high for low values of samples and decrease very fast initially as the learning algorithm works and stabilizes very fast. Seeing that the regression error decreases means that the algorithm is learning.

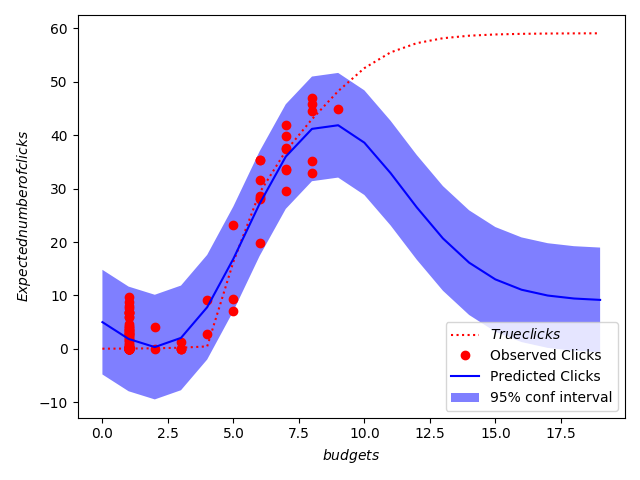


Facebook and YouTube sub-campaigns errors are not decreasing over time, since we have forced the algorithm to work also with these two not profitable campaigns. This will lead to a solution in which we don’t consider anymore these two sub-campaigns, focusing our attention only on the most profitable ones. Since the error does not decrease over time, we can expect that the algorithms is not choosing those two sub- campaigns for learning.

1. Gaussian Processes Regression of 1 experiment (optimum 4-1-6-1-7)



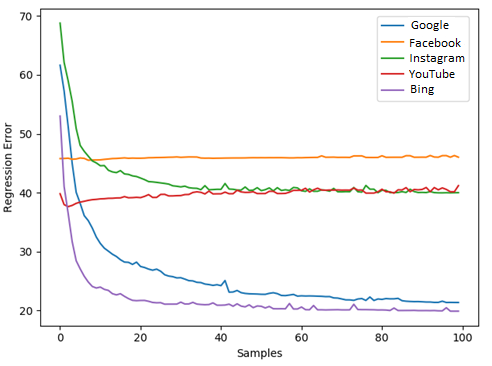




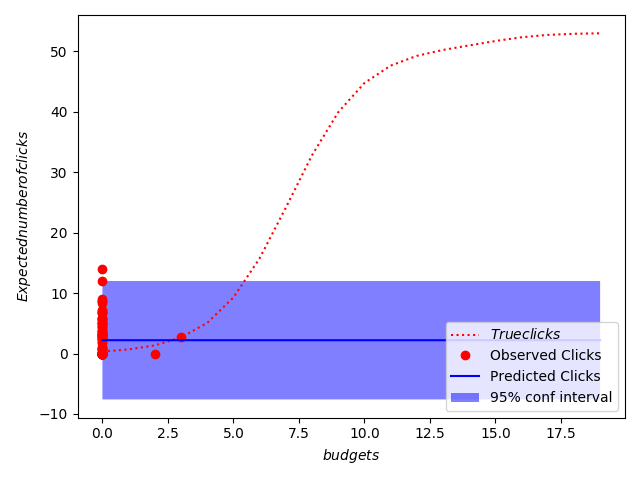
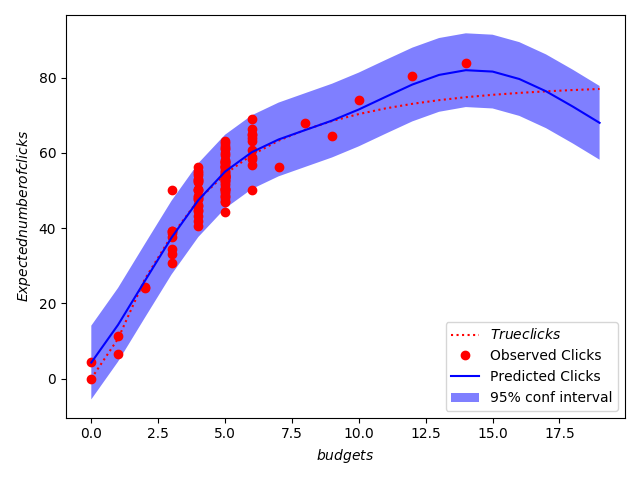
The second and the fourth GP has been estimated as a straight line, because the algorithm pulls low budget values from these two sub-campaigns, being these two not profitable. So since the sampled values are all in the same interval of budget, the GP estimates those curves as strict lines, and the AVG regression error does not decrease.

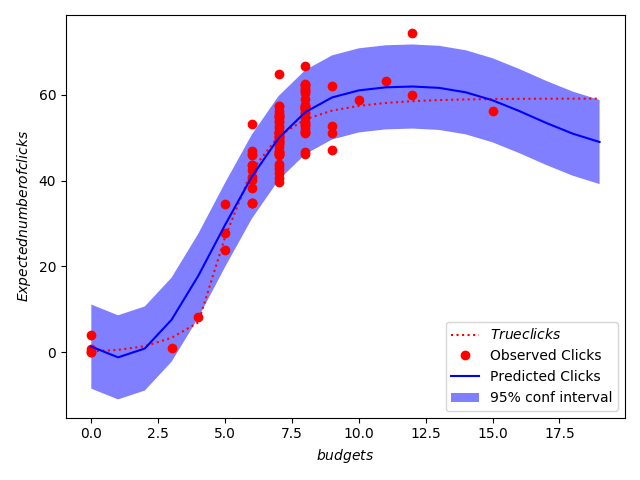
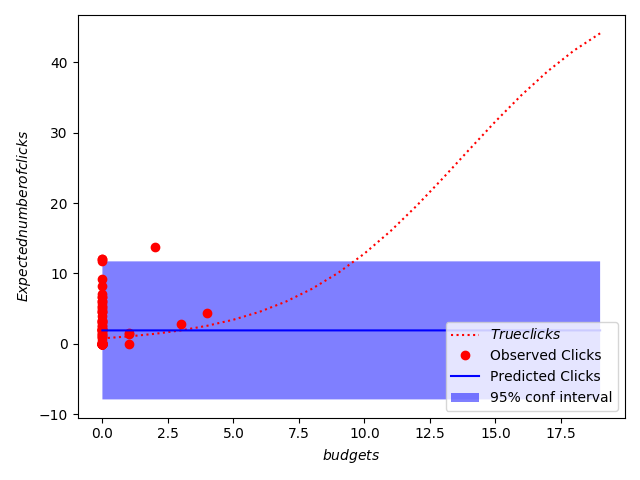
While for the three remaining sub-campaigns, the GP learns more correctly the true curves, approximating better their behaviour, so the AVG error decrease and the curves are correctly learned.

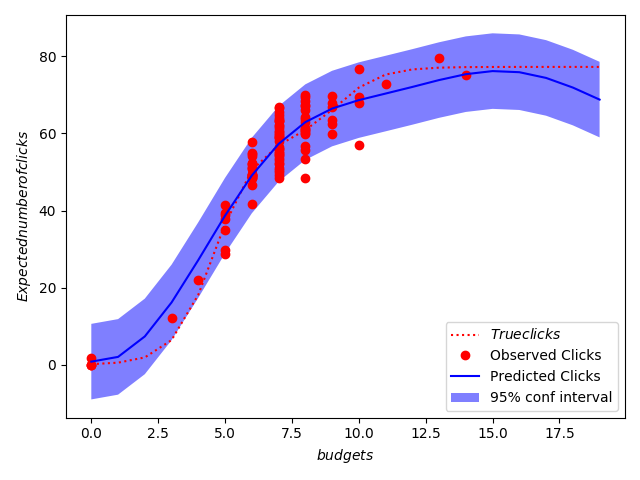
## 4.2 Average regression error of the GP without strict assignment rule

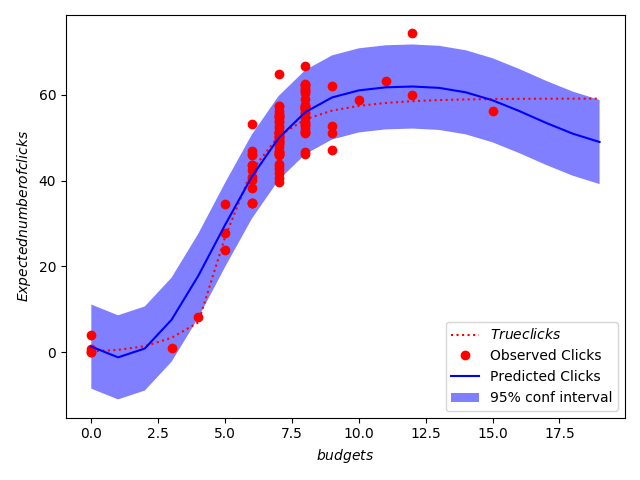
-Also in this case the two not profitable campaigns are modelled as straight lines, so their error does not decrease. But the error of the other 3 is significantly lower with respect to the first case. This because the curves can be better approximate the real ones in this case.

1. Gaussian Processes Regression (optimum 5-0-7-0-7)









Allowing the possibility to have 0-budget sub-campaign, we can approximate better the other three sub-campaigns, and the curves are more accurate, so the AVG regression error is lower with respect to the first case.

# 5. Disaggregation and context identification

In this section we are launching the algorithm considering the choice of decomposing some aggregated curve into some partition of its components. For example and for simplicity we are considering the first sub-campaign, Google. At the beginning we launch the algorithm for two weeks without any concern on the possible disaggregation (as we have seen from previous points, the algorithm requires few days to stabilize its rewards).

After two weeks, it fixes the reward for all the other four sub-campaign and solve a context generation problem and a combinatorial problem, to see if one of the possible decomposition of the first sub-campaign ({c1,c2,c3},{c1c2,c3},{c1c3,c2},{c2c3,c1}) with respect to maintaining the sub-campaign aggregated, can have a larger reward.

If the algorithm finds out that the reward of a disaggregation is better than the aggregated one, from now on, it considers the disaggregation instead of the aggregated curve.

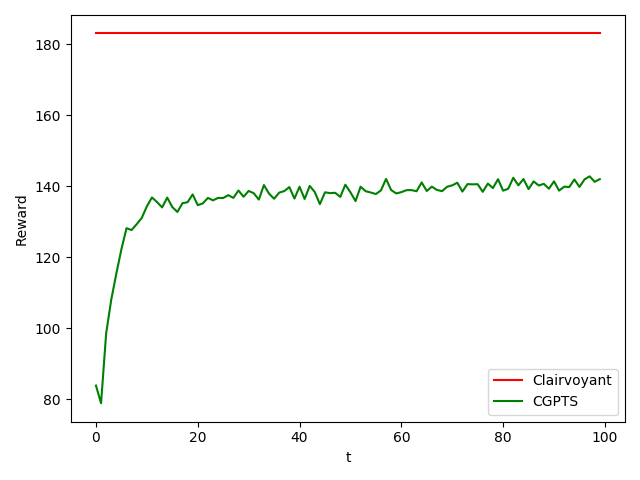
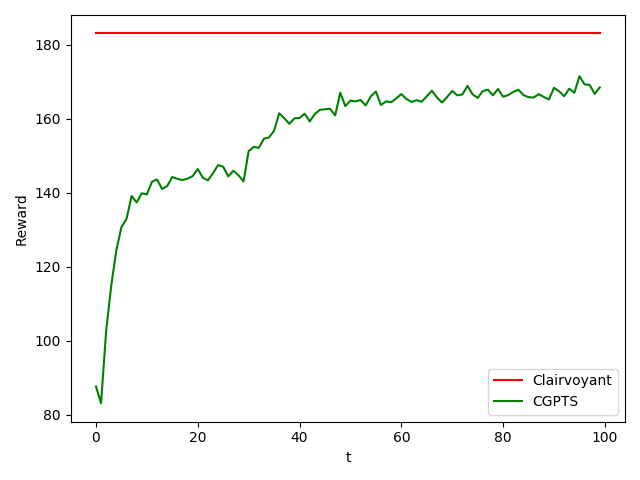
If the algorithm has generated a partition with two classes together and a single one, it can check if disaggregating again the formed two class partition, to see if it could get a higher reward.

In these example we consider checking the disaggregation of the first sub-campaign, maintaining the other all aggregated (for computational reason we consider the disaggregation only for the first sub-campaign, the complete algorithm has the same functioning, but it’s very computationally expensive, since it has to consider all the possible combination at each round).

We consider these settings, and the possibility to assign to 0 the budget for a sub-campaign (since from the previous points we have gathered the information that we can get a higher reward).

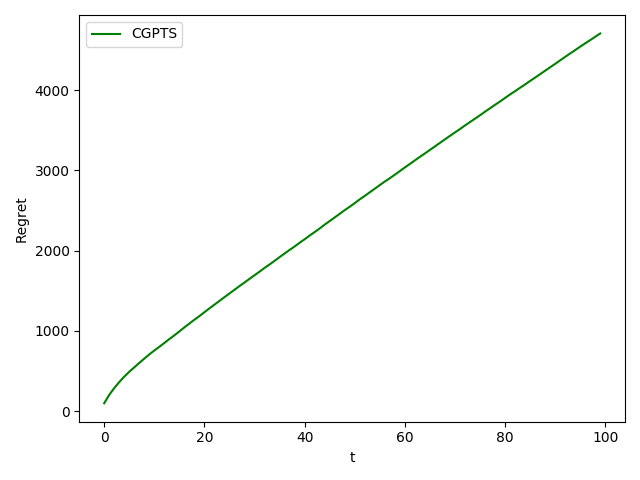
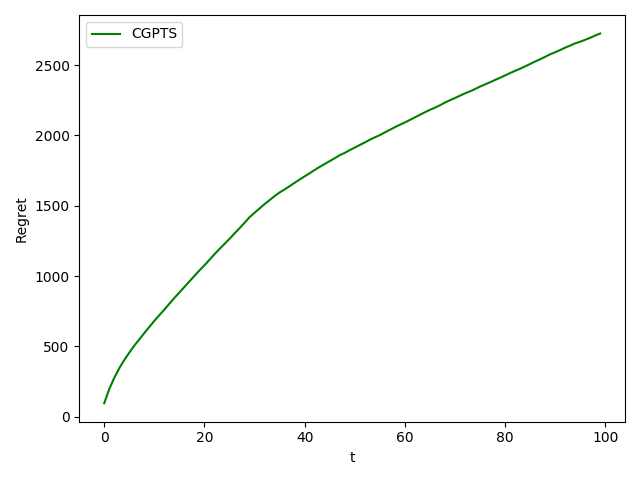
* T = 100
* Exp = 100
* first disaggregation day ~28 c2, c1c3
* second disaggregation day ~ 42 c1,c2,c3

Aggregate reward vs Disaggregate reward



As we can see from the picture, after the disaggregation there is a step in the reward in the disaggregated model, so disaggregating the Google sub-campaign is a very profitable choice.

Aggregate regret vs Disaggregate regret



The regret of the second model is much slower with respect to the one from the aggregated model, because the performance of the second one is more similar to the clairvoyant performances.