DATA INTELLIGENCE APPLICATIONS

PRICING AND ADVERTISING

Battistini Camilla
Di Luozzo Giovanni Antonio
Frigeri Michela
Scaramuzza Riccardo

SCENARIO

- VegStrenghty, e-commerce in the market of protein and vitamine shakes.
- Born to provide aid to vegan people, got appreciated by athletes.
- AllMightyShake©: new product designed to address the whole market
- Social Advertising: advertising campaign is pursued through Instagram Ads

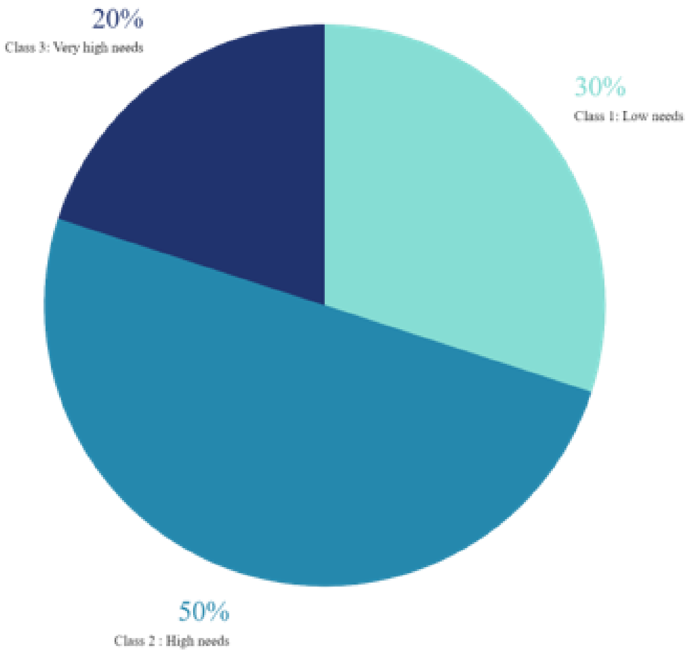
Instagram Advertising

SCENARIO

- Market segmentation: three underlying customer classes

- Two binary features:
  - Diet Profile
  - Physical activity Profile

		DIET PROFILE	
PHYSICAL ACTIVITY PROFILE		LOW	HIGH
	LOW	C1	C2
	HIGH	C3	C3

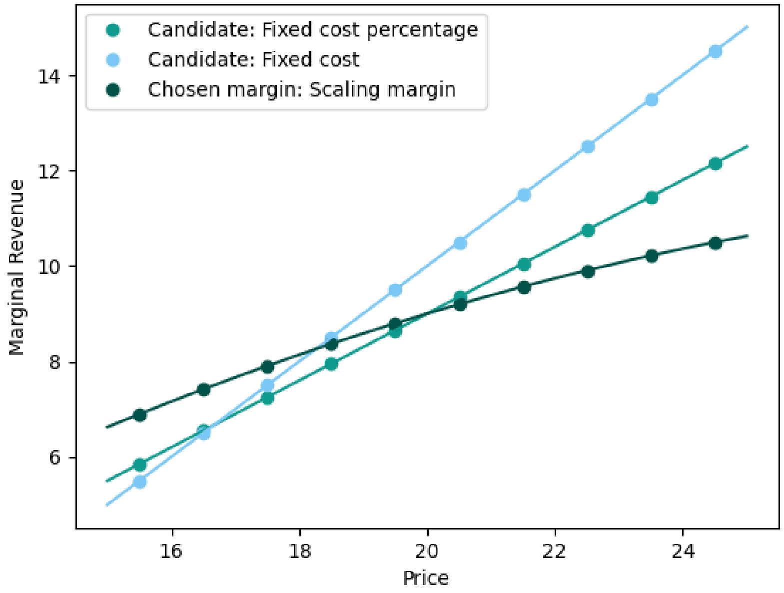


SIMULATED DATA: FUNCTIONS OF THE PRICE

MARGINAL REVENUE FUNCTION
Three candidates:

- Pure fixed cost
- Fixed and price-percentage cost
- "Real-life" designed function

$$MR(p) = p - 5 - \frac{1.5}{100}p^2$$

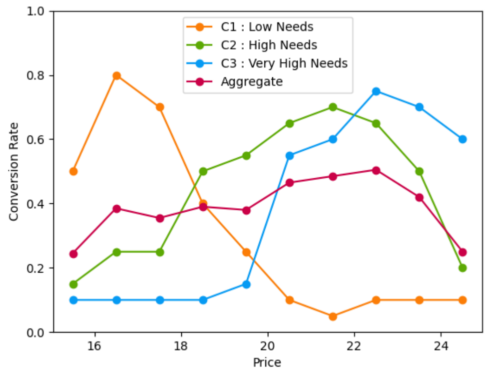


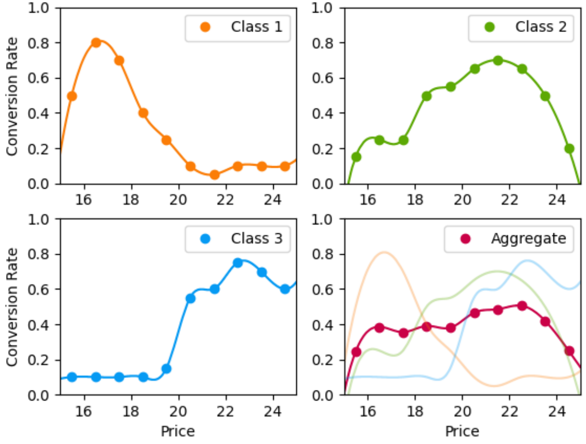
SIMULATED DATA: FUNCTIONS OF THE PRICE

CONVERSION RATE FUNCTIONS

Assumptions:

- Class C1 customers are more likely to finalize their purchase if the price is low
- Class C3 may consider a cheap price indicative of a not so worthy product
- Class C2 follows a behaviour in-between the above





SIMULATED DATA: FUNCTIONS OF THE BID

COST PER CLICK STOCHASTIC FUNCTION

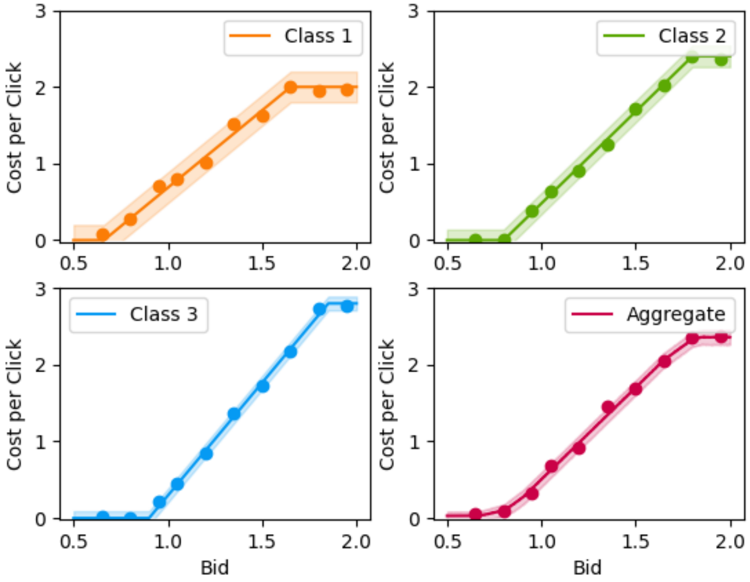
Assumptions:

- Class-specific mean levels are considered
- Saturation may happen both at small and big values of the bid
- Introduction of stochasticity through a zero-mean gaussian noise:

  $CPC_i(b) = CPC^*(b) + \varepsilon_i$

  $CPC^*(b)$: $[0.5, 2] \rightarrow \mathbb{R}$

  $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$



SIMULATED DATA: FUNCTIONS OF THE BID

NUMBER OF DAILY CLICKS STOCHASTIC FUNCTION

Assumptions:

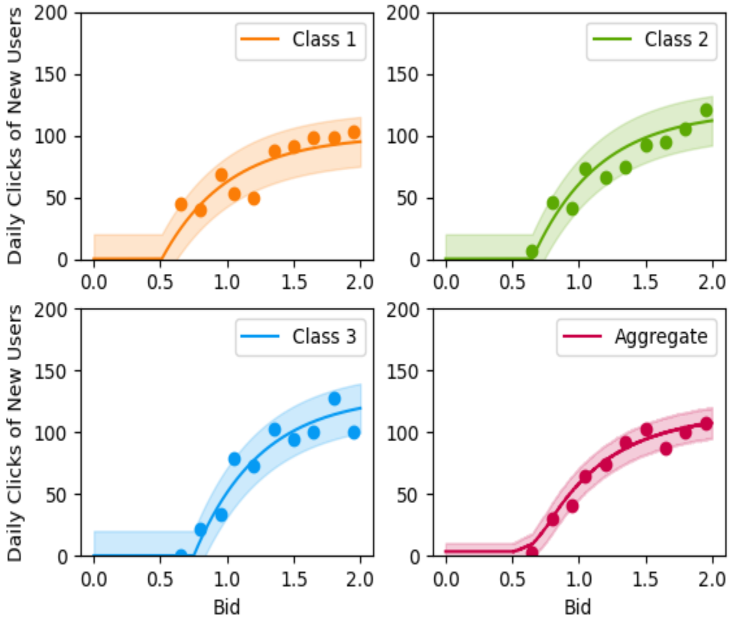
- Function domain: $\mathbb{N}$
- Too low values of the bid win no auction: zero costumers reached
- High values of the bid lead to better ads but this effect is losen for too high bids.

Stochastic representation:

- $NDC_i(b) = NDC^*(b) + \varepsilon_i$

$NDC^*(b): [0.5, 2] \rightarrow \mathbb{N}$

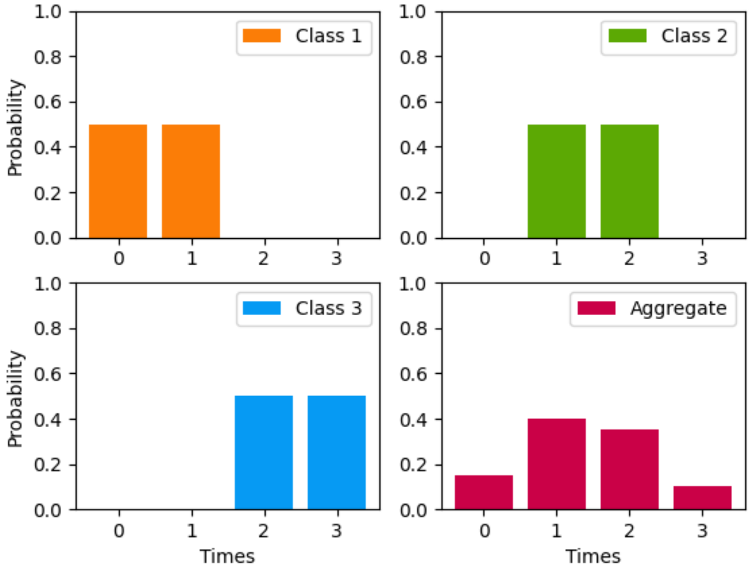
$\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$



SIMULATED DATA

CUSTOMER RETURNS WITHIN 30 DAYS
Assumptions:

- Discrete distribution whose support is $\{0, 1, 2, 3\}$
- Any customer is likely to buy again the product proportionally to his/her needs (and so, to his/her class)



OPTIMIZATION PROBLEM

GENERAL FORMULATION

$$\max_{p_i,b_i} \sum_{i=1}^{K} \mathbb{E}[NDC_i(b_i)] \cdot CR_i(p_i) MR(p_i)(1+1/30 \cdot \mathbb{E}[R_i]) - \mathbb{E}[NDC_i(b_i)] \cdot \mathbb{E}[CPC_i(b_i)]$$

Variable	Description
t	time instant
T	length of observation period
i	customer class
K	total number of customer classes
$b_{i,t}$	proposed bid for class i at time t
$p_{i,t}$	proposed price for class i at time t
NDC_i	stochastic number of clicks of new users of subcampaign i given $b_{i,t}$
R_i	stochastic number of purchases within next 30 days of users of class i given $b_{i,t}$
CR_i	conversion rate for class i given $p_{i,t}$
MR	marginal revenue for price $p_{i,t}$
CPC_i	stochastic cost per click of subcampaign i given $b_{i,t}$

OPTIMIZATION PROBLEM

AGGREGATED DATA

$$\max_{p,b} \mathbb{E}[NDC_{agg}(b)] \cdot CR_{agg}(p)MR(p)(1+1/30 \cdot \mathbb{E}[R_{agg}]) - \mathbb{E}[NDC_{agg}(b)] \cdot \mathbb{E}[CPC_{agg}(b)]$$

CONTEXTUAL PRICING

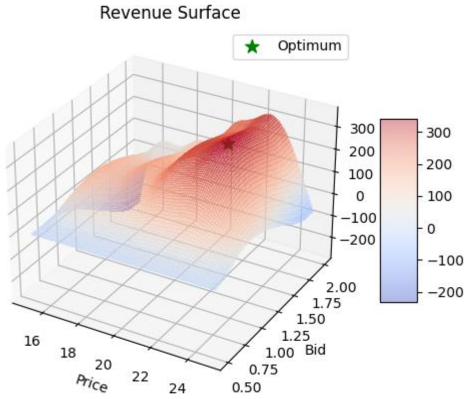
$$\max_{p_i,b} \sum_{i=1}^{K} \mathbb{E}[NDC_{agg}(b)] \cdot CR_i(p_i)MR(p_i)(1+1/30 \cdot \mathbb{E}[R_i]) - \mathbb{E}[NDC_{agg}(b)] \cdot \mathbb{E}[CPC_{agg}(b)]$$

OPTIMIZATION PROBLEM

BRUTE FORCE SOLUTION

- Candidates

Price	15.50	16.50	17.50	18.50	19.50	20.50	21.50	22.50	23.50	24.50	€
Bid	0.65	0.80	0.95	1.05	1.20	1.35	1.50	1.65	1.80	1.95	€



MODEL COMPLEXITY

$$\mathcal{O}(N \cdot B \cdot P)$$

Where:

- N = number of costumer classes
- B = total number of bid values
- P = total number of prices values

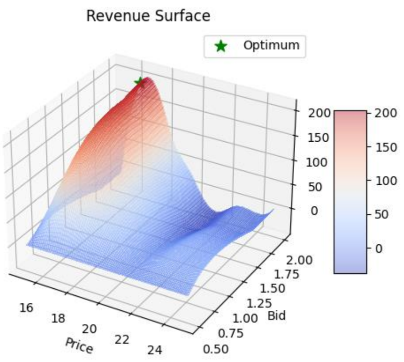
- Aggregated Opt solution

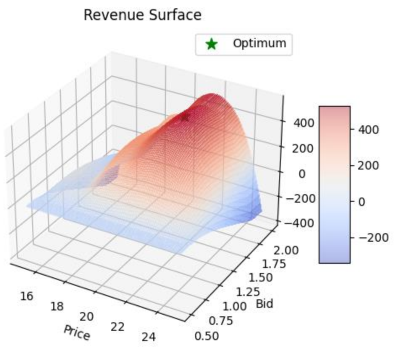
Maximum expected revenue	337.91€
$p_{agg}*$	22.5
$b_{agg}*$	1.35

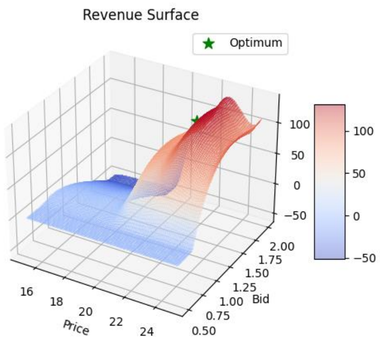
OPTIMIZATION PROBLEM

BRUTE FORCE SOLUTION

- Class 1 opt. solution
- Class 2 opt. solution
- Class 3 opt. solution







Maximum expected revenue	380.56€
p_1*	16.5
b_1*	1.95

Maximum expected revenue	524.83€
p_2*	21.5
b_2*	1.5

Maximum expected revenue	634.28€
p_3*	22.5
b_3*	1.5

OPTIMIZATION PROBLEM: ONLINE-LEARNING

WHAT WE OBSERVE

- Random Variables

CPC	cost per click (fixed the bid)
ACC	draw from a Bernoulli of mean $CR(price)$
NDC	number of daily clicks (fixed the bi)
R	number of future purchases within 30 days

- Time horizon: 365 days
- Potential delays in the feedback
- Algorithm: MAB

MAB's MATHEMATICAL FORMULATION

$$\min T \cdot \mu * - \sum_{t=0}^{T} \mu_t$$

$$\mu * := \max_{p,b} DailyExpectedRevenue(CR(p), NDC(b), CPC(b))$$

$$\mu_t := DailyExpectedRevenue(CR(p_t), NDC(b_t), CPC(b_t))$$

FIXED BID AND ONLINE PRICING

ASSUMPTIONS

- We consider the **aggregated case** in which no contextual pricing is applied
- We consider the value of mean returns to be known
- Bid related values are considered as fixed

ENVIRONMENT

- The environment is modeled by using a **binomial sampler**

LEARNER

- Thompson Sampling and UCB1 learners modified for a Binomial reward
- What the learner registers is the expected daily revenue, computed using daily rewards

$$\max_{p,b} \mathbb{E}[NDC_{agg}(b)] \cdot CR_{agg}(p) MR(p)(1 + 1/30 \cdot \mathbb{E}[R_{agg}]) - \mathbb{E}[NDC_{agg}(b)] \cdot \mathbb{E}[CPC_{agg}(b)]$$

FIXED BID AND ONLINE PRICING

TS-ALGORITHM

Updated parameters at the end of the day using the realization of a binomial distribution

$$\begin{cases} \alpha = \alpha + r \\ \beta = \beta + (n - r) \end{cases}$$

Number of daily positive realizations

Total number of daily costumers

UCB1-ALGORITHM

It works as the classical one, but it performs the update of parameters necessary to compute the upper Hoeffding bound in a batch manner.

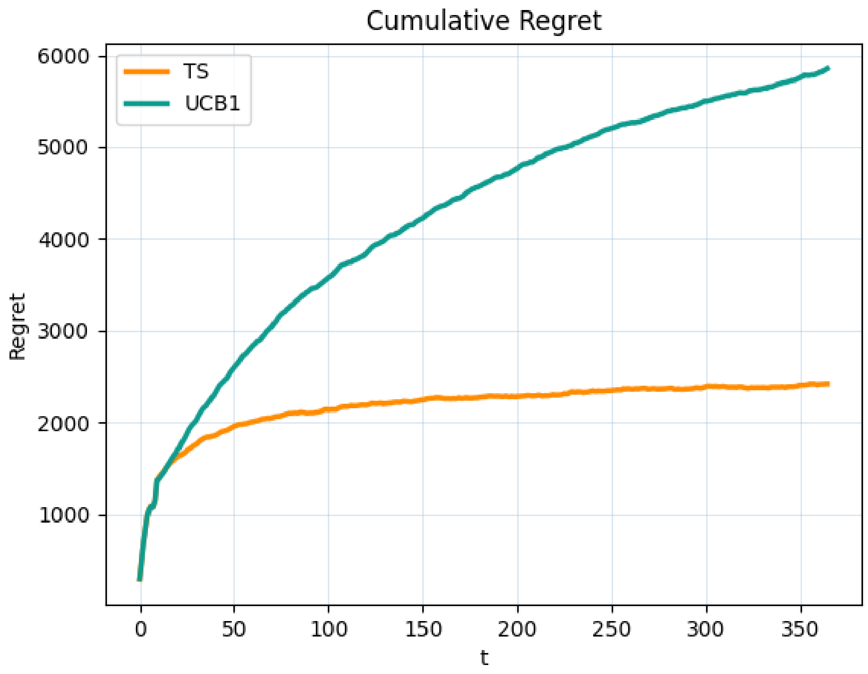
FIXED BID AND ONLINE PRICING

PERFORMANCE METRIC

We repeat 100 times a one year simulation in order to obtain the estimate of:

- **Cumulative Regret,** to identify the better performing algorithm

- **Daily Expected Revenue,** to understand if the algorithm actually converges to the optimum value.



PERFORMANCE EVALUATION

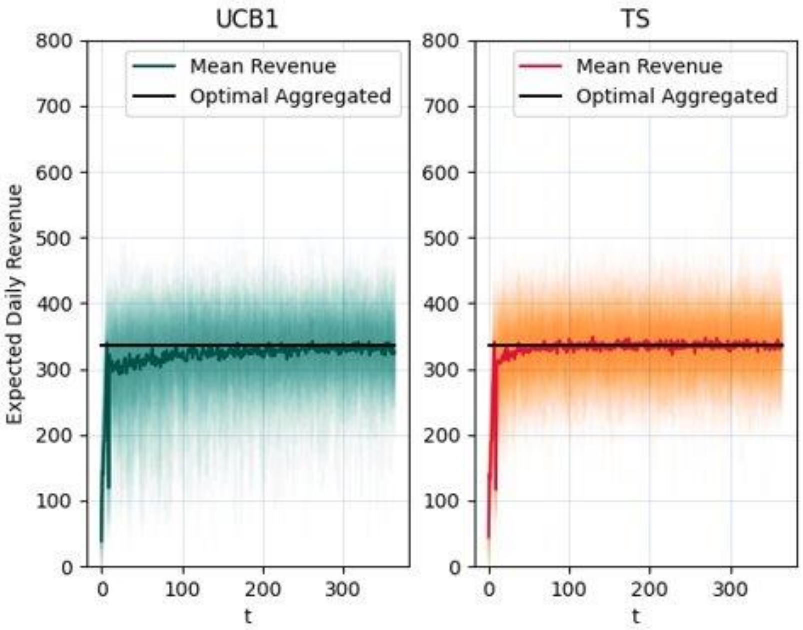
- Both learners show sublinear Cumulative Regret

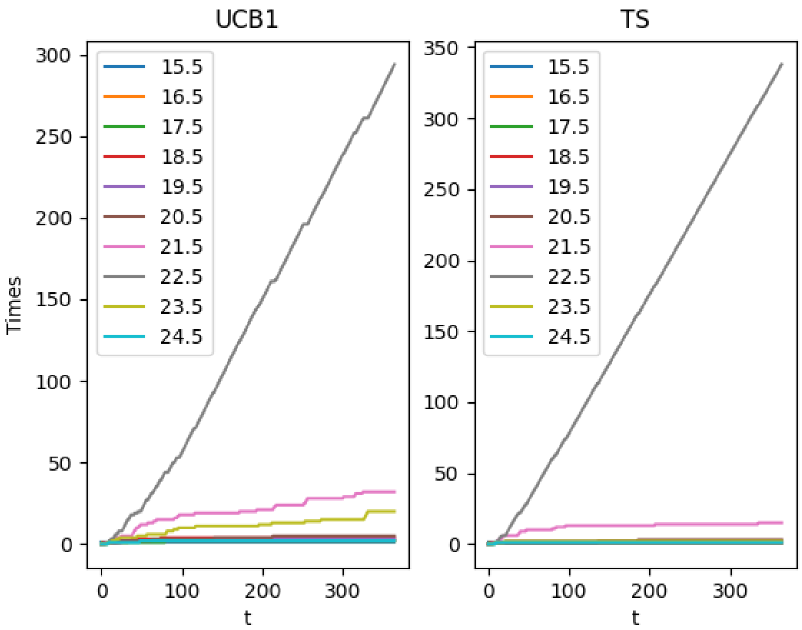
- TS achieves much better performances

FIXED BID AND ONLINE PRICING

PERFORMANCE EVALUATION

- Fast convergence for both learners
- TS achieves perfect convergence to the optimum value; UCB1 seems to converge to slightly suboptimal policy



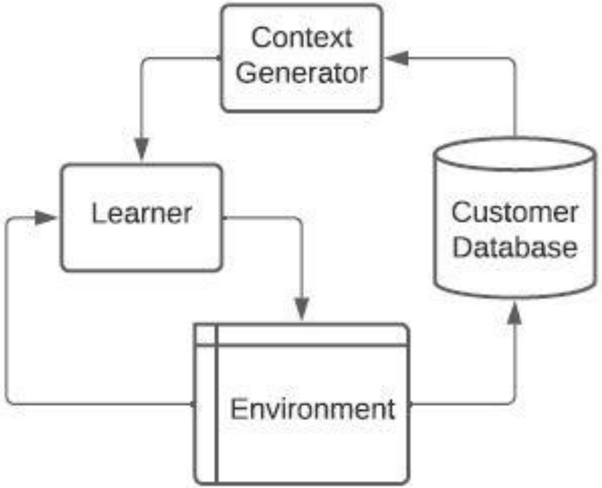


FIXED BID AND CONTEXTUAL PRICING

OPTIMAL SOLUTION

- Different price to each true underlying class

Maximum Expected Revenue	485.37€
p_1*	16.50€
p_2*	21.50€
p_3*	22.50€
$b*$	1.35€

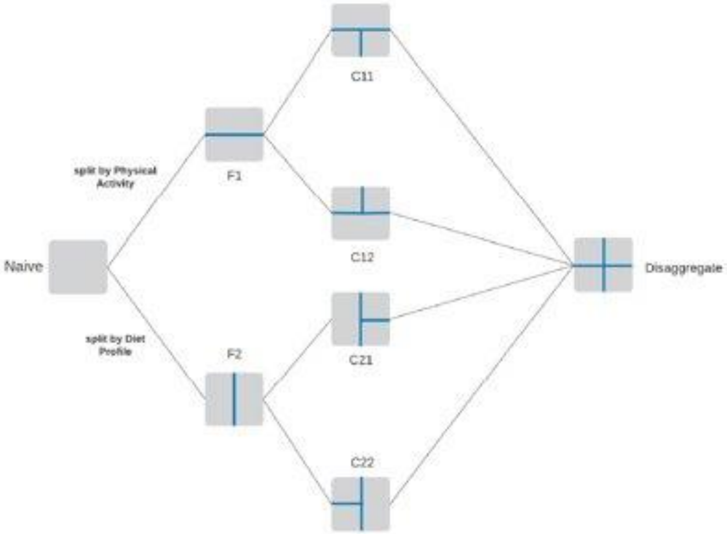


$$\max_{p_i,b} \sum_{i=1}^{K} \mathbb{E}[NDC_{agg}(b)] \cdot CR_i(p_i) MR(p_i)(1 + 1/30 \cdot \mathbb{E}[R_i]) - \mathbb{E}[NDC_{agg}(b)] \cdot \mathbb{E}[CPC_{agg}(b)]$$

FIXED BID AND CONTEXTUAL PRICING

IMPLEMENTATION

- Customer
- Feature Generation
- Context Mapping



- Context Generators

1. Naïve
2. Brute Force
3. Greedy

$$V_{context} = \sum p_{segment} * \mu^{*}_{segment}$$

FIXED BID AND CONTEXTUAL PRICING

IMPLEMENTATION

- Customer Database

1. Daily variables
2. Collection of daily customers
3. Context history
4. Mean Returns Estimation

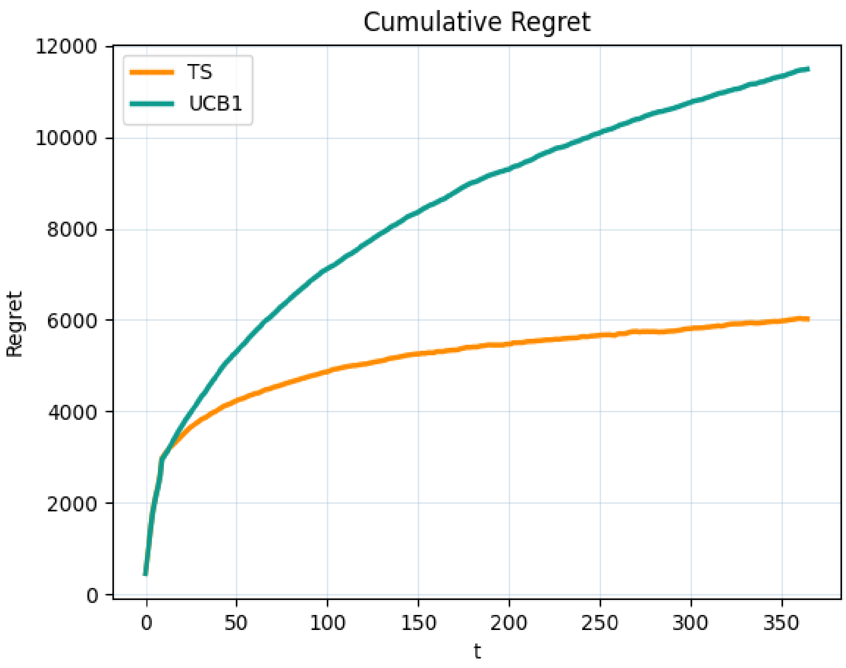
- Learners

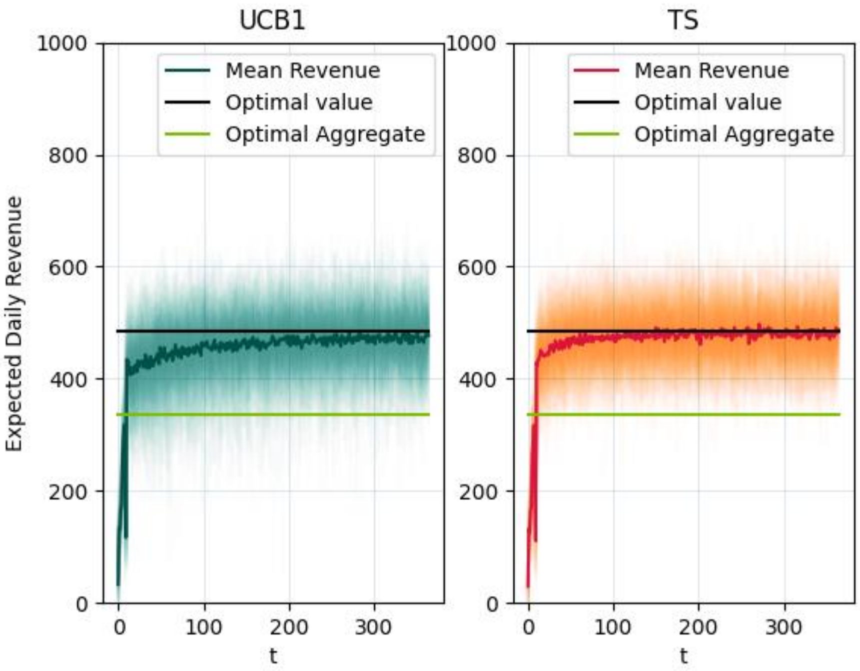
1. TS & UCB1
2. Offline-Training

```
Pseudocode for experiment workflow
Initialize:
Database
Context ← Naive
Learner
Environment
Forall day:
  if new week:
        new context ← ContextGenerator
        update Database.ContextHistory
        if Learner.Context != new context:
             update Learner.Context
             perform Learner.OfflineTraining
  Daily Routine:
        prices ← Learner.pull-arms
        Database.dailyVariables ←prices, bid, costperclick
        N ← sample from NDC_agg
        for each customer (1 to N):
             create Customer (ID, feature vector)
             acceptance ← Environment.round(feature, context)
             record Customer in Database
        update Learner
        update Database
```

FIXED BID AND CONTEXTUAL PRICING

PERFORMANCE EVALUATION





FIXED PRICE AND ONLINE BIDDING

ASSUMPTIONS

- No budget constraint
- No class diversification wrt bidding
- Price related values considered as fixed

ENVIRONMENT

At each round:

- Samples values of NDC and CPC according to the given bid
- Compute the revenue expression using the sampled values

LEARNER

- MAB setting with a bid value for each arm
- 10 candidates (possible bid values)
- Models Considered:
  - Gaussian TS (GTS)
  - Gaussian Process TS (GPTS)

FIXED PRICE AND ONLINE BIDDING

GTS

- Same as TS Algorithm, but on the bid values
- Prior and Sampling distributions assumed to be Gaussian

GPTS

- Using Gaussian Process as a regressor to model the Revenue distribution, assuming:

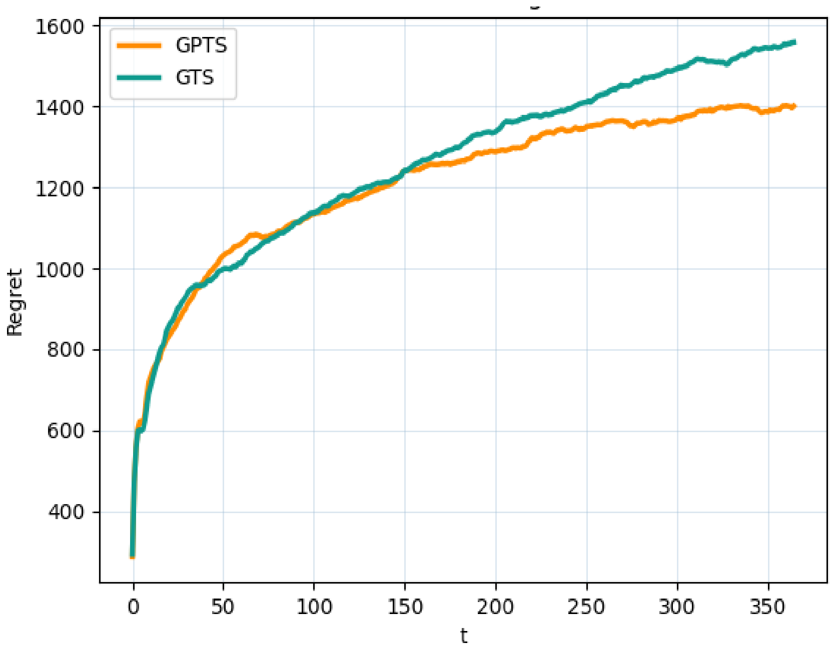
  - Zero mean GP
  - covariance function: squared exponential kernel function $k(x, x')$ plus a white noise:

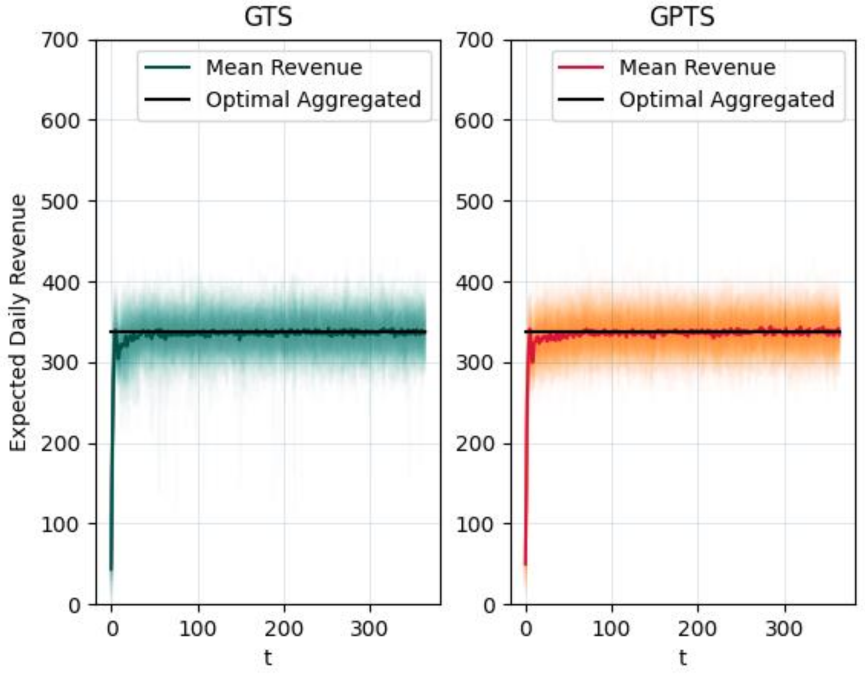
$$k(x, x') = \theta^2 e^{-\frac{(x-x')^2}{2l^2}} + WN(\sigma^2)$$

- Mean and Variance are updated according to the daily collected revenue.

FIXED PRICE AND ONLINE BIDDING

PERFORMANCE EVALUATION





JOINT BIDDING & PRICING

We now want to apply the two algorithms sequentially in order to discover the optimal joint bidding-pricing strategy.

ADVERTISING :

- GPTS selects a bid value at the beginning of each day (according to collected data)
- The environment samples NDC and CPC values according to the bid value

PRICING :

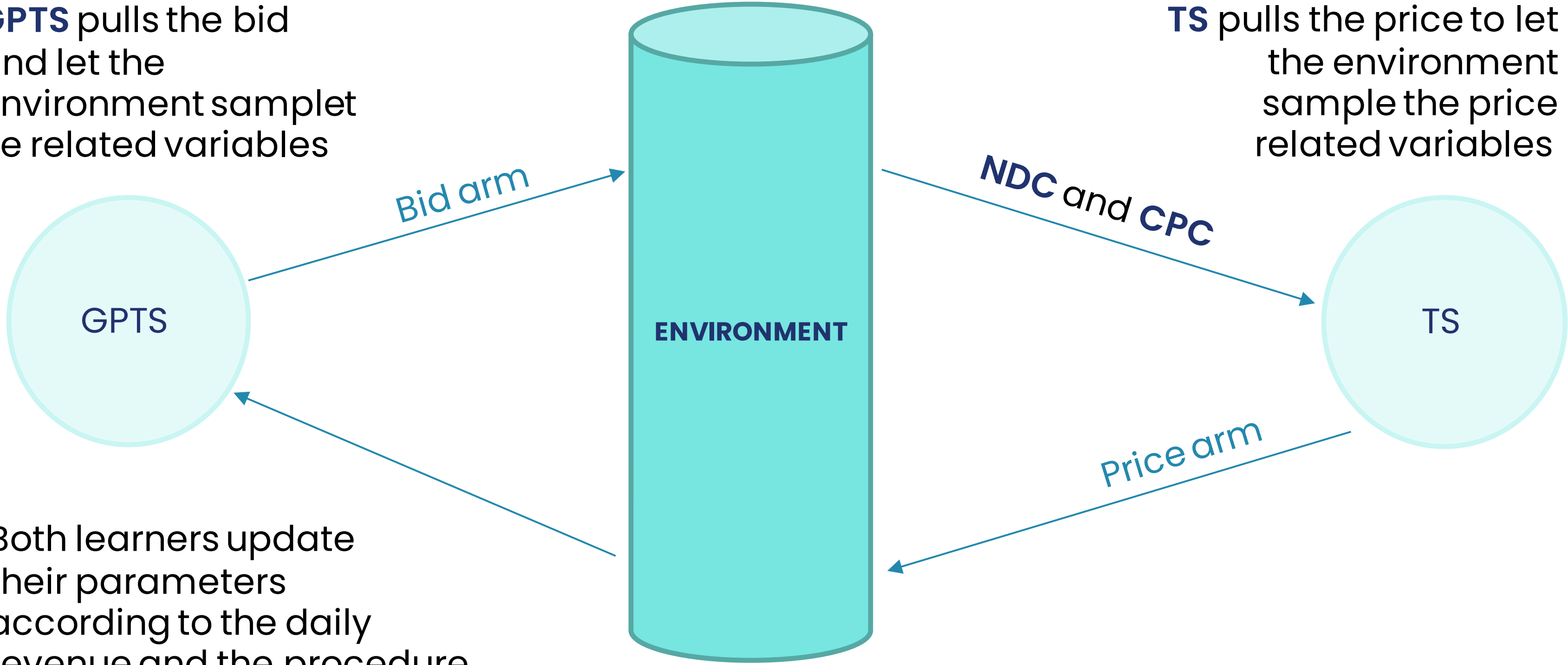
- TS selects the price value for the day
- The environment decides for each user if they accept to buy at given price

- Given the full simulation of the day , the expected daily revenue can be computed.
- GPTS and TS parameters are updated at the end of each day, according to the collected revenue.

JOINT BIDDING & PRICING

GPTS pulls the bid and let the environment samplet he related variables

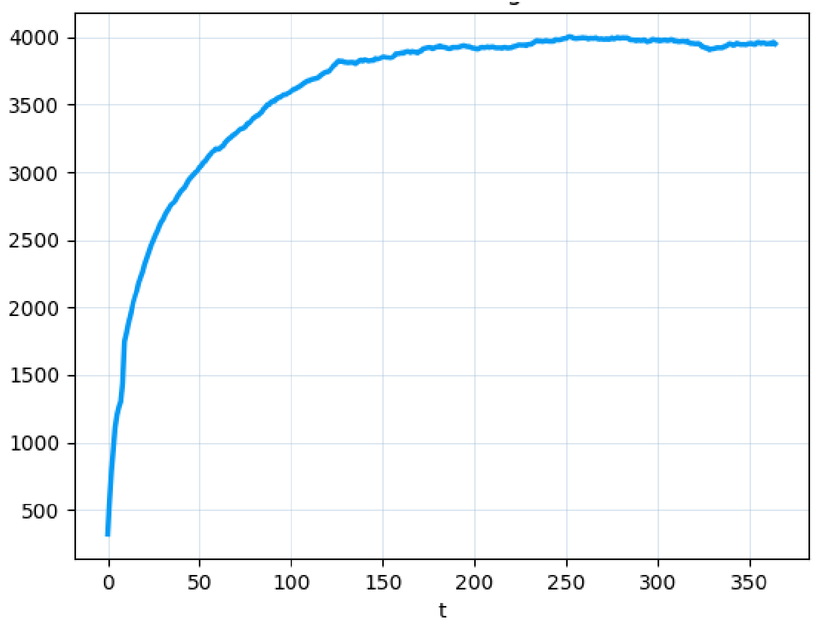
TS pulls the price to let the environment sample the price related variables

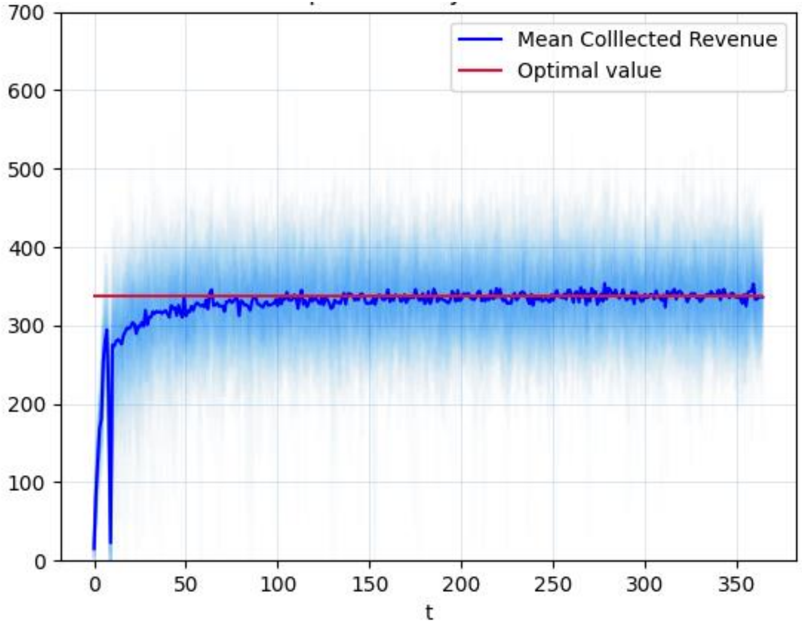


Both learners update their parameters according to the daily revenue and the procedure is repeated for the next day

JOINT BIDDING & PRICING

PERFORMANCE EVALUATION





JOINT BIDDING & CONTEXTUAL PRICING

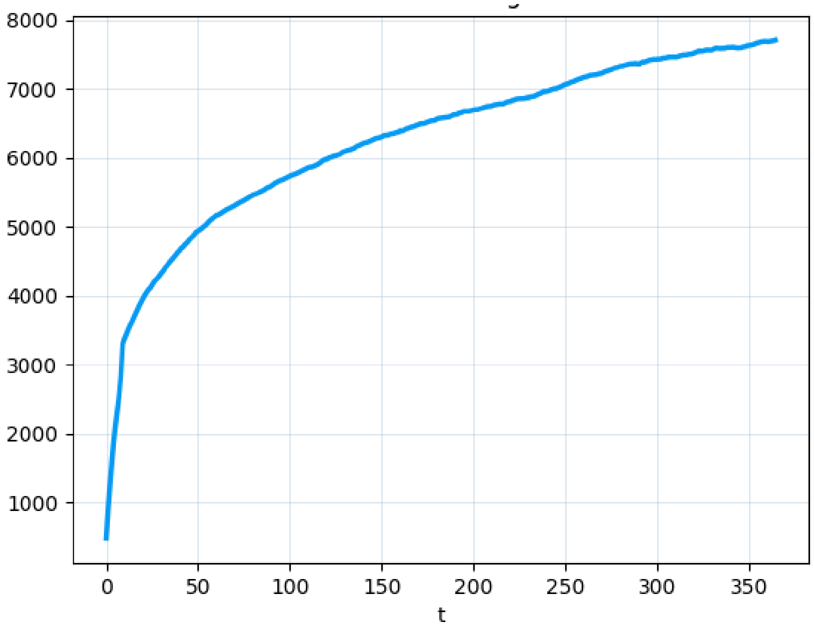
CONTEXTUAL SCENARIO

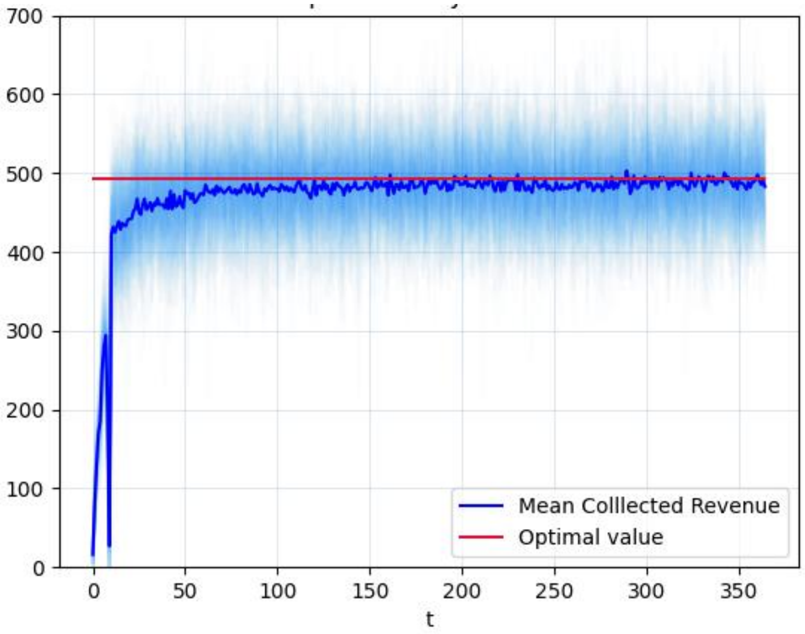
ASSUMPTIONS

- Same mechanism as before for each day
- Context diversification only wrt pricing
- Each day the drawn price is proposed to the 3 classes (context C12)
- The optimal context is assumed to be known beforehand

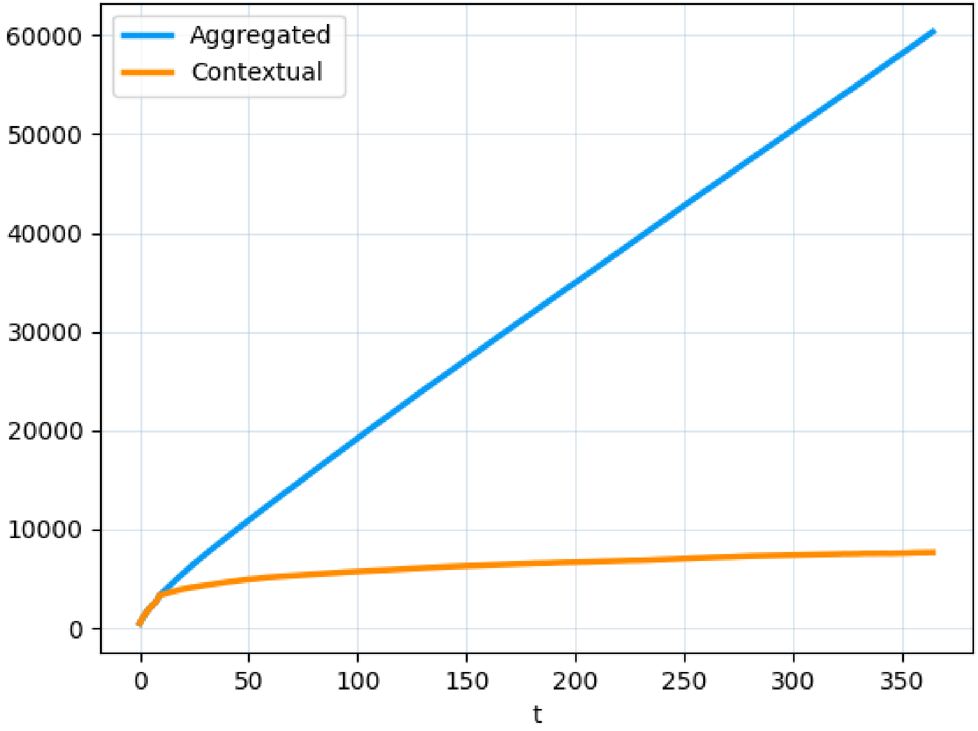
JOINT BIDDING & CONTEXTUAL PRICING

PERFORMANCE EVALUATION





CONCLUSIONS



- Aggregated Optimum:
  EDR = 337.9 €
- Contextual Optimum:
  EDR = 492.5 €
- Contextual solution has a slower convergence that may lead to a loss
- Comparing regrets computed wrt the highest revenue possible (492.5 €) it's clear that the contextual algorithm overcomes the aggregated one