



**POLITECNICO**  
MILANO 1863

# Data Intelligence Applications - Presentazione del progetto

Pricing and Advertising  
d'Amato - Frantuma - Fucci

Lo scopo del progetto è ricercare, per uno specifico prodotto sul mercato, la miglior combinazione di prezzo proposto al pubblico e offerta (per click) per la pubblicizzazione online, al fine di massimizzare i profitti in un orizzonte temporale di un anno.

Le variabili controllate nel modello sono:

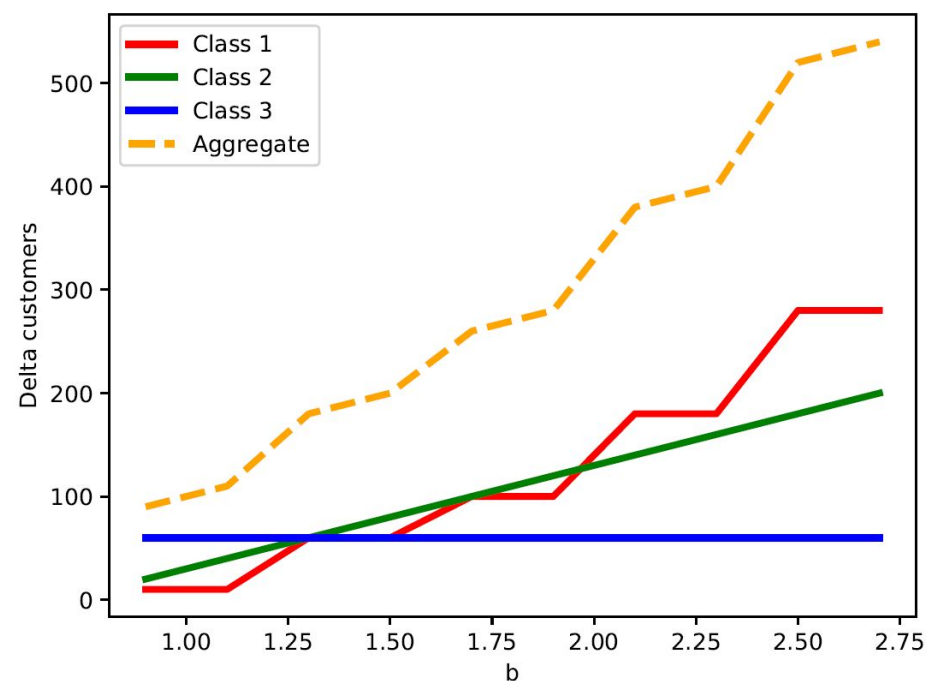
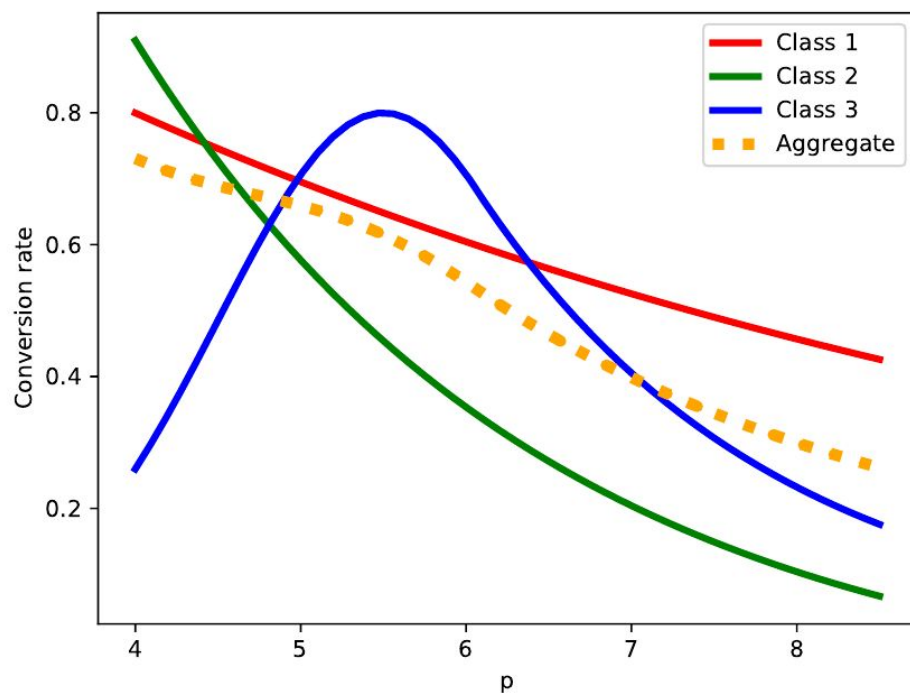
- Prezzo,  $p$
- Offerta,  $b$

Mentre le variabili aleatorie da esse dipendenti sono:

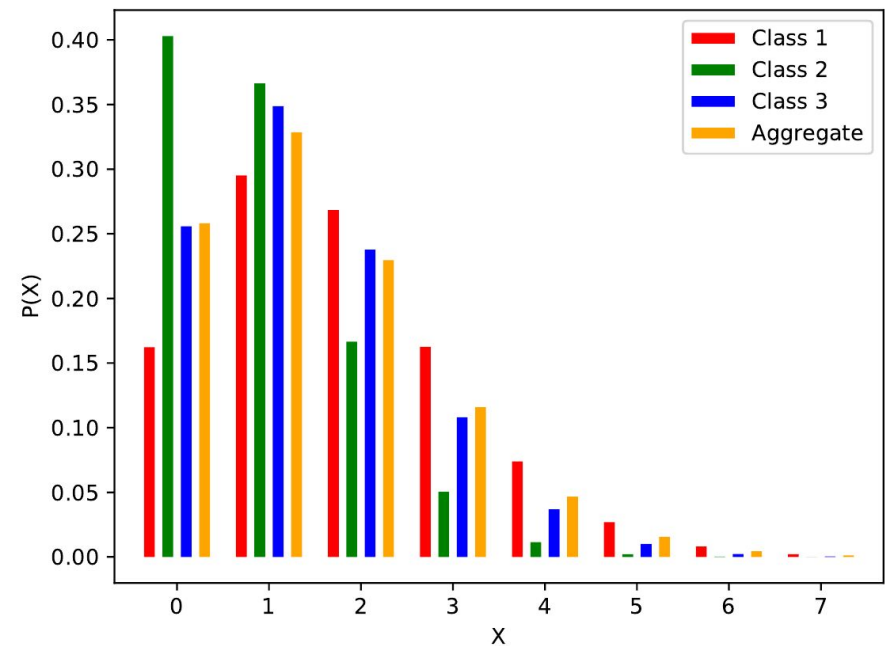
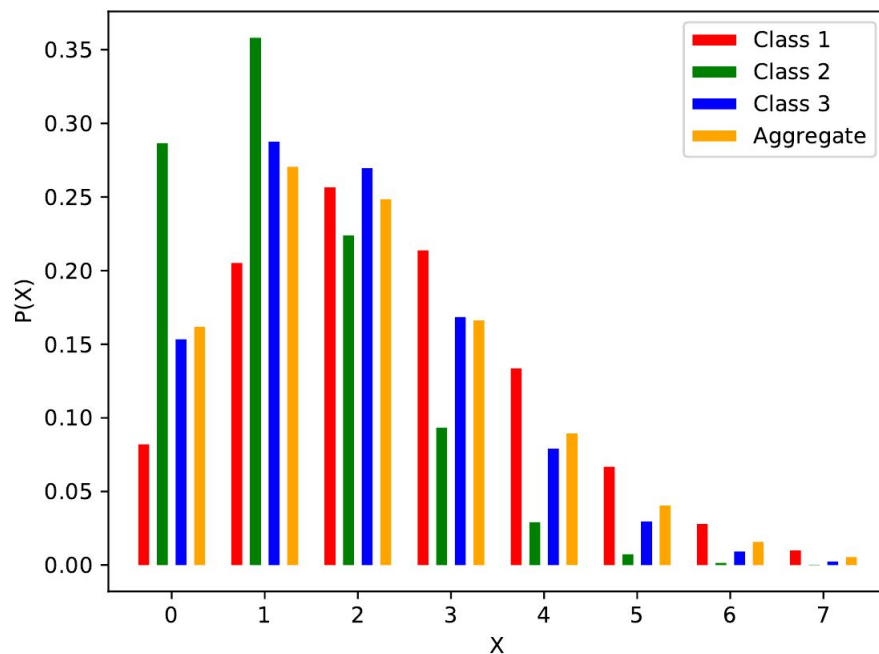
- Numero di click giornalieri
- Visite future
- Costo per click
- Numero di vendite attraverso pubblicità (tasso di conversione)

- Nuovi click giornalieri:  $d_t(b_t) \sim \mathcal{N}(\mu_{0,i} + \mu_{bid,i}(b_t), \sigma^2)$
- Vendite attraverso pubblicità al giorno  $t$ :  $q_t(b_t, p_t) = \sum_{i=1}^{d_t(b_t)} X(p_t)$ , dove  $X(p_t) \sim \mathcal{Be}(r(p_t), 1 - r(p_t))$  and  $r(p_t)$  è il tasso di conversione valutato al prezzo  $p_t$ .
- Costo per click:  $c_i(b_t) = b_t - |X(b_t)|$ ,  $X \sim \mathcal{N}(\frac{b_t}{\alpha}, 0.1)$ , dove  $\alpha$  è una costante di normalizzazione ottenuta pesando le relative costanti di ogni classe.
- Visite future di ogni utente:  $n_i(p) \sim \mathcal{Poisson}(\lambda)$

Caratteristiche	anziano	giovane
attivo	Sportivo (classe 1)	Sportivo (classe 1)
sedentario	Pensionato (classe 2)	Programmatore (classe 3)



Le visite nei 30 giorni successivi all'acquisto sono modellate con una distribuzione di Poisson, il cui parametro lambda varia in base alla classe di utenti e al prezzo di acquisto.



Confezione da 6 lattine di energy drink, con relativo dash button incluso al primo acquisto



La classe aggregata corrisponde all'aggregazione delle classi 1,2 e 3. Pertanto i suoi parametri sono derivati dalla media pesata tra le classi, basata sul numero di nuovi clienti giornalieri per ogni classe, valore che dipende dall'offerta  $b$ . Per esempio:

$$X_{agg} \sim \mathcal{Poisson}(\lambda_{agg})$$

$$\lambda_{agg} = \frac{n_1(b)\lambda_1(p) + n_2(b)\lambda_2(p) + n_3(b)\lambda_3(p)}{n_1(b) + n_2(b) + n_3(b)}$$

$$n_i \sim \mathcal{N}(\mu_i, \sigma^2 | b)$$



$$\text{maximize } \mathcal{R}(p, b) = \sum_{i=1}^{q(b,p)} [(p - c_{\text{prod}})(1 + n_i(p)) - c_i(b)]$$

```
 $\mathcal{A} :$   max  $\leftarrow -\infty$   
        for  $p$  in  $P$   
          for  $b$  in  $B$   
            if  $\mathcal{R}(p, b) > \text{max}$   
              then max  $\leftarrow \mathcal{R}(p, b)$   
        return max
```



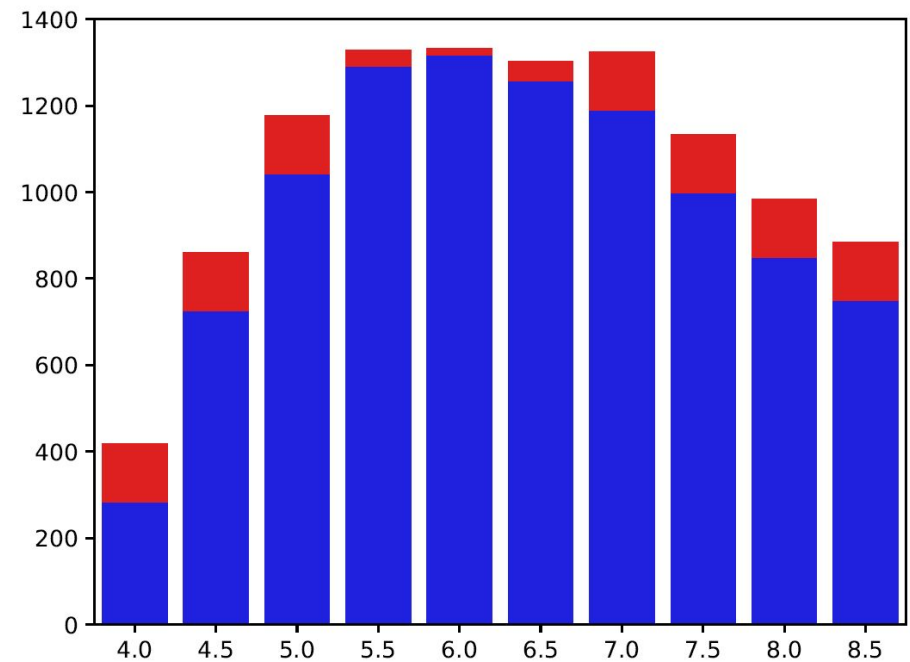
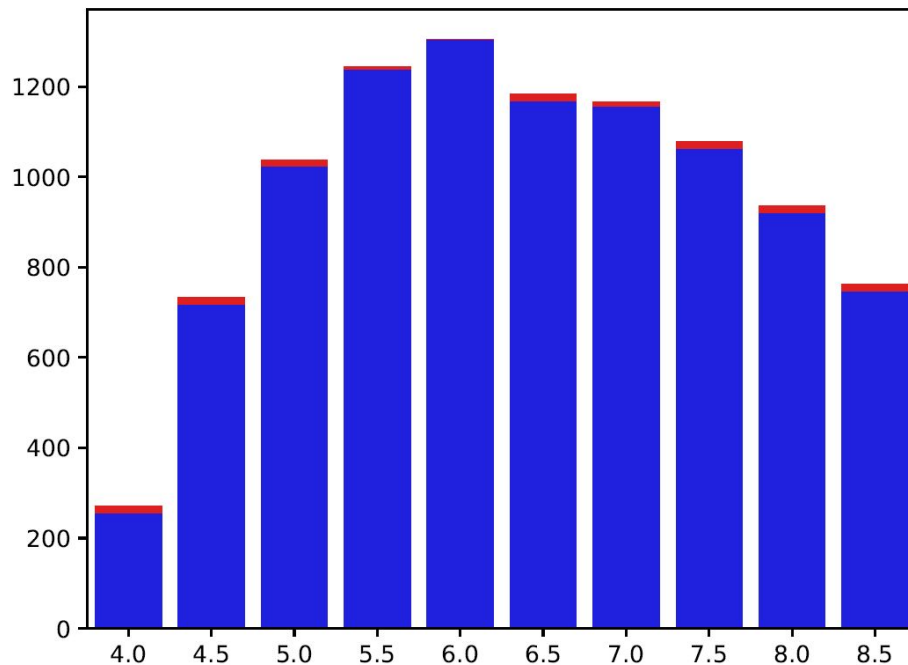
$$\text{maximize} \quad \sum_{t=1}^{365} \mathcal{R}_t(p, b) = \sum_{t=1}^{365} \sum_{i=1}^{q_t(b,p)} [(p_t - c_{prod})(1 + n_i(p_t)) - c_i(b_t)]$$

- Nuovi click giornalieri:  $d_t(b_t) \sim \mathcal{N}(\mu_{0,i} + \mu_{bid,i}(b_t), \sigma^2)$
- Vendite attraverso pubblicità al giorno  $t$ :  $q_t(b_t, p_t) = \sum_{i=1}^{d_t(b_t)} X(p_t)$ , dove  $X(p_t) \sim \mathcal{Be}(r(p_t), 1 - r(p_t))$  and  $r(p_t)$  è il tasso di conversione valutato al prezzo  $p_t$ .
- Costo per click:  $c_i(b_t) = b_t - |X(b_t)|$ ,  $X \sim \mathcal{N}(\frac{b_t}{\alpha}, 0.1)$ , dove  $\alpha$  è una costante di normalizzazione ottenuta pesando le relative costanti di ogni classe.
- Visite future di ogni utente:  $n_i(p) \sim \mathcal{Poisson}(\lambda)$

# Upper Confidence Bound

9

$$a_t \leftarrow \operatorname{argmax}_{a \in A} \left\{ \bar{x} + 80 \sqrt{\frac{2 \log(t)}{n_a(t-1)}} \right\}$$



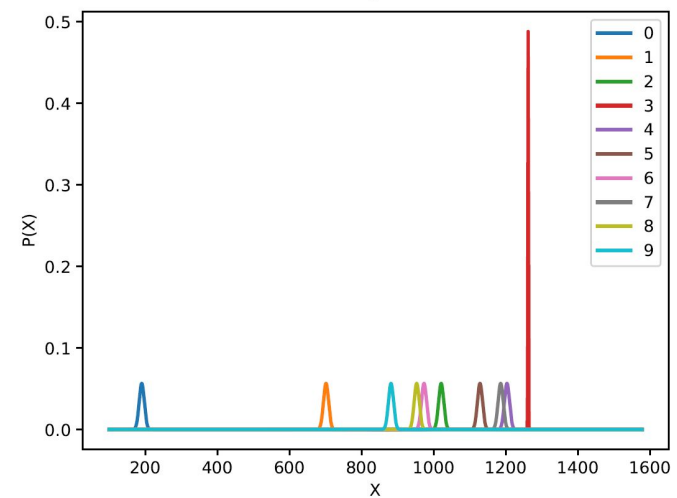
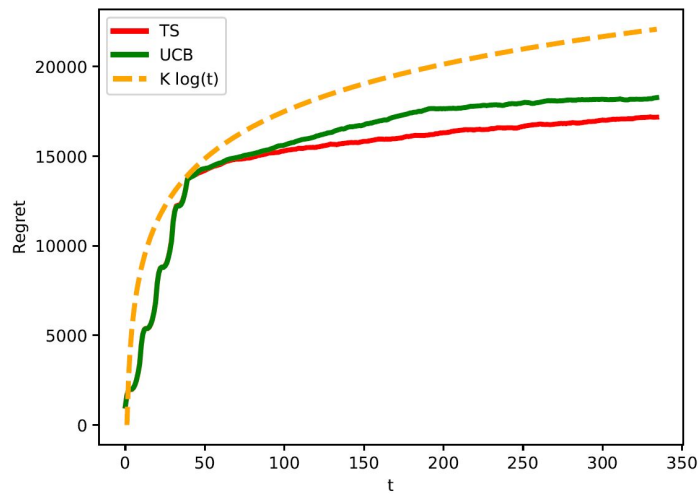
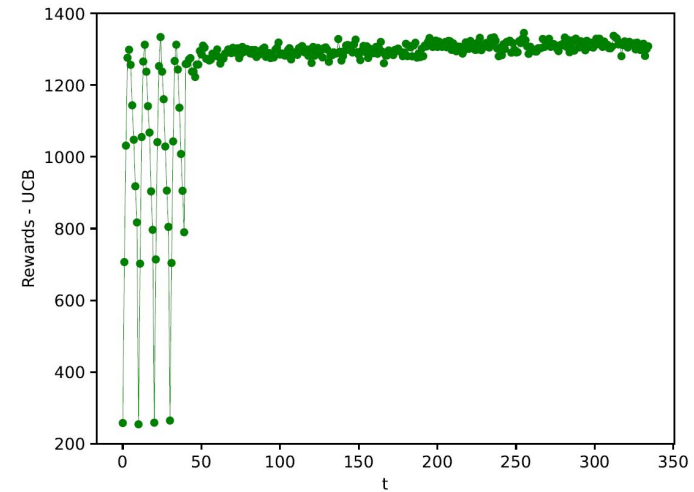
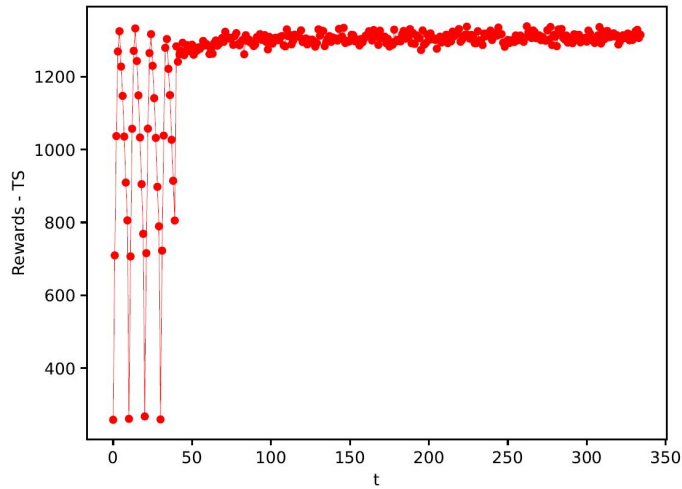
- Nell'algoritmo GTS abbiamo scelto di aggiornare i parametri delle distribuzioni di probabilità a posteriori attraverso la definizione di precisione, invece di calcolare direttamente la deviazione standard dall'insieme dei campioni.

$$\tau_{0,a} \leftarrow \tau_{0,a} + n_a(t-1)\tau_a$$

$$\mu_{0,a} \leftarrow \frac{\tau_{0,a}\mu_{0,a} + \tau_a \sum_{i=1}^n x_i}{\tau_{0,a} + n_a(t)\tau_a}$$

# Step 3: risultati

11



Il vincolo di sicurezza richiesto in questo punto è implementato tramite una maschera binaria che esclude dai candidati i bracci valutando la funzione di ripartizione della distribuzione a posteriori, secondo la condizione:

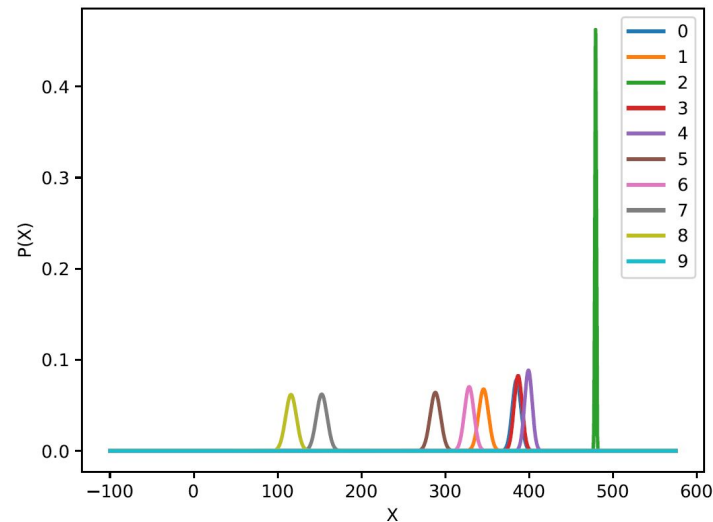
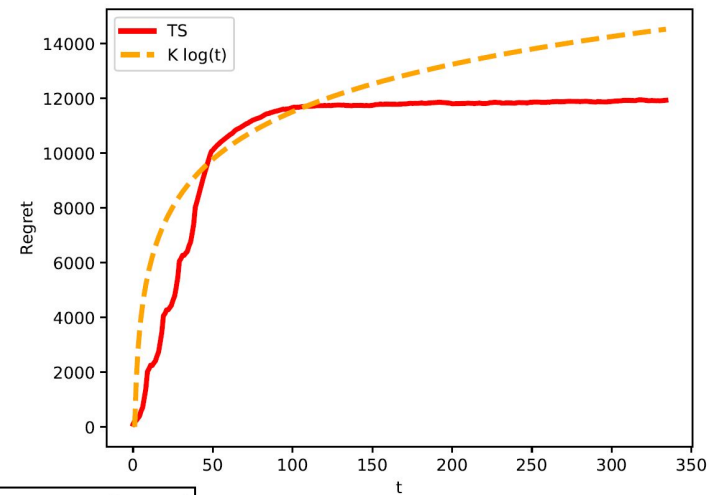
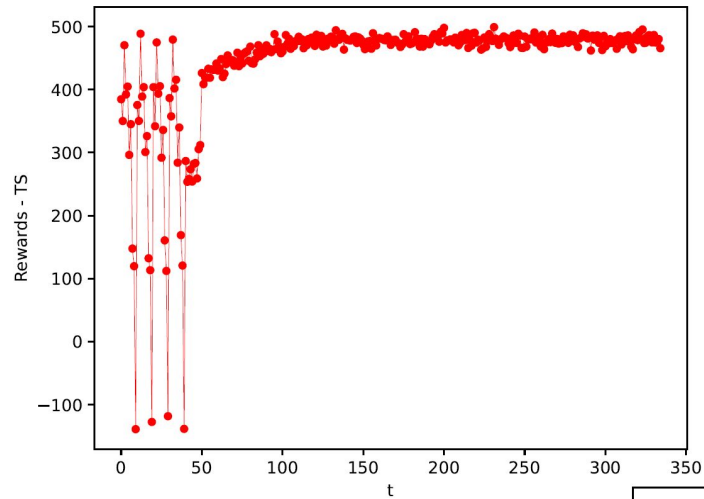
$$\Phi_a(0) = \Phi(0|\mu_a, \sigma_a^2) = P(x \leq 0|\mu_a, \sigma_a^2) \geq 0.2$$

verificata tramite

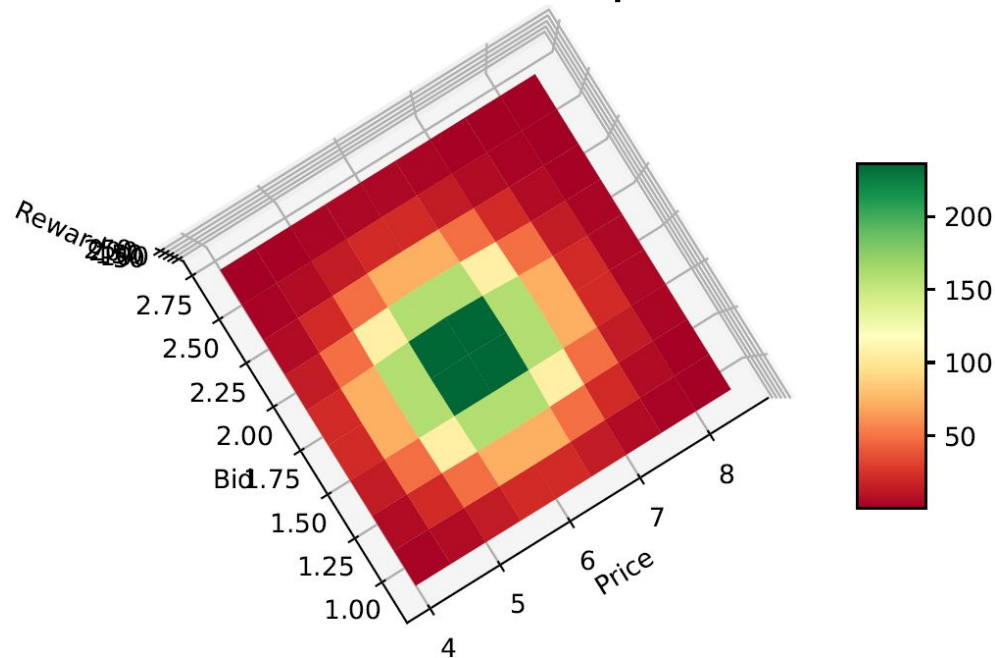
```
scipy.stats.norm.cdf(0, mu_a, sigma_a)
```

# Step 5: risultati Thompson Sampling

13



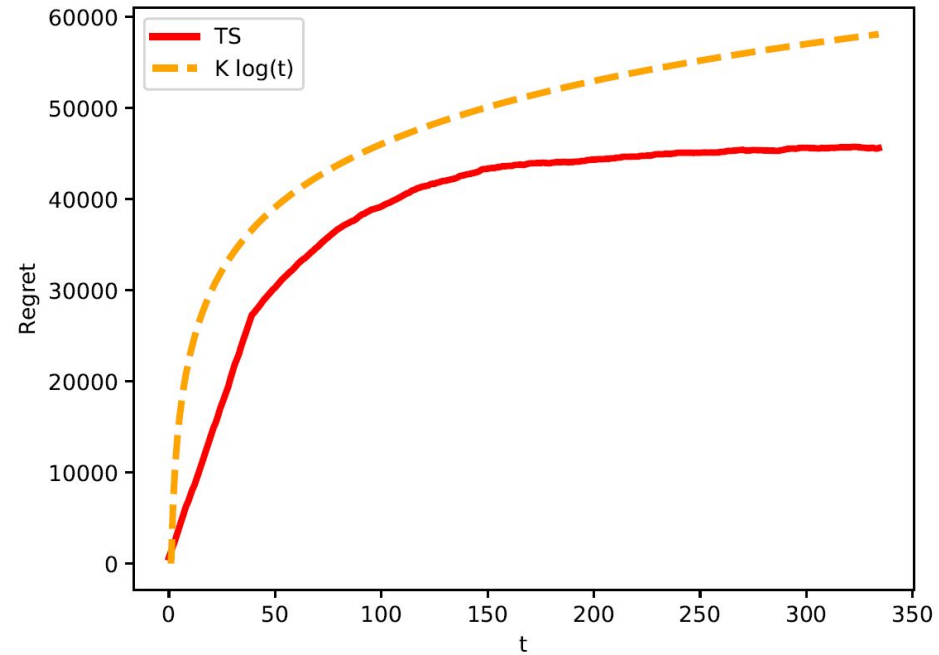
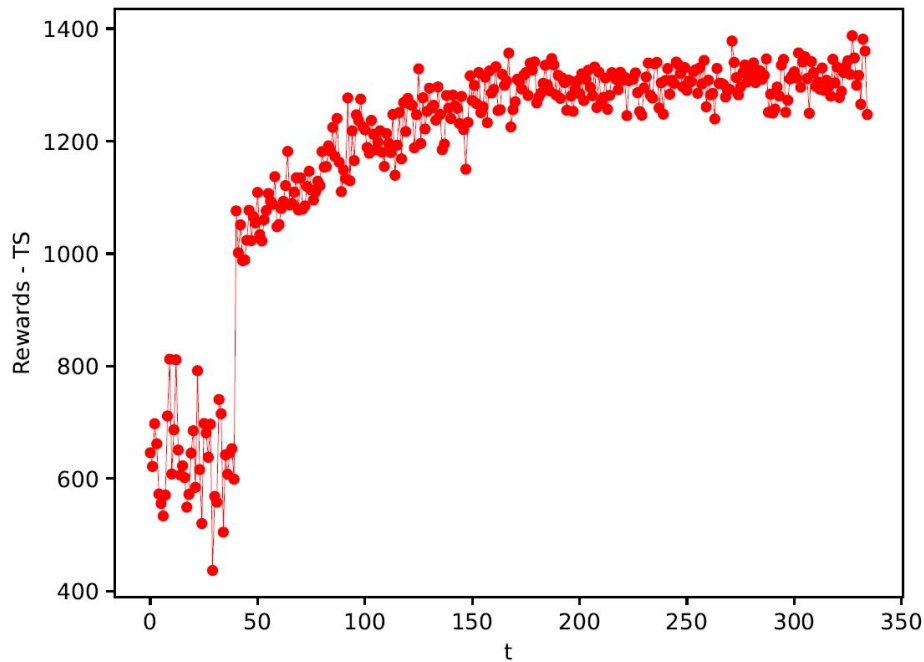
- Per affrontare il problema di una combinazione di  $10 \times 10$  bracci, abbiamo utilizzato un algoritmo di Gaussian Process Thompson Sampling. Il kernel utilizzato tiene conto della differenza di scala dei vettori di input:





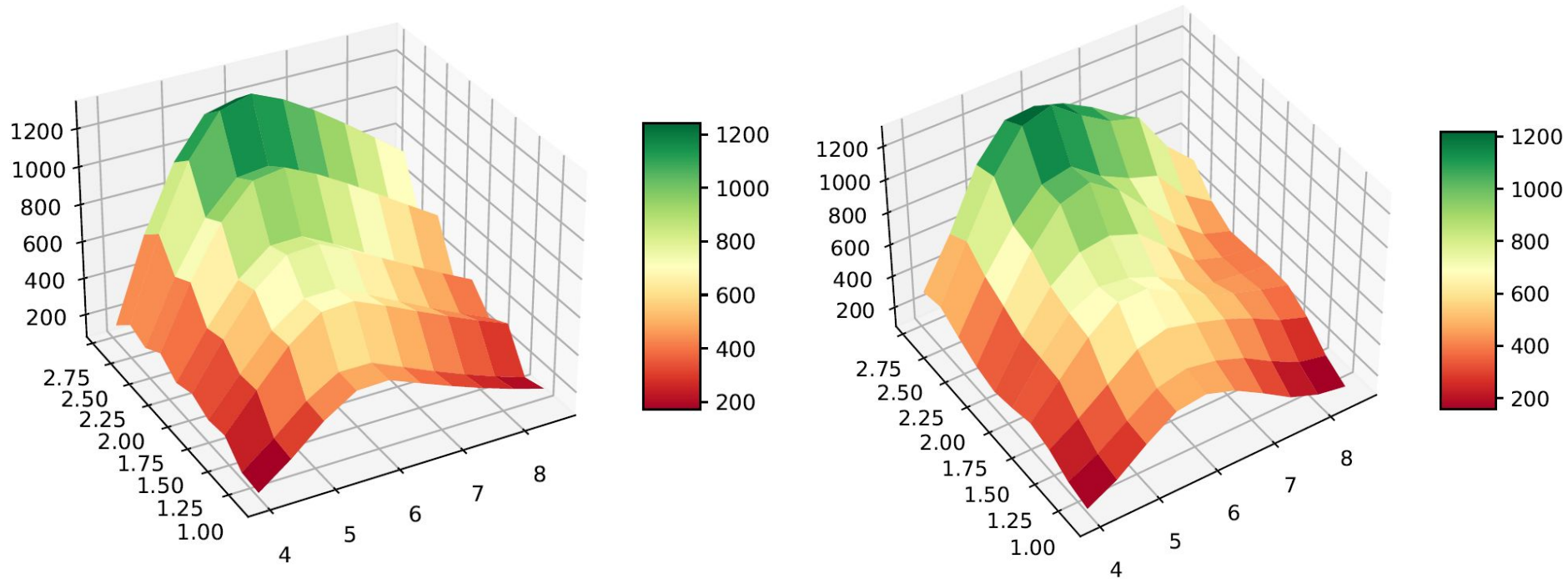
# Step 6: risultati

15



# Step 6: valore atteso e appreso

16



# Context generation (Step 4 & Step 7)

17

Per poter effettuare lo splitting 9 learner operano in parallelo per i diversi contesti, questo permette di valutare, in fase di splitting, il valore atteso prima e dopo la divisione

Per poter generare delle ricompense autonome sono stati necessari 4 environments, ciascuno dei quali produce ricompense per gruppi di learners tra loro indipendenti

# Context generation (Step 4 & Step 7)

18

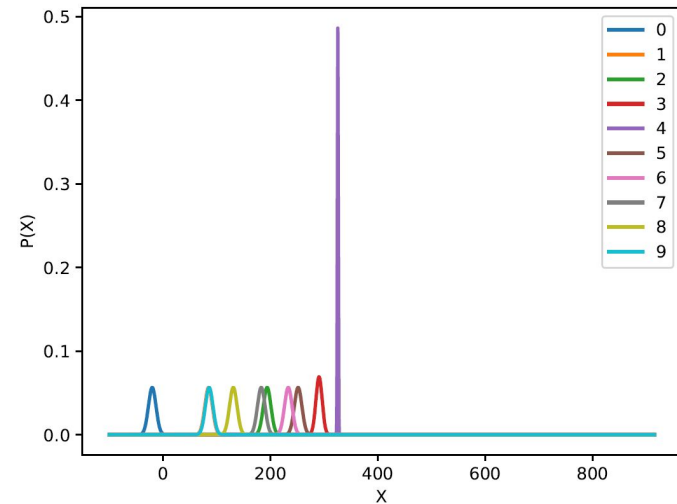
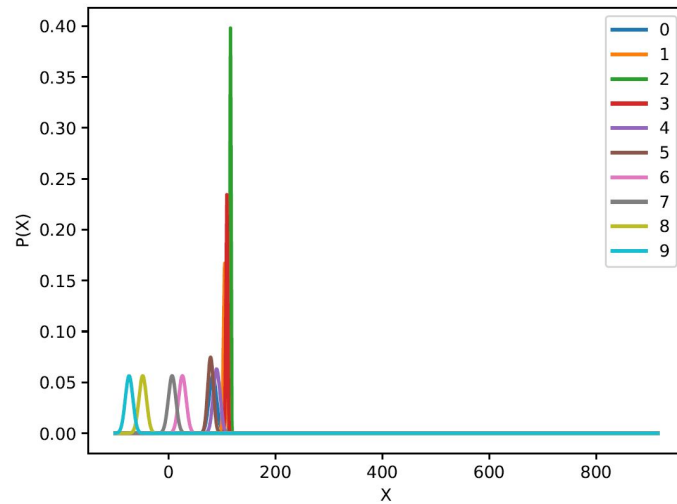
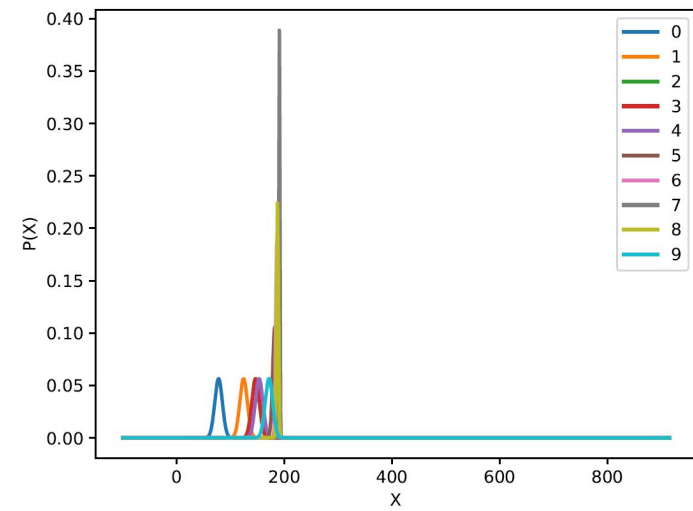
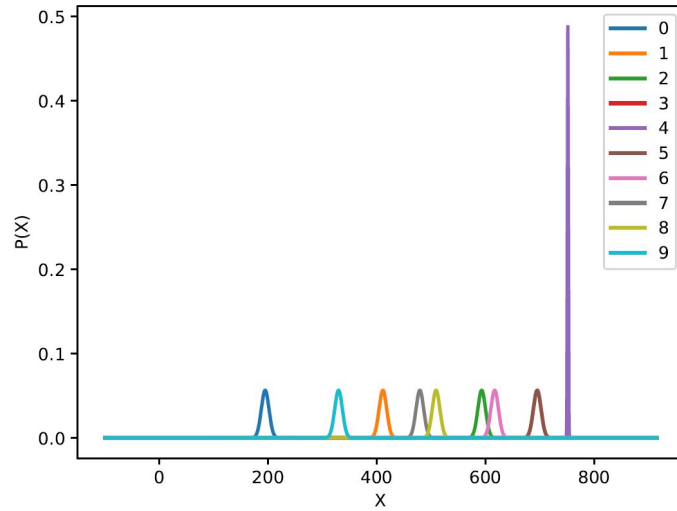
Come lower bound in entrambi i casi si è scelto di usare  $(\mu - \sigma)$  di ciascun braccio

La condizione per lo splitting è la seguente:  $\underline{R}_1 + \underline{R}_2 > \underline{R}_0$

Dove  $\underline{R}$  è il lower bound del miglior braccio per ciascuno dei 3 learner.

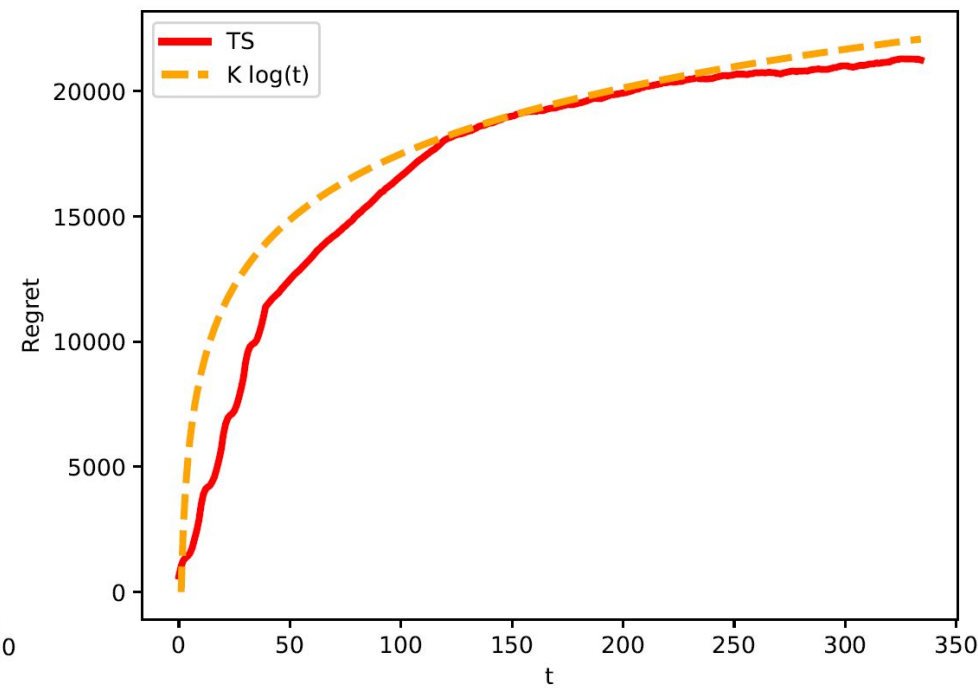
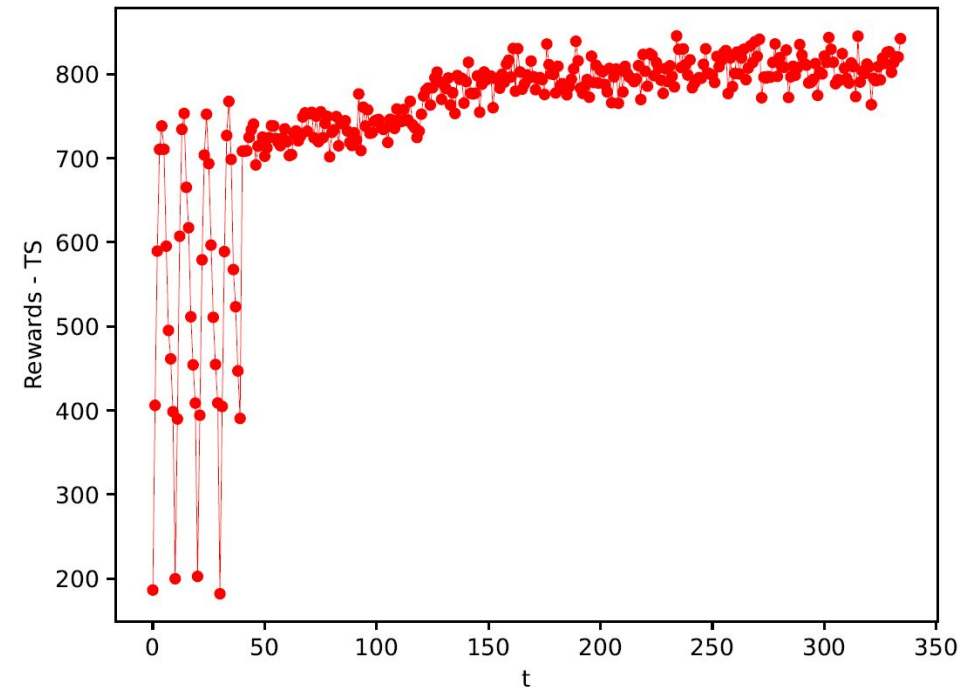
# Step 4: risultati

19



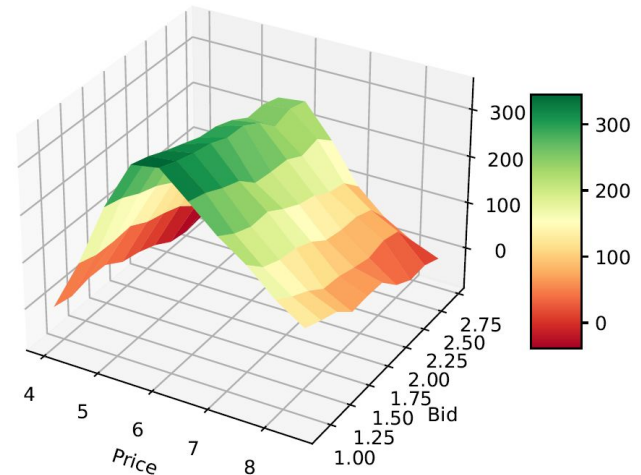
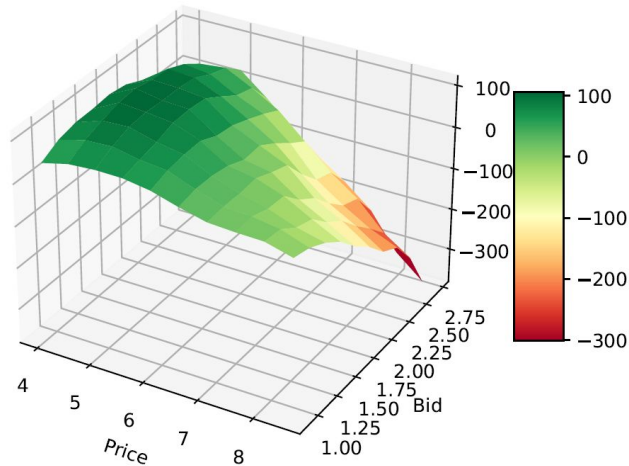
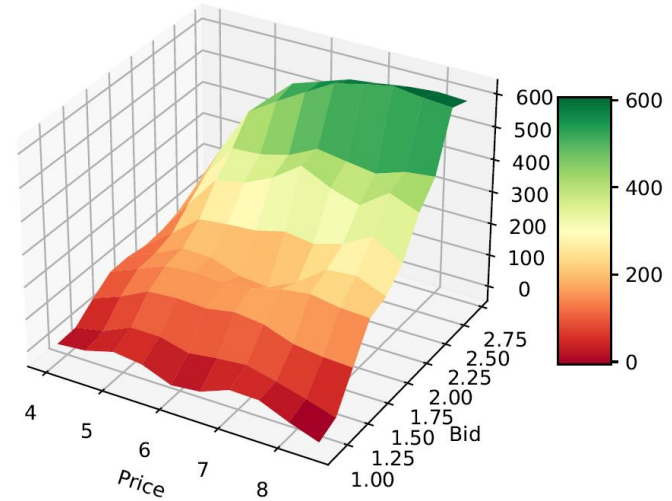
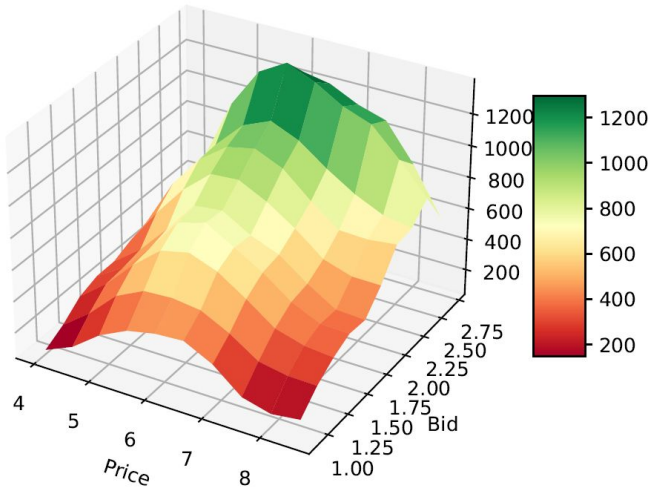
# Step 4: risultati

20



# Step 7: risultati

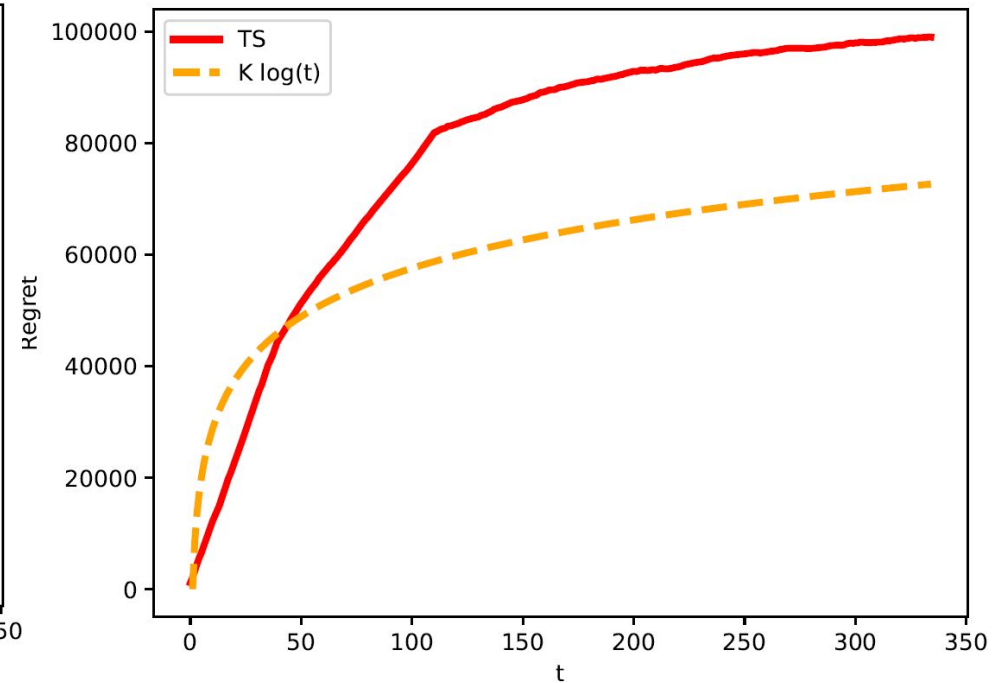
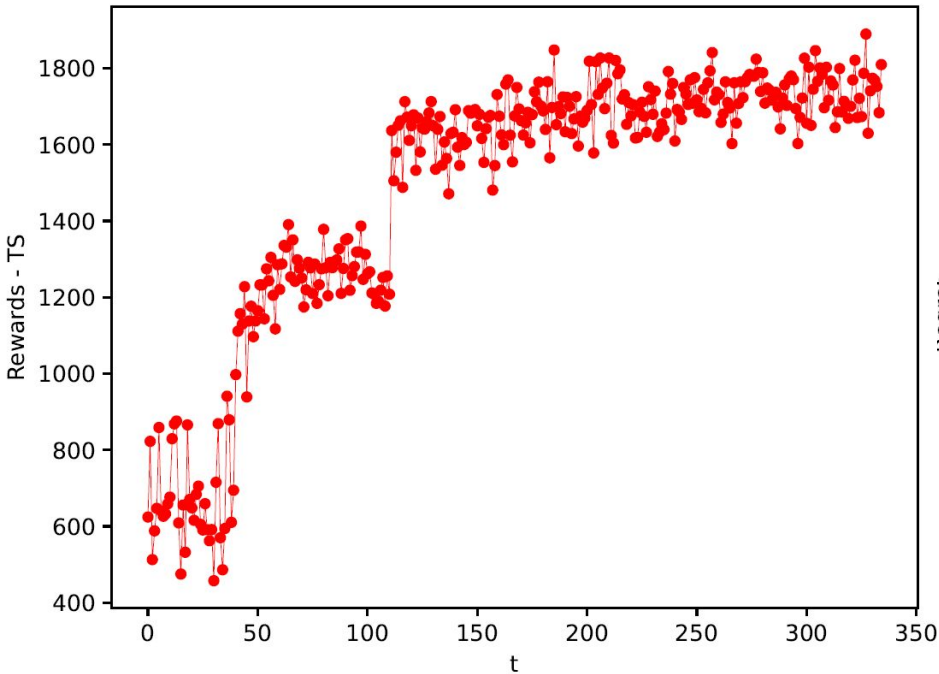
21





# Step 7: risultati

22



- tempo di inizio splitting scelto con un trade-off correttezza vs regret
- La scelta di un contesto verosimile, porta a situazioni in cui si hanno più bracci con reward simili, aumentando la varianza

