USER OWNED

Lista utilities = [0.5;0.7…]

Lista product IDs = [1;2…]

GLOBALE

Lista utenti = [x;y;z]

Lista ID di best product = [2;4;2]

Lista ID di best companies = [1;3;2]

set users-list []  
set best-products-list []  
set best-companies-list []

net revenue list

utility function management

1. Ask products di normalizzare le variabili usate per calcolo di utilità
2. Finito l’ask products, apriamo ask users
   1. Creiamo lista utilities e p-ids
   2. Chiediamo ai prodotti di calcolare la propria utilità e inserire il valore finale nelle liste
   3. L’utente deve scegliere il miglior prodotto e inserire i dati nelle liste
   4. I prodotti aggiornano la lista net revenues
   5. Caro utente, scegli se comprare qualcosa🡪

* Se si:incrementiamo lo stock; chiediamo alla company di incrementare le proprie earnings

La funzione di acquisto era inizialmente così:

if ((i-stock / i-stock-threshold ) <= (utility-of-best-product )\*trigger)

con stock-threshold= 1

con trigger= 1

Ragionandoci siamo giunte alla conclusione che:

* scritta così stiamo confrontando 2 quantità non confrontabili
* ragionando su excel con varie casistiche, risultava che gli utenti a volte comprassero nei momenti sbagliati e non si riesce a trovare un valore di trigger tale da far funzionare l’equazione

A screenshot of a graph

Description automatically generated

Successivamente abbiamo aggiunto il budget (in quanto è realistico che un utente non abbia capacità infinita di acquisto) e un adjustment factor

* if (i-price <= tot-budget) [
* let adjustment-factor 0  
   if (i-stock / i-stock-threshold) <= 1 [  
   ;es 0.3  
   ;adj sarebbe 1.7  
   ; se fosse 1+ stock/trhs --> adj = 1.3  
   set adjustment-factor ( 1 + (1 - i-stock / i-stock-threshold) ) ; Amplifica proporzionalmente a quanto manca per raggiungere il threshold  
   ]
* set p-utility p-init-utility \* ( adjustment-factor )
* l’adjustment factor considerate inialmente era (1-stock-i/stock-threshold-i)
* anche questa formula è sbagliata in quanto porta gli utenti ad acquistare sempre

infine abbiamo unito le 2 modifiche e siamo giunte a

if ((1 - i-stock / i-stock-threshold ) \* trigger <= (utility-of-best-product )) and (i-price <= tot-budget) [

che però continua ad avere dei problemi

)

Poi anche qualità influenzerà anche il costo di produzione🡪 più il prodotto è sostenibile, più il costo di produzione si avvicinerà a baseline\*(1+delta)

Se sust=1 allora costo prod aumenta del 10%

Price-min= costo prod-baseline

Exp = esponente compreso tra 0 e 1

Iniziamo a settare exp = 0.5 e poi vediamo

Di conseguenza mettiamo vincolo su che non può essere <0

C-memory = sottolista 1 (c1-1 c2-1 c3-1..) ;; sottolista 2 (c1-2 c2-2;; sottolista 3…..

Weight associato alla sottolista

While k < c-memory -length (k<10)

(((

Target annuale: 200€

c-revenues= 10+15+15+10+10+10+10+10+10+10+10+10 +16

vendite mese 1: 10€

vendite mese 2: 15€

vendite mese 3: 15

vita reale:

lista di 12 elementi= 1 2 3 4 5 6 7 8 9 10 11 12

counter: tick = se length list = length desiderata (12 o simili) 🡪reset list

Esselunga: revenue media per prodotto: 10€

Eurospin: revenue media per prodotto: 5€

Conad: revenue media per prodotto: 7€

Carrefour: revenue media per prodotto: 9€

Alla fine dei sei 6 mesi: devo fare la rev-media per prodotto degli ultimi 6 mesi

C-revenue-list (company1) =( revenue gg1; revenue gg2; ….revenue gg180)

Revenue media per prodotto di oggi= revenue di oggi / numero di prodotti venduti oggi

METODO 1

1. Lista con c-revenue di ogni gg (c-revenue-list)
   1. Veniva creata variabile dummy c-revenue (= revenues di oggi per company 1)
   2. La variabile dummy c-revenue veniva aggiornata durante il gg man mano che gli user comprano
   3. a fine giornata (a fine tick) 🡪 la revenue veniva prima aggiunta alla lista c-revenue-list, e poi azzerata (pronta per il tick successivo
2. Lista con numero di prodotti venduti ogni gg (n° prod venditi gg1; n° prod venditi gg2;… n° prod venditi gg180)
3. Creami una lista con item 1 = item 1 di Lista con c-revenue / item 1 di Lista con numero di prodotti venduti

METODO 2

1. Ad ogni tick compilo la c-revenues-list, aggiungendo un elemento per volta: =( revenue gg1; revenue gg2; ….revenue gg180)
   1. Veniva creata variabile dummy c-revenue (= revenues di oggi per company 1)
   2. La variabile dummy c-revenue veniva aggiornata durante il gg man mano che gli user comprano
   3. a fine giornata (a fine tick) 🡪 la revenue veniva prima aggiunta alla lista c-revenue-list, e poi azzerata (pronta per il tick successivo
2. Ad ogni tick uso una variabile dummy - period-revenues-per-unit= c-revenues di oggi / numero prodotti acquistati di oggi (questa variabile dummy viene riazzerata a fine tick perche? 🡪 )
3. period-revenues-per-unit-list = viene aggiornata ad ogni tick aggiungendo un elemento

period-revenues-per-unit= (rev-media-per-prod gg1; rev-media-per-prod gg2; ….; rev-media-per-prod gg180)

Revenue per product min= 5

Revenue per product max = 10



Target revenues assolute = min + random (max-min ) \* fattore smorzamento ??

Fattore smorzamento= Mie revenue / revenues max

Revenue target = min+ random (max-min)

* 5
  + Calcolare il min revenue per unit

Pr(target =6) 20%

Pr(target =7) 20%

* qui c’è conad

Pr(target =8) 20%

Pr(target =9) 20%

* qui c’è carrefour

Pr(target =10) 20%

Revenue unitaria

Revenues list= (10 20 15 10 13 …fino a 12 elementi)

Rev per unit list= (2.5 3 6 1.5 …fino a 12 elementi)

A noi serve Revenue per product min e Revenue per product max 🡪 derivano da revenue media per product (media calcolata sui 6 mesi)

COMPANY 1 🡪 Revenue per product list= (rev per unit gg1, gg2,…gg180) 🡪 media di questi 180 valori = rev media per prodotto degli ultimi 6 mesi

COMPANY 2 🡪 Revenue per product list= (rev per unit gg1, gg2,…gg180) 🡪 media di questi 180 valori = rev media per prodotto degli ultimi 6 mesi

COMPANY 3 🡪 Revenue per product list= (rev per unit gg1, gg2,…gg180) 🡪 media di questi 180 valori = rev media per prodotto degli ultimi 6 mesi

* prima dobbiamo avere questi risultati per tutte le companies

Company1🡪 media di questi 180 valori

Company2🡪 media di questi 180 valori

Company3🡪 media di questi 180 valori

Calcoliamo min e max

NUOVA IDEA

mi confronto con la migliore e punto alla compagnia migliore

Poi questo mio obiettivo viene smorzato in base alle mie capacità= mio posizionamento rispetto ai competitors

Volevamo p-utility compresa tra 0 e 1

Se p-utility fosse stata = alpha\*Q + beta\*S + gamma\*P

🡪 data alpha + beta + gamma= 1, è scontato che p-utility è compreso tra 0 e 1

(caso p-utility = 1 si ha quando sia Q che S che P =1)

Tuttavia noi abbiamo

* p-utility= alpha\*Q + beta\*S - gamma\*P
* poi, in media, I valori di Q,S e P (norm) non sono =1
* in media i valori di p-utility erano < 0.5

TUTTAVIA

La nostra funzione di utilità e di scelta di acquisto: ha senso a livello di comportamento logico con valori medi di utility > 0.5

* per far funzionare l’equazione, dobbiamo alzare il valore medio di utilità
* di conseguenza eleviamo l’utilità a 0.5