

Sviluppo di un Algoritmo di Trading utilizzando il Q-learning per Massimizzare i Profitti

Progetto di reinforcement learning

Obiettivi del progetto

<u>Raccogliere i dati</u>	Raccogliere i dati storici di un asset finanziario
<u>Progettazione dell'agente</u>	Implementare un agente di reinforcement learning per il trading azionario
<u>Ambiente</u>	Utilizzo di una repository da GitHub per lo sviluppo dell'ambiente
<u>Addestramento dell'agente</u>	Addestrare il modello su dati di training
<u>Valutazione delle performance</u>	Valutare le performance ottenute su dati di test attraverso il ROI (Return of investment)
<u>Sfide affrontate</u>	Diverse problematiche affrontate durante lo sviluppo del progetto

Table of contents

01

Raccogliere i dati

02

Progettazione dell'agente

03

Ambiente e ricompense

04

Addestramento dell'agente

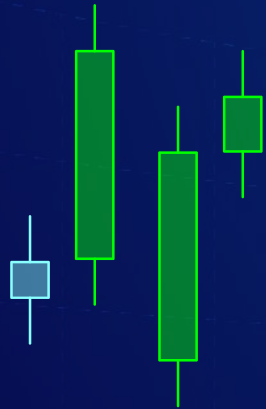
05

Valutazione delle performance

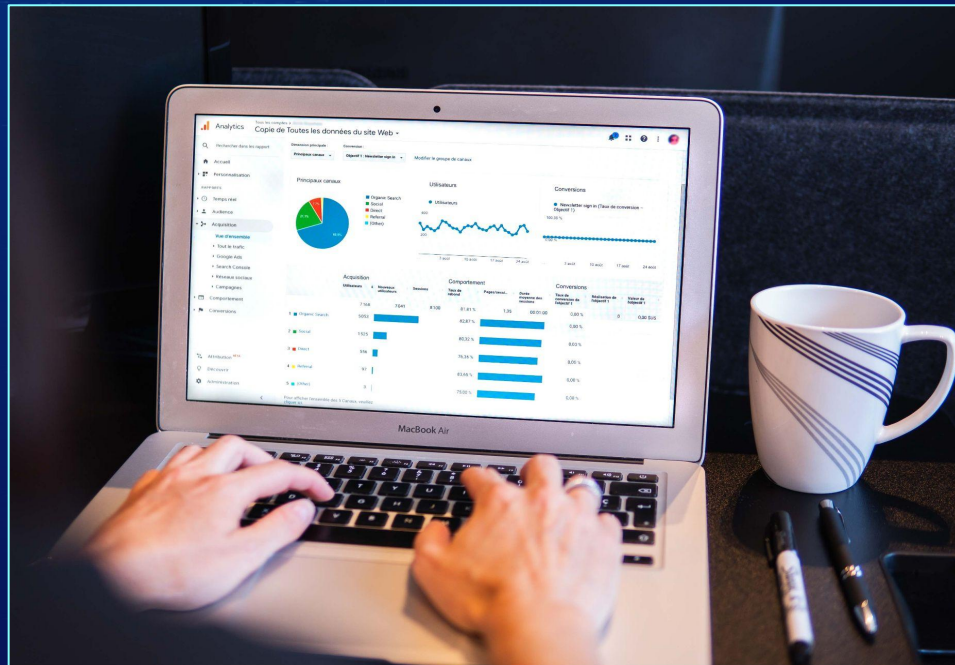
06

Sfide affrontate





01



Raccogliere i dati

Scelta dell'asset

S&P500

S&P500 è un indice azionario che rappresenta le performance di 500 delle più grandi aziende quotate negli Stati Uniti. Utilizzato come indicatore chiave del mercato azionario, fornisce una panoramica della salute generale dell'economia statunitense.



Dataset S&P 500

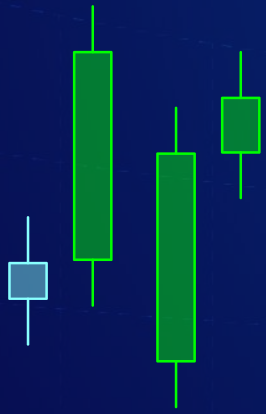
Date	Open	High	Low	Close	Adj Close	Volume
1973-02-19	0.000000	116.260002	112.769997	113.160004	113.160004	58920000
1973-02-26	0.000000	113.260002	109.449997	112.279999	112.279999	85860000
1973-03-05	0.000000	115.230003	111.330002	113.790001	113.790001	79910000
1973-03-12	0.000000	115.610001	112.839996	113.540001	113.540001	72060000
1973-03-19	0.000000	113.500000	107.410004	108.879997	108.879997	77390000
1973-03-26	0.000000	113.220001	108.290001	111.519997	111.519997	78120000
1973-04-02	0.000000	111.699997	107.440002	109.279999	109.279999	62080000
1973-04-09	0.000000	113.650002	108.739998	112.080002	112.080002	76150000
1973-04-16	0.000000	112.930000	109.989998	112.169998	112.169998	52630000
1973-04-23	0.000000	112.660004	106.760002	107.230003	107.230003	72310000

Dal 19 Febbraio del 1973 al 15 Dicembre 2023 (50 anni)

Osservazioni settimanali

Dataset train: 1856 (70%)

Dataset test: 796 (30%)



02



Progettazione dell'agente

Metodi dell'agente



`--init--`

Vengono inizializzati gli iperparametri del modello, ossia:

- learning rate;
- initial epsilon;
- epsilon decay;
- final epsilon
- discount factor;



`get_action`

Restituisce la migliore azione, calcolata con il Q-value, con probabilità $1 - \epsilon$.



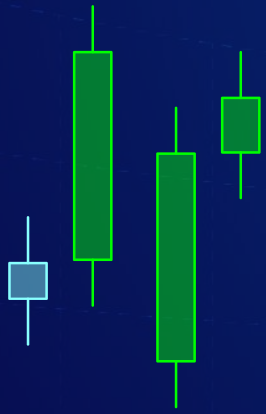
`update`

Calcolare il temporal difference e dunque aggiorna il q-value per scegliere l'azione migliore.



`decay epsilon`

Diminuisce il valore epsilon alla fine di un episodio per ridurre l'esplorazione nel tempo.

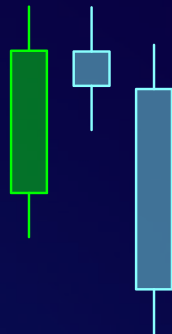


03

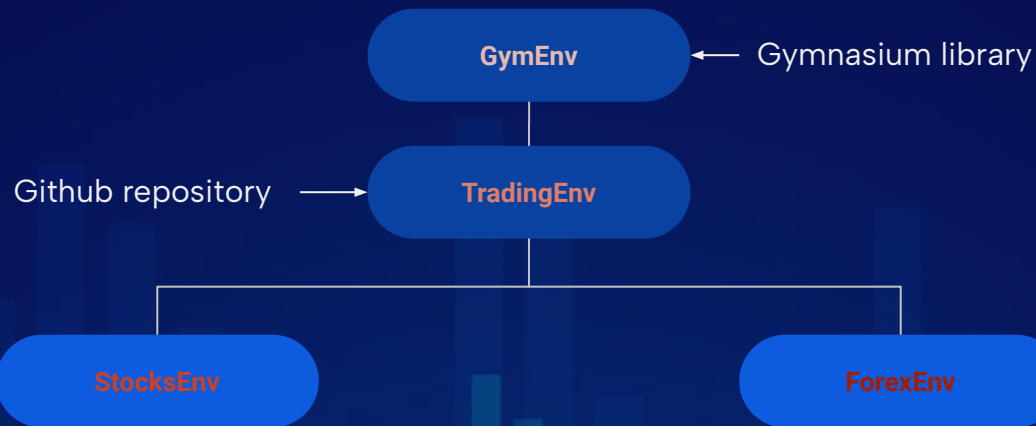


Ambiente e Ricompense

Ambiente



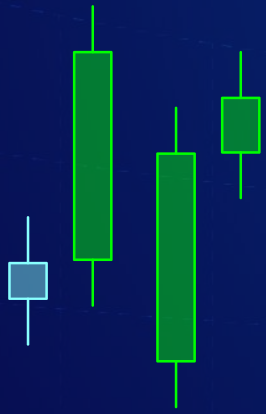
OpenAI Gym environment per fare trading con algoritmi di reinforcement learning.



TradingEnv classe astratta che eredita da `gym.Env`, imposta l'ambiente generale, definisce il prezzo la grandezza dell'ambiente, lo spazio delle azioni e delle osservazioni.

StocksEnv eredita da **TradingEnv** e implementa i metodi astratti, come il calcolo delle ricompense, l'aggiornamento dei profitti ecc

Repository GitHub: <https://github.com/AminHP/gym-anytrading/tree/master>

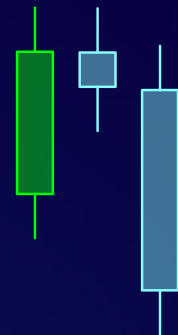


04



Addestramento dell'agente

Parametri



Wallet

Il totale dei soldi posseduti dall'agente. Il valore scelto è di 100.000 mila.

Budget

Il budget è la percentuale da investire rispetto al wallet, il valore è il 2% del wallet. Dunque è dinamico.

Stop loss

La cifra massima che può perdere prima di chiudere il trade. Il valore è pari al 2% del budget investito.

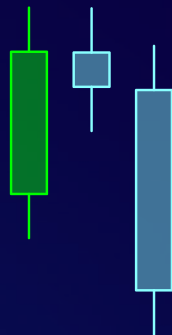
Tassa ingresso

Indica la percentuale che va al broker e che dunque deve essere sottratta al budget al momento dell'inizio di un trade. Il valore è del 0.1%

Tassa uscita

Indica la tassa percentuale che viene sottratta al budget al termine del trade. Il valore è del 0.05%.

Iper-parametri



discount_factor

Varia tra 0 e 1, indica quanto peso dare alle ricompense future

learning_rate

Definisce la grandezza del passo di aggiornamento durante la ricerca dell'ottimo

start_epsilon

Il valore epsilon iniziale determina il livello di esplorazione dell'agente

epsilon_decay

Determina quanto velocemente deve decrescere epsilon alla fine di ogni episodio

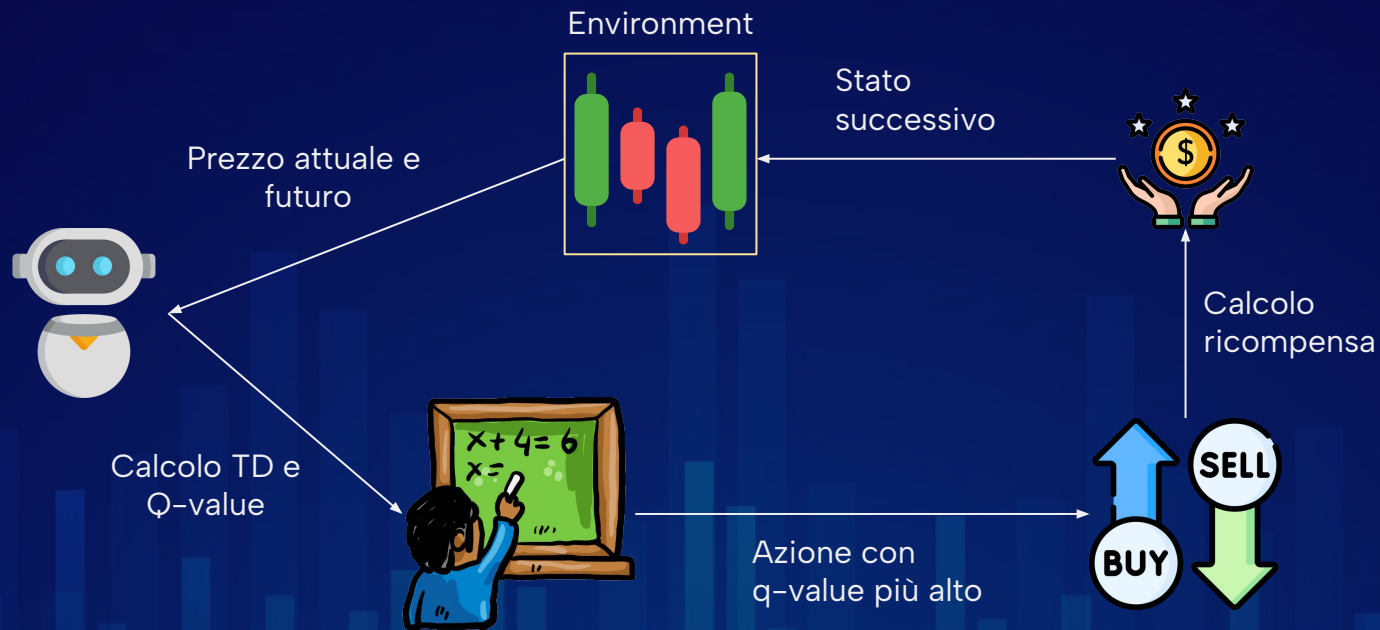
final_epsilon

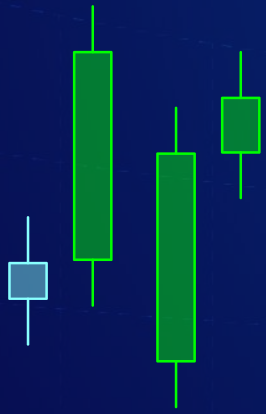
Determina la probabilità minima di esplorazione dell'agente

numero di episodi

Il numero di episodi si riferisce al numero di iterazioni complete attraverso il set di dati durante il processo di apprendimento

Addestramento





05



Valutazione performance

Dati e grafici

Presentiamo dati e grafici che mostrano i profitti generati dall'algorithm.

Grafico training set

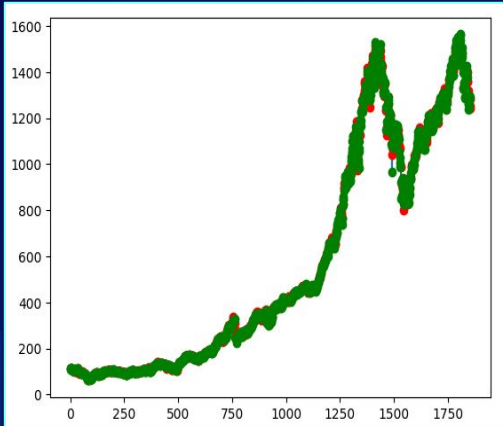
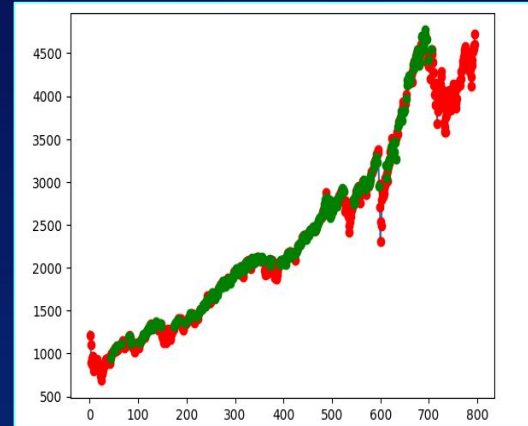


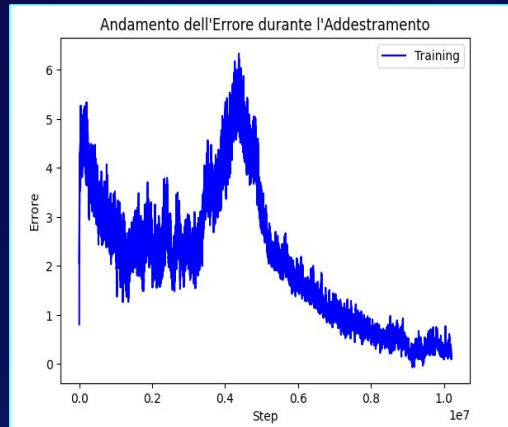
Grafico test set



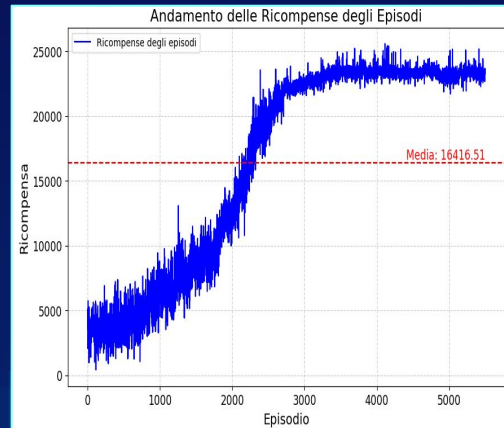
Dati e grafici

Presentiamo i grafici che mostrano l'error rate, (TD Error), l'andamento delle rewards e l'andamento del ROI al termine dei **5500** episodi.

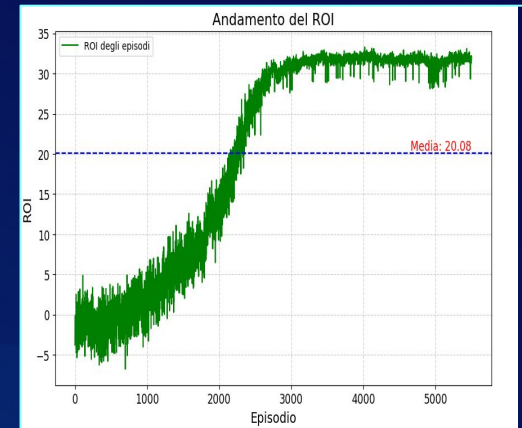
Error training set



Rewards test set



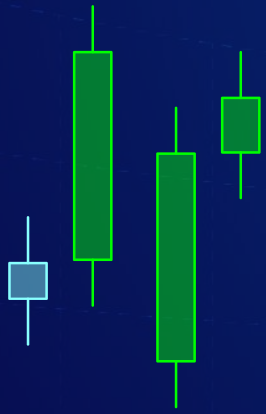
ROI test set



Risultati finali

	Training
ROI	31.80
Rewards	16416.51
Discount factor	0.98
Numero episodi	5500
Learning rate	0.1
Final epsilon	0.01

	Test
ROI	2.83
Wallet	102830.62\$



06



Sfide affrontate

Sfide affrontate

Adattamento Agente

1

Scelta del dataset

2

Problemi con il test set

3

4

ROI altissimo

7

Implementazione stop
loss

6

Nuova funzione
ricompensa

5

Ricompense ambiente
sbagliate



Grazie per l'attenzione!

Progetto di Giuseppe Genito e Rosaria Leone

