

Università
Ca'Foscari
Venezia

Corso di Laurea Magistrale
in
Economia e Finanza

Tesi di Laurea

Algoritmo di PPO applicato alla gestione di portafoglio

Relatore
Ch.mo Prof. Marco Corazza

Laureando
Davide Di Benedetto
Matricola 889025

Anno Accademico
2025 / 2026

INDICE

INDICE 1
INTRODUZIONE 5
CAPITOLO 1 - QUADRO TEORICO 8
1.1. Costruzione di Portafoglio 8
1.2. Efficient Market Hypothesis 9
1.3. Il paradosso dell'efficienza informativa 12
1.4. I limiti dell'arbitraggio 13
1.5. Dall'Efficient Market Hypothesis all'Adaptive Markets Hypothesis 14
1.6. Implicazioni per la costruzione di portafoglio 18
CAPITOLO 2 - TECNICHE DI MACHINE LEARNING IN CONTESTI ECONOMICI E FINANZIARI 20
2.1. Reinforcement Learning: 21
2.2. Markov Decision Process (MDP) 23
2.3. Funzione di ricompensa e ritorno 24
2.4. Funzione di valore e politiche 26
2.5. Valore ottimale 28
CAPITOLO 3 - ALGORITMI DI REINFORCEMENT LEARNING 31
3.1. Modalità di apprendimento 31
3.2. Dynamic Programming 32
3.3. Monte Carlo Methods 34
3.4. Temporal Difference Methods 37
3.5. Value-Based Methods 39
3.6. Policy-Based Methods 41
3.7. Metodi Actor-Critic 47
CAPITOLO 4 - PROXIMAL POLICY OPTIMIZATION 51
4.1. Instabilità degli aggiornamenti e TPRO 51
4.2. Introduzione tecnica alla PPO 53
4.3. La funzione obbiettivo completa nella PPO 54
4.4. Ciclo di apprendimento della PPO 55
CAPITOLO 5 - APPLICAZIONE EMPIRICA 56
5.1. Obiettivo del modello baseline PPO-MLP 56

5.2. Dataset e universo investibile 57
5.3. Costruzione dello stato 60
5.4. Struttura dell'ambiente e azioni 62
5.5. Funzione di ricompensa 64
5.6. Algoritmo PPO e rete MLP 68
5.7. Benchmark di confronto 73
5.8. Risultati in validation 74
5.9. Risultati in test 76
5.10. Valutazione del modello 79
CAPITOLO 6 - INTEGRAZIONE DEL SENTIMENT FINANZIARIO NEL MODELLO MEDIANTE FINBERT 82
6.1. Introduzione al modello 82
6.2. Natural Language Processing 84
6.3. Dataset di notizie finanziarie utilizzato 87
6.4. Implementazione di FinBERT e metodologie 88
6.5. Integrazione del sentiment nello stato del modello PPO 90
6.6. Risultati in validation 92
6.7. Risultati in test 94
6.8. Valutazione del modello 97
CONCLUSIONI: 99
Bibliografia consultata 103
Appendice 105

Dichiarazione del ricorso a strumenti di Intelligenza Artificiale (IA)

Il sottoscritto Davide Di Benedetto, matricola numero 889025, iscritto presso Ca' Foscari al corso di Laurea magistrale in Economia e Finanza, autore della tesi di laurea Algoritmo di PPO applicato alla gestione di portafoglio, relatore Marco Corazza, consapevole delle sanzioni previste dall'art. 76 del Testo unico, D.P.R. 28/12/2000 n.445, e della decadenza dei benefici prevista dall'art. 75 del medesimo Testo unico in caso di dichiarazioni false o mendaci, sotto la propria personale responsabilità ai sensi degli artt. 46 e 47 del D.P.R. 28/12/2000 n. 445,

DICHIARA

☐ di NON aver fatto ricorso a tecnologie di intelligenza artificiale generativa per supportare vari aspetti della ricerca e della scrittura del suo elaborato finale/della sua tesi di laurea;

X di aver fatto ricorso a tecnologie di intelligenza artificiale generativa per supportare vari aspetti della ricerca e della scrittura della mia tesi di laurea, in particolare:

- indicare il nome dello strumento con riferimento alla versione specifica: Chat gpt versione 5.6
- se sono stati utilizzati più strumenti, specificare ciascuno di essi indicandone nome e versione;
- fornire informazioni su come è stato utilizzato lo strumento:
  - x revisione e ottimizzazione dei testi per migliorare chiarezza e coerenza;
  - o generazione di grafici e diagrammi per la presentazione dei dati;
  - o supporto alla scrittura automatica di frasi;
  - o produzione di immagini illustrative per integrare la parte metodologica;
  - x supporto nello sviluppo, revisione e debug del codice Python
- indicare le sezioni/i capitoli interessate/interessati: capitoli 5 e 6
- indicare come è stata verificata la correttezza, la qualità e l'attendibilità delle informazioni o dei testi eventualmente generati: attraverso la documentazione ufficiale delle librerie utilizzate

DICHIARA INOLTRE

che tutto il materiale generato dall'IA è stato attentamente verificato, rivisto e rielaborato dall'autore, che si assume la piena responsabilità per la qualità, l'accuratezza e l'integrità scientifica del lavoro.

INTRODUZIONE

La costruzione e la gestione di portafoglio rappresentano uno dei temi centrali e di interesse comune della finanza moderna, per i quali, a partire dal contributo fondamentale di Markowitz, la letteratura finanziaria ha sempre cercato di individuare criteri adatti a combinare attività caratterizzate da differenti profili di rischio e rendimento, con l'obbiettivo comune di ottenere un'allocazione il più efficiente possibile del capitale. Tuttavia, l'applicazione di questi principi all'interno dei mercati reali ha sempre dovuto scontrarsi con un elemento di particolare complessità, ossia le condizioni instabili nelle quali gli investitori operano nel tempo. Rendimenti attesi, volatilità, correlazioni e disponibilità informativa tendono infatti a variare significativamente al variare del tempo e del contesto economico-finanziario, rendendo particolarmente complessa l'individuazione di una strategia di investimento universale che mantenga la propria efficacia nel lungo periodo.

Questa considerazione si inserisce poi all'interno di un più ampio dibattito relativo all'efficienza dei mercati finanziari: a partire dalla tradizionale teoria dei mercati efficienti, formulata da Fama, la quale sostiene che i prezzi riflettano direttamente l'informazione disponibile, la letteratura successiva ha evidenziato come la presenza di costi informativi, limiti all'arbitraggio, comportamenti non pienamente razionali e differenze nella capacità degli investitori di acquisire ed elaborare le informazioni disponibili, renda più complesso il processo attraverso il quale queste vengano incorporate nei prezzi.

In tale prospettiva assumono particolare rilevanza le Adaptive Markets Hypothesis proposte da Andrew Lo, il quale interpreta i mercati finanziari come sistemi dinamici e competitivi nei quali gli operatori modificano progressivamente le proprie strategie in risposta alle variazioni dell'ambiente circostante. L'efficienza viene quindi traslata da una condizione assoluta ed immutabile al risultato di un processo evolutivo, nel quale opportunità di profitto, strategie e relazioni tra gli strumenti finanziari possono emergere, modificarsi e successivamente scomparire. Proprio da questa impostazione ne deriva una conseguenza fondamentale per la costruzione di portafoglio: se l'ambiente di mercato è

dinamico e non stazionario, allora anche le strategie di allocazione dovrebbero essere in grado di adattarsi al variare delle condizioni osservate.

Lo sviluppo delle tecniche di machine learning e in particolare del reinforcement learning offre strumenti potenzialmente adatti ad affrontare queste tipologie di problemi. In particolare, a differenza degli approcci tradizionali, il reinforcement learning considera la presenza di un agente, chiamato a prendere una serie di decisioni attraverso l'interazione con l'ambiente che lo circonda, modificando progressivamente il proprio comportamento sulla base delle conseguenze delle azioni intraprese in precedenza. Il problema della gestione di portafoglio può quindi essere rappresentato come un processo decisionale sequenziale nel quale, ad ogni istante temporale, l'agente osserva le condizioni di mercato, sceglie una nuova allocazione del capitale e riceve successivamente un segnale attraverso il quale valutare la qualità della scelta effettuata.

Il presente lavoro si concentra in particolare sull'algoritmo di Proximal Policy Optimization (PPO), appartenente alla famiglia dei metodi actor-critic. La scelta di un algoritmo PPO deriva dalla capacità di questo di effettuare aggiornamenti progressivi della propria politica, limitando variazioni eccessive ampie tra una configurazione e la successiva, caratteristica particolarmente rilevante in ambito finanziario dove l'elevato livello di rumore presente nei dati potrebbe determinare aggiornamenti instabili e strategie eccessivamente sensibili alle singole osservazioni.

L'obbiettivo principale dell'elaborato consiste pertanto nel verificare empiricamente se un agente basato sulla PPO possa apprendere una politica di allocazione dinamica capace di produrre risultati economicamente significativi e competitivi rispetto ad altre strategie di investimento più semplici e tradizionali. In una prima fase iniziale viene sviluppato un modello base, basato su una rete neurale Multi-Layer Perceptron e applicato ad un universo investibile composto da dodici ETF rappresentativi di diversi mercati, asset class e fattori di rischio. Il modello osserva dunque un insieme di variabili quantitative relative a: prezzi, volatilità, momentum, volumi, drawdown, correlazioni e composizione corrente di portafoglio, determinando conseguentemente la nuova composizione del portafoglio.

Successivamente l'analisi viene estesa introducendo una seconda fonte informativa, ossia le aspettative degli investitori estratte dal flusso di notizie finanziarie pubbliche. Per questa ragione viene integrato FinBERT, un modello linguistico specializzato nell'analisi del sentiment delle notizie a carattere finanziario, utilizzato per trasformare ampi set di

notizie in variabili quantitative che possono essere inserite nello stato osservato dall'agente. In questo modo è possibile verificare se l'ampliamento delle informazioni disponibili all'agente possa migliorare le decisioni di allocazione, mantenendo sostanzialmente invariata la struttura del modello PPO di base.

L'elaborato persegue dunque due interrogativi strettamente collegati. Il primo riguarda la possibilità di utilizzare il reinforcement learing, in particolare la PPO, come strumento per la gestione dinamica di un portafoglio. Il secondo riguarda la valutazione del beneficio apportato dall'integrazione del sentiment derivato dalle notizie finanziarie, al fine di costruire una rappresentazione più completa dello stato del mercato, misurando la migliore o peggiore performance rispetto ad una configurazione basata esclusivamente su informazioni quantitative.

Il lavoro è strutturato secondo una progressione che prosegue dalla teoria all'applicazione empirica. Inizialmente viene definito il quadro finanziario di riferimento e viene discussa l'evoluzione dal paradigma dei mercati efficienti a quello dei mercati adattivi, successivamente vengono presentati i principi fondamentali del reinforcement learning e le principali famiglie di algoritmi sviluppate dalla letteratura, terminando l'analisi con la descrizione della PPO e delle caratteristiche che ne giustificano l'utilizzo nel problema considerato. Nella parte empirica viene quindi sviluppato e valutato il modello baseline PPO-MLP ed infine, viene analizzata e valutata l'estensione mediante l'integrazione del sentiment finanziario attraverso l'applicativo FinBERT.

CAPITOLO 1 - QUADRO TEORICO

1.1. Costruzione di Portafoglio

Uno dei principali temi trattati dalla finanza moderna è rappresentato dalla costruzione di portafoglio, la quale gioca un ruolo fondamentale nell'ambito delle decisioni di investimento. L'obbiettivo principale di tale processo consiste nella massimizzazione del rendimento atteso degli investimenti, minimizzando al contempo il livello di rischio assunto. In questa prospettiva, all'investitore è richiesto il compito di bilanciare opportunamente rischi e rendimenti, due elementi strettamente connessi che assieme costituiscono il fondamento di tutte le principali teorie di gestione degli investimenti.

In questo settore, il contributo teorico più rilevante è rappresentato dalla pubblicazione nel 1952 della "Modern Portfolio Theory"[1] (MPT), elaborata da Harry Markowitz, che ha segnato l'inizio di una nuova era nella finanza quantitativa.

L'intuizione principale introdotta da Markowitz risiede nello spostare l'analisi dal singolo titolo alla valutazione del portafoglio nel suo complesso. Markowitz dimostra che il valore di un investimento non può essere analizzato isolatamente, bensì deve essere considerato in relazione agli altri asset detenuti, poiché la correlazione presente tra i diversi asset può influenzare i livelli di rischio-rendimento del portafoglio nel suo complesso.

L'ipotesi sulla quale si basa il modello è l'avversione al rischio degli investitori, ossia che questi preferiscano, a parità di rendimento atteso, l'investimento con volatilità minore. In questo modello, la covarianza tra gli asset assume un ruolo cruciale, poiché il rischio di un portafoglio non è semplicemente la media ponderata dei rischi dei singoli asset che lo compongono, ma dipende anche dal loro grado di correlazione.

Nel caso di un portafoglio composto da due asset, il rendimento atteso è dato da:

$$E(R_{PF}) = w_1 E(R_1) + w_2 E(R_2)$$

Mentre la varianza del portafoglio è:

$$\sigma_{PF}^2 = w_1^2 \sigma_1^2 + w_2^2 \sigma_2^2 + 2 w_1 w_2 \sigma_{1,2}$$

[1] Markowitz, H. (1952). Modern Portfolio Theory.

Ed è proprio il termine di covarianza che dimostra che la combinazione di attività non perfettamente correlate consente di ridurre la volatilità complessiva in modo da ottenere, a parità di rendimento atteso, un rischio inferiore. Dunque, maggiore è la diversificazione tanto più significativo sarà l'effetto di riduzione del rischio.

Il modello di Markowitz si traduce in un problema di ottimizzazione vincolata, nel quale l'investitore può alternativamente minimizzare la varianza del portafoglio, dato un certo livello di rendimento atteso, oppure massimizzare il rendimento atteso, dato un livello prefissato di rischio. La soluzione del problema fornisce dunque un vettore di pesi che definisce la composizione del portafoglio ottimale.

Nonostante la sua enorme rilevanza teorica, la MPT presenta alcune limitazioni oggettive: oltre all'ipotesi di avversione al rischio degli investitori, che risulta essere del tutto razionale e poco restrittiva, il modello si basa su una serie di altre ipotesi che non sempre risultano verificate nei mercati reali, come la normalità dei rendimenti, che spesso si dimostrano asimmetrici e con code spesse; il comportamento sempre razionale degli investitori, spesso sfatato dalla finanza comportamentale; ed, infine, la perfetta simmetria informativa tra tutti gli operatori di mercato.

1.2. Efficient Market Hypothesis

Proprio questi ultimi due punti hanno dato luogo ad uno dei dibattiti più discussi nel panorama economico-finanziario, ovvero l'efficienza dei mercati finanziari. Il concetto di mercato efficiente è spiegato in "Efficient Capital Markets"[2] di Eugene F. Fama, pubblicato nel 1970, nel quale l'autore introduce quelle che ad oggi prendono il nome di Efficient Market Hypotesis (EMH).

La nozione di mercato efficiente può essere sintetizzata nell'affermazione di Fama secondo cui "in un mercato efficiente i prezzi riflettono tutta l'informazione disponibile".[3]

[2] Fama, E.F. (1970). Efficient Capital Markets: A Review Of Theory And Empirical Work.

[3] A livello formale, il concetto di efficienza identificato da Fama nel quale i prezzi riflettono l'informazione disponibile può essere rappresentato in termini di rendimenti attesi condizionati, uguagliando il prezzo di un'attività finanziaria al tempo t con il suo valore atteso futuro condizionato all'informazione disponibile in t: $E(\widehat{p_{j,t+1}}|\phi_t) = [1 + E(\widehat{r_{j,t+1}}|\phi_t)]p_{j,t}$ dove $p_{j,t}$ è il valore del prezzo dell'asset j al tempo t; $\widehat{r_{j,t+1}}$ è la stima del rendimento uniperiodale $(p_{j,t+1} - p_{j,t})/p_{j,t}$ e ϕ_t rappresenta l'insieme informativo disponibile al tempo t.

A partire da questa definizione, la letteratura ha distinto tre forme di efficienza nel mercato:

1. Efficienza in forma debole: secondo la quale i prezzi incorporano solamente l'informazione contenute nelle serie storiche dei prezzi. In questo contesto, l'analisi tecnica non consentirebbe di ottenere extra profitti dal mercato, poiché le dinamiche già passate sono riflesse nei prezzi correnti. Resterebbe però valida la possibilità, per i soggetti che possiedono informazioni non incorporate nei prezzi storici di mercato, di ottenere degli extra rendimenti;
2. Efficienza in forma semi-forte: secondo la quale i prezzi incorporano tutte le informazioni pubblicamente disponibili, come: bilanci societari, dati macroeconomici e comunicati ufficiali. In questo caso, né l'analisi tecnica né l'analisi fondamentale consentirebbero di generare extra profitti dal mercato. Eventuali extra rendimenti potrebbero quindi derivare esclusivamente dal possesso di informazioni private e non ancora diffuse nel mercato;
3. Efficienza in forma forte: secondo la quale i prezzi incorporano tutta l'informazione disponibile, sia privata che pubblica. In tale configurazione, nessun investitore può ottenere extra rendimenti, poiché tutta l'informazione è già riflessa nei prezzi di mercato.

Fama propone successivamente una riformulazione di questa condizione, nel quale viene preso in considerazione l'errore di previsione, ossia la differenza tra il prezzo effettivamente realizzato e il prezzo atteso condizionato. Questa nuova condizione di efficienza implica che tale errore abbia valore atteso nullo.[4]

Questa condizione esprime il processo di formazione dei prezzi come un gioco equo rispetto all'informazione disponibile. Data l'informazione in t non è quindi possibile prevedere sistematicamente deviazioni positive o negative rispetto al rendimento atteso di equilibrio.

Nella formulazione delle ipotesi di efficienza dei mercati, Fama individua una serie di condizioni sufficienti, ma non necessarie, che garantirebbero il pieno funzionamento del

[4] $E(p_{t+1} - E(p_{t+1}|\phi_t)|\phi_t) = 0$ che può essere riscritta in termini di rendimenti attesi come:
$E(r_{t+1} - E(r_{t+1}|\phi_t)|\phi_t) = 0$

meccanismo informativo dei prezzi. In particolare, vengono richiamate tre condizioni fondamentali:

1. Assenza di costi di transazione nelle operazioni di scambio;
2. Perfetta simmetria informativa, ossia accessibilità gratuita e simultanea di tutte le informazioni, pubbliche e private, del mercato;
3. Omogeneità nelle aspettative e nella valutazione degli investitori nella valutazione dell'impatto che l'informazione ϕ_t ha su prezzi correnti e distribuzione dei prezzi futuri.

Qualora queste condizioni risultino soddisfatte, il sistema dei prezzi incorporerebbe coerentemente tutta l'informazione disponibile al tempo t, rendendo impossibile per gli investitori conseguire extra rendimenti in modo sistematico. Tuttavia, Fama sottolinea che si tratta di condizioni sufficienti ma non necessarie, per questo motivo l'assenza di una o più di queste condizioni non implica automaticamente la presenza di possibili inefficienze di mercato. Infatti, anche in presenza di costi di transazione, asimmetrie informative o divergenze nelle aspettative sull'informazione disponibile, il mercato può comunque tendere all'efficienza qualora un numero sufficientemente ampio di investitori sia in grado di accedere alle informazioni e di utilizzale nelle proprie decisioni di investimento.

Infine, nel proprio lavoro del 1970, Fama divide le analisi empiriche secondo le forme di efficienza precedentemente elencate: debole, semi-forte e forte.

Pe quanto riguarda l'efficienza in forma debole, i test si concentrano sulla prevedibilità delle variazioni dei prezzi a partire dalle serie storiche dei prezzi stessi. Nonostante sia possibile individuare una certa correlazione seriale positiva nei rendimenti giornalieri, Fama sottolinea come tali dipendenze siano economicamente trascurabili e, soprattutto, non sufficienti a generare extra rendimenti a causa dei costi di transazione. L'assenza di prevedibilità dei prezzi, che sia economicamente sfruttabile, appare quindi coerente con la definizione di efficienza in forma debole.

Nei test relativi all'efficienza in forma semi-forte, dove l'analisi viene estesa all'insieme delle informazioni pubbliche, le evidenze presentate da Fama dimostrano che il mercato reagisce quasi istantaneamente e in modo sistematico alle nuove informazioni. Ciò suggerisce anche in questo caso l'impossibilità di ottenere extra rendimenti sfruttando informazioni pubbliche.

Infine, per quanto riguarda l'efficienza in forma forte, i test empirici su questa forma risultano essere più problematici. Fama riconosce che alcune categorie di operatori, come i corporate insider, possano effettivamente beneficiare di un accesso privilegiato alle informazioni non disponibili al pubblico. Tuttavia, l'autore interpreta queste situazioni come casi circoscritti, utili più a livello teorico che come descrizione realistica del funzionamento quotidiano dei mercati.

Analizzando questa condizione di mercato efficiente in un'ottica di costruzione di portafoglio, ne conseguirebbe che qualsiasi strategia di trading, costruita sulla base dell'informazione disponibile, genererebbe un profitto atteso nullo.

1.3. Il paradosso dell'efficienza informativa

Sicuramente la teoria dei mercati efficienti elaborata da Fama ha influenzato e ha fornito un benchmark interpretativo nello studio dei mercati finanziari. Tuttavia, con l'evoluzione della letteratura, si è sempre più cercato di far conciliare nuove ipotesi di efficienza con scenari più concreti e con ipotesi meno restrittive. In particolare, l'introduzione di costi informativi, asimmetrie tra operatori e frizioni operative ha portato ad una riformulazione del concetto stesso di efficienza, senza abbandonare il modello di Fama, ma bensì integrandolo con strutture teoriche capaci di spiegare il funzionamento di queste variazioni in uno scenario competitivo come quello dei mercati finanziari.

In questa prospettiva, risulta utile citare il contributo di Grossman e Stiglitz, che nel 1980 pubblicarono "On the Impossibility of Informationally Efficient Markets"[5] dimostrando come l'ipotesi di perfetta efficienza informativa, non possa convivere in presenza di costi di acquisizione dell'informazione, aprendo la strada ad una concezione dell'efficienza come condizione di equilibrio imperfetto ma sostenibile.

Il punto di partenza della loro dimostrazione è l'incoerenza tra due assunzioni fondamentali: la prima, effettuata da Fama, secondo la quale i prezzi riflettano completamente tutta l'informazione disponibile; la seconda, data dal fatto che l'informazione è costosa e richieda un certo investimento di risorse.[6]

---

[5] Grossman, S. and Stiglitz, J. (1980). On the Impossibility of Informationally Efficient Markets.

[6] Formalmente, il modello descrive il rendimento di un'attività rischiosa come: $u = \theta + \varepsilon$
dove θ rappresenta la componente informativa osservabile ad un costo c, mentre ε costituisce una componente di rumore non osservabile. Gli investitori possono quindi scegliere se sostenere il costo c per raccogliere le informazioni disponibili, oppure non formarsi osservando solamente il prezzo di mercato. Di conseguenza, il prezzo di equilibrio può essere espresso come: $P_\lambda(\theta, x) = \alpha_1 + \alpha_2 w_\lambda(\theta, x)$

Il risultato fondamentale risiede nel fatto che l'informatività del prezzo dipende direttamente dalla quota di operatori λ che hanno sostenuto dei costi per informarsi. Se nessuno è informato ($\lambda = 0$), il prezzo non riflette l'informazione del mercato e quindi ciascun investitore ha un incentivo ad informarsi. Se invece tutti sono informati ($\lambda = 1$) o l'informazione è perfettamente rilevata dal prezzo (assenza di rumore), allora il rendimento atteso dell'investimento in informazione diventa nullo e nessuno è incentivato a sostenerne il costo della formazione.

Formalmente, l'equilibrio richiederebbe che il rapporto tra l'utilità attesa degli informati e l'utilità attesa dei non informati sia pari ad uno[7], ma tale condizione non può essere soddisfatta se il prezzo riflette integralmente l'informazione del mercato. Da ciò deriva che l'esistenza di un equilibrio in questo modello, porterebbe ad un disequilibrio del modello tradizionale di Fama, generando una situazione in cui i prezzi incorporano solo parzialmente l'informazione disponibile, garantendo così una remunerazione per chi sostiene il costo informativo.

1.4. I limiti dell'arbitraggio

Un'altra critica che può essere mossa all'ipotesi di efficienza dei mercati formulata da Fama è che questa individua nei meccanismi concorrenziali degli operatori razionali il principale strumento di correzione dei prezzi. In particolare, eventuali deviazioni dai valori teorici dei prezzi vengono corretti dall'intervento degli arbitraggisti, i quali, sfruttando le discrepanze tra prezzo e valore intrinseco, riportano i prezzi di mercato al loro valore di equilibrio. Questa condizione presuppone quindi implicitamente che tali forze correttive siano sufficientemente potenti e celeri da eliminare immediatamente e sistematicamente ogni opportunità di profitto anomalo.

Proprio su questa riflessione, Shleifer e Vishny nel 1997, nel loro articolo "The Limits of Arbitrage"[8] mettono in discussione questo meccanismo correttivo, ponendo maggiore attenzione a quelli che sono i vincoli finanziari e istituzionali che caratterizzano le attività di arbitraggio nei mercati reali. Gli autori sostengono infatti che l'arbitraggio non è svolto

---

dove α_1 e α_2 sono numeri reali e w_λ è una combinazione lineare tra l'informazione osservabile θ e il rumore di mercato x. Ne consegue che il prezzo trasmette l'informazione, ma in modo imperfetto.

[7] $$\frac{E[V(W^I)]}{E[V(W^U)]} = 1$$

[8] Shleifer, A. and Vishny, R.W. (1997). The Limits of Arbitrage.

da investitori con capitale illimitato, bensì da professionisti che operano principalmente mediante la raccolta di capitale di terzi, il quale può seguire diversi flussi, in entrata ed in uscita, in base alle necessità dell'intermediario.[9]

Proprio tale struttura introduce un vincolo fondamentale nella teoria di correzione dei prezzi attraverso l'arbitraggio, secondo il quale perdite temporanee determinano deflussi di capitale, che vanno a ridurre la capacità dell'arbitraggista di eseguire operazioni correttive dei prezzi di mercato.

Ne deriva un modello che riconosce come degli errori di prezzo possano aumentare nel breve periodo prima di correggersi, generando a loro volta delle perdite per l'arbitraggista. In presenza di performance negative, gli investitori potrebbero decidere di ritirare i fondi, costringendo gli arbitraggisti a liquidare le loro posizioni e facendo allontanare ulteriormente il prezzo al suo valore di equilibrio. Questo meccanismo produce quindi un effetto paradossale: l'arbitraggio, anziché stabilizzare il mercato, può amplificare temporaneamente le deviazioni dei prezzi.

Ciò spiega il motivo per il quale i mercati finanziari possano essere caratterizzati da temporanei disallineamenti dei prezzi rispetto a quelli di equilibrio, non necessariamente eliminati istantaneamente dagli arbitraggisti. Tali scostamenti non implicano l'assenza di razionalità del mercato, bensì riflettono l'esistenza di rischi di liquidità e di agenzia che caratterizzano la gestione del capitale di terzi e che causano il rallentamento del processo di aggiustamento dei prezzi. Dunque, in questo contesto è necessario interpretare l'efficienza come una proprietà dinamica e graduata, piuttosto che una condizione assoluta che modifica immediatamente i prezzi di mercato.

1.5. Dall'Efficient Market Hypothesis all'Adaptive Markets Hypothesis

Le numerose critiche provenienti dal mondo della finanza hanno progressivamente messo in discussione l'interpretazione più rigida dell'EMH formulata da Fama, soprattutto per quanto riguarda le ipotesi di razionalità degli investitori nel mercato.

---

[9] Nel modello proposto, il capitale gestito nel periodo successivo a quello di riferimento (F_2), dipende dal rendimento ottenuto nel periodo precedente, secondo una relazione del tipo:

$$F_2 = F_1 * G\left(\frac{D_1}{F_1}\frac{p_2}{p_1} + \left(1 - \frac{D_1}{F_1}\right)\right)$$

dove D1 rappresenta la posizione detenuta e G() è la funzione che lega i flussi futuri alla performance realizzata.

Un tentativo particolarmente influente di superare queste incompatibilità tra efficienza e finanza comportamentale è rappresentato dalle Adaptive Markets Hypothesis (AMH) proposte da Andrew W. Lo nel 2004 nella sua pubblicazione "The Adaptive Market Hypothesis: Market Efficiency from an Evolutionary Perspective"[10]. Lo sviluppa un quadro teorico alternativo basato su principi evoluzionistici, come: competizione, selezione naturale e adattamento, applicati alle interazioni nei mercati finanziari, con l'obbiettivo che non è quello di confutare le EMH, bensì quello di proporre una reinterpretazione dinamica del concetto di efficienza basata sui principi dell'evoluzione biologica.

Nel suo lavoro, Lo apre l'analisi ripercorrendo la formulazione classica dell'EMH e lo fa con una celebre battuta che spiega efficacemente il significato alla base del modello di Fama. L'aneddoto racconta di due economisti che camminando per strada si imbattono in una banconota da 100$ a terra: quando il primo si abbassa per raccoglierla, l'altro lo ferma affermando che se fosse realmente una banconota autentica, qualcuno l'avrebbe già presa.

Questo esempio riesce perfettamente a far cogliere l'idea che in un mercato di investitori razionali e competitivi, ogni informazione disponibile viene immediatamente sfruttata a fini di profitto e, proprio attraverso tale processo, incorporata nei prezzi. Ne consegue che la competizione elimina sistematicamente ogni opportunità di profitto anomalo, eliminando la possibilità di guadagno basato sull'informazione disponibile, proprio perché qualsiasi vantaggio informativo sarebbe già stato utilizzato da altri investitori.

Successivamente, Lo affronta direttamente il centro della contestazione verso l'EMH, criticando i presupposti di razionalità degli investitori, coerenza delle preferenze e corretta formulazione delle probabilità soggettive. Secondo la teoria tradizionale, gli individui possiedono preferenze stabili e coerenti, con l'obbiettivo di massimizzare la loro utilità attesa. Tuttavia, la psicologia e la finanza comportamentale hanno dimostrato come il processo decisionale umano è fortemente influenzato da bias cognitivi, come overconfidence, overreaction, loss avversion e miscalibration delle probabilità, sottolineando come questi comportamenti non siano eventi sporadici, ma veri e propri pattern osservabili sperimentalmente.

[10] Lo, A. (2004). The Adaptive Markets Hypothesis: Market Efficiency from an Evolutionary Perspective.

Uno degli esperimenti più significativi riportati da Lo è il test di Kahneman e Tversky (1979)[11], che evidenzia la natura asimmetrica dell'aversione al rischio degli individui. Gli investitori tenderebbero infatti ad essere avversi al rischio in presenza di guadagni certi e, al contrario, propensi al rischio quando devono affrontare delle perdite. Questo implicherebbe una funzione di utilità che non risulta né lineare né simmetrica, in contrasto con l'ipotesi razionalità degli investitori.

In questo contesto, il problema non è semplicemente che gli individui commettano degli errori, bensì che tali errori siano sistematici e prevedibili. Se dunque le preferenze degli investitori non sono pienamente razionali, l'ipotesi che i prezzi di mercato riflettano pienamente e correttamente tutta l'informazione disponibile diventa meno solida, poiché se una parte significativa degli operatori è soggetta a distorsioni cognitive, l'aggiustamento dei prezzi, che dovrebbe essere immediato e successivo a qualsiasi variazione dal valore fondamentale, potrebbe non essere istantaneo né completo.

A questo punto si aggiunge un'ulteriore critica, trattata in precedenza anche in questo elaborato, ossia la dimostrazione di Grossman e Stiglitz (1980), secondo la quale mercati perfettamente efficienti sono logicamente impossibili.

Infine, Lo discute il ruolo dei cosiddetti "noise traders", ossia investitori che agiscono sulla base di segnali ritenuti informativi, ma che sono in realtà privi di contenuto fondamentale. La loro presenza introduce del rumore nei prezzi, generando temporali deviazioni dai valori fondamentali. Come visto in precedenza, queste deviazioni creano opportunità di arbitraggio per gli investitori più razionali, ma dall'altra parte potrebbero scaturire quei limiti dell'arbitraggio trattati anche da Shleifer e Vishny (1997).

Dopo aver esposto le critiche comportamentali all'EMH, Lo sostiene che il conflitto tra efficienza e finanza comportamentale non sia rappresentato solamente da incoerenze empiriche, ma che fondi le sue radici nel contrasto metodologico tra economia e psicologia. Mentre la psicologia accetta infatti l'idea che il comportamento umano sia eterogeneo, incoerente e influenzato dal contesto, l'economia tende a ricercare una teoria unificata del comportamento, seppur fondata su assunzioni più restrittive di razionalità.

[11] Kahneman, D. and Tversky, A. (1979). Prospect theory: an Analysis of Decision Under Risk. Econometrica, 47(2), pp.263–292.

Secondo Lo, il principale contributo a questa teoria unificata della finanza è attribuibile a Paul Samuelson e alla sua opera "Foundations of Economic Analysis"[12] del 1947. In questo lavoro Samuelson introdusse un approccio fortemente matematico, ispirato ai metodi della fisica, per costruire i principi dell'economia e della finanza come se derivassero da una scienza deduttiva capace di produrre teoremi rigorosi. Nonostante questa impostazione abbia prodotto degli straordinari risultati teorici, come: modelli di asset pricing, CAPM, teoria dell'equilibrio generale, teoria di portafoglio, … la stessa ha anche creato una forte dipendenza scientifica sulla coerenza formale.

Lo utilizza il concetto di "physics envy", ossia l'ambizione della finanzia di emulare la precisione e l'universalità delle leggi fisiche. Ma la principale problematica, secondo Lo, è che i sistemi economici sono completamente differenti dai sistemi fisici, poiché è presente una componente sociale che non appartiene invece alla fisica. Se infatti le leggi della fisica sono stabili e invarianti, i comportamenti umani sono adattivi e influenzati dall'ambiente.

Se l'analogia con la fisica è limitante, Lo suggerisce che un'analogia più appropriata possa essere quella effettuata con la biologia evolutiva. I mercati finanziari, infatti, non sono sistemi in equilibrio permanente, bensì sono rappresentati da ecosistemi dinamici popolati da agenti che apprendono e si adattano. L'obbiettivo, dunque, non è più quello di ricercare leggi universali e invarianti, ma di comprendere come le dinamiche evolutive vanno ad inficiare i mercati finanziari.

Un punto fondamentale per l'introduzione dell'AMH risiede nel richiamo del concetto di "bounded rationality" di Herbert Simon nel suo articolo "A Behavioral Model of Rational Choice"[13]. In questo lavoro Simon introduce l'idea che gli individui non dispongano né delle informazioni complete né della capacità computazionale necessaria per massimizzare l'utilità attesa. Questo porta gli individui a ricorrere a regole decisionali semplificate, che consentono di affrontare problemi complessi.

La novità introdotta dalle AMH consiste nell'integrare questa razionalità limitata con la dinamica evolutiva. Le strategie di investimento selezionate non vengono scelte perché ottimale in senso strettamente teorico, ma perché funzionali alla sopravvivenza in un

[12] Paul A. Samuelson (1947). Foundations of economic analysis.
[13] Simon, H.A. (1955). A Behavioral Model of Rational Choice. The Quarterly Journal of Economics, 69(1), pp.99–118.

determinato contesto competitivo. Questo perché gli operatori che adottano strategie inefficaci vengono progressivamente cacciati dal mercato, mentre quelli che riescono ad adattarsi sopravvivono ottenendo risultati positivi.

Lo descrive quindi il mercato come un ecosistema naturale, nel quale convivono diverse "specie" che competono per risorse limitate, ad esempio: investitori retail, fondi pensione, hedge fund, … ciascuno con caratteristiche, strategie e orizzonti temporali differenti. L'efficienza dipende quindi dall'intensità della competizione tra queste specie. Quando molti operatori competono per sfruttare le stesse opportunità di profitto, l'efficienza tende ad aumentare e di conseguenza le opportunità di profitto si riducono. Al contrario, quando la competizione è bassa o emergono nuovi segmenti di mercato, possono svilupparsi inefficienze persistenti. Da ciò ne deriva che l'efficienza non è una proprietà intrinseca del mercato, ma bensì il risultato di una complessa struttura competitiva dell'ambiente intorno a questo.

Proprio come le specie in un ecosistema naturale, le strategie di investimento attraversano cicli di successo e di declino. Lo osserva che determinate strategie, dopo aver generato redimenti significativi, attraggono nuovi capitali, aumentando la competizione e riducono progressivamente la loro redditività. Successivamente, quando l'interesse diminuisce, il capitale si ritira e il mercato riduce la sua competizione, le opportunità riemergono. Da questo ne deriva che l'efficienza dinamica è ciclica ed è quindi possibile spiegare fenomeni come bolle speculative, crisi finanziari e fasi di forte inefficienza senza dover abbandonare l'idea di mercati razionali.

Il risultato teorico più importante della AMH è che l'EMH non viene rigettata, ma bensì viene reinterpretata come un caso particolare nel quale, in un ambiente stabile, con alta competizione e abbondanza di capitale per l'arbitraggio, i mercati possano avvicinarsi ad una situazione di perfetta efficienza informativa, come quella descritta da Fama.

1.6. Implicazioni per la costruzione di portafoglio

Se si accetta che i mercati siano sistemi adattivi e non stazionari, ne derivano conseguenze rilevanti per la teoria di portafoglio. I premi per il rischio non possono essere considerati costanti, le correlazioni tra asset possono mutare in funzione del mercato e le strategie che hanno funzionato in passato possono perdere efficacia.

L'ottimizzazione di portafoglio non può quindi essere concepita come un esercizio statico, ma come un processo dinamico di adattamento continuo. In questa prospettiva, strumenti

innovativi come il machine learning ed il reinforcement learning risultano coerenti con l'AMH, poiché permettono di incorporare aggiornamento, revisione dei parametri e riallocazione in risposta a condizioni di mercato mutevoli nel tempo, consentendo di poter operare in mercati sempre più interconnessi, rapidi e complessi.

Nel presente capitolo è stato delineato il quadro teorico di riferimento, rappresentato dall'AMH. A partire da tale impostazione, l'obbiettivo è ora quello di verificare empiricamente se, operando in coerenza con la teoria proposta da Lo, sia possibile conseguire risultati economici soddisfacenti, attraverso lo sviluppo di un algoritmo adattivo basato su meccanismi di apprendimento per rinforzo.

Nei capitoli successivi verranno introdotte delle classi di tecniche computazionali, note come "reinforcment learning", impiegate per affrontare problemi di ottimizzazione dinamica mediante processi iterativi. Tali tecniche risultano particolarmente adatte ad operare in contesti in continua evoluzione, in cui l'agente deve adattare progressivamente le proprie decisioni sulla base dei feedback ricevuti dall'ambiente esterno.

Se, come sostiene Lo, la teoria dei mercati deve evolversi incorporando concetti quali apprendimento, adattamento ed evoluzione, allora l'impostazione del reinforcement learning appare pienamente coerente con questa filosofia. In questi modelli, infatti, un agente sviluppa strategie operative attraverso l'interazione continua con l'ambiente, modificando il proprio comportamento in funzione dei risultati ottenuti. Essi rappresentano pertanto uno strumento computazionale idoneo a dare un'applicazione concreta ai principi teorici dell'AMH, traducendo l'idea di mercato evolutivo in un procedimento operativo formalizzato digitalmente.

CAPITOLO 2 - TECNICHE DI MACHINE LEARNING IN CONTESTI ECONOMICI E FINANZIARI

Negli ultimi decenni, la crescente evoluzione delle tecnologie informatiche e l'aumento della disponibilità di dati di alta qualità hanno determinato un'evoluzione esponenziale nell'utilizzo delle tecniche di machine learning (ML) in moltissimi settori di ricerca.

Per quanto riguarda il contesto economico-finanziario, Galo et al., in "Machine learning in business and finance: a literature review and research opportunities"[14] descrivono come il progressivo consolidamento delle tecniche di ML abbiano fornito uno strumento metodologico fondamentale per l'analisi di fenomeni economici complessi, tra cui la gestione del rischio, il pricing nei mercati azionari e la costruzione di portafoglio.

Rispetto ai modelli econometrici tradizionali, le tecniche di ML presentano alcune caratteristiche distintive che permettono di trattare efficacemente grandi quantità di dati, sia strutturati che non, ma soprattutto di evitare l'imposizione di forti assunzioni sulla forma strutturale del modello o sulla distribuzione dei parametri. Tali caratteristiche rendono questi modelli maggiormente orientati alla massimizzazione della performance predittiva al di fuori del campione di riferimento.

Nella loro sezione introduttiva, Galo et al. definiscono il ML come un insieme di algoritmi finalizzati a modellare relazioni tra variabili esplicative e variabili dipendenti in contesti in cui il modello teorico sottostante non è facilmente esprimibile in forma chiusa e/o sotto forma di relazioni lineari.

In merito al processo di apprendimento di questi modelli, questo si basa tipicamente sulla divisione del campione in due sotto campioni. Il primo, chiamato campione di addestramento (o training sample), viene utilizzato per stimare i parametri del modello attraverso la minimizzazione di una funzione di perdita. Il secondo campione utilizza la restante parte dei dati, non ancora analizzati dal modello, per valutare in termini di capacità predittiva l'accuratezza e la precisione di questo.

[14] Gao, H., Kou, G., Liang, H., Zhang, H., Chao, X., Li, C.-C. and Dong, Y. (2024). Machine learning in business and finance: a literature review and research opportunities. Financial Innovation, 10(1).

Gli autori distinguono il ML in tre principali macrocategorie: supervised learning, unsupervised learning e reinforcement learning.

Nel supervised learing si osservano sia le variabili esplicative (X), sia la variabile target (Y), con l'obbiettivo di trovare una funzione capace di minimizzare l'errore di previsione.

Nell'unsupervised learning si osservano invece esclusivamente le variabili esplicative (X) con l'obbiettivo di individuare pattern specifici all'interno del dataset.

Infine, come citato nel capitolo precedente, merita particolare attenzione il reinforcement learning (RL) che, a differenza delle metodologie precedenti rappresenta un paradigma distinto, incentrato sull'apprendimento tramite interazione con un ambiente esterno e orientato alla massimizzazione di una ricompensa nel tempo.

Poiché lo sviluppo futuro del presente lavoro si concentra sull'applicazione del RL alla gestione di portafoglio, si è deciso di non sviluppare ulteriormente l'analisi delle tecniche di supervised learning e unsupervised learning. Per approfondire tali argomenti si rimanda alla letteratura specialistica e a lavori dedicati.

2.1. Reinforcement Learning:

Il principale riferimento teorico per questo e i successivi capitoli è rappresentato dal lavoro di Sutton e Barto, "Reinforcement Learning: An Introduction"[15], che presentano il RL come un processo di apprendimento basato sull'interazione tra un agente e un ambiente, nel quale l'agente apprende autonomamente una politica decisionale attraverso un processo di prova ed errore finalizzato alla massimizzazione della ricompensa cumulata nel tempo. Tale meccanismo richiama esattamente le dinamiche di apprendimento osservabili in natura, in cui l'esperienza guida l'adattamento e migliora il comportamento di un individuo.

Per inquadrare correttamente il concetto di RL, conviene partire dalla sua idea fondamentale, ossia quella di un agente che interagisce nel tempo e in un ambiente incerto, scegliendo le azioni che non solo influenzano la ricompensa immediata, ma anche gli stati futuri e quindi l'opportunità di ricevere maggiori ricompense successive. In questa prospettiva, l'obbiettivo non è semplicemente quello di predire o classificare sulla base dei dati a disposizione, bensì decidere sotto incertezza, cercando di considerare le

[15] Sutton, R.S. and Barto, A. (2018). Reinforcement learning: An introduction. 2nd ed. Cambridge, Ma ; London: The Mit Press.

conseguenze indirette e ritardate delle scelte che vengono effettuate in un determinato istante. Spesso, infatti, un'azione è utile non tanto perché produce subito un esito favorevole, ma perché apre la strada a stati potenzialmente più favorevoli in futuro, e ciò richiede oltre alla capacità predittiva anche una qualche forma di pianificazione.

All'interno dei sistemi di RL, è possibile identificare quattro elementi fondamentali, oltre all'agente e l'ambiente, che caratterizzano questo tipo di strutture: policy, reward signal, value function e, facoltativamente, model of the environment.

La policy (o politica) identifica come l'agente opera in un determinato stato e in un determinato tempo, rappresenta il nucleo dell'agente e ne determina il comportamento.

Il reward signal (o segnale di ricompensa) determina invece cosa l'agente deve perseguire. Ad ogni istante temporale, l'ambiente invia all'agente un numero chiamato reward (o ricompensa) e l'obbiettivo principale dell'agente è massimizzare il totale delle ricompense nel lungo periodo. Il segnale di ricompensa permette all'agente di capire se l'azione selezionata dalla propria politica ha avuto successo o meno a seconda della ricompensa ottenuta, di conseguenza l'agente saprà se in futuro questa azione può essere ripetuta o se deve essere cambiata.

La value function (o funzione di valore) introduce il valore fondamentale del tempo. Mentre la ricompensa rappresenta la bontà immediata delle scelte dell'agente, il valore misura la validità di lungo periodo, ossia l'aspettativa delle ricompense future cumulabili a partire da uno stato a seconda dell'azione selezionata dall'agente. Per spiegare al meglio questa differenza tra ricompensa e valore, Sutton e Barto utilizzano un'analogia con l'essere umano. La ricompensa può essere associata al piacere o al dolore che proviamo immediatamente dopo aver compiuto un'azione, ma il valore rappresenta un sentimento molto più astratto, come la soddisfazione o l'insoddisfazione associata nel trovarsi in un determinato stato piuttosto che in un altro.

Infine, il model of environment (o modello ambientale), è un elemento che, se presente, cerca di replicare il comportamento dell'ambiente. Dunque, dato uno stato e un'azione, il modello tenta di prevedere lo stato successivo e la corrispondente ricompensa. I metodi di RL che fanno uso di questi modelli sono definiti model-based, mentre i modelli più semplici che apprendono esclusivamente mediante meccanismi di prova ed errore prendono il nome di model-free.

2.2. Markov Decision Process (MDP)

Prima di introdurre formalmente le funzioni che caratterizzano il RL, è necessario soffermarsi sulla struttura del processo decisionale che caratterizza questo tipo di analisi.

L'agente, come già discusso, non opera in un contesto statico, bensì interagisce con un ambiente dinamico dal quale riceve, ad ogni istante discreto t, una serie di informazioni che vengono sintetizzate nello stato S_t. Tale stato rappresenta dunque l'insieme delle informazioni disponibili all'agente in un preciso istante temporale e costituisce il riferimento sulla base del quale verrà selezionata un'azione A_t, che genererà a sua volta la ricompensa R_{t+1}. Questa interazione porta dunque a generare una sequenza del tipo $(S_0, A_0, R_1); (S_1, A_1, R_2);$ ..., nella quale stato, azione e ricompensa sono strettamente legati tra loro.

Occorre tuttavia chiarire che lo stato non coincide necessariamente con la realtà assoluta dell'ambiente, bensì questo rappresenta la porzione di informazione effettivamente accessibile all'agente. In molti contesi può succedere che parte delle variabili rilevanti possa essere non osservabile, l'agente apprenderà progressivamente attraverso l'esperienza ma in ogni istante è richiesta una decisione sulla base di ciò che conosce. In questo senso, sarebbe idealmente auspicabile disporre di una rappresentazione dello stato capace di riassumere in maniera compatta tutta l'informazione passata rilevante per prevedere i possibili scenari futuri, scartando al contempo i dettagli non informativi. E proprio questa esigenza conduce all'introduzione della proprietà di Markov.

Nel formalismo del Markov Decision Process (MDP), uno stato si dice Markoviano se la distribuzione condizionata del prossimo stato e della ricompensa dipendono esclusivamente dalla coppia corrente stato-azione e non da tutta la storia delle interazioni precedenti, formalmente:

$$p(s', r|s, a) = \Pr(S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a)$$

Questa funzione di transizione probabilistica descrive la probabilità che, dato lo stato $S_{t-1} = s$ e l'azione $A_{t-1} = a$, il sistema evolva verso $S_t = s'$ producendo una ricompensa $R_t = r$. Questo implica che lo stato attuale contenga tutte le informazioni necessarie per determinare le distribuzioni future, ovvero che il passato sia completamente riassunto nello stato presente.

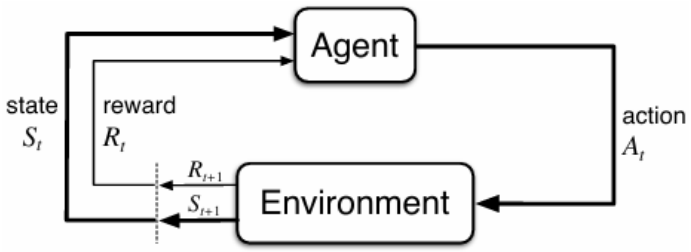

Figura 2.1: Interazione stato-agente in un MDP[16]

Proprio questa proprietà consente di trattare il problema in modo ricorsivo, facendo sì che le decisioni future dipendano solamente dallo stato corrente, rendendo così il problema matematicamente trattabile. Infatti, una volta garantita la proprietà di Markov, l'intero processo decisionale può essere analizzato attraverso relazioni ricorsive tra stati successivi.

Si possono infine esplicitare due proprietà fondamentali derivanti dalla proprietà Markoviana:

- Probabilità di transizione:

$$p(s'|s,a) = Pr(S_t = s'|S_{t-1} = s, A_{t-1} = a) = \sum_{r\in\mathcal{R}} p(s', r|s, a)$$

- Rendimenti attesi, data la coppia stato-azione:

$$r(s,a) = E(R_t|S_{t-1} = s, A_{t-1} = a) = \sum_{r\in\mathcal{R}} r \sum_{s\in\mathcal{S}} p(s', r|s, a)$$

Risulta comunque importante sottolineare come la proprietà Markoviana non è una caratteristica intrinseca dell'ambiente, ma dipende dalla rappresentazione scelta dello stato. Se lo stato omette informazioni rilevanti, la dinamica non sarà Markoviana; se invece lo stato incorpora tutte le variabili necessarie a prevedere la distribuzione del futuro, allora la proprietà di Markov risulta soddisfatta.

2.3. Funzione di ricompensa e ritorno

Una volta definito il processo, è necessario esplicitare la funzione obbiettivo perseguita dall'agente. Come già introdotto precedentemente, la ricompensa non è altro che un segnale numerico che l'ambiente comunica all'agente ogni istante temporale a seconda

[16] Sutton, R.S. and Barto, A. (2018). Reinforcement learning: An introduction. 2nd ed. Cambridge, Ma ; London: The Mit Press.

dell'azione scelta. L'obbiettivo dell'agente è dunque quello di massimizzare il totale delle ricompense ricevute, ma questo non avviene massimizzando la ricompensa immediata ad ogni istante, bensì massimizzando la somma scontata di tutte le ricompense attese future.

Indicando con $R_{t+1}, R_{t+2}, R_{t+3}$, ..., la sequenza delle ricompense successive all'istante t, è possibile definire, tramite un'apposita funzione, la nuova grandezza da massimizzare: l'expected return (o ritorno atteso), definito come G_t. Nello scenario più semplice, questa funzione è rappresentata dalla somma delle ricompense future:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

dove T rappresenta l'ultimo istante temporale, che può essere finito, se le interazioni ambiente-agente hanno una nozione naturale di tempo finale, oppure può tendere all'infinito qualora non esista un termine alle interazioni.

A questa prima formulazione del ritorno atteso, è necessario integrare la nozione di attualizzazione. Seguendo questo approccio, l'agente avrà come obbiettivo la massimizzazione del ritorno atteso attualizzato:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{K=0}^{\infty} \gamma^k R_{t+k+1}$$

dove γ indica il fattore di sconto, $0 \leq \gamma \leq 1$. Più γ si avvicina a 0 e più l'agente diventa "miope", ossia si concentra più ad una visione di breve termine, attribuendo un peso maggiore alle ricompense di breve periodo. Al contrario, più γ si avvicina a 1, più l'agente diventa lungimirante, concentrandosi maggiormente sui ritorni attesi futuri. Inoltre, se $\gamma < 1$, la sommatoria infinita descritta da G_t risulta convergente e assume pertanto un valore finito.

Infine, è possibile rappresentare la formula del ritorno atteso attualizzato in forma ricorsiva, più appropriata a livello teorico e per i successivi algoritmi di RL:

$$G_t = R_{t+1} + \gamma G_{t+1}$$

Formalizzate le definizioni di ricompensa e ritorno, è possibile introdurre una delle principali difficolta del RL, assente nelle altre forme di supervised e unsupervised learning. Per ottenere ricompense elevate, l'agente dovrebbe privilegiare le azioni che in passato si sono rivelate efficaci nel generare ricompense positive. Tuttavia, per

individuare tali azioni, o azioni migliori, è necessario sperimentare anche nuove azioni che non sono mai state selezionate prima. L'agente si trova quindi davanti ad una scelta:

- Da un lato deve sfruttare (exploit) le conoscenze già acquisite per massimizzare le ricompense
- Dall'altro lato deve esplorare (explore) possibili alternative potenzialmente migliori, così da poter aumentare le possibili ricompense future

Questo dilemma è il cosiddetto trade-off tra exploration ed exploitation, nel quale perseguire esclusivamente una delle due strategie porta inevitabilmente a risultati subottimali: un'eccesiva esplorazione impedisce di sfruttare le informazioni già disponibili, mentre un eccessivo sfruttamento delle conoscenze pregresse può impedire di scoprire azioni migliori. Proprio questo problema rappresenta una delle sfide più complesse del RL, la cui gestione richiede la definizione di un equilibrio adeguato tra esplorazione e sfruttamento.

2.4. Funzione di valore e politiche

Oltre alla definizione della funzione di ritorno, un altro aspetto fondamentale per gli algoritmi di RL è rappresentato dalla stima della funzione di valore, dove la nozione di valore è da intendersi come la quantificazione di quanto vantaggioso sia per l'agente trovarsi in un determinato stato. Sicuramente questo può essere definito in termini di ritorno atteso, che dipende dunque dalle azioni che l'agente sceglierà di intraprendere nel tempo. Per questo motivo le funzioni di valore sono sempre definite con riferimento alle modalità di comportamento dell'agente che, come visto nella parte introduttiva, prendono il nome di politiche e raccolgono tutte le possibili probabilità che in un determinato stato l'agente selezioni una determinata azione.

Dunque, la funzione di valore associata ad una determinata politica π misura quanto sia vantaggioso per l'agente trovarsi in uno certo stato, assumendo che questo continui a seguire quella politica nel futuro. Formalmente è possibile definire la funzione di valore come:

$$v_\pi(s) = E_\pi[G_t \mid S_t = s]$$

Ossia il valore atteso del ritorno a partire dallo stato s, quando ad ogni spazio temporale t, tutte le azioni vengono selezionate dalla politica π.

In modo analogo, è possibile definire la funzione di azione-valore:

$$q_\pi(s, a) = E_\pi[G_t \mid S_t = s,\ A_t = a]$$

Questa funzione rappresenta il valore atteso del ritorno ottenuto scegliendo inizialmente l'azione a nello stato s e seguendo successivamente la politica π per ogni istante t.

L'introduzione delle funzioni di valore consente di riformulare il problema dell'apprendimento sequenziale in termini più trattabili. Invece di determinare direttamente la sequenza ottimale di decisioni lungo l'intero orizzonte temporale, l'agente può concentrarsi sulla stima della qualità associata ai singoli stati o alle coppie stato-azione. Tali quantità sintetizzano infatti il valore atteso delle ricompense future che possono essere ottenute a partire da una determinata situazione. Una volta stimati questi valori, l'agente può migliorare progressivamente il proprio comportamento, selezionando, in ciascuno stato, l'azione che conduce al valore atteso più elevato.

Come osservato precedentemente, il ritorno può essere espresso in forma ricorsiva come la somma tra la ricompensa immediata e il ritorno futuro scontato. Proprio questa proprietà implica dunque che il valore associato ad uno stato non dipenda esclusivamente dalla ricompensa ottenuta, ma anche dal valore degli altri stati che possono essere raggiunti successivamente, creando così un'ulteriore struttura ricorsiva. Questa struttura permette di descrivere la relazione tra uno stato e il suo stato successivo e prende il nome di equazione di Bellman per la funzione di valore, formalmente:

$$v_\pi(s) = E_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s]$$

La relazione evidenzia come il valore di uno stato, per una policy π, sia determinato dalla combinazione tra ricompensa immediata R_{t+1} e valore atteso scontato degli stati successivi γG_{t+1}.

Esplicitando quindi il valore atteso rispetto alle probabilità di selezione delle azioni e delle transizioni ambientali, l'equazione può essere riscritta come:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a)\,[r + \gamma v_\pi(s')]$$

dove $\pi(a|s)$ rappresenta la probabilità di scegliere l'azione a nello stato s, mentre $p(s', r \mid s, a)$ descrive la dinamica probabilistica con cui si evolve l'ambiente dato lo stato attuale s e l'azione scelta a, producendo una ricompensa r. In questa formulazione appare

dunque più chiaro come il valore di uno stato dipenda dai valori degli stati successivi che possono essere raggiunti a partire dallo stesso a seconda dell'azione scelta dall'agente.

In modo del tutto analogo è possibile definire l'equazione di Bellman per la funzione di azione-valore:

$$q_\pi(s,a) = \sum_{s',r} p(s', r \mid s, a)\,[r + \gamma \sum_{a'} \pi(a'|s')\, q_\pi(s', a')]$$[17]

La quale evidenzia come il valore di una coppia azione-stato dipenda da tutte le possibili transizioni che possono verificarsi dopo aver eseguito l'azione a. In particolare, dopo aver eseguito l'azione a e raggiunto lo stato successivo s', l'agente selezionerà una nuova azione a' secondo la politica π.

In questo contesto, le equazioni di Bellman possiedono un'importanza teorica fondamentale per la soluzione delle equazioni del RL poiché permettono di esprimere il problema decisionale intertemporale dell'agente, potenzialmente molto complesso, come una serie di relazioni tra stati consecutivi.

2.5. Valore ottimale

A partire da queste relazioni, è possibile introdurre il concetto di valore ottimale. Se l'obbiettivo dell'agente è massimizzare il ritorno atteso, allora esiste una funzione di valore ottimale v_* che rappresenta il massimo valore ottenibile, partendo dallo stato s e scegliendo le azioni nel miglior modo possibile, ovvero seguendo la miglior politica.

È possibile, infatti, definire una politica π migliore o uguale di un'altra politica π' se il suo ritorno atteso è maggiore o uguale a quello della politica π' in tutti gli stati. In altre parole, $\pi \geq \pi'$ se e solo se $v_\pi(s) \geq v_{\pi'}(s)$ per ogni $s \in \mathcal{S}$. Queste politiche prendono il nome di politiche ottimali (o optimal policies) e, nonostante possano essere più di una, tutte soddisfano la stessa funzione di valore ottimale (o optimal state-value function), definita come:

$$v_*(s) = \max_\pi v_\pi(s)\ ; \forall s \in \mathcal{S}$$

[17] Kabbani, T. and Duman, E. (2022). Deep Reinforcement Learning Approach for Trading Automation in the Stock Market. IEEE Access, 10, pp.93564–93574.

In modo analogo, le politiche ottimali condividono anche la stessa funzione di azione-valore ottimale (o optimal action-value function), dove $\pi \geq \pi'$ se e solo se $v_\pi(s,a) \geq v_{\pi'}(s,a)$ per ogni $s \in \mathcal{S}$ e per ogni $a \in \mathcal{A}$, definita come:

$$q_*(s,a) = \max_\pi q_\pi(s,a) \ ; \ \forall s \in \mathcal{S} \ e \ \forall a \in \mathcal{A}$$

A differenza delle equazioni di Bellman precedentemente indicate, che esprimono il valore degli stati o delle coppie stato-azione rispetto ad una policy data, è possibile introdurre ora le equazioni ottimali di Bellman (o Bellman optimality equations), le quali descrivono il valore massimo ottenibile dall'agente assumendo che questo scelga sempre l'azione che massimizza il ritorno atteso. Poiché si parla di funzioni di ottimo, non è più necessario fare riferimento esplicito ad una politica, in quanto sarà direttamente l'agente a determinare la migliore. Dunque, il problema di ottimizzazione della policy viene automaticamente incorporato nella struttura stessa dell'equazione.

L'equazione ottima di Bellman per $v_*(s)$ può essere formalizzata come:

$$v_*(s) = \max_a \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma v_*(s') \right]$$

Mentre l'equazione ottima di Bellman per $q_*(s,a)$ può essere espressa come:

$$q_*(s,a) = \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma \max_{a'} q_*(s', a') \right]$$

Le equazioni ottime di Bellman rappresentano un risultato fondamentale nella teoria del RL, poiché formalizzano il legame tra il valore degli stati e la scelta ottimale delle azioni. Queste equazioni non rappresentano in realtà una singola relazione, bensì un sistema di equazioni, uno per ogni stato dell'ambiente. A livello teorico, se fosse possibile conoscere fin dal principio la dinamica di tutti gli stati, si potrebbe determinare immediatamente la policy ottimale. Una volta ottenuta la funzione di valore ottimale v_*, sarebbe sufficiente selezionare per ogni stato s, l'azione che massimizza il suo ritorno atteso nell'equazione di Bellman. In questo modo qualsiasi politica che assegni probabilità positiva esclusivamente a tali azioni risulterebbe essere una politica ottimale. Se invece si disponesse direttamente della funzione di valore-azione ottimale, la scelta diventerebbe ancora più semplice in quanto l'agente potrebbe selezionare direttamente l'azione che massimizza q_* per lo stato corrente, senza dover effettuare alcuna ricerca sugli stati successivi.

Tuttavia, nella maggior parte dei problemi reali, la dinamica dell'ambiente è ignota e/o troppo complessa per essere conosciuta inizialmente. Per questo motivo si sono sviluppati con il tempo numerosi algoritmi di apprendimento che hanno l'obbiettivo di approssimare e stimare progressivamente queste funzioni di valore, attraverso l'interazione e la graduale scoperta dell'ambiente.

CAPITOLO 3 - ALGORITMI DI REINFORCEMENT LEARNING

Nel capitolo precedente è stata introdotta la teoria fondamentale sulla quale si basano i sistemi di RL, attraverso la formalizzazione dei processi decisionali alla Markov (MDP), delle funzioni di valore e delle equazioni ottimali di Bellman. L'insieme di questi strumenti consente dunque di definire la base delle regole decisionali che permettono all'agente di trovare la politica ottimale che massimizza il rendimento atteso nel lungo periodo attraverso l'interazione con l'ambiente.

Nel corso degli anni la letteratura sul RL ha sviluppato numerosi algoritmi che permettessero di approssimare le equazioni di Bellman attraverso diverse tipologie di procedure. In generale, è possibile classificare gli algoritmi di RL in tre principali famiglie[18]:

- Metodi basati sulla funzione di valore (value-based methods)
- Metodi basati sulla politica decisionale (policy-based methods)
- Metodi ibridi (actor-critic methods)

Il presente capitolo ha quindi l'obbiettivo di presentare queste principali categorie di algoritmi che hanno contribuito alla formazione della letteratura moderna. In particolare, verranno analizzati per ogni categoria alcuni degli algoritmi più rilevanti, seguendo una progressione logica adatta a riflettere l'evoluzione storica effettuata dalla ricerca.

L'analisi di questi algoritmi consentirà di analizzare le caratteristiche che li rendono più o meno adatti alla risoluzione di determinate categorie di problemi, tra cui quelli legati alla gestione di portafoglio.

3.1. Modalità di apprendimento

Nonostante la grande varietà di algoritmi sviluppati negli ultimi anni, molti di essi possono essere ricondotti ai tre paradigmi fondamentali introdotti dalla letteratura classica del RL per le modalità con le quali risolvono i MDP:

- Dynamic Programming (DP)

[18] Terven, J. (2025). Deep Reinforcement Learning: A Chronological Overview and Methods. AI, 6(3), p.46.

- Monte Carlo Methods (MCM)
- Temporal-Difference learning (TD)

Queste tre approcci rappresentato differenti modalità di apprendimento e costituiscono le fondamenta sulle quali si basano gran parte degli algoritmi di RL. Sebbene i metodi più recenti possano presentare strutture più complesse e tecniche più avanzate, i loro principi di funzionamento rimangono riconducibili alle idee introdotte da queste classi.

È importante sottolineare come questi tre paradigmi non vengano necessariamente utilizzati in maniera isolata, bensì vengano spesso combinati o generalizzati al fine di formare strutture più complesse e performanti. In particolare, molti algoritmi value-based moderni possono essere ricondotti ai metodi di TD; al contrario, numerosi algoritmi policy-based utilizzano stime ottenute tramite MCM o attraverso combinazioni tra MCM e TD, come avviene nelle strutture actor-critic.

Alla luce di queste considerazioni risulta utile introdurre preventivamente i principi teorici alla base di ciascuna di queste tre classi, analizzando per ciascuna di queste le caratteristiche, i punti di forza e i relativi limiti.

3.2. Dynamic Programming

Il Dynamic Programming (DP), rappresenta uno dei primi approcci sviluppati per la risoluzione di problemi di decisione sequenziale e rappresenta la base teorica fondamentale per lo studio degli algoritmi di RL. In particolare, il DP indica una famiglia di metodi utilizzati per la risoluzione dei MDP quando è disponibile una descrizione completa dell'ambiente, ovvero sono note le probabilità di transizione tra stati e le ricompense associate alle diverse azioni.

Date le sue peculiari ipotesi restrittive, gli algoritmi di DP non sono particolarmente utilizzati nell'ambito pratico ma, nonostante ciò, ricoprono un importante ruolo a livello teorico per la loro capacità di convergenza.

L'idea fondamentale alla base degli algoritmi di DP è l'utilizzo della struttura ricorsiva delle equazioni di Bellman per esprimere il valore di uno stato in funzione dei valori degli stati successivi. Conoscendo dunque le probabilità di transizione e le ricompense associate alle azioni, è possibile trovare la politica che soddisfa le equazioni ottime.

Il primo problema affrontato dai metodi di DP è la valutazione di una politica (o policy evaluation), ossia il calcolo della funzione di valore $v_*(s)$ associata ad una politica

predeterminata. In questo contesto, il sistema di equazioni definito dall'equazione di Bellman viene sostituito da un processo iterativo noto come iterative policy evaluation. La funzione di valore viene dunque aggiornata iterativamente, seconda la regola:

$$v_{k+1}(s) = \sum_{a} \pi(a|s) \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma v_k(s') \right]$$

dove $v_k(s)$ rappresenta la stima del valore dello stato alla k-esima iterazione. Partendo da un valore arbitrario di v_0 e ripetendo questo aggiornamento per un numero sufficiente di stadi, è possibile dimostrare che la funzione di valore converga alla funzione v_π associata alla politica considerata.

Una volta stimata la funzione di valore, è possibile migliorare la politica selezionando, per ciascuno stato, l'azione che massimizza il valore atteso. Questo secondo passaggio prende il nome di policy improvement ed è formalmente espresso come:

$$\pi'(s) = \arg\max_{a} \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma v_\pi(s') \right]$$

Questa equazione identifica per ogni stato l'azione che massimizza il valore atteso delle ricompense future. Il processo si arresta quando la nuova politica π' coincide con quella precedente, identificando così la politica ottimale.

Una volta che la politica π è stata migliorata utilizzando v_π, trovando una politica π', è possibile trovare nuovamente v'_π per migliorare ulteriormente la politica π''. In questo modo è possibile ottenere una sequenza monotona di politiche migliorate e funzioni di valore del tipo:

$$\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi_* \xrightarrow{E} v_*$$

dove $\xrightarrow{E}$ rappresenta la valutazione della politica e $\xrightarrow{I}$ rappresenta il miglioramento della politica. Ogni passaggio garantisce un miglioramento rispetto al precedente, fino al raggiungimento della politica ottimale. Questa alternanza tra fasi di policy evaluation e policy improvement dà origine all'algoritmo noto come policy iteration, uno dei principali metodi di DP per l'ottenimento della politica ottimale.

In alternativa, è possibile combinare in un unico passaggio le operazioni di valutazione e miglioramento della politica, attraverso un algoritmo che prende il nome di value

iteration. In questo caso, l'aggiornamento della funzione di valore avviene secondo l'equazione ottimale di Bellman:

$$v_{k+1}(s) = \max_a \sum_{s',r} p(s', r \mid s, a)\,[r + \gamma v_k(s')]$$

Partendo sempre da uno stato arbitrario v_0 è possibile dimostrare che la sequenza v_k converge a v_* per un numero sufficiente di iterazioni.

Gli algoritmi di DP consentono dunque di determinare una politica ottimale attraverso l'alternanza di due fasi fondamentali: la valutazione della politica e il miglioramento di questa. Tuttavia, l'idea alla base di questo meccanismo può essere generalizzata in uno dei principi fondamentali del RL che prende il nome di Generalized Policy Iteration (GPI). La GPI descrive infatti il processo attraverso il quale un'agente migliora progressivamente il proprio comportamento alternando, da un lato, la stima della funzione di valore, per misurare la qualità degli stati o delle azioni rispetto ad una determinata politica e dall'altro, l'aggiornamento della politica utilizzata per selezionare azioni sempre più vantaggiose.

Questo principio assume un ruolo centrale nel RL poiché costituisce la base concettuale per gran parte dei suoi algoritmi. Sebbene i metodi di DP rappresentino l'esempio più diretto di questo schema, anche gli approcci successivi come i metodi Monte Carlo e i metodi Temporal-Difference, possono essere interpretati come modalità alternative della realizzazione dello stesso principio generale, con la differenza che in tali algoritmi la stima delle funzioni di valore non avviene più necessariamente in un modello noto dell'ambiente.

3.3. Monte Carlo Methods

I Monte Carlo Methods (MCM) rappresentano uno dei primi approcci sviluppati per affrontare il problema della mancata conoscenza diretta dell'ambiente. In questi casi diventa infatti necessario ricorrere a modelli che permettano all'agente di apprendere direttamente dall'esperienza, senza disporre dell'informazione completa della dinamica dell'ambiente circostante.

Questi modelli non richiedono la conoscenza delle probabilità di transizione tra i vari stati, bensì si basano esclusivamente sull'osservazione delle sequenze di interazione tra l'agente e l'ambiente. In questo senso l'apprendimento avviene utilizzando degli episodi,

ossia delle sequenze di stati, azioni e ricompense che cominciano allo stato iniziale e si concludono in uno stato finale. L'idea fondamentale consiste nello stimare il valore degli stati a partire dai ritorni effettivamente osservati durante gli episodi di interazione con l'ambiente.

Come per gli algoritmi di DP, anche nei MCM è possibile applicare l'idea generale della GPI, considerando quindi separatamente i problemi di valutazione e di miglioramento della politica.

Per stimare la funzione di valore $v_\pi(s)$ viene utilizzata la media dei ritorni osservata ogni volta che, all'interno di un episodio, viene incontrato lo stato s, dove ogni passaggio per lo stato prende il nome di visita. Esistono quindi due metodi per calcolare il ritorno per ciascuno stato: il metodo prima visita (o first-visit method) e il metodo ogni visita (every-visit method). Nel first-visit method la funzione di valore è stimata dalle medie dei ritorni date dalla prima visita allo stato s, mentre nel every-visit method la stima è data dalle medie dei ritorni di tutte le visite allo stato s. In entrambi i casi, si converge a $v_\pi(s)$ se il numero di visite a s è sufficientemente elevato.

Una possibile approssimazione di un algoritmo MCM può essere formalizzata come:

$$V(S_t) \leftarrow V(S_t) + \alpha[G_t - V(S_t)]$$

dove G_t rappresenta il ritorno osservato al tempo t, mentre α, noto come tasso di apprendimento (o learning rate), determina l'intensità dell'aggiornamento, attraverso il controllo del peso attribuito alla nuova informazione rispetto alla stima precedente. In particolare, il parametro $\alpha \in [0,1]$, dove per $\alpha = 0$ la funzione non viene aggiornata, mentre per $\alpha = 1$ la funzione viene completamente rimpiazzata dalla nuova.

In assenza della conoscenza dell'ambiente, risulta più utile stimare le funzioni di azione-valore $q_\pi(s,a)$ piuttosto che la sola funzione di valore. Questo perché, conoscendo come l'ambiente si evolve è sufficiente selezionare l'azione che porta alla maggiore ricompensa nello stato successivo ma, se non si conosce l'ambiente, allora è necessario esplicitare la stima del valore per ogni azione in modo da poter selezionare una politica ottimale.

Il processo per stimare la funzione di azione-valore è identico a quello visto per il calcolo della sola funzione di valore, con l'aggiunta che i ritorni sono calcolati per ogni coppia stato-azione piuttosto che solo per ogni stato. Formalmente:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[G_t - Q(S_t, A_t)]$$

Un aspetto critico dell'applicazione dei MCM alla stima delle funzioni di azione-valore riguarda il problema dell'esplorazione delle possibili azioni percorribili. In particolare, può accadere che molte coppie stato-azione non vengano mai osservate durante il processo di apprendimento. Questo problema emerge soprattutto quando la politica π è deterministica. In tal caso, seguendo sistematicamente tale politica, in ciascuno stato verrebbe eseguita sempre la stessa azione e, di conseguenza, i ritorni osservati riguarderebbero esclusivamente quella specifica azione, senza quindi raccogliere informazioni su possibili azioni migliori nello stesso stato.

Questa problematica rappresenta una criticità significativa, poiché lo scopo dell'apprendimento della funzione $q_\pi(s, a)$ risiede proprio nel confrontare diverse azioni disponibili in uno stato, al fine di individuare la scelta ottimale. Tale problema è strettamente legato al più generale trade-off tra esplorazione e sfruttamento, analizzato anche nei capitoli precedenti; dunque, per permettere una corretta valutazione della politica è necessario garantire che tutte le coppie stato-azione vengano esplorate nel corso dell'apprendimento.

Una possibile soluzione consiste nell'introdurre l'ipotesi dei cosiddetti exploring starts, dove si assume che ogni episodio inizi da una coppia stato-azione selezionata casualmente con probabilità positiva tra tutte le possibili combinazioni. In alternativa è possibile utilizzare una politica ε-greedy, definita come:

- Con probabilità 1- ε: si seleziona l'azione migliore (greedy)
- Con probabilità ε: si seleziona un'azione casuale

Questa ipotesi garantisce che, in un numero infinito di episodi, ogni coppia stato-azione venga visitata un numero infinito di volte, consentendo dunque una stima accurata della funzione $q_\pi(s, a)$.

Una volta stimata la funzione $q_\pi(s, a)$, è possibile migliorare la politica adottando una scelta migliorativa rispetto ai valori stimati. La nuova politica viene infatti definita come:

$$\pi'(s) = \arg\max_a \; q_\pi(s, a)$$

dove anche in questo caso, è garantito che la nuova politica sia almeno migliore della precedente. Il processo viene quindi ripetuto fino a quando la politica converge a quella

ottimale π_* ottenendo, come visto precedentemente, una sequenza monotona di politiche e funzioni di azione-valore migliorate:

$$\pi_0 \xrightarrow{E} q_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} q_{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \dots \xrightarrow{I} \pi_* \xrightarrow{E} q_*$$

Tuttavia, una caratteristica fondamentale dei MCM risiede nel fatto che l'aggiornamento della funzione di azione-valore avviene solo dopo che il ritorno completo G_t è stato osservato, ovvero alla fine di un episodio. Per tale ragione questi metodi risultano particolarmente adatti ai problemi episodici, ossia scenari nei quali l'interazione tra l'agente e l'ambiente termina in uno stato finale ben preciso e identificabile.

La necessità di dover attendere il termine dell'episodio rende i MCM inutilizzabili in presenza di problemi continui, nei quali non è presente uno stato finale. Questo limite ha portato col tempo allo sviluppo dei Temporal-Difference methods, che permettono di aggiornare le stime delle funzioni di valore in maniera incrementale durante l'interazione con l'ambiente, senza dover attendere il termine dell'episodio.

3.4. Temporal Difference Methods

I metodi TD rappresentano la classe centrale degli algoritmi di RL, poiché combinano le caratteristiche dei metodi di DP e MCM. In particolare, essi apprendono direttamente dall'esperienza, senza richiedere la conoscenza dell'ambiente (come MCM), ed allo stesso tempo aggiornano le stime utilizzando altre stime correnti, senza la necessità di attendere la fine di un episodio per effettuare un aggiornamento (come DP).

Anche in questo caso, è possibile seguire il principio generale della GPI e dividere l'apprendimento dell'algoritmo in due fasi: la fase di valutazione della politica e la fase di miglioramento della stessa.

Per quanto riguarda la fase di valutazione, a differenza dei MCM, i metodi TD attendono solamente la fine di un passo temporale, invece che il termine dell'intero episodio, utilizzando informazioni parziali per aggiornare le stime di $V(S_t)$.

Nella sua versione più semplice $TD(0)$, o TD ad un passo, l'aggiornamento di $V(S_t)$ può essere espresso come:

$$V(S_t) \leftarrow V(S_t) + \alpha[R_{t+1} + \gamma V(S_{t+1}) - V(S_t)]$$

dove i parametri α e γ sono rispettivamente i già noti tasso di apprendimento e fattore di sconto, mente il termine tra parentesi quadre rappresenta il cosiddetto errore TD (o TD

error), ossia la differenza tra la stima corrente del valore di S_t e la stima aggiornata dello stesso valore ottenuta sfruttando l'informazione della transizione osservata $R_{t+1} + \gamma V(S_{t+1})$. Formalmente, è possibile riscrivere il TD error come:

$$\delta_t = R_{t+1} + \gamma V(S_{t+1}) - V(S_t)$$

Questa quantità permette all'algoritmo di aggiornare la stima del valore di uno stato sulla base dell'informazione osservata nella transizione successiva.

Inoltre, è dimostrabile che per una politica fissata π, l'algoritmo $TD(0)$ converge alla funzione di valore ottima $v_\pi(s)$. In particolare, se il learning rate α viene mantenuto costante, ma sufficientemente piccolo, la convergenza avviene in media, ossia le stime oscillano attorno al valore corretto. Se invece il learning rate decresce nel tempo secondo le condizioni dell'approssimazione stocastica, ovvero garantendo sia l'apprendimento durante tutto il processo sia il progressivo azzeramento dello stesso, allora la convergenza avviene con probabilità pari ad uno.

Per quanto riguarda la fase di miglioramento della politica, come per i MCM anche in questo caso è necessario basarsi sulla stima della funzione di azione-valore $Q(s, a)$, che consente di confrontare direttamente le azioni disponibili in ciascuno stato. Tuttavia, nei metodi TD, il miglioramento della politica non avviene in modo separato dalla valutazione, ma è integrato e continuo per ogni interazione con l'ambiente. L'algoritmo infatti aggiorna le stime della funzione di azione-valore ad ogni interazione e contestualmente modifica il cambiamento della politica.

Come visto nei MCM, per garantire un corretto processo di apprendimento, è necessario mantenere un adeguato livello di esplorazione. Per questo motivo anche nei metodi TD si utilizzano politiche ε-greedy, che consentono di bilanciare entrambe le fasi di esplorazione e sfruttamento, permettendo all'algoritmo di migliorare valutando tutte le possibili coppie di stato-azione e di non fermarsi ai soli scenari già conosciuti.

Introdotte dunque le diverse modalità di apprendimento, è possibile ora analizzare le tre principali classi di algoritmi elencate all'inizio del capitolo, ossia i metodi: value-based, policy-based ed actor-critic, le quali verranno analizzate singolarmente nei paragrafi successivi.

3.5. Value-Based Methods

I value-based methods costituiscono una classe di algoritmi di RL in cui il problema decisionale viene affrontato attraverso la stima della funzione di valore $v(s)$ o di azione-valore $q(s, a)$. In particolare, tali metodi si basano sull'idea di associare a ciascuno stato, oppure a ciascuna coppia stato-azione, un valore numerico che rappresenti il ritorno atteso di lungo periodo.

Una volta ottenuta una stima accurata delle funzioni di valore o azione-valore, questa può essere utilizzata per guidare l'intero processo decisionale, selezionando per ogni stato le azioni associate a valori più elevati. In questo senso, la funzione di valore rappresenta l'elemento centrale di queste tipologie di algoritmi da cui deriva la denominazione stessa della categoria.

I due approcci fondamentali di questa categoria sono gli algoritmi SARSA e Q-learning, entrambi basati su tecniche di TD. In particolare, i due algoritmi si differenziano per le modalità con le quali viene aggiornata la funzione di azione-valore.

L'approccio SARSA rappresenta un approccio on-policy ovvero dove la politica utilizzata per generare i dati e quella oggetto di apprendimento coincidono. Operativamente, il metodo SARSA apprende quanto buona è una decisione, considerando che l'algoritmo continuerà a comportarsi seguendo la stessa politica.

In merito all'aggiornamento della funzione di azione-valore, questo può essere formalizzato con le stesse modalità utilizzate per il calcolo della funzione di valore viste in precedenza:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

Allo stesso modo, è possibile applicare il teorema di convergenza visto per $TD(0)$ al relativo algoritmo per la funzione di azione-valore.

Per ogni aggiornamento temporale, si configurano una serie di elementi $(S_t, A_t, R_{t+1}, S_{t+1}, A_{t+1})$, che formano la transizione da una coppia azione-stato alla successiva e dai quali deriva il nome di questa tipologia di algoritmi. Ad ogni passo, la funzione di azione-valore $Q(S_t, A_t)$ viene aggiornata in modo da stimare $q_\pi(s, a)$, ossia il valore della politica corrente. Contestualmente, la politica stessa viene definita come ε-greedy rispetto a tali stime, risultando dunque progressivamente più orientata verso le azioni con valore più elevato, formalmente:

$$\pi(S_t) = \arg\max_a \ Q(S_t, A_t)$$

In questo modo, valutazione e miglioramento della politica avvengono simultaneamente: da un lato viene migliorata la funzione di azione-valore, dall'altro la politica viene gradualmente resa più ε-greedy rispetto ad essa.

Questo approccio converge alla politica ottimale π^*, se tutte le coppie stato-azioni vengono visitate un numero infinito di volte.

Dall'altro lato, l'algoritmo Q-learning rappresenta invece un approccio off-policy, in cui la politica seguita per generare i dati può differire da quella che si sta migliorando. Operativamente, il metodo Q-learning impara quanto buona è un'azione considerando che l'algoritmo agirà sempre in modo ottimale.

In questo caso l'aggiornamento della funzione di azione-valore avviene massimizzando direttamente Q. Formalmente:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$$

In questo modo, l'algoritmo apprende direttamente la funzione di azione-valore ottima q^*, indipendentemente dalla politica seguita durante l'esplorazione. È pertanto importante sottolineare che, per agire, l'algoritmo utilizza una politica ε-greedy, in modo da poter visitare tutte le possibili coppie di stato-azione, ma l'aggiornamento della politica avviene in modo greedy, ovvero ricercando esclusivamente la migliore azione possibile al fine di massimizzare Q.

Di conseguenza, ogni aggiornamento non contribuisce solo alla valutazione, ma anche al contestuale miglioramento della politica, fino alla convergenza alla politica ottimale π^*.

Entrambi gli algoritmi rispecchiano a pieno la logica dei value-based method, poiché è proprio la funzione di azione-valore $Q(s, a)$ ad assumere un ruolo centrale nelle logiche di apprendimento e guida del processo decisionale. In questo contesto la politica emerge infatti solamente in modo implicito, attraverso la selezione delle azioni con valore più elevato, ma senza una sua derivazione esplicita.

Tuttavia, nella loro formulazione originaria, i metodi value-based come SARSA e Q-learning, si basano su rappresentazioni tabulari della funzione valore-azione. Questo implica che essi risultano naturalmente adatti a problemi con spazi di stato e azione discreti e di dimensione limitata. In presenza di spazi continui o ad alta dimensionalità,

tali metodi diventano difficilmente applicabili, sia per ragioni computazionali, sia per la necessità di visitare un numero elevatissimo di coppie stato–azione. Tali limitazioni hanno portato allo sviluppo di metodi alternativi, in particolare gli approcci policy-based, che consentono di operare direttamente su spazi di azione continui senza richiedere una discretizzazione esplicita.

3.6. Policy-Based Methods

A differenza dei metodi value-based che, come visto nel paragrafo precedente, istruiscono l'agente in modo indiretto attraverso la stima della funzione di valore (o azione-valore), i metodi policy-based mirano ad ottimizzare direttamente la politica decisionale dell'agente in modo da poter selezionare delle azioni senza dover consultare la funzione di valore.

Formalmente, in questi approcci la politica viene rappresentata come una funzione parametrica:

$$\pi(a|s,\theta)$$

dove θ rappresenta il vettore dei parametri da apprendere.

L'obbiettivo dell'apprendimento consiste dunque nell'individuare il vettore ottimale di θ che massimizza una misura di performance predeterminata. In questo senso, l'ottimizzazione avviene direttamente tra le diverse politiche, senza la necessità di passare attraverso le stime esplicite delle funzioni di valore.

Questo tipo di impostazione rappresenta un cambiamento concettuale rispetto ai metodi precedentemente trattati. Nei metodi value-based, infatti, la politica ottimale emerge implicitamente come una politica greedy rispetto alla funzione di valore stimata; con i metodi policy-based, invece, la politica è l'oggetto primario dell'apprendimento dell'agente e viene aggiornata direttamente sulla base della sua performance.

Nonostante i metodi policy-based comprendano diverse tecniche di ottimizzazione diretta della politica, la classe più rilevante e diffusa in letteratura è rappresentata dai policy gradient methods, ossia algoritmi che aggiornano i parametri della policy $J(\theta)$ seguendo il gradiente[19] di una funzione obbiettivo di performance.

[19] Il gradiente è il vettore delle derivate parziali di una funzione rispetto ai suoi parametri e individua la direzione di massima crescita della funzione stessa.

I metodi policy-based presentano inoltre una serie di vantaggi teorici e pratici rispetto ai metodi value-based, che ne giustificano l'utilizzo in numerosi contesti applicativi più complessi. Tali vantaggi derivano principalmente dalla scelta di parametrizzare direttamente la politica e di ottimizzarla attraverso tecniche di ascesa del gradiente.

Come già visto nel paragrafo precedente, in problemi ad alta dimensionalità e/o strutture decisionali complesse, la funzione di valore può risultare complessa da approssimare rispetto alla politica stessa. In questi casi, parametrizzare direttamente la politica può portare ad una maggiore efficienza dell'apprendimento e a migliori prestazioni del modello. Inoltre, nei metodi value-based la gestione degli spazi di azione continui può diventare intrattabile, mentre una politica parametrica può modellare direttamente una distribuzione continua sulle azioni, rendendo l'ottimizzazione più naturale ed efficiente.

A questi vantaggi si aggiungono: la possibilità di rappresentare politiche stocastiche, ovvero di assegnare, attraverso le politiche, una probabilità a ciascuna azione, invece che sceglierne solo una. In questo modo è possibile apprendere direttamente con quale probabilità scegliere ogni azione, senza dover ricorrere a meccanismi artificiali di esplorazione; la continuità rispetto ai parametri, che permette alla politica di cambiare in modo graduale e continuo al variare dei parametri θ, in modo da evitare che, come avviene invece nei metodi value based, anche una minima variazione possa portare ad un cambiamento improvviso dell'azione scelta; ed infine, la possibilità di incorporare conoscenza a priori nel modello, progettando e scegliendo esplicitamente la funzione parametrica più adatta al problema che si vuole affrontare.

Affinché sia possibile applicare le tecniche di ottimizzazione basate sul gradiente, è necessario che la politica sia differenziabile rispetto ai parametri θ, ovvero che esista il gradiente $\nabla_\theta \pi(a|s,\theta)$. Questa condizione è fondamentale poiché consente di modificare i parametri della politica in modo sistematico per migliorare la performance del modello.

Nel caso di spazi di azione discreti e sufficientemente limitati, una delle parametrizzazioni più comuni si basa sull'utilizzo di preferenze numeriche $h(s|a,\theta)$, associate a ciascuna coppia azione-stato. Tali preferenze vengono poi trasformate in probabilità attraverso una funzione "softmax", che garantisce che le probabilità siano positive e che sommino a uno.

$$\pi(a|s,\theta) = \frac{\exp(h(s,a,\theta))}{\sum_b \exp(h(s,b,\theta))}$$

In questo modo, le azioni con preferenze più elevate vengono selezionate con probabilità maggiore, ma tutte le azioni mantengono probabilità non nulla, favorendo così l'esplorazione.

Per quando riguarda le preferenze $h(s|a,\theta)$, queste possono essere parametrizzate in diversi modi. Una possibile semplificazione è quella di utilizzare una combinazione lineare delle caratteristiche dello stato. In alternativa, è possibile utilizzare modelli più complessi, come le reti neurali.

Da questa parametrizzazione ne risulta che la politica ottenuta è intrinsecamente stocastica, poiché assegna una probabilità a ciascuna azione. Inoltre, la dipendenza continua della policy ai parametri θ garantisce che piccole variazioni nei parametri producano variazioni graduali nella probabilità di selezione delle azioni, facilitando così l'ottimizzazione tramite metodi di ascesa del gradiente.

L'apprendimento della politica è formulato come un problema di ottimizzazione, nel quale si introduce una funzione obbiettivo $J(\theta)$, che misura la qualità della politica parametrizzata $\pi(a|s,\theta)$, e si ricercano i valori dei parametri θ che massimizzano la performance.

Nel caso più semplice di problemi episodici, la funzione obbiettivo può essere definita come il valore atteso del ritorno a partire dallo stato iniziale:

$$J(\theta) = v_{\pi_\theta}(s_0)$$

dove $v_{\pi\theta}(s_0)$ rappresenta il valore dello stato iniziale sotto la politica π_θ.

L'obbiettivo è quindi quello di massimizzare $J(\theta)$ e a tal fine si utilizzano tecniche di ottimizzazione basate sul gradiente, che aggiornato iterativamente i parametri della politica nella direzione di massimo incremento della funzione obbiettivo. In particolare, l'aggiornamento dei parametri segue la regola dell'ascesa del gradiente:

$$\theta_{t+1} = \theta_t + \alpha\widehat{\nabla J(\theta_t)}$$

dove: α rappresenta il tasso di apprendimento, mentre $\widehat{\nabla J(\theta_t)}$ è il gradiente della funzione obbiettivo rispetto ai parametri della politica.

Questo tipo di aggiornamento riflette l'idea che il gradiente $\nabla J(\theta_t)$ fornisca la direzione lungo la quale la funzione $J(\theta)$ cresce più rapidamente. Pertanto, muovendosi in tale direzione, è possibile migliorare progressivamente la qualità della politica.

Tuttavia, come già visto all'interno del contesto del RL, la funzione obbiettivo dipende dalla dinamica dell'ambiente che risulta spesso sconosciuta, di conseguenza il calcolo esatto del gradiente non è generalmente possibile. Per questo motivo, i policy gradient methods utilizzano una stima stocastica del gradiente, ottenuta attraverso l'esperienza dell'agente durante le interazioni con l'ambiente.

Un aspetto critico è che la performance della politica dipende non solo dalle azioni selezionate, ma anche dalla distribuzione degli stati visitati, che a sua volta dipende dalla politica stessa e proprio questa connessione sembrerebbe rendere il problema di ottimizzazione apparentemente molto complesso.

Il policy gradient theorem fornisce però una soluzione a questa difficolta, permettendo di esprimere il gradiente della funzione obbiettivo senza dover derivare esplicitamente la distribuzione degli stati. In particolare, nel caso episodico, il teorema stabilisce che il gradiente può essere espresso come:

$$\nabla J(\theta) \propto \sum_{s} \mu(s) \sum_{a} q_{\pi}(s,a)\, \nabla \pi(a|s,\theta)$$

dove: $\mu(s)$ rappresenta la frequenza con cui gli stati vengono visitati sotto la politica π; $q_{\pi}(s,a)$ rappresenta la funzione valore-azione associata alla politica; $\nabla_{\theta}\pi(a|s,\theta)$ rappresenta il gradiente della probabilità di selezionare l'azione a nello stato s.

Il simbolo di proporzionalità indica che il gradiente è definito a meno di una costante moltiplicativa che, nei i casi episodici, questa è pari alla lunghezza media di un episodio, mentre nei casi continui questa è pari a 1, rendendo la relazione un'eguaglianza.

A partire da questo risultato teorico, è possibile costruire un primo algoritmo di apprendimento. Questo algoritmo prende il nome di REINFORCE[20] ed è stato introdotto nel 1992 da Ronald J. Williams con l'obbiettivo di ottenere una stima campionaria del gradiente attraverso l'utilizzo dell'algoritmo stesso.

[20] Williams, R.J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine Learning, 8(3-4), pp.229–256.

Partendo dal policy gradient theorem, il gradiente della funzione obbiettivo è espresso come una somma su tutti gli stati, pesata dalla distribuzione di visita indotta dalla politica, e per ciascuno stato come una somma su tutte le azioni possibili. Un primo passaggio consiste nel reinterpretare la distribuzione degli stati $\mu(s)$ come una distribuzione di visita di questi indotta dalla politica. Questo permette di riscrivere la somma degli stati, come un valore atteso rispetto alle traiettorie generate seguendo la politica stessa, introducendo quindi la variabile S_t, che rappresenta lo stato visitato al tempo t. Tuttavia, l'espressione ottenuta contiene ancora una somma sulle azioni, che la rende non ancora campionabile. Per questo motivo si inserisce la probabilità che la politica assegni l'azione a nello stato s, moltiplicando e dividendo per $\pi(a|s,\theta)$, introducendo così la variabile A_t, ovvero l'azione selezionata al tempo t secondo la politica. Infine, è necessario sostituire il valore dell'azione $q_\pi(S_t, A_t)$, con una quantità osservabile. A tal fine, si sfrutta la proprietà del ritorno G_t che costituisce una stima campionaria di tale quantità in quanto il suo valore atteso condizionato allo stato e all'azione coincide con il valore dell'azione stessa. In questo modo si ottiene un'espressione del gradiente completamente formulata come valore atteso rispetto alla politica e quindi stimabile direttamente tramite i campioni osservati. Formalmente:

$$\nabla J(\theta) \propto \sum_s \mu(s) \sum_a q_\pi(s,a)\, \nabla_\theta \pi(a|s,\theta) = E_\pi\left[\sum_s q_\pi(S_t,a)\, \nabla\pi(a|S_t,\theta)\right]$$
$$= E_\pi\left[\sum_s q_\pi(S_t,A_t)\frac{\nabla\pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)}\right] = E_\pi\left[G_t\frac{\nabla\pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)}\right]$$

Il risultato fondamentale è che il gradiente della funzione obbiettivo può essere espresso in termini di quantità osservabili (stati, azioni, ritorni) della politica stessa, senza richiedere la conoscenza esplicita della dinamica dell'ambiente o della derivata della distribuzione degli stati. In questo modo è possibile stimare il gradiente $\nabla J(\theta)$ attraverso l'esperienza diretta dell'agente, utilizzando i dati ottenuti durante l'interazione con l'ambiente stesso.

A partire dall'espressione ottenuta, è possibile definire la regola di aggiornamento dei parametri della politica, applicando l'approccio dell'ascesa stocastica del gradiente, ottenendo:

$$\theta_{t+1} = \theta_t + \alpha G_t \frac{\nabla \pi(A_t|S_t,\theta)}{\pi(A_t|S_t,\theta)} = \theta_t + \alpha\, G_t \nabla \ln \pi(A_t|S_t,\theta)$$[21]

dove: α rappresenta il tasso di apprendimento e G_t il ritorno osservato a partire dall'istante t.

L'interpretazione dell'aggiornamento è dunque la seguente: il termine $\nabla \ln \pi(A_t|S_t,\theta)$ indica come modificare i parametri della policy al fine di aumentare la probabilità di selezionare l'azione A_t nello stato S_t, mentre il fattore G_t modula l'intensità dell'aggiornamento in funzione della qualità del risultato ottenuto. In questo modo, le azioni che portano ritorni elevati vengono progressivamente favorite, mentre quelle associate a ritorni inferiori tendono a essere penalizzate.

Infine, è importante osservare che l'algoritmo utilizza il ritorno completo G_t, che include tutte le ricompense future fino al termine dell'episodio. Per questo motivo, REINFORCE appartiene alla classe dei metodi Monte Carlo ed è naturalmente definito per contesti episodici.

Nonostante, la sua correttezza a livello teorico, l'utilizzo di stime Monte Carlo per il calcolo del valore dell'azione e la necessità di dover aspettare il ritorno dell'intero episodio per l'aggiornamento comportano lentezza nell'apprendimento del modello e ampie oscillazioni, rendendo difficile la convergenza della politica verso una soluzione ottimale. Infatti, a parità di stato e azione, il valore G_t può variare significativamente tra diverse traiettorie, producendo aggiornamenti del gradiente instabili e spesso poco coerenti tra loro.

Per affrontare questo limite è possibile introdurre un termine di confronto, chiamato "baseline", che consente di ridurre la variabilità della stima del gradiente, senza alterarne il valore atteso. L'idea alla base consiste nel sottrarre al ritorno G_t una quantità che dipende dallo stato, ottenendo dunque una misura relativa della qualità dell'azione selezionata.

In particolare, si può dimostrare che il policy gradient theorem rimane valido anche introducendo una funzione arbitraria $b(s)$, purché essa non dipenda dall'azione. Questo passaggio consente di definire una nuova regola di aggiornamento:

[21] Tale espressione deriva dall'equivalenza tra la forma frazionaria del gradiente e il gradiente logaritmico, che consente una rappresentazione più compatta e computazionalmente efficiente.

$$\theta_{t+1} = \theta_t + \alpha\,(G_t - b(S_t))\nabla \ln \pi(A_t|S_t, \theta)$$

L'introduzione della baseline non altera il valore atteso del gradiente, ma può ridurne significativamente la varianza. Questo perché il termine sottratto (baseline) agisce come riferimento rispetto al quale valutare la qualità dell'azione, anziché considerare il ritorno assoluto.

Un valore tipicamente utilizzato per la baseline è la stima del valore dello stato $V(S_t)$, poiché il termine $G_t - V(S_t)$ misura proprio il vantaggio dell'azione selezionata rispetto al valore atteso dello stato. Questa quantità prende il nome di vantaggio (o advantage) e consente di discriminare le azioni tra migliori o peggiori rispetto alla media dello stato. In questo modo se l'azione è superiore alla media viene rafforzata, se al contrario è inferiore alla media viene penalizzata.

L'introduzione di questo concetto evidenzia dunque la necessità di affiancare alla stima della politica, un ulteriore modello dedicato alla determinazione di una baseline adeguata, al fine di ridurre la varianza dell'aggiornamento. Questo passaggio ha portato allo sviluppo dei metodi actor-critic, nei quali la politica (actor) viene affiancata da una funzione di valore (critic) appresa esplicitamente e utilizzata per fornire una stima del baseline.

3.7. Metodi Actor-Critic

I metodi actor-critic rappresentano una classe di algoritmi nei quali coesistono due agenti: il primo ha il compito di apprendere la politica; mentre il secondo ha il compito di apprendere una funzione di valore, utilizzata per valutare gli stati e fornire un segnale di feedback all'actor, in modo da guidarne l'aggiornamento.

La differenza rispetto ai metodi policy-based con baseline risiede proprio in questo passaggio. Mentre in questi ultimi la funzione di valore viene utilizzata esclusivamente come termine di correzione, nei metodi actor-critic questa è utilizzata attivamente per guidare l'aggiornamento della politica.

Si introduce innanzitutto una stima incrementale del ritorno basata su un singolo passo, attraverso il calcolo del valore allo stato successivo e del cosiddetto one-step return:

$$G_{t:t+1} = R_{t+1} + \gamma \hat{v}(S_{t+1}, w)$$

Questa quantità viene quindi confrontata con la stima corrente $\hat{v}(S_t)$, dando origine al temporal-difference error:

$$\delta_t = R_{t+1} + \gamma \hat{v}(S_{t+1}, w) - \hat{v}(S_t, w)$$

In questo modo si introduce una stima del ritorno reale, che presenta una varianza inferiore rispetto al ritorno completo e che rappresenta una misura efficace per valutare l'azione intrapresa.

L'aggiornamento della politica avviene dunque sostituendo il ritorno completo, utilizzato nei metodi Monte carlo, con il TD error δ_t, ottenendo una regola di aggiornamento del tipo:

$$\theta_{t+1} = \theta_t + \alpha\, \delta_t\, \nabla \ln \pi(A_t|S_t, \theta)$$

Parallelamente, il critic aggiorna i propri parametri utilizzando un metodo di tipo TD(0), sfruttando lo stesso TD error.

È eventualmente possibile estendere l'utilizzo delle stime del ritorno a più passi temporali, definendo un n-step return come:

$$G_{t:t+n} = R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \hat{v}(S_{t+n}, w)$$

In questo modo l'aggiornamento della politica può essere sostituito con la quantità $G_{t:t+n} - \hat{v}(S_{t+n}, w)$, lasciando inalterata la struttura generale dell'algoritmo.

La caratteristica fondamentale che ne deriva per questi algoritmi è che sono in grado di aggiornare i parametri ad ogni passo, senza la necessità di attendere il termine di un episodio.

Questa caratteristica permette di risolvere il problema riscontrato nei paragrafi precedenti della dipendenza a contesti episodici, nei quali l'interazione tra l'agente e l'ambiente è suddivisa in traiettorie finite e la performance della politica è definita in funzione del valore dello stato iniziale $J(\theta) = v_{\pi_\theta}(s_0)$. Questa impostazione risulta infatti limitante in numerosi contesti applicativi, nei quali non è possibile individuare una naturale suddivisione degli eventi in episodi. In questi casi, l'interazione con l'ambiente si sviluppa lungo orizzonti temporali indefiniti ed è richiesto all'agente di operare in maniera continuativa.

Per affrontare questa tipologia di problemi è necessaria quindi una nuova definizione della funzione obbiettivo, nella quale si considera la ricompensa media per unità di tempo:

$$J(\theta) = \bar{r}_\pi = \lim_{h\to\infty} \frac{1}{h} \sum_{t=1}^{h} E[R_t \,|S_0, A_{0:t-1}\sim\pi]$$

In questo modo è possibile valutare la qualità della politica in termini di rendimento nel lungo periodo, ottenendo una misura più adatta a contesti nei quali l'obbiettivo dell'agente è quello di massimizzare una performance sostenuta nel tempo, piuttosto che il risultato di una singola traiettoria.

Tale riformulazione non incide sulla validità del policy gradient theorem, ma richiede un adattamento dei segnali utilizzati per stimare il gradiente. In questo contesto, i metodi actor–critic risultano particolarmente efficaci, in quanto consentono di aggiornare le stime ad ogni passo temporale, senza la necessità di attendere la conclusione di un episodio.

Infine, un ulteriore elemento di rilevanza dei policy gradient methods riguarda la loro capacità di operare in spazi di azione continui. Nelle formulazioni precedenti dei value-based methods, le azioni venivano considerate appartenenti a insiemi discreti, tuttavia, in numerosi contesti operativi, le decisioni prese dall'agente devono poter assumere valori continui.

Questo limite viene superato attraverso una parametrizzazione diretta della politica. In particolare, la politica $\pi(a|s,\theta)$, può essere definita come una distribuzione di probabilità continua sulle azioni, rendendo possibili l'apprendimento di strategie in spazi continui.

Una formulazione comunemente adottata consiste nel modellare la politica come una distribuzione Gaussiana, in cui i parametri dipendono dallo stato osservato.

$$a\sim N\left(\mu_\theta(s), \sigma_\theta^2(s)\right)$$

dove: $\mu_\theta(s)$ rappresenta la media della distribuzione, ossia l'azione più probabile, mentre σ_θ ne descrive la deviazione standard, permettendo di gestire il livello di esplorazione. Entrambi questi parametri dipendono dallo stato s e sono funzione del vettore dei parametri θ.

In questo modo l'agente non seleziona direttamente un'azione deterministica, ma campiona l'azione da una distribuzione parametrizzata:

$$\pi(a|s,\theta) = \frac{1}{\sigma(s,\theta)\sqrt{2\pi}} exp\left(-\frac{\left(a-\mu(s,\theta)\right)^2}{2\sigma(s,\theta)^2}\right)$$

Questo approccio consente di trattare spazi di azione infiniti e di incorporare direttamente nella politica il livello di esplorazione.

CAPITOLO 4 - PROXIMAL POLICY OPTIMIZATION

4.1. Instabilità degli aggiornamenti e TPRO

Nonostante i metodi policy gradient rappresentino un approccio efficace per l'ottimizzazione diretta della politica, essi presentano una principale criticità legata alla stabilità degli aggiornamenti. In particolare, sebbene il gradiente della funzione obbiettivo individui la direzione di miglioramento della politica, questo non fornisce alcuna indicazione sulla dimensione dell'aggiornamento, con il rischio che variazioni troppo ampie dei parametri conducano ad un peggioramento progressivo della performance.

Questa problematica emerge chiaramente nel lavoro di Schulman et al. (2017)[22], che per affrontare tale problematica introducono il concetto di Trusted Region Policy Optimization (TRPO), basato sull'idea di limitare l'aggiornamento della policy imponendo un vincolo sulla distanza tra la politica corrente e quella aggiornata.

L'ottimizzazione della politica viene quindi formulata come la massimizzazione di una funzione obbiettivo surrogata, ottenuta dal confronto tra la differenza di performance tra una politica nuova $\tilde{\pi}$ e una politica vecchia π. È possibile, infatti, definire la performance di una politica come:

$$J(\theta) = E\left[\sum_{t=0}^{\infty} \gamma^t r(s_t)\right]$$

E dunque la differenza di performance tra le politiche risulta:

$$J_{\tilde{\pi}}(\theta) = J_{\pi}(\theta_{old}) + \sum_{s} \mu_{\tilde{\pi}}(s) \sum_{a} \tilde{\pi}(a|s) A^{\pi}(s,a)$$

dove: $J(\theta)$ rappresenta la performance della politica, $\mu_{\tilde{\pi}}(s)$ rappresenta la frequenza con cui lo stato viene visitato sotto la nuova politica, $\tilde{\pi}(a|s)$ è la nuova politica e $A^{\pi}(s,a)$ è il vantaggio della vecchia politica.

Ma la dipendenza della formula alla distribuzione degli stati sotto la nuova politica $\mu_{\tilde{\pi}}(s)$, rende complessa l'ottimizzazione diretta della formula. Per questo motivo si introduce

[22] Schulman, J., Levine, S., Moritz, P., Jordan, M. and Abbeel, P. (2017). Trust Region Policy Optimization.

un'approssimazione locale della performance, ipotizzando che se la nuova politica è molto simile alla vecchia, allora anche la distribuzione degli stati visitati non subisce particolari variazioni:

$$L_\pi(\theta) = J_\pi(\theta_{old}) + \sum_s \mu_\pi(s) \sum_a \tilde{\pi}(a|s) A^\pi(s,a)$$ [23]

Il problema principale della funzione surrogata è che essa rappresenta solo un'approssimazione locale della vera performance. Di conseguenza, se l'aggiornamento della politica è troppo ampio, questa approssimazione può diventare inaccurata. Per questo motivo si introduce un vincolo nella massimizzazione che limita quanto la nuova politica può discostarsi da quella corrente:

$$\max_\theta L_\pi(\theta) \; sotto \; vincolo: \overline{D}_{KL}^{p\theta_{old}}(\theta_{old}, \theta) \leq \delta$$

dove $\overline{D}_{KL}^{p\theta_{old}}(\theta_{old}, \theta)$ indica la divergenza di Kakade-Langford media tra la politica vecchia e quella nuova. Questa grandezza misura quanto le due distribuzioni di probabilità divergono tra loro, ovvero quanto cambia il comportamento delle due politiche. Il termine δ rappresenta invece il limite di variazione massima consentito, che prende anche il nome di "trust region". Questa misura è una soglia positiva, scelta dall'algoritmo, che impone che la nuova politica non si discosti troppo dalla vecchia, in modo da evitare che variazioni eccessive rendano inaffidabile l'approssimazione locale. In particolare, se δ è piccolo, l'aggiornamento risulterà più prudente; viceversa, se δ è grande, l'aggiornamento risulterà più aggressivo.

Questo approccio consente di ottenere proprietà tecniche rilevanti, tra tutte il miglioramento monotono della performance della policy. Tuttavia, sebbene il TRPO rappresenti un significativo, avanzamento in termini di stabilità e affidabilità, esso risulta computazionalmente complesso e richiede l'utilizzo di tecniche e procedure di ottimizzazione avanzate e difficili da implementare in contesti pratici.

Alla luce di tali limitazioni, emerge la necessità di sviluppare metodi che mantengano i benefici della restrizione sugli aggiornamenti della politica, ma che risultino al contempo più semplici ed efficienti dal punto di vista computazionale. In questa direzione si colloca la Proximal Policy Optimization (PPO), che può essere interpretata come una

[23] Formula originale del paper: $L_\pi(\tilde{\pi}) = \eta(\pi) + \sum_s p_\pi(s) \sum_a \tilde{\pi}(a|s) A^\pi(s,a)$

riformulazione semplificata del principio di trust region, ottenuta sostituendo il vincolo esplicito sulla divergenza tra politiche con una funzione obiettivo modificata che penalizza implicitamente aggiornamenti troppo ampi.

4.2. Introduzione tecnica alla PPO

L'idea centrale degli algoritmi di PPO consiste nel riformulare la funzione obbiettivo dei policy gradient methods, con l'obbiettivo di penalizzare variazioni eccessive tra la politica corrente e quella aggiornata. In particolare, come proposto da Schulman et al., (2017)[24], tale approccio si basa sull'introduzione di una funzione obbiettivo surrogata, modificata rispetto a quella vista nel TRPO, progettata per limitare l'entità degli aggiornamenti della politica senza ricorre all'utilizzo di vincoli espliciti.

A tal fine, si introduce il rapporto di probabilità tra la nuova politica e quella precedente, formalmente:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$$

Questo termine misura quanto la probabilità di selezionare una determinata azione a_t nello stato s_t cambia a seguito dell'aggiornamento della politica. Quando $r_t(\theta) = 1$, la nuova politica coincide con quella precedente, mentre valori diversi da 1 indicano variazioni nel comportamento dell'agente.

Attraverso questo nuovo rapporto, è possibile riscrivere la funzione obbiettivo degli algoritmi policy gradient come:

$$L(\theta) = \hat{E}_t\left[r_t(\theta)\hat{A}_t\right]$$

Questa formulazione coincide con la funzione obbiettivo surrogata, già introdotta con i metodi di TPRO, e costituisce una stima della variazione di performance tra la politica aggiornata e quella precedente.

Tuttavia, anche in questo caso l'ottimizzazione diretta della funzione può condurre ad aggiornamenti troppo ampi, analogamente a quanto osservato con i metodi di policy gradient standard. Per questo motivo, la PPO introduce una modifica fondamentale basata

[24] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Openai, O. (2017b). Proximal Policy Optimization Algorithms.

su un'operazione di clipping (o limitazione) del rapporto di probabilità, definendo la seguente funzione obbiettivo:

$$L^{CLIP}(\theta) = \hat{E}_t\left[\min(r_t(\theta)\hat{A}_t\,, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

dove ϵ è un iperparametro che definisce l'ampiezza massima consentita per la variazione del rapporto di probabilità.

L'intuizione alla base di questa nuova formulazione è che, per valori di $r_t(\theta)$ compresi nell'intervallo $[1-\epsilon, 1+\epsilon]$, la funzione obbiettivo coincide con quella standard; qualora invece il rapporto di probabilità esca dall'intervallo, e quindi si discosti eccessivamente da 1, il termine di clipping limita il contributo dell'aggiornamento, impedendo variazioni troppo ampie della politica.

Questo approccio consente dunque di ottenere un comportamento analogo a quello dei metodi di TPRO, ma con una formulazione computazionalmente più semplice e facilmente ottimizzabile.

4.3. La funzione obbiettivo completa nella PPO

Nella formulazione pratica dell'algoritmo, la funzione obbiettivo non si limita al solo termine di clipping, ma viene estesa per includere altre componenti che ne migliorano la stabilità e l'efficienza dell'apprendimento. In particolare, la funzione obbiettivo completa è definita come una combinazione di tre termini: la funzione clippata, la funzione di valore e un termine di entropia. Formalmente:

$$L^{CLIP+VF+S}(\theta) = \hat{E}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)]$$

dove:

- $L_t^{CLIP}(\theta)$ rappresenta il termine principale, basato sul clipping del rapporto di probabilità;
- $L_t^{VF}(\theta)$ è un termine di errore quadratico associato alla funzione di valore;
- $S[\pi_\theta](s_t)$ rappresenta l'entropia della politica;
- c_1 e c_2 sono dei coefficienti che bilanciano il contributo dei diversi termini

Il primo termine, già introdotto in precedenza, costituisce il cuore dell'algoritmo e consente di controllare implicitamente l'ampiezza degli aggiornamenti della politica. Il secondo termine, relativo alla funzione di valore, è definito come un errore quadratico tra la stima della funzione di valore $V_\theta(s_t)$ e un target V_t^{targ}, e contribuisce a rendere più

accurata la stima del vantaggio utilizzata nell'aggiornamento della politica. Infine, il termine di entropia ha la funzione di favorire l'esplorazione, evitando che la politica converga troppo velocemente. Una maggiore entropia corrisponde infatti ad una maggiore incertezza nella selezione delle azioni, incentivando l'agente ad esplorare strategie diverse durante il processo di apprendimento.

L'integrazione di questi tre elementi consente alla PPO di mantenere un equilibrio tra stabilità degli aggiornamenti, accuratezza delle stime e capacità di esplorazione, rendendo la PPO particolarmente adatto a contesti adattivi complessi.

4.4. Ciclo di apprendimento della PPO

Dal punto di vista operativo, la PPO si basa su un processo iterativo che alterna fasi di raccolta dati e fasi di ottimizzazione della politica. In particolare, ad ogni iterazione, l'agente interagisce con l'ambiente seguendo la politica corrente $\pi_{\theta old}$, generando una sequenza di traiettorie dalle quali vengono calcolate le stime del vantaggio $\hat{A}_t$.

Una volta raccolti i dati, la PPO non si limita ad un singolo aggiornamento, ma esegue più epoche di ottimizzazione sugli stessi campioni, suddividendoli in minibatch. In ciascuna epoca, i parametri della politica vengono aggiornati, massimizzano la funzione obbiettivo con clipping, mantenendo fisso il riferimento alla politica precedente.

Al termine del processo di ottimizzazione, i parametri aggiornati θ sostituiscono quelli precedenti, costituendo la nuova politica di riferimento per l'iterazione successiva. Il ciclo dunque si ripete, alternando nuovamente raccolta dati e aggiornamento dei parametri, consentendo un miglioramento progressivo della performance dell'agente.

CAPITOLO 5 - APPLICAZIONE EMPIRICA

Avendo analizzato nei capitoli precedenti le basi finanziarie legate alle problematiche della costruzione di portafoglio e i principali elementi teorici del reinforcement learing, il presente capitolo ha l'obbiettivo di unire tali prospettive attraverso lo sviluppo di un modello sperimentale basato sulla Proximal Polixy Optimization.

La scelta di usare un algoritmo di PPO è motivata dalla sua capacità di aggiornamento della politica in modo progressivo, evitando modifiche troppo ampie tra una versione della politica e quella successiva. Proprio questa caratteristica è particolarmente adatta ai mercati finanziari, dove la natura instabile e il rumore dei dati potrebbe portare il modello ad aggiornamenti eccessi della politica, peggiorandone i risultati nel lungo termine.

L'idea fondamentale alla base dell'esperimento è valutare se un agente artificiale, libero da limiti cognitivi e operativi tipici del decisore umano e supportato dalla capacità computazionale moderna, possa adattare dinamicamente e con profitto la composizione di un portafoglio alle condizioni di mercato. Tale impostazione si collega alla visione dei mercati finanziari come sistemi complessi ed in continua evoluzione, nei quali una strategia flessibile potrebbe risultare più competitiva rispetto a regole allocative più statiche e tradizionali.

Il presente capitolo tratterà l'introduzione di un modello baseline che consentirà di ottenere un ambiente sperimentale controllato, interpretabile e coerente con il problema finanziario analizzato. A questo verranno poi collegate architetture più complesse e fonti informative aggiuntive con lo scopo di ottenere un modello il più completo possibile. Questa formulazione assume dunque un duplice ruolo all'interno dell'elaborato. Da un lato, costituisce una prima verifica empirica della capacità della PPO di affrontare un problema di gestione di portafoglio multi-asset. Dall'altro, rappresenta un termine di confronto per le successive integrazioni del modello.

5.1. Obiettivo del modello baseline PPO-MLP

Il modello baseline utilizza un agente PPO basato su una rete neurale di tipo Multi-Layer Perceptron, utilizzata per approssimare la politica dell'agente e la sua funzione di valore. In questo modo è possibile mantenere un'architettura relativamente semplice e facilmente

interpretabile, evitando che la complessità del modello renda difficile la comprensione del contributo dell'algoritmo da quello di componenti più sofisticate.

Dal punto di vista finanziario, il problema viene formulato come una sequenza di decisioni di investimento. A ogni data di ribilanciamento, l'agente osserva una serie di variabile descritte dallo stato del mercato e dalla composizione corrente del portafoglio, per poi scegliere una nuova allocazione tra gli strumenti disponibili.

In questo scenario l'agente ha come obbiettivo la massimizzazione di una funzione di ricompensa, che però non considera solamente il rendimento di portafoglio, ma anche altri elementi legati al rischio e ai costi di ribilanciamento del portafoglio stesso. In questo modo, il problema non viene ridotto alla semplice ricerca del massimo rendimento, ma si avvicina ad una logica più coerente con la gestione finanziaria, nella quale più elementi devono essere valutati congiuntamente.

5.2. Dataset e universo investibile

Prima di trattare le componenti interne che determinano il comportamento dell'agente è indispensabile definire l'universo investibile all'interno del quale l'agente potrà operare.

Per questo elaborato si è voluto scegliere un insieme di Exchange Traded Funds (ETF), poiché gli stessi rappresentano strumenti generalmente liquidi, facilmente negoziabili e in grado di offrire esposizioni immediate verso diverse asset class, aree geografiche e fattori di rischio. Proprio questa caratteristica li rende particolarmente adatti ad un contesto di costruzione di portafoglio, in quanto offrono la possibilità di utilizzare strumenti già diversificati senza dover selezionare singoli titoli, permettendo dunque di costruire portafogli diversificati anche disponendo di pochi asset.

L'universo considerato è composto da dodici ETF:

Ticker	Esposizione principale	Possibile ruolo nel portafoglio
SPY	S&P500 / Azionario USA large cap	Rappresenta il mercato azionario statunitense nel suo complesso
QQQ	Nasdaq 100 / tecnologia USA	Esposizione alla componente tecnologia
IWM	Small cap USA	Esposizione al fattore dimensionale e alle imprese a minore capitalizzazione

EFA	Mercati sviluppati ex-USA	Diversificazione geografica verso economie sviluppate non statunitensi
EEM	Mercati emergenti	Esposizione ad aree geografiche con maggiore potenziale di crescita e rischio
TLT	Treasury USA a lunga scadenza	Componente obbligazionaria e potenzialmente difensiva
GLD	Oro	Bene rifugio e diversificazione rispetto agli asset finanziari tradizionali
DBC	Commodities	Esposizione a materie prime e dinamiche inflazionistiche
XLE	Settore energetico USA	Esposizione settoriale a energia
UUP	Dollaro USA	Esposizione valutaria e possibile componente difensiva
VNQ	Real estate USA	Esposizione al settore immobiliare
MTUM	Azionario momentum USA	Esposizione al fattore momentum[25]

Tabella 5.1: composizione dell'universo investibile

La composizione dell'universo investibile è stata definita in modo da includere asset class caratterizzate da differenti profili di rischio, rendimenti e correlazioni, così da consentire all'agente di individuare una politica di allocazione dinamica tra strumenti eterogenei in funzione delle condizioni di mercato.

La componente azionaria comprende: SPY, che replica l'andamento del mercato azionario statunitense a grande capitalizzazione; QQQ, maggiormente esposto ai titoli tecnologici; XLE, che investe nelle principali società del settore energetico statunitensi; IWM, rappresentativo delle società statunitensi a bassa capitalizzazione; EFA, che fa riferimento ai mercati sviluppati al di fuori degli Stati Uniti; ed EEM, che fornisce esposizione verso i mercati emergenti. A questi si aggiunge MTUM, che consente un'esposizione al fattore momentum del mercato azionario statunitense. Tali strumenti costituiscono dunque la componente maggiormente orientata alla crescita di portafoglio, che dovrebbe consentire al modello di beneficiare di eventuali fasi espansive del mercato, diversificando allo stesso tempo l'esposizione per area geografica, settore e dimensione delle imprese.

[25] Il fattore momentum identifica la tendenza dei titoli che hanno registrato performance relativamente elevate nel recente passato a mantenere, almeno nel breve-medio periodo, una dinamica positiva.

La componente obbligazionaria è rappresentata da TLT, ossia un ETF composto da titoli di stato statunitensi a lunga scadenza. Tale strumento può svolgere una funzione di diversificazione rispetto al rischio azionario, offrendo maggiore stabilità in fasi di mercato particolarmente avverse al rischio.

L'esposizione alle materie prime è ottenuta attraverso GLD e DBC. GLD replica l'andamento dell'oro, asset tipicamente utilizzato come strumento di diversificazione e, in determinate condizioni di mercato, come bene rifugio, mentre DBC offre un'esposizione più ampia verso un paniere di commodities, consentendo al portafoglio di beneficiare anche di dinamiche legate ai prezzi delle materie prime, introducendo una fonte di potenziale rendimento differente rispetto a quella azionaria ed obbligazionaria.

La componente valutaria è rappresentata da UUP, un ETF che permette di assumere esposizioni verso l'andamento del dollaro statunitense rispetto ad un paniere di principali valute internazionali. Tale strumento introduce una fonte di rendimento legata alle dinamiche di tassi di cambio, e può contribuire alla diversificazione complessiva del modello, soprattutto nelle fasi caratterizzate da un rafforzamento della valuta statunitense.

Infine, VNQ permette di includere un'esposizione al settore immobiliare statunitense attraverso società immobiliari dove, seppur mantenendo una natura azionaria, questa componente potrebbe riflettere delle fasi economiche legate a specifiche condizioni del mercato immobiliare, come flussi degli affitti e tassi di interesse, contribuendo dunque ad ampliare le fonti di rischio e i rendimenti disponibili per il modello.

I dati utilizzati sono a frequenza giornaliera, poiché rispetto a dati settimanali o mensili, consentono di aumentare il numero di osservazioni disponibili per l'addestramento dell'agente. Tuttavia, il modello non effettua ribilanciamenti del portafoglio giornalieri ma le decisioni di allocazione vengono aggiornate ogni cinque sedute di mercato, approssimando quindi una frequenza di ribilanciamento settimanale. Questa scelta riduce l'eccessiva movimentazione del portafoglio, rendendo il modello più coerente con una logica di gestione effettivamente implementabile anche per investitori retail.

Il periodo campionario viene suddiviso in tre sottoinsiemi distinti: training set, validation set e test set. Il training set viene utilizzato per l'apprendimento della politica da parte dell'agente; il validation set consente di selezionare la migliore configurazione tra i diversi addestramenti effettuati; infine, il test set viene impiegato per valutare la performance del modello out-of-sample.

All’interno del modello la suddivisione temporale è la seguente:

- Training set: 2015-2020
- Validation set: 2021-2022
- Test set: 2023-2026 (ultimo dato disponibile)

5.3. Costruzione dello stato

Nel contesto del RL, la definizione dello stato rappresenta una delle scelte più rilevanti all’interno dell’intero modello sperimentale. Come visto precedentemente lo stato costituisce l’insieme delle informazioni osservate dall’agente al momento della decisione e, di conseguenza, ne determina la base informativa sulla quale viene costruita la politica di allocazione dell’agente.

All’interno del modello lo stato è costruito utilizzando esclusivamente informazioni di mercato relative agli ETF inclusi nell’universo investibile. La scelta di non inserire in questa prima configurazione variabili esterne è dovuta all’obbiettivo di valutare se l’agente sia in grado di apprendere una politica di allocazione dinamica partendo da segnali quantitativi direttamente ricavabili dalle serie storiche dei dati di mercato.

Per ciascun ETF vengono calcolate diverse variabili, finalizzate a cogliere aspetti differenti del comportamento finanziario del singolo asset. È bene precisare che tali variabili non hanno lo scopo di prevedere deterministicamente il rendimento futuro, ma di fornire all’agente una sintesi delle condizioni di mercato, sulla base della quale aggiornare dinamicamente la struttura del portafoglio. La struttura dello stato può essere sintetizzata come segue:

Componente dello stato	**Descrizione**
Rendimento giornaliero	Variazione percentuale del prezzo nell’ultima seduta per ciascun ETF
Volatilità giornaliera	Escursione tra massimo e minimo giornaliero per ciascun ETF
Volume	Quantità scambiata sul mercato per ciascun ETF
Momentum a 21 sedute	Performance cumulata recente per ciascun ETF
Prezzo/media mobile a 21 sedute	Posizione del prezzo rispetto alla media recente per ciascun ETF
Drawdown rolling a 21 sedute	Perdita dal massimo recente per ciascun ETF

Correlazioni rolling a 21 sedute	Relazioni dinamiche tra coppie di ETF
Pesi correnti	Allocazione detenuta prima del ribilanciamento

Tabella 5.2: composizione dello stato

Una prima componente dello stato è rappresentata dal rendimento giornaliero degli ETF. Questa variabile consente all'agente di osservare la variazione più recente del prezzo e fornisce una misura immediata della direzione assunta da ciascun asset nell'ultima seduta.

$$r_{i,t} = \frac{P_{i,t}}{P_{i,t-1}} - 1$$

Accanto al rendimento viene inclusa una misura di volatilità giornaliera, calcolata dalla variazione tra il prezzo massimo e il prezzo minimo della seduta. La volatilità rappresenta un'approssimazione del rischio recente dell'asset e consente all'agente di distinguere strumenti che, a parità di rendimento, presentano livelli di incertezza diversi.

$$vol_{i,t} = \frac{H_{i,t} - L_{i,t}}{P_{i,t}}$$

Una terza componente è costituita dal volume di scambio. L'inclusione del volume consente di arricchire lo stato con una indicazione dell'intensità dell'attività del mercato sul singolo ETF. Il volume può infatti riassumere variazioni nell'interesse degli investitori, nella liquidità dello strumento o nella forza di un determinato movimento di prezzo.

Lo stato include inoltre una misura di momentum, calcolata su una finestra di 21 sedute, corrispondente a circa un mese di mercato. L'utilizzo del momentum è coerente con l'idea che i mercati possano presentare fenomeni di persistenza dei rendimenti su alcuni orizzonti temporali, che l'agente potrebbe sfruttare con una allocazione dinamica.

$$mom_{i,t}^{(21)} = \frac{P_{i,t}}{P_{i,t-21}} - 1$$

Un'ulteriore informazione è data dal rapporto tra prezzo corrente e la relativa media mobile a 21 sedute. Questa variabile consente di valutare se un ETF si trovi al di sopra o al di sotto della propria media recente, offrendo una rappresentazione sintetica rispetto alla tendenza di breve periodo.

$$ma_ratio_{i,t}^{(21)} = \frac{P_{i,t}}{MA_{i,t}^{(21)}} - 1 \text{ ; dove: } MA_{i,t}^{(21)} = \frac{1}{21}\sum_{k=0}^{20} P_{i,t-k}$$

Infine, per ciascun ETF viene calcolato il drawdown rolling sempre su una finestra di 21 sedute. Tale variabile misura la perdita rispetto al massimo recente e fornisce all'agente un'informazione sul rischio di ribasso sperimentato dall'asset nel periodo immediatamente recente e sulle possibili inversioni di trend.

$$dd_{i,t}^{(21)} = \frac{P_{i,t}}{\max\left(P_{i,t-20}, P_{i,t-19}, \dots, P_{i,t}\right)} - 1$$

Oltre alle variabili calcolate sui singoli ETF lo stato include anche le correlazioni rolling a 21 sedute tra i diversi strumenti dell'universo investibile. Questa scelta è legata alla teoria della costruzione di portafoglio, che non prevede solamente la selezione degli asset considerati individualmente migliori, ma anche alla loro combinazione con gli altri strumenti, al fine di ottenere rendimenti elevati e rischi contenuti sfruttando gli effetti di diversificazione. Fornire all'agente questa informazione permette di avvicinare maggiormente lo stato osservato alla logica effettiva della costruzione di portafoglio.

$$\rho_{ij,t}^{(21)} = \frac{cov(r_{i,t-21:t}, r_{j,t-20:t})}{\sigma_{i,t}^{(21)} \sigma_{j,t}^{(21)}}$$

Un'ultima componente dello stato è costituita dai pesi correnti del portafoglio. Questa informazione è essenziale perché la decisione ottimale non dipende solamente dalle condizioni di mercato ma anche dalla posizione di partenza dell'agente. Inoltre, poiché la funzione di ricompensa penalizza il turnover, l'agente deve conoscere la composizione corrente del portafoglio per valutare implicitamente il costo associato ad una modifica dell'allocazione.

Prima di essere utilizzate dall'agente, le variabili vengono standardizzate per evitare che le diverse scale incluse nello stato possano dominare artificialmente il processo di apprendimento dell'agente.

5.4. Struttura dell'ambiente e azioni

Una volta definito l'universo investibile e lo stato del modello è necessario formalizzare l'ambiente con il quale l'agente interagirà. Ad ogni data di ribilanciamento l'agente osserva lo stato corrente, sceglie una nuova allocazione tra gli ETF disponibili e riceve una ricompensa in funzione della performance successiva del portafoglio.

All'interno del modello, ogni step dell'ambiente corrisponde ad un intervallo di cinque sedute di mercato, approssimando quindi una frequenza di ribilanciamento settimanale.

Questa scelta, come descritto in precedenza, consente di utilizzare la granularità informativa dei dati giornalieri, evitando tuttavia che il portafoglio venga modificato quotidianamente, con il rischio di generare turnover eccessivi e poco realistici dal punto di vista operativo.

Formalmente, il problema può essere descritto come un processo decisionale sequenziale in cui, ad ogni istante decisionale t, l'agente osserva uno stato s_t, seleziona un'azione a_t, ottiene una ricompensa R_t e passa allo stato successivo s_{t+1}. Nel presente modello lo stato sintetizza le condizioni recenti dei mercati, attraverso gli indicatori visti nel paragrafo precedente, mentre l'azione rappresenta la nuova allocazione scelta dall'agente.

L'azione dell'agente è rappresentata dunque dalla scelta dei nuovi pesi di portafoglio. Tuttavia, nel modello implementato, l'output della rete neurale non viene interpretato direttamente come un vettore finale di pesi, bensì l'agente produce inizialmente un vettore di valori reali, uno per ciascun ETF. Tali valori possono essere interpretati come uno score di preferenza assegnato dall'agente ai diversi asset, attraverso il quale vengono selezionati i cinque ETF con valore più elevato. Successivamente, su tale sottoinsieme viene applicata una funzione softmax, in modo da ottenere pesi positivi e normalizzati.

Indicando con $z_{i,t}$ lo score prodotto dall'agente per l'ETF i al tempo t, viene individuato l'insieme K_t contenente i cinque asset con score maggiore:

$$K_t = top5(z_{1,t}, z_{2,t}, .. z_{N,t})$$

Per gli asset inclusi in K_t, il peso viene calcolato tramite la funzione di softmax:

$$w_{i,t} = \frac{e^{z_{i,t}}}{\sum_{j \in K_t} e^{z_{j,t}}} \quad \text{se } i \in K_t$$

Mentre per gli asset esclusi dal sottoinsieme selezionato il peso viene posto pari a zero:

$$w_{i,t} = 0 \quad \text{se } i \notin K_t$$

Questa funzione garantisce il rispetto di due vincoli fondamentali: in primo luogo il portafoglio è long-only, poiché tutti i pesi risultano non negativi:

$$w_{i,t} \geq 0$$

In secondo luogo, il capitale è sempre interamente investito, poiché la somma dei pesi è pari a uno:

$$\sum_{i=1}^{N} w_{i,t} = 1$$

La scelta di imporre un vincolo long-only rende il modello più coerente con una logica di investimento tradizionale e non levereged. L'agente non può infatti assumere posizioni corte e non può utilizzare la leva finanziaria, bensì si limita solamente a scegliere come distribuire il capitale tra gli ETF disponibili. Inoltre, l'assenza di una componente di liquidità, implica che il capitale sia sempre allocato interamente tra tutti gli strumenti utilizzabili.

Il vincolo di selezione dei cinque migliori asset è stato integrato con il fine di evitare che la funzione di softmax possa assegnare piccole allocazioni positive a ciascun ETF, generando portafogli di difficile interpretazione e potenzialmente meno coerenti con una gestione ordinaria di portafoglio. Limitando dunque l'allocazione a massimo cinque strumenti l'agente è costretto ad esprimere una scelta più netta tra le diverse asset class, che consente di osservare con maggiore chiarezza quali esposizioni vengano privilegiate nei diversi periodi di mercato. Al tempo stesso, tale scelta presenta anche un possibile limite: la selezione dei migliori ETF può generare discontinuità nella composizione del portafoglio in funzione di piccole variazioni negli score prodotti dalla rete, soprattutto nei casi in cui gli score di alcuni asset siano molto vicini tra loro, determinando l'ingresso o l'uscita di uno o più ETF nel portafoglio e generando così aumenti nel turnover e nei costi di transazione. Per questo motivo, oltre all'introduzione di una penalizzazione alla variazione dei pesi all'interno della funzione di ricompensa, è stata implementata una soglia minima di miglioramento. In particolare, un ETF esterno al portafoglio può sostituire un ETF già detenuto solo se il suo score supera quello dell'altro di una quantità prefissata δ, formalmente:

$$z_{j,t} > \min_{i \, in \, Z_t} z_{i,t} + \delta$$

Nel presente modello, la soglia minima di miglioramento è stata fissata ad un valore pari a 0,075, in ottica di garantire un trade-off tra reattività e stabilità di portafoglio e disincentivare modifiche eccessivamente frequenti della composizione del portafoglio che comporterebbero eccessivi costi di transazione.

5.5. Funzione di ricompensa

La definizione della funzione di ricompensa rappresenta sicuramente un passaggio centrale nella costruzione del modello. All'interno del modello, la ricompensa non può

essere identificata esclusivamente con il rendimento del portafoglio, poiché una strategia finanziaria deve essere valutata anche in relazione al rischio assunto, alla stabilità dei risultati e ai costi generati dall'attività di ribilanciamento. La ricompensa deve dunque tradurre in forma numerica l'obbiettivo economico-finanziario che si vuole far apprendere all'agente.

Nel presente modello, la funzione di ricompensa è stata costruita combinando tre componenti principali:

- Il rendimento netto del portafoglio
- Lo sharpe ratio rolling
- Il sortino ratio rolling[26]

L'obbiettivo è dunque quello di incentivare l'agente non soltanto a generare rendimento, ma anche a preferire allocazioni caratterizzate da un migliore rischio-rendimento e ad una minore esposizione alla volatilità negativa.

All'interno della funzione di ricompensa, la componente principale è rappresentata dal rendimento netto di portafoglio. A ogni data di ribilanciamento, l'agente sceglie una nuova allocazione tra gli ETF disponibili che porterà il portafoglio a realizzare un certo rendimento nel periodo successivo. Indicando con $w_{i,t}$ il peso assegnato all'ETF i al tempo t e con $r_{i,t+1:t+5}$ il relativo rendimento cumulato nelle cinque sedute successive, il rendimento lordo del portafoglio può essere espresso come:

$$R_{p,t}^{lordo} = \sum_{i=1}^{N} w_{i,t} * r_{i,t+1:t+5}$$

Tuttavia, poiché la strategia dell'agente può modificare la composizione del portafoglio ad ogni ribilanciamento, è necessario considerare anche i costi associati a tali variazioni. Per questo motivo viene calcolato il turnover, definito come la somma delle variazioni assolute dei pesi rispetto all'allocazione del periodo precedente:

$$TO_t = \sum_{i=1}^{N} |w_{i,t} - w_{i,t-1}|$$

[26] Con il termine rolling si indica il calcolo dell'indicatore su una finestra temporale mobile di ampiezza prefissata dove ad ogni nuova osservazione la finestra viene aggiornata eliminando il dato più remoto ed includendo quello più recente.

Il costo di transazione viene quindi calcolato associando un coefficiente di costo c, pari a 0,001, per ogni unità di turnover[27]:

$$C_t = c * TO_t$$

Ottenendo quindi un rendimento netto di portafoglio pari a:

$$R_{p,t}^{netto} = R_{p,t}^{lordo} - C_t$$

L'introduzione dei costi associati al turnover all'interno della funzione di ricompensa è rilevante dal punto di vista finanziario poiché impedisce all'agente di perseguire strategie eccessivamente instabili o difficilmente implementabili. In assenza di tale penalizzazione, il modello potrebbe essere incentivato a modificare continuamente la composizione del portafoglio, ottenendo potenzialmente buoni risultati teorici ma generando costi operativi troppo elevati che ne ridurrebbero il vantaggio.

Accanto al rendimento netto, la funzione di ricompensa include una componente basata sullo Sharpe ratio rolling. Lo Sharpe ratio misura il rendimento ottenuto per unità di rischio complessivo e rappresenta una delle metriche più utilizzate nella valutazione finanziaria di una strategia di investimento. Nel modello, tale indicatore viene calcolato su una finestra mobile di 12 ribilanciamenti, corrispondente approssimativamente a tre mesi di mercato, in modo da incorporare un orizzonte temporale sufficientemente ampio per permettere al modello di utilizzare dati più stabili nelle valutazioni.

Indicando con $R_{p,t}^{netto}$ il rendimento netto del portafoglio al tempo t, lo Sharpe ratio rolling può essere definito come:

$$SR_t^{(12)} = \frac{\mu_t^{(12)}}{\sigma_t^{(12)}}$$

dove $\mu_t^{(12)}$ rappresenta la media dei rendimenti netti di portafoglio negli ultimi 12 ribilanciamenti:

$$\mu_t^{(12)} = \frac{1}{12} \sum_{t-11}^{t} R_{p,t}^{netto}$$

[27] È stato scelto un coefficiente pari allo 0,10% in quanto in questo elaborato vengono trattati ETF molto liquidi, per i quali il mercato offre costi di transazione particolarmente bassi.

mentre $\sigma_t^{(12)}$ rappresenta la deviazione standard dei rendimenti nella stessa finestra:

$$\sigma_t^{(12)} = \sqrt{\frac{1}{12}\sum_{t-11}^{t}(R_{p,t}^{netto} - \mu_t^{(12)})^2}$$

Infine, la terza componente è rappresentata dal Sortino ratio rolling, anch'esso calcolato su una finestra mobile di 12 ribilanciamenti. Il Sortino ratio è simile allo Sharpe ratio, ma considera al denominatore soltanto la volatilità negativa, ossia la variabilità dei rendimenti inferiori ad una soglia di riferimento. Nel modello, questa soglia è assunta pari a zero, così da penalizzare in modo specifico le oscillazioni negative del portafoglio:

$$SOR_t^{(12)} = \frac{\mu_t^{(12)}}{\sigma_{-,t}^{(12)}}$$

dove $\sigma_{-,t}^{(12)}$, rappresenta la deviazione downside, calcolata considerando solo i rendimenti negativi:

$$\sigma_{-,t}^{(12)} = \sqrt{\frac{1}{12}\sum_{t-11}^{t} min\,(R_{p,t}^{netto}, 0)^2}$$

L'inclusione del Sortino ratio è motivata dal fatto che, nella prospettiva dell'investitore, non tutta la volatilità ha lo stesso significato. Le oscillazioni positive non rappresentano un rischio nella stessa misura di quelle negative, per questo motivo una metrica basata sulla deviazione downside consente di incentivare l'agente all'apprendimento verso politiche meno esposte a rischio di ribasso persistente.

Per evitare valori indefiniti, lo Sharpe Ratio rolling e il Sortino Ratio rolling vengono posti pari a zero quando la volatilità dei rendimenti e/o la volatilità negativa nella finestra temporale considerata risultano nulle.

La funzione di ricompensa complessiva utilizzata dal modello può quindi essere definita formalmente come:

$$Ricompensa_t = R_{p,t}^{netto} + \alpha * SR_t^{(12)} + \beta * SOR_t^{(12)} \qquad \text{con: } \alpha = 0{,}005 \text{ e } \beta = 0{,}005$$

I coefficienti α e β sono stati fissati entrambi pari a 0,005 a seguito di una fase preliminare di sperimentazione su differenti configurazioni della funzione di ricompensa. Le prove

effettuate hanno evidenziato come valori più elevati tendessero ad attribuire un peso eccessivo alle metriche risk-adjusted, riducendo la centralità del rendimento netto, mentre valori inferiori ne limitavano eccessivamente l'influenza. Tale valore è stato quindi scelto come compromesso al fine di incentivare allocazioni caratterizzate da un equilibrio tra performance e qualità dei rendimenti, senza consentire a Sharpe ratio e Sortino ratio di dominare la funzione di ricompensa.

Dal punto di vista economico-finanziario, questa struttura della ricompensa riflette l'idea che una strategia di portafoglio debba essere valutata secondo più dimensioni. Il rendimento netto spinge l'agente verso allocazioni profittevoli; la penalizzazione del turnover introduce una disciplina operativa; lo Sharpe rolling incentiva un miglior rapporto tra rendimento e rischio complessivo; il Sortino rolling concentra l'attenzione sul rischio downside. In questo modo, l'agente non viene addestrato a inseguire semplicemente il rendimento massimo, ma a costruire una politica di allocazione dinamica più coerente con i criteri tradizionali di valutazione della performance finanziaria.

5.6. Algoritmo PPO e rete MLP

Dopo aver definito l'ambiente e la funzione di ricompensa utilizzata dal modello è necessario specificare il funzionamento dell'algoritmo utilizzato per l'apprendimento della politica d'investimento.

Come visto nei capitoli precedenti, la PPO utilizza una funzione obbiettivo surrogata, basata sul rapporto tra la probabilità assegnata dalla nuova politica e quella assegnata dalla vecchia politica alla stessa azione. Formalmente:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$$

La funzione obbiettivo clipped viene quindi espressa come:

$$L^{CLIP}(\theta) = \hat{E}_t\left[\min(r_t(\theta)\hat{A}_t\,, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)\right]$$

dove $\hat{A}_t$ rappresenta la stima del vantaggio, mentre ϵ è il parametro di clipping, che nel modello utilizzato assume valore pari a 0,2. Questo termine fa sì che il rapporto $r_t(\theta)$ non si allontani eccessivamente da 1, limitando la variazione tra la vecchia e la nuova politica e di conseguenza evitando cambiamenti troppo drastici che potrebbero compromettere la qualità della politica appresa.

Nel presente lavoro, la politica e la funzione di valore sono approssimate tramite una rete neurale di tipo Multi-Layer Perceptron (MLP), ossia una rete neurale in grado di modellare relazioni non lineari tra le variabili che compongono lo stato[28]. Più precisamente, l'implementazione adottata segue un'architettura di tipo actor-critic, nella quale l'osservazione dello stato viene elaborata da due componenti distinte: l'actor, responsabile della definizione della politica di allocazione; e il critic, utilizzato per stimare il valore dello stato[29].

Dal punto di vista operativo, l'elaborazione effettuata da ciascuno strato della MLP applica una trasformazione all'output dello strato precedente, seguita da una funzione di attivazione non lineare. Indicando con $h^{(l-1)}$ l'output dello strato precedente, l'elaborazione effettuata dallo strato l-esimo può essere espressa come:

$$h^{(l)} = \phi(W^{(l)}h^{(l-1)} + b^{(l)})$$

dove: $W^{(l)}$ rappresenta la matrice dei pesi dello stato l, $b^{(l)}$ il relativo vettore di bias e $\phi()$ una funzione di attivazione non lineare.

In particolare: i pesi contenuti nella matrice $W^{(l)}$ determinano l'importanza attribuita alle diverse informazioni provenienti dallo stato precedente e vengono modificati durante tutto il processo di apprendimento; il bias $b^{(l)}$ viene anch'esso appreso durante l'addestramento e rappresenta un termine additivo che consente di introdurre un grado di libertà aggiuntivo durante la trasformazione dello stato; infine, la funzione di attivazione non lineare $\phi()$ utilizzata nel modello è la funzione tangente iperbolica $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ che consente di trasformare i valori in input in un intervallo compreso tra -1 e 1, introducendo così la non linearità necessaria affinché la rete possa rappresentare relazioni complesse tra le variabili.

La successione di tali trasformazioni permette alla rete di costruire rappresentazioni progressivamente più complesse dello stato osservato, cogliendo eventuali relazioni non lineari tra le variabili finanziarie.

Nel modello implementato, il vettore di stato comprende le feature di mercato standardizzate, le correlazioni rolling tra gli ETF e i pesi correnti di portafoglio, ottenendo

[28] Goodfellow, I., Bengio, Y. and Courville, A. (2016). Deep learning - Chapter 6 "Deep Feedforward Networks." [online] Deeplearningbook.org. Available at: https://www.deeplearningbook.org/.
[29] Per una più approfondita analisi dei modelli actor-critic si rimanda al capitolo 3.7 "metodi actor-critic"

un vettore di input composto da 150 elementi. A partire da questo, l'Actor determina i parametri della distribuzione associati alla politica $\pi_\theta(a_t|s_t)$, dalla quale viene ricavata un'azione continua pari al numero di ETF considerati. Successivamente tale azione viene trasformata attraverso un processo di selezione degli asset e una funzione softmax, in modo da ottenere un'allocazione long-only, come descritto nei paragrafi precedenti.

Parallelamente, il Critic approssima la funzione di valore come:

$$V_\phi(s_t) = E_{\pi_\theta}\left[\sum_{k=0}^{\infty} \gamma^k \, r_{t+k} \mid s_t\right]$$

fornendo una stima del rendimento futuro atteso a partire dallo stato corrente, che viene utilizzata dall'algoritmo PPO per calcolare il vantaggio conseguente all'azione intrapresa durante l'aggiornamento della politica.

Una particolarità della rete MLP è la sua natura feed-forward, ciò implica che ogni osservazione viene elaborata come un vettore autonomo e di conseguenza la rete non dispone di nessuno strato persistente attraverso il quale conservare direttamente le informazioni in merito alle osservazioni precedenti. Tuttavia, per superare questo limite, nel presente modello la dipendenza temporale dei dati finanziari non avviene attraverso una vera e propria memoria interna della rete, come avverrebbe con altre tipologie di reti neurali, ma viene sintetizzata dalle componenti rolling dello stato.

In termini finanziari, l'actor può essere visto come il modello decisionale che propone la nuova composizione del portafoglio, mentre il critic svolge una funzione di valutazione della qualità dello stato e delle prospettive future. In questo senso l'agente non si limita ad osservare il rendimento realizzato immediatamente successivo alle sue scelte, ma cerca piuttosto di apprendere una politica che massimizzi la ricompensa cumulata nel tempo, coerentemente con una logica tradizionale di gestione degli investimenti.

Analizzando la funzione obbiettivo completa della PPO, questa può essere scomposta in tre termini: la funzione clippata, la funzione di valore e un termine di entropia. Formalmente:

$$L^{CLIP+VF+S}(\theta) = \hat{E}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)]$$

All'interno del modello sono stati impostati i seguenti parametri: coefficiente di clipping pari a 0,2; coefficiente associato alla funzione di valore pari a 0,5; fattore di sconto pari a

0,99 e tasso di apprendimento pari a 0,0003. Tali valori rappresentano uno standard ampliamento utilizzato nelle implementazioni PPO, nonché i valori predefiniti utilizzati dalla libreria implementata[30].

L'unico valore modificato rispetto alla configurazione standard è il termine di entropia, fissato a 0,005. La scelta di un coefficiente così contenuto è motivata dalla necessità di introdurre un incentivo moderato all'esplorazione della politica, soprattutto visto il numero relativamente limitato di osservazioni e ribilanciamenti nel periodo di training, condizione che rende preferibile una maggiore stabilità del processo di apprendimento.

La procedura di addestramento avviene sul training set, dove ogni 64 interazioni consecutive con l'ambiente l'agente aggiorna i parametri della rete neurale. Poiché quindi ogni step corrisponde a cinque sedute di mercato, una raccolta di 64 interazioni equivale a 320 sedute di mercato, ossia dopo oltre un anno di mercato. Il numero di step prima di un aggiornamento esprime quindi la quantità di esperienza raccolta dal modello prima di cambiare: un valore più basso porterebbe ad un modello più dinamico ma anche più esposto al rumore; mentre valori più alti porterebbero a risultati potenzialmente più stabili ma meno sensibili alle novità. Un ulteriore parametro rilevante è la dimensione del batch, fissata anch'essa a 64. La dimensione del batch indica quante osservazioni vengono utilizzate contemporaneamente per calcolare un aggiornamento dei parametri della rete durante l'ottimizzazione. In questo caso, avendo numero di step e dimensione del batch identici, l'intera sequenza di esperienza raccolta viene usata per aggiornare il modello.

Il terzo elemento da considerare è il numero complessivo di timesteps di addestramento, fissato pari a 50.000 timesteps. Questo parametro indica il numero totale di interazioni tra agente e ambiente durante la fase di training, dove ogni step corrisponde ad una decisione di ribilanciamento e permette all'agente di ripercorrere più volte il training set, nonostante questo sia composto da un numero finito di osservazioni storiche. Durante l'addestramento viene poi utilizzata una procedura di checkpoint validation, poiché il modello non viene selezionato automaticamente alla fine dei 50.000 timesteps, ma viene valutato periodicamente. In particolare, ogni 1.024 timesteps di training, la politica viene valutata e salvata, così che al termine del training venga selezionato il modello migliore. Per evitare il rischio che un addestramento troppo lungo su un periodo porti il modello ad

[30] Tali valori sono stati ad ogni modo confrontati sperimentalmente con configurazioni alternative, che hanno confermato la configurazione standard adottata come quella più adatta a fornire risultati complessivamente più soddisfacenti in termini di performance e stabilità sul set di validation.

adattarsi eccessivamente al training set, la performance finale non viene valutata su quest'ultimo, bensì sul set di validation, separato temporalmente rispetto a quello di training.

Infine, per ridurre la dipendenza dei risultati da una singola inizializzazione casuale, il modello viene addestrato utilizzando un totale di cinque seed. Il seed determina la sequenza di numeri pseudo-casuali utilizzata durante il processo di apprendimento e interviene, inoltre, nell'inizializzazione dei parametri della rete neurale nel campionamento delle azioni della politica stocastica e nelle operazioni casuali associate all'ottimizzazione. L'utilizzo di più seed consente quindi di verificare la robustezza dei risultati rispetto alla componente puramente stocastica dell'addestramento e di ridurre il rischio che i risultati ottenuti siano riconducibili esclusivamente ad una particolare inizializzazione favorevole.

Tra i diversi modelli ottenuti, viene quindi selezionato quello che mostra la migliore performance sul validation set secondo un criterio basato sulla combinazione tra sharpe ratio e massimo drawdown:

$$sharpe\ drawdown\ score = sharpe\ ratio + \max drawdown$$

dove essendo max drawdown una misura negativa, l'agente premia i modelli con sharpe ratio elevato, penalizzando invece quelli che mostrano drawdown più ampi.

È opportuno precisare che lo sharpe drawdown score non coincide con la funzione di ricompensa utilizzata durante l'addestramento. La ricompensa costituisce infatti il segnale attraverso il quale l'agente apprende a ogni periodo di ribilanciamento, mentre lo sharpe drawdown score viene utilizzato esclusivamente in fase di validazione per la selezione del checkpoint migliore. La scelta di una metrica distinta permette di valutare il modello sulla base di criteri complessivi di performance corretta per il rischio, considerando contemporaneamente l'efficienza dei rendimenti, attraverso lo Sharpe Ratio, e l'entità delle perdite massime subite, attraverso il Maximum Drawdown. In questo modo, la selezione finale non dipende direttamente dalla stessa funzione utilizzata per l'apprendimento, ma da una misura di valutazione esterna applicata ai risultati ottenuti sul validation set.

5.7. Benchmark di confronto

Al fine di valutare in modo coerente la performance del modello, i risultati ottenuti sono stati confrontati con un insieme di benchmark passivi e semi-passivi. La scelta di includere più strategie di riferimento consente infatti di analizzare il comportamento dell'agente non solo rispetto ad un tradizionale indice di mercato, ma anche rispetto a portafogli diversificati e regole di allocazioni semplici e tradizionali, utilizzate tutt'oggi da diversi investitori.

Il primo benchmark considerato è rappresentato dallo SPY, ETF che replica l'andamento dell'indice S&P 500. La scelta dello SPY è motivata dal fatto che il mercato azionario statunitense rappresenta storicamente uno dei mercati più rilevanti, liquidi e performanti a livello globale, ed è frequentemente utilizzato come riferimento nella valutazione di strategie di investimento. In questo senso, lo SPY costituisce un benchmark particolarmente sfidante, poiché rappresenta una strategia passiva esposta ad un mercato caratterizzato, almeno nel lungo periodo, da rendimenti medi particolarmente elevati.

Il secondo benchmark di riferimento è costituito da un portafoglio 60/40, composto per il 60% da SPY e per il 40% da TLT, ossia l'ETF rappresentativo dei Treasury bond statunitensi a lunga scadenza. Tale portafoglio riproduce una delle allocazioni che risultano tradizionalmente più utilizzate nella gestione di portafoglio, basata su una combinazione tra componente azionaria e obbligazionaria. Il portafoglio viene inoltre ribilanciato periodicamente e viene valutato al netto dei costi di transazione.

Sono inoltre inclusi due benchmark equally weighted. Il primo è un portafoglio equal weight ribilanciato periodicamente, che assegna lo stesso peso a tutti gli ETF dell'universo investibile e viene riportato periodicamente alla ricomposizione iniziale. Il secondo invece, è un portafoglio equal weight buy and hold, costituito con pesi iniziali uguali ma lasciato evolvere senza successivi ribilanciamenti. Questi due benchmark hanno dunque la funzione di confrontare direttamente il modello utilizzato attraverso strategie estremamente più semplici e non ottimizzate.

Come già indicato, tutti i risultati sono calcolati al netto dei costi di transazione, fissati pari allo 0,10% per singola operazione. Tale aspetto è particolarmente rilevante soprattutto per il confronto con il modello analizzato, poiché, replicando una strategia dinamica, presenta un turnover sensibilmente superiore rispetto ai suoi benchmark.

5.8. Risultati in validation

La fase di validation copre il periodo compreso tra il 1° gennaio 2021 e il 31 dicembre 2022, una finestra temporale particolarmente significativa poiché include sia una fase di mercato positiva, legata alla prosecuzione del recupero post-pandemico, sia una forte fase negativa del 2022, caratterizzata da un aumento dell'inflazione, rialzo dei tassi di interesse e ribasso simultaneo di molte asset class.

I risultati ottenuti nella fase di validation sono riportati nella seguente tabella:

Strategia	Rendimento cumulato	Volatilità annualizzata	Sharpe ratio	Sortino ratio	Max drawdown	Turnover medio	Costi totali
PPO MLP	14,89%	13,78%	0,592	1,031	-11,61%	34,04%	3,40%
SPY	3,44%	17,94%	0,187	0,282	-23,15%	0,00%	0,00%
60/40 SPY-TLT	-11,06%	13,39%	-0,388	-0,616	-27,38%	1,07%	0,11%
Equal weight ribilanciato	5,11%	12,58%	0,268	0,441	-15,65%	1,64%	0,16%
Equal weight buy and hold	8,52%	12,88%	0,394	0,619	-14,32%	0,00%	0,00%

Tabella 5.3: risultati in fase di validation

Nel periodo di validation il modello ottiene la migliore performance complessiva tra tutte le strategie considerate. Il rendimento cumulato è pari al 14,89%, superiore sia allo SPY, che registra un rendimento del 3,44%, sia ai due benchmark equally weighted. Il confronto è ancora più evidente rispetto al portafoglio 60/40 che nel periodo considerato presenta una performance negativa pari a -11,06%.

Anche in termini di rendimento corretto per il rischio il modello PPO mostra risultati favorevoli. Lo Sharpe ratio è pari a 0,592, superiore a quello dello SPY e dei benchmark diversificati. Analogamente, il Sortino ratio raggiunge un valore pari a 1,031, indicando una migliore capacità del modello di generare rendimento rispetto alla volatilità negativa. Questo risultato è particolarmente rilevante in un contesto come quello del 2022, nel quale molte asset class hanno subito correzioni significative.

Un ulteriore elemento positivo riguarda il contenimento del rischio di perdita massima. Il massimo drawdown del modello PPO è pari a -11,61%, valore sensibilmente inferiore rispetto allo SPY, che registra un drawdown massimo di -23,15%, e rispetto al portafoglio 60/40, il cui drawdown raggiunge -27,38%. Anche rispetto ai benchmark equally weighted, il modello presenta una perdita massima più contenuta.

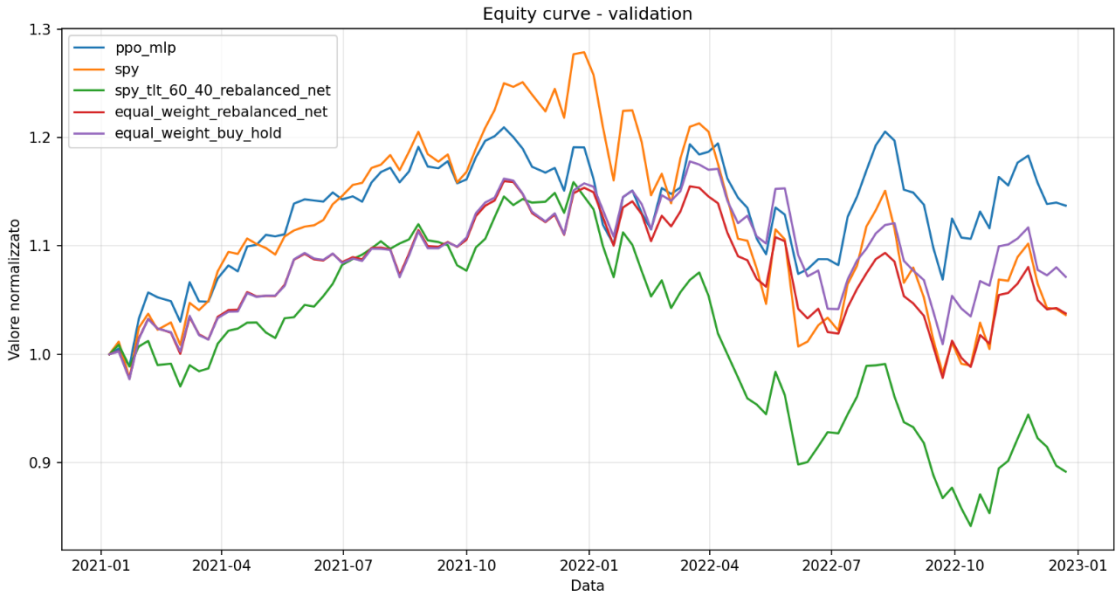

Figura 5.1: equity curves in fase di validation

Come si può osservare dalla Figura 5.1, la curva del modello analizzato mostra una dinamica più favorevole rispetto ai benchmark, soprattutto nelle fasi di maggiore instabilità del mercato. In particolare, il modello sembra riuscire a contenere meglio le perdite durante le fasi negative, mantenendo una traiettoria più regolare rispetto allo SPY e al portafoglio 60/40.

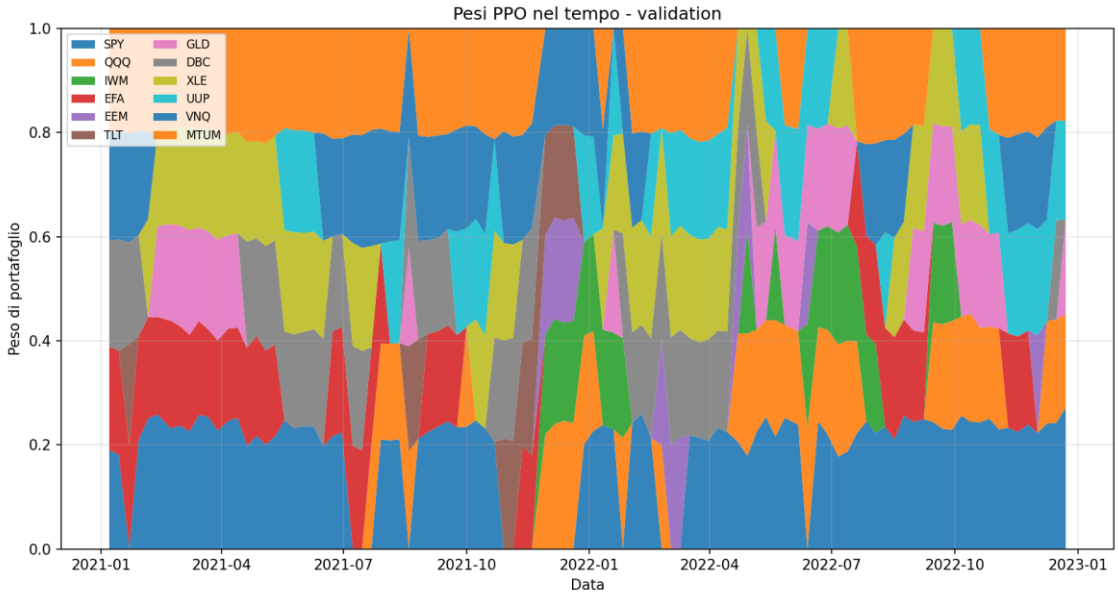

Figura 5.2: pesi di portafoglio in fase di validation

La figura 5.2 relativa ai pesi di portafoglio evidenzia invece la natura dinamica della strategia appresa dall'agente. Il modello non mantiene una composizione statica, bensì modifica nel tempo l'esposizione alle diverse asset class in funzione dello stato osservato del mercato.

È tuttavia necessario sottolineare che il periodo di validation è stato utilizzato anche per la selezione del modello finale. Di conseguenza, i risultati ottenuti in questa fase devono essere interpretati con prudenza, in quanto essi indicano che la strategia appresa presenta buone proprietà sul campione di validation, ma non sono sufficienti da soli per dimostrare la capacità del modello a generalizzare su dati completamente non osservati.

5.9. Risultati in test

La fase di test rappresenta la vera prova del modello, poiché valuta la strategia su un periodo successivo a quello utilizzato per l'addestramento e per la selezione del modello. Il test ha inizio il 1° gennaio 2023 e prosegue fino ad oggi, ciò consente di osservare il comportamento dell'agente in una fase di mercato diversa, caratterizzata da una forte ripresa dei mercati azionari.

I risultati ottenuti nella fase di test sono riportati nella seguente tabella:

Strategia	Rendimento cumulato	Volatilità annualizzata	Sharpe ratio	Sortino ratio	Max drawdown	Turnover medio	Costi totali
PPO MLP	75,67%	11,97%	1,486	1,751	-13,53%	30,16%	5,19%
SPY	101,89%	14,50%	1,540	2,078	-16,88%	0,00%	0,00%
60/40 SPY-TLT	51,96%	10,91%	1,215	2,057	-12,24%	0,91%	0,16%
Equal weight ribilanciato	72,16%	10,55%	1,612	2,083	-9,70%	1,41%	0,24%
Equal weight buy and hold	74,19%	11,28%	1,546	1,983	-10,64%	0,00%	0,00%

Tabella 5.3: risultati in fase di test

Nel periodo di test il modello mantiene una performance complessivamente positiva e competitiva. Il rendimento cumulato è pari al 75,67%, superiore a quello del portafoglio 60/40, pari al 51,96%, leggermente superiore rispetto al portafoglio equal weight ribilanciato, pari al 72,16%, e all'equal weight buy and hold, pari al 74,19%. Tuttavia, il modello non riesce a superare lo SPY, che nello stesso periodo registra un rendimento cumulato pari al 101,89%.

Questo risultato deve essere interpretato tenendo conto della straordinaria performance del mercato azionario statunitense durante il periodo di test. Lo SPY rappresenta infatti un benchmark particolarmente difficile da battere in una fase di mercato così favorevole all'azionario USA. Per questo motivo, il modello costituito su un universo multi-asset e vincolato da una logica di allocazione dinamica, tende a mantenere una maggiore diversificazione e, di conseguenza, non beneficia pienamente della concentrazione sul mercato azionario statunitense.

Dal punto di vista del rischio, il modello mostra comunque proprietà interessanti. La volatilità annualizzata è pari all'11,97%, inferiore dunque a quella dello SPY, pari al 14,50%. Anche il massimo drawdown risulta più contenuto rispetto allo SPY: -13.53% contro -16.88% per lo SPY. Complessivamente, questo suggerisce che, pur rinunciando ad una parte di rendimento, la strategia appresa dall'agente è stata in grado di mantenere un profilo di rischio più moderato.

Il confronto con i benchmark equally weighted è invece più equilibrato, in quanto il portafoglio equal weight ribilanciato ottiene uno sharpe ratio pari a 1,612, superiore a quello del modello analizzato, pari a 1,486. Anche il sortino ratio e il massimo drawdown risultano migliori per l'equal weight ribilanciato, con valori rispettivamente di 2,083 e -9,70% contro 1,751 e -13.53%.

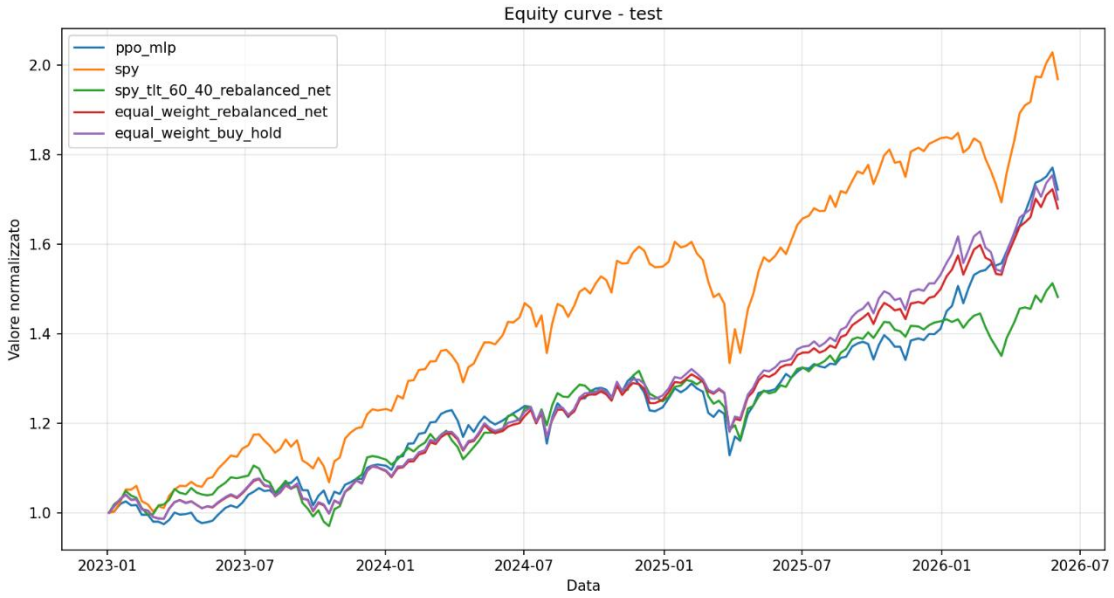

Figura 5.3: equity curves in fase di test

La figura 5.3 relativa alle equity curves mostra come il modello segua una traiettoria positiva, ma non riesca a replicare la crescita dello SPY nelle fasi più favorevoli di mercato azionario, soprattutto nei periodi in cui il mercato statunitense registra una forte accelerazione. Al contrario, il comportamento del modello risulta più vicino a quello dei portafogli diversificati, confermando la natura più bilanciata della strategia appresa.

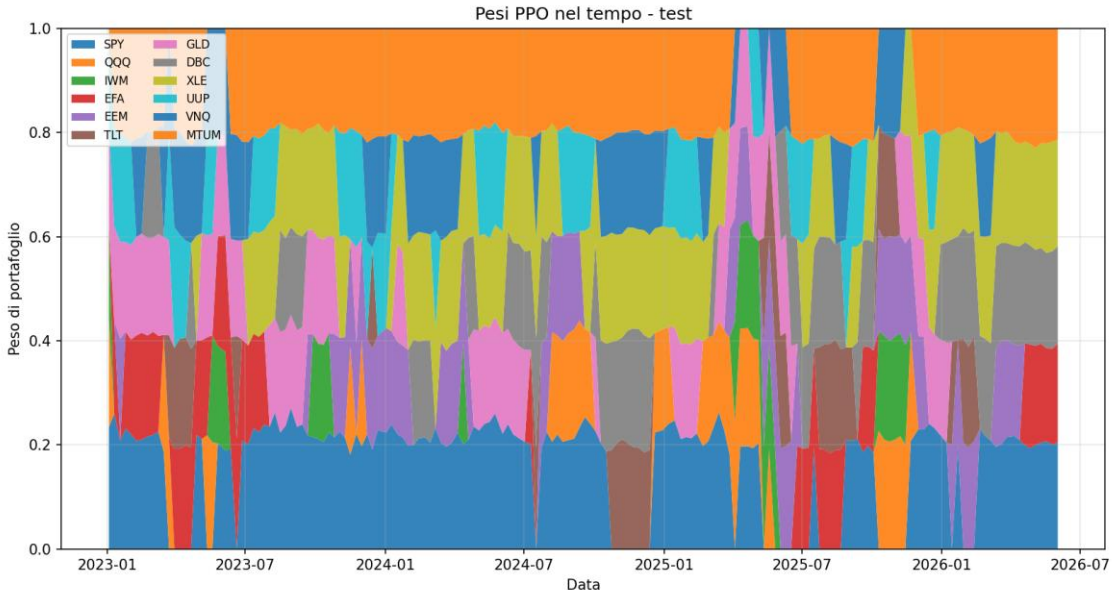

Figura 5.4: pesi di portafoglio in fase di test

La figura 5.4 relativa ai pesi di portafoglio permette invece di osservare come l'agente continui ad adottare una gestione dinamica di portafoglio anche fuori dal campione di validation. Le allocazioni non rimangono costanti, ma si modificano nel tempo in risposta delle variazioni dello stato di mercato. Questo conferma che il modello non si limita a replicare una strategia statica, ma applica una politica di investimento che dipende direttamente dalle condizioni di mercato osservate.

Un possibile elemento critico è sicuramente rappresentato dall'elevato turnover. Nel periodo di test, il turnover medio del modello è pari a circa 30,16%, con costi totali di transazione pari al 5,19%.

5.10. Valutazione del modello

Nel complesso, i risultati mostrano come il modello analizzato sia in grado di costruire una strategia dinamica di gestione di portafoglio competitiva rispetto ai principali benchmark considerati. La superiorità del modello emerge in modo netto nella fase di validation, nella quale ottiene il miglior rendimento cumulato, il miglior Sharpe ratio, il miglior Sortino ratio e il drawdown più contenuto. Questo suggerisce che l'agente sia riuscito ad apprendere una politica efficace nel periodo immediatamente successivo al training, caratterizzato da condizioni di mercato complesse e da una significativa instabilità nel 2022

Nel test, tuttavia, il quadro risulta più articolato. Il modello continua a generare una performance elevata e supera il portafoglio 60/40 e, in termini di rendimento cumulato, anche i benchmark equally weighted. Tuttavia, non riesce a battere lo SPY in termini di rendimento assoluto e non supera l'equal weight ribilanciato in termini di Sharpe ratio, Sortino ratio e massimo drawdown. Di conseguenza, i risultati non consentono di affermare una superiorità robusta e sistematica del modello rispetto a strategie passive semplici.

Il modello mostra quindi una buona capacità di adattamento e una gestione del rischio più prudente rispetto allo SPY, ma il vantaggio rispetto a benchmark diversificati e poco costosi appare meno netto. In particolare, l'equal weight ribilanciato rappresenta un benchmark molto severo che pur adottando una strategia elementare, beneficia della diversificazione tra asset class e di un turnover estremamente contenuto. Il fatto che tale benchmark ottenga nel test metriche risk-adjusted superiori al PPO suggerisce che la complessità del modello non si traduca automaticamente in un miglioramento netto della performance.

Un ulteriore elemento da considerare riguarda i costi di transazione. La strategia del modello analizzato presenta un turnover sensibilmente più elevato rispetto ai benchmark passivi, aspetto coerente con la natura dinamica dell'approccio adottato, ma che pone un tema rilevante in termini di efficienza operativa. Nell'ambito della presente analisi, la performance finale del modello rimane comunque positiva e competitiva anche al netto dei costi di transazione, evidenziando come la maggiore dinamicità della strategia comporti costi operativi non trascurabili, ma sia al tempo stesso associata a rendimenti sufficientemente elevati da compensare tali oneri. Tuttavia, in un'applicazione reale, un turnover elevato potrebbe incidere in misura significativa sulla performance netta, soprattutto in presenza di bid-ask spread, commissioni, e vincoli di liquidità non pienamente catturati dal modello utilizzato.

Dal punto di vista metodologico, il principale risultato positivo dell'esperimento è che il modello non produce una strategia casuale o instabile, ma una politica economicamente interpretabile, capace di generare rendimenti elevati e di mantenere il rischio sotto controllo. L'utilizzo di una rete MLP actor-critic, pur relativamente semplice, si dimostra sufficiente per apprendere una relazione tra stato di mercato e decisioni di allocazione. Questo è un aspetto rilevante, poiché conferma la possibilità di applicare tecniche di RL

alla gestione di portafoglio anche senza ricorrere, in questa prima fase, ad architetture più complesse come reti ricorrenti o modelli basati su dati testuali.

In conclusione, l'esperimento finale evidenzia sia il potenziale sia i limiti dell'approccio adottato. Il modello dimostra di poter apprendere una politica di allocazione capace di competere con benchmark tradizionali e diversificati, mantenendo un profilo di rischio relativamente controllato. Tuttavia, la performance fuori validation non è tale da giustificare in modo definitivo la maggiore complessità del modello rispetto a strategie passive robuste e poco costose. Il contributo principale dell'analisi consiste quindi nel mostrare che la PPO può essere applicata con successo a un problema di gestione di portafoglio, ma che la sua utilità pratica dipende dalla capacità di migliorare ulteriormente la robustezza, ridurre il turnover e verificare la stabilità dei risultati su periodi di mercato alternativi.

CAPITOLO 6 - INTEGRAZIONE DEL SENTIMENT FINANZIARIO NEL MODELLO MEDIANTE FINBERT

6.1. Introduzione al modello

Il modello sviluppato nel capitolo precedente affronta il problema della gestione dinamica di portafoglio utilizzando esclusivamente un insieme di variabili quantitative ricavate dall'andamento storico dei mercati finanziari, consentendo al modello di apprendere una politica di allocazione che trasforma segnali numerici disponili in decisioni di investimento.

Sebbene questo approccio rappresenti una base metodologicamente solida, esso presenta un limite informatico rilevante: le variabili utilizzate descrivono gli effetti già manifestatisi sui prezzi e sui volumi di mercato, ma non incorporano direttamente il contenuto delle informazioni che contribuiscono alla formazione delle aspettative degli investitori. I prezzi costituiscono infatti il risultato finale delle decisioni degli investitori, mentre le notizie economiche, societarie e geopolitiche rappresentano uno dei principali canali attraverso cui le nuove informazioni raggiungono il mercato.

L'integrazione di dati testuali risponde dunque all'obbiettivo di ampliare lo spazio informativo disponibile per l'agente. Se gli indicatori quantitativi descrivono l'evoluzione osservata dei mercati, le notizie possono fornire una rappresentazione più diretta delle informazioni, delle valutazioni e delle aspettative che possono precedere o accompagnare le variazioni dei prezzi.

Secondo l'ipotesi dei mercati efficienti, i prezzi delle attività finanziarie dovrebbero riflettere rapidamente l'insieme delle informazioni disponibili. Tuttavia, nella realtà il processo di incorporazione delle notizie nei prezzi può non essere immediato o uniforme e questo è dovuto alla capacità limitata di elaborazione degli investitori, che possono attribuire un diverso grado di attenzione alle informazioni e possono interpretarne gli effetti in maniera differente.

L'utilizzo di notizie finanziarie permette quindi di rappresentare il sentiment associato ai singoli strumenti o ai rispettivi settori di appartenenza. Inoltre, essendo l'universo investibile del presente modello non costituito da singole azioni, ma da ETF

rappresentativi di differenti classi di attività e segmenti di mercato, le decisioni di allocazioni tra le diverse tipologie di strumenti possono essere influenzate non soltanto dalle informazioni specifiche relative ad una singola società, ma anche da notizie riguardanti l'inflazione, la politica monetaria, la crescita economica, il rischio geopolitico, i mercati delle materie prime e le aspettative sui tassi di interesse.

L'intento non è sostituire le variabili numeriche già presenti nel modello baseline, ma affiancare a esse una componente informativa differente e potenzialmente complementare. L'introduzione del sentiment non presuppone necessariamente che le notizie consentano di prevedere in modo sistematico i rendimenti futuri, il loro contributo può manifestarsi anche attraverso una migliore identificazione del regime di mercato, una più efficace valutazione del rischio o una diversa distribuzione dei pesi tra gli asset.

Dal punto di vista computazionale, l'estensione dello stato mediante variabili di sentiment è coerente con la natura sequenziale del problema. L'aggiunta delle informazioni ricavate dalle notizie permette solamente di costruire uno stato più ampio, all'interno del quale, ad ogni data di ribilanciamento, l'agente selezionerà un'azione di portafoglio.

Tuttavia, la scelta di inserire variabili non numeriche all'interno del modello comporta diverse difficoltà metodologiche. I testi finanziari sono dati non strutturati, caratterizzati da ambiguità semantica, dipendenza dal contesto e utilizzo di una terminologia specialistica.

Le tecniche tradizionali di sentiment analysis si basano spesso su dizionari predefiniti o su algoritmi addestrati mediante rappresentazioni semplificate del testo e sebbene tali metodi siano poco onerosi dal punto di vista computazionale, essi possono riscontrare difficoltà nel riconoscere relazioni, ambiguità e significati dipendenti dal contesto.

Per affrontare tali criticità, si è deciso di introdurre l'applicativo FinBERT, un modello linguistico derivato da BERT e specializzato nell'analisi del linguaggio finanziario. BERT è un algoritmo che produce rappresentazioni contestuali delle parole considerando congiuntamente gli elementi che precedono e seguono ciascun termine[31]. FinBERT adatta questa architettura al dominio economico-finanziario e consente di classificare un testo secondo tre categorie di sentiment: positivo, neutrale e negativo ed è stato sviluppato proprio per superare i limiti degli strumenti linguistici generici in un ambito nel quale la

[31] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018). bert: pre-training of deep bidirectional transformers for language understanding. arXiv.

terminologia specialistica rende particolarmente complessa la classificazione del sentiment[32].

Nel presente lavoro, FinBERT viene utilizzato esclusivamente come strumento di trasformazione dei dati testuali e non come componente diretta di addestramento dell'agente PPO. Il modello linguistico elabora preventivamente le notizie e restituisce, per ciascun testo, le probabilità associate alle tre classi di sentiment, queste vengono successivamente aggregate e convertite in variabili numeriche che entrano nello stato osservato dall'agente assieme alle variabili già utilizzati nel modello baseline.

All'interno del modello sono stati mantenuti invariati gli algoritmi di apprendimento, le funzioni di ricompensa, i vincoli e gli iperparametri, in modo da poter giudicare se l'integrazione di informazioni ricavate dal sentiment delle notizie finanziarie consente di migliorare le decisioni di allocazione apprese dal modello PPO rispetto all'utilizzo delle sole variabili quantitative. Il confronto non riguarda quindi algoritmi di reinforcement learining differenti, bensì due configurazioni dello stesso modello che si distinguono esclusivamente per il contenuto informativo dello stato.

6.2. Natural Language Processing

La crescente disponibilità di documenti digitali ha reso le informazioni testuali una componente sempre più rilevante dell'analisi economica e finanziaria. Da sempre gran parte delle informazioni utilizzate dagli investitori non è espressa in forma numerica, bensì sono contenute in articoli di giornale, comunicati societari, relazioni finanziari, dichiarazioni delle banche centrali, report di analisti e altri documenti, che per essere utilizzati all'interno di un modello quantitativo rendono necessaria la conversione in rappresentazioni strutturate interpretabili da un algoritmo.

In questo ambito il Natural Language Processing (NLP), rappresenta il settore dell'intelligenza artificiale che studia le metodologie attraverso le quali i sistemi informatici possono elaborare, rappresentare e analizzare il linguaggio umano, con l'obbiettivo di trasformare testi non strutturati in informazioni utilizzabili per compiti specifici.

A differenza dei dati numerici impiegati tradizionalmente nell'analisi finanziaria, un testo non presenta una struttura immediatamente utilizzabile, bensì richiede che le parole

[32] Araci, D. (2019). finbert: financial sentiment analysis with pre-trained language models. arXiv:1908.10063 [cs]. [online] Available at: https://arxiv.org/abs/1908.10063.

vengano recepite, rappresentate e interpretate tenendo conto delle relazioni che le legano con gli altri elementi della frase. Tale processo è particolarmente complesso poiché il linguaggio naturale può essere caratterizzato da ambiguità, negazioni, espressioni e significati che dipendono dal contesto dell'intero documento.

Nel contesto finanziario, il sentiment non coincide necessariamente con lo stato emotivo dell'autore, ma rappresenta piuttosto la valutazione economica implicita nel contenuto della notizia e delle potenziali conseguenze che l'informazione può determinare per il mercato. Questa dipendenza dal contesto distingue l'analisi del sentiment finanziario dalle applicazioni di NLP più comuni, dove il sentiment può essere immediatamente riconosciuto dalla presenza di parole positive o negative. Il linguaggio economico-finanziario tende ad essere più tecnico, indiretto ed informativo e le informazioni recepite devono essere pesate anche rispetto alle aspettative del mercato.

Ad esempio, l'affermazione secondo cui i ricavi di una società sono aumentati del 5%, può non essere classificata correttamente senza considerare il contesto. Se gli analisti prevedevano una crescita del 2%, il risultato può essere interpretato come favorevole; d'altra parte, se le attese erano pari al 10%, la stessa informazione può generare una reazione negativa.

I primi metodi di NLP si basavano principalmente su dizionari lessicali, nei quali venivano utilizzati elenchi di parole classificate come positive, negative o neutre e il sentiment veniva determinato come frequenza di questi termini. Tuttavia, la loro efficacia dipendeva dalla qualità e dalla pertinenza del dizionario utilizzato; inoltre, la corretta gestione delle negazioni, richiedeva regole aggiuntive che appesantivano il processo di analisi.

Ad oggi, i metodi di NLP trasformano il testo in una rappresentazione numerica che possa essere elaborata da un algoritmo utilizzando dei word embeddings, ossia dei vettori numerici associati ad ogni parola. Gli embeddings tradizionali assegnavano ad ogni parola lo stesso vettore, indipendentemente dalla frase in cui compariva, ma i modelli di NLP più recenti adottano rappresentazioni contestuali, nelle quali il significato di ciascun termine viene determinato considerando l'intera frequenza della frase.

Formalmente, data una frase composta dalle parole:

$$X = (x_1, x_2, \dots, x_n)$$

Il modello genera un insieme di vettori:

$$H = (h_1, h_2, \dots, h_n)$$

dove ogni rappresentazione h_i dipende sia dalla parola x_i, sia dalle altre parole presenti nella stessa frase. In questo modo uno stesso termine può assumere significati differenti in base al contesto nel quale viene utilizzato.

La costruzione di queste rappresentazioni contestuali è stata resa possibile soprattutto dall'architettura Transformer[33], che a differenza delle precedenti reti ricorrenti analizza direttamente le relazioni tra tutti i termini della sequenza attraverso il meccanismo della self-attention. La self-attention attribuisce ad ogni parola un diverso livello di importanza rispetto alle altre parole della frase, consentendo di determinare quanto una parola sia rilevante per interpretarne un'altra.

Su questa architettura si basa il modello di Biderectional Encoder Rappresentation from Transformers (BERT), che riesce a costruire rappresentazione bidirezionali, considerando contemporaneamente sia le parole che precedono sia quelle che seguono il termine analizzato.

Successivamente, nel caso della sentiment analysis, la rappresentazione complessiva del testo viene fornita dall'analisi delle probabilità delle diverse classi di sentiment:

$$p_i = [p_i^+, p_i^0, p_i^-] \quad con \;\; p_i^+ + p_i^0 + p_i^- = 1$$

dove: $p_i^+, p_i^0 \; e \; p_i^-$ rappresentano rispettivamente le probabilità che il testo esprima un sentiment positivo, neutrale o negativo e il sentiment finale corrisponde a quello caratterizzato dalla probabilità più elevata.

BERT venne originariamente pre-addestrato su un corpo di documenti linguistici generali, per questo motivo sebbene sia in grado di interpretare il contesto di un documento, può incontrare difficoltà quando viene applicato a testi caratterizzati da una terminologia altamente specialistica come quella del linguaggio finanziario.

Da questa esigenza nasce FinBERT, un modello derivato da BERT e adatto specificatamente a trattare testi economico-finanziari e alla classificazione del relativo

[33] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017). Attention Is All You Need. [online] Cornell University. Available at: https://arxiv.org/abs/1706.03762.

sentiment. Il modello sfrutta le conoscenze linguistiche già apprese da BERT e le specializza attraverso un ulteriore addestramento su testi finanziari, migliorando sensibilmente i risultati rispetto ai tradizionali metodi applicati allo stesso compito.

6.3. Dataset di notizie finanziarie utilizzato

Per integrare il sentiment all'interno del modello è stato utilizzato il dataset fnspid[34], un dataset ampio e strutturato che costituisce una base informativa omogenea, necessaria per l'addestramento di FinBERT lungo l'intero periodo considerato.

Il dataset originario comprende complessivamente oltre 15,6 milioni di osservazioni, dal 1999 al 2023. In fase di preparazione si è limitato il periodo di analisi all'intervallo compreso tra il 1° gennaio 2014 e il 31 dicembre 2023; inoltre, sono stati eliminati gli articoli caratterizzati da testi mancanti e non in lingua inglese, poiché il modello FinBERT utilizzato è stato addestrato per elaborare esclusivamente testi in lingua inglese.

Un passaggio particolarmente importante ha riguardato la deduplicazione degli articoli. Lo stesso articolo poteva infatti comparire più volte nel dataset, poiché associato a ticker diversi. Le osservazioni sono quindi state considerate duplicate quando presentavano lo stesso timestamp originario e lo stesso contenuto testuale, mentre le diverse associazioni ai ticker sono state conservate unendo le copie dello stesso articolo in un'unica osservazione.

Al termine della procedura sono stati ottenuti 4.442.301 articoli unici tra il 2014 e il 2023, di cui 2.465.870 associati ad almeno un ticker dell'universo investibile.

Nel presente lavoro, FinBERT è stato applicato esclusivamente al titolo della notizia. Questa scelta presenta diversi vantaggi: in primo luogo, il titolo sintetizza generalmente l'informazione principale dell'articolo e rappresenta la parte immediatamente più osservabile dagli investitori; in secondo luogo, l'utilizzo dei soli titoli riduce notevolmente il costo computazionale dell'elaborazione. D'altra parte, l'utilizzo dei soli titoli può portare inevitabilmente alla perdita di alcune informazioni, alcune notizie possono presentare formulazioni troppo sintetiche o ambigue e possono non rappresentare correttamente l'intero articolo. Tale scelta rappresenta dunque un trade-off tra ricchezza informativa e sostenibilità computazionale.

[34] Dong, Z., Fan, X. and Peng, Z. (2024). fnspid: a comprehensive financial news dataset in time series. [online] arXiv.org. Available at: https://arxiv.org/abs/2402.06698.

La copertura temporale del dataset è limitata al 31 dicembre 2023, per questo motivo il modello integrato con FinBERT non è stato valutato fino al 2026, come avvenuto per la configurazione baseline originaria. L'estensione oltre il 2023 avrebbe richiesto l'unione di più dataset differenti, che avrebbe potuto generare un cambiamento nella distribuzione dei dati, nella qualità e nelle modalità di raccolta delle notizie, rendendo difficile stabilire se eventuali variazioni nelle prestazioni dipendessero dall'effettivo sentiment o dalla diversa struttura della fonte utilizzata. Per preservare l'omogeneità e la riproducibilità dell'esperimento, si è quindi scelto di mantenere la copertura originaria del dataset e di terminare l'analisi nel 2023.

Ai fini del confronto, anche il modello baseline è stato nuovamente addestrato e valutato sul medesimo intervallo temporale. In questo modo, le due configurazioni condividono lo stesso periodo di training, validation e test.

6.4. Implementazione di FinBERT e metodologie

Nel presente lavoro è stato utilizzato il modello pre-addestrato ProsusAI/finbert, ulteriormente addestrato su un corpus finanziario più coerente con il dataset a disposizione.

A differenza dei metodi tradizionali, che limitano l'analisi del sentiment alla classificazione in tre categorie: positiva, neutrale e negativa, attribuendo al testo la classe associata alla probabilità più elevata, nel presente modello le probabilità prodotte dal modello vengono conservate, poiché consentono di rappresentare anche l'intensità e il grado di incertezza della classificazione.

A partire dalle probabilità, viene quindi costruito uno score continuo di sentiment:

$$SentimentScore = p_i^+ - p_i^-$$

Lo score può assumere valori compresi tra -1 e +1, dove valore prossimi a +1 indicano una forte prevalenza della componente positiva, mentre valori prossimi a -1 indicano una marcata componente negativa. I valori vicini allo zero possono invece derivare da una classificazione prevalentemente neutrale, oppure da una maggiore incertezza tra la classe positiva e negativa.

Una volta calcolato il sentiment ogni notizia viene associata alla prima seduta nella quale l'informazione può essere effettivamente utilizzata dall'agente, in questo modo si evita

che l'agente utilizzi informazioni che non erano ancora note al momento della decisione di portafoglio.

I timestamp delle notizie sono stati convertiti al fuso orario di New York, dove le notizie pubblicate in un giorno di negoziazione sono state considerate disponibili alla chiusura della seduta stessa, mentre le notizie pubblicate durante il fine settimana o in un giorno di chiusura del mercato sono state trasferite alla seduta successiva. I timestamp registati esattamente alle ore 00:00 sono stati trattati in modo prudenziale, considerando l'informazione disponibile esclusivamente dalla seduta successiva.

Successivamente all'allineamento con il calendario di negoziazione, le notizie sono state aggregate utilizzando una finestra mobile pari a cinque sedute di mercato, in modo coerente alla frequenza settimanale di ribilanciamento adottata dal modello principale. In questo modo è possibile calcolare il sentiment medio dato dall'insieme delle notizie disponibili nelle cinque sedute concluse entro la data t come:

$$SentimentMean_t = \frac{1}{N_t} * \sum_i \mathcal{N}_{t,5} * SentimentScore_i$$

dove N_t rappresenta il numero di notizie comprese nella finestra e $\mathcal{N}_{t,5}$ rappresenta l'insieme delle notizie disponibili nelle cinque sedute.

Accanto al sentiment è stato mantenuto anche il volume informativo, misurato attraverso il numero di articoli pubblicati nella finestra temporale. Inoltre, poiché tale variabile può assumere valori molto elevati e può presentare una distribuzione fortemente asimmetrica in base al periodo di riferimento, è stata applicata la trasformazione logaritmica:

$$LogNewsCount_t = \log(1 + N_t)$$

Nel conteggio, l'aggiunta dell'unità consente di attribuire peso pari a zero alle finestre prive di notizie.

Infine, per stabilire a quali strumenti del portafoglio l'informazione debba essere collegata è stato necessario costruire una procedura di associazione dedicata, in quanto tale operazione non può basarsi esclusivamente sul ticker riportato nel dataset, poiché quest'ultimo identifica frequentemente una singola società, mentre l'universo investibile è composto da ETF rappresentativi di mercati, settori, aree geografiche e classi di attività differenti.

Per ciascuno dei dodici ETF sono stati individuati:

- Ticker diretto dell'ETF
- Ticker societari rappresentativi
- Insieme di parole chiave riferite all'esposizione economica di base

Una notizia viene quindi associata ad un ETF quando trova riscontro in uno dei tre punti precedentemente elencati. In quest'ambito, il confronto dei ticker utilizza quelli già presenti nel database, mentre le parole chiave vengono ricercate nel titolo della notizia. Inoltre, l'assegnazione può essere multipla, poiché una stessa notizia può risultare rilevante per più ETF contemporaneamente.

È importante sottolineare come i criteri utilizzati non mirano a ricostruire le composizioni storiche di ciascun ETF in modo completo. Per farlo sarebbe necessario disporre delle effettive partecipazioni per ogni istante temporale, d'altra parte utilizzare retroattivamente la composizione attuale dell'ETF introdurrebbe dei bias poiché alcune società presenti oggi nel fondo potrebbero non averne fatto parte nei precedenti anni. Per questo motivo si è preferito utilizzare un insieme di ticker e parole chiave relativamente stabili nel tempo e rappresentativi dell'ETF analizzato.

Questa scelta presenta comunque alcuni limiti, il sistema può infatti non riconoscere articoli rilevanti che utilizzano formulazioni differenti da quelle incluse nel sistema, oppure può associare una notizia ad un ETF anche quando questa in realtà non produce un effetto significativo.

Nonostante tali criticità, il risultato di tale processo consente di ottenere variabili giornaliere che sintetizzano il sentiment e il volume delle notizie disponibili nelle cinque sedute precedenti, sia a livello di mercato complessivo, sia con riferimento alle esposizioni dei singoli ETF

6.5. Integrazione del sentiment nello stato del modello PPO

Una volta costruite le variabili di sentiment a livello generale e per ciascun ETF, queste vengono inserite nello stato osservato dall'agente PPO con l'obbiettivo di ampliare l'insieme informativo disponibile al momento della decisione di portafoglio.

Nel modello originario, lo stato era composto da 138 variabili quantitative, relative all'andamento storico degli asset, agli indicatori tecnici, alle misure di rischio e alle

condizioni complessive del mercato. Con l'introduzione di FinBERT vengono aggiunte allo stato 27 nuove feature, portando la dimensione complessiva a 165 variabili.

Le nuove informazioni comprendono, per ciascuno dei 12 ETF:

- Sentiment medio delle notizie associate
- Logaritmo del numero di notizie associate

Queste variabili producono complessivamente 24 feature, alle quali si aggiungono tre variabili riferite all'intero mercato:

- Sentiment medio complessivo
- Probabilità negativa media
- Logaritmo del numero totale di notizie

La componente relativa al singolo ETF consente all'agente di distinguere le informazioni per le diverse esposizioni nel portafoglio, mentre il sentiment generale fornisce una misura sintetica del clima informativo del mercato.

Il numero di notizie viene mantenuto come variabile separata perché può contribuire a dare un significato differente al sentiment medio in base alla quantità di informazioni disponibili.

La probabilità media negativa viene invece inserita tra le feature generali per rappresentare in modo più diretto la presenza di notizie sfavorevoli. In questo modo, è possibile differenziare gli scenari con score neutri dati dalla contemporanea presenza di notizie positive e negative o dalla presenza di notizie prevalentemente neutrali.

L'integrazione avviene mediante l'allineamento delle feature di sentiment con le date di osservazione utilizzate dall'ambiente dove, ad ogni data di ribilanciamento, l'agente riceve solamente le informazioni risultati dalle notizie disponibili fino a quel momento, aggregate sulla finestra temporale definita in precedenza.

Per valutare correttamente il contributo delle nuove feature, il modello è stato confrontato con una nuova configurazione del modello baseline basata sul medesimo orizzonte temporale. Questa scelta si è resa necessaria poiché il dataset di notizie termina il 31 dicembre 2023, mentre l'esperimento baseline descritto nel capitolo precedente disponeva di dati finanziari fino al 2026, dunque un confronto diretto tra due modelli su periodi differenti non sarebbe stato metodologicamente corretto.

Entrambe le configurazioni sono pertanto state addestrate e valutate utilizzando gli stessi intervalli temporali:

- Training set: 2015-2020
- Validation set: 2021-2022
- Test set: 1° gennaio 2023- 31 dicembre 2023

Il modello integrato con FinBERT mantiene invariati rispetto alla baseline l'algoritmo PPO, l'architettura MLP, la funzione di ricompensa, i vincoli di portafoglio, la frequenza di ribilanciamento, i costi di transazione e i principali iperparametri. La differenza tra le due configurazioni riguarda quindi esclusivamente il contenuto dello stato, che passa da 138 a 165 variabili mediante l'aggiunta delle 27 feature derivate dalle notizie

Per ridurre la dipendenza dei risultati dall'inizializzazione casuale, l'esperimento è stato ripetuto utilizzando i cinque seed usati anche in precedenza: 1, 7, 21, 42 e 100, dove per ciascun seed, il modello è stato addestrato per 50.000 timestep con checkpoint ogni 1.024, utilizzando sempre la metrica sharpe drowdown score.

6.6. Risultati in validation

Come per il modello baseline la fase di validation copre il periodo compreso tra il 1° gennaio 2021 e il 31 dicembre 2022, ed è stata utilizzata per confrontare i diversi checkpoint prodotti durante l'addestramento e selezionare la configurazione da valutare successivamente sul periodo di test. Tra le simulazioni effettuate, il miglior risultato del modello è stato ottenuto con il seed 7, al checkpoint corrispondente a 39.936 timestep.

Il confronto con la configurazione baseline sullo stesso periodo è riportato nella tabella che segue:

Modello	Rendimento cumulato	Volatilità annualizzata	Sharpe ratio	Sortino ratio	Max drawdown	Turnover medio	Costi totali
Baseline	14,89%	13,78%	0,592	1,031	-11,61%	34,04%	3,40%
Baseline + FinBERT	26,80%	11,19%	1,16	1,82	-6.37%	64,60%	6,46%

Tabella 6.1: risultati in fase di validation

L'aggiunta delle variabili di sentiment determina un miglioramento significativo delle principali metriche. In particolare, il rendimento cumulato aumenta dal 14,89% al 26,80%, con una differenza positiva di 11,91 punti percentuali, mentre la volatilità

annualizzata diminuisce dal 13,78% all'11,19%. Il miglioramento risulta evidente anche nelle misure corrette per il rischio, dove lo Sharpe ratio aumenta da 0,59 a 1,16, mentre il Sortino ratio aumenta da 1,03 a 1,82. Il modello con FinBERT produce quindi un rendimento maggiore per unità di rischio complessivo e una performance migliore rispetto alla sola volatilità negativa.

Un altro aspetto rilevante è la riduzione del massimo drawdown, che passa da -11,61% a solamente -6,37%, dimostrando come l'integrazione del sentiment nel periodo di validation consente non solo al modello di ottenere rendimenti più elevati, ma anche di limitare maggiormente le perdite nelle fasi sfavorevoli di mercato.

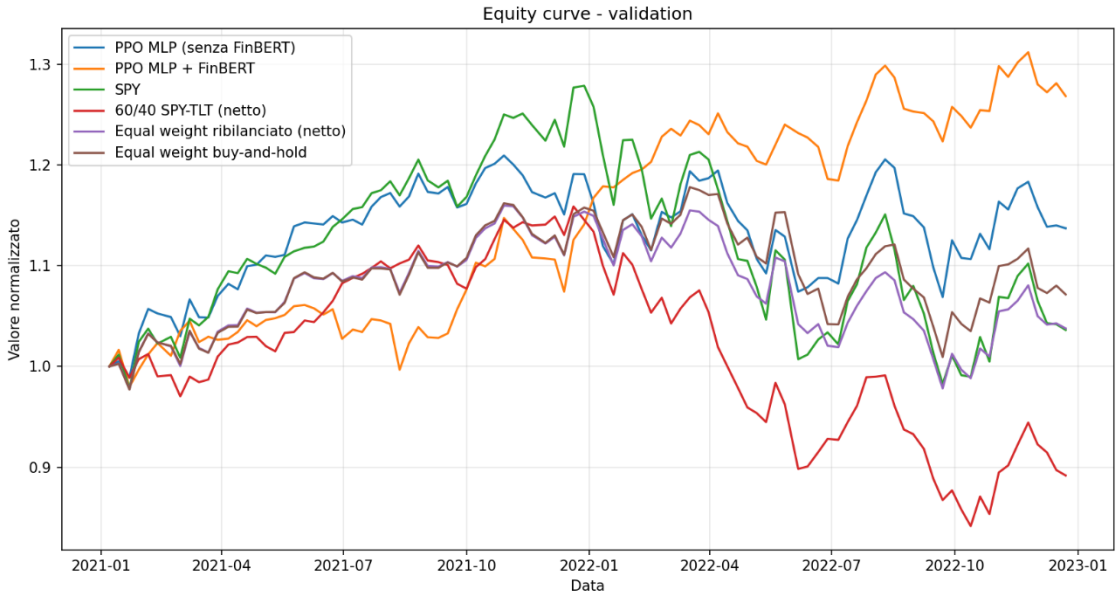

Figura 6.1: equity curves in fase di validation

La figura 6.1 mostra come il modello con FinBERT, dopo un andamento iniziale peggiore rispetto alla baseline, inizia a sovraperformare in modo netto nel corso del 2022. Nella fase di maggiore debolezza dei benchmark, la curva del modello risulta molto più robusta e riesce a chiudere il periodo con un rendimento sensibilmente superiore rispetto agli altri.

D'altra parte, il miglioramento delle prestazioni è accompagnato da una crescita della frequenza di negoziazione, con un turnover medio che aumenta dal 34% al 64,6% portando i costi complessivi di transazione dal 3,40% al 6,46% del capitale iniziale.

Tale risultato può essere interpretato come una maggiore reattività del modello data dal flusso informativo proveniente dalle notizie, che induce l'agente a rivedere più

frequentemente le proprie scelte di allocazione, aumentando così la propria capacità di adattamento.

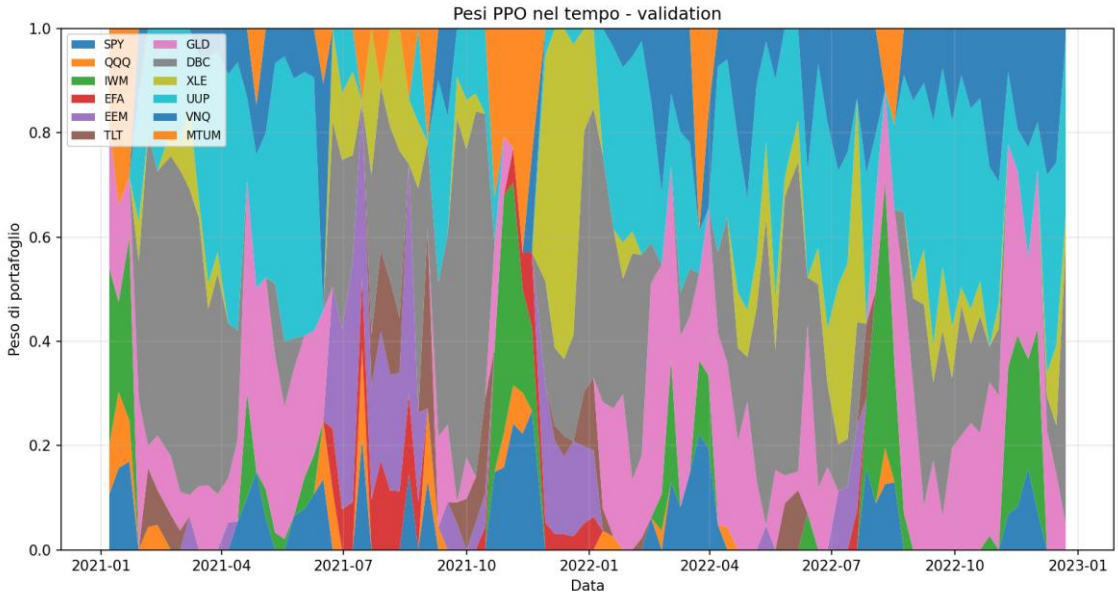

Figura 6.2: pesi di portafoglio in fase di validation

Nel complesso, i risultati in validation indicano che l'integrazione di FinBERT migliora sensibilmente la configurazione baseline in termini di rendimento, rischio e drawdown. Tali evidenze devono tuttavia essere interpretate con cautela poiché, come visto nel capitolo precedente, il validation set partecipa alla selezione del miglior modello. Per questo motivo, la verifica principale delle capacità del modello deve essere effettuata sul periodo di test, rimasto estraneo all'addestramento e alla selezione del modello.

6.7. Risultati in test

Il test set ricompre il periodo compreso tra il 1° gennaio e il 31 dicembre 2023 e rappresenta la verifica principale delle capacità di generalizzazione delle politiche apprese dal modello.

Il confronto tra la configurazione baseline e il modello integrato con FinBERT è riportato nella tabella seguente:

Modello	Rendimento cumulato	Volatilità annualizzata	Sharpe ratio	Sortino ratio	Max drawdown	Turnover medio	Costi totali
Baseline	13,02%	9,47%	1,39	1,77	−5,85%	36,8%	1,84%
Baseline + FinBERT	16,37%	9,67%	1,68	3,35	−6,04%	53%	2,65%

Tabella 6.2: risultati in fase di test

Anche nel periodo di test l'integrazione delle feature informative produce un rendimento superiore rispetto al modello baseline, con un rendimento cumulato che passa dal 13,02% al 16,37%, mentre la volatilità annualizzata aumenta solo marginalmente, passando dal 9,47% al 9,67%. Questo aumento di rendimento, a parità di volatilità, si traduce in un miglioramento dello Sharpe ratio, che passa da 1,39 a 1,68, risultato che indica che il nuovo modello genera una performance migliore anche dopo aver considerato il rischio complessivamente sostenuto.

Il miglioramento più marcato riguarda però il Sortino ratio, che aumenta da 1,77 a 3,35. Tale risultato suggerisce che la nuova configurazione abbia prodotto una distribuzione dei rendimenti più favorevole nelle fasi di ribasso.

Peggiora invece il massimo drawdown, passando da -5,85% a -6,04%. Nonostante la differenzia sia contenuta, non si confermano i risultati osservati in validation dove l'introduzione del sentiment riduceva in maniera significativa la perdita massima, bensì questa viene quasi mantenuta analoga rispetto a quella del modello baseline.

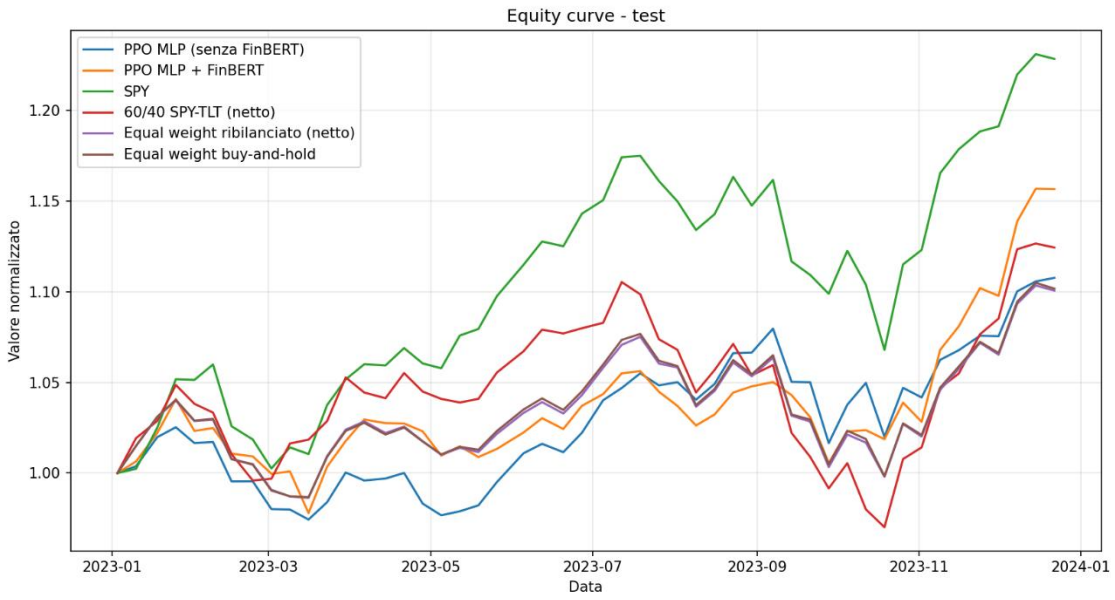

Figura 6.3: equity curves in fase di test

La figura 6.3 mostra come il modello con FinBERT mantenga un andamento complessivamente migliore rispetto alla baseline, chiudendo il periodo con un rendimento cumulato finale più elevato. La distanza tra le due curve rimane tuttavia più contenuta rispetto a quanto osservato in validation, segnalando un vantaggio fuori campione meno marcato.

Si conferma invece anche sul test set la maggiore attività di negoziazione del modello con FinBERT, il turnover medio aumenta infatti dal 36,8% al 53%, mentre i costi totali di transazione passano dall'1,84% al 2,65% del capitale inziale. Nonostante l'incremento dei costi, il nuovo modello mantiene un vantaggio nel rendimento cumulato, già al netto dei costi di transazione, dimostrando come la migliore performance non venga annullata dalla maggiore frequenza di negoziazione.

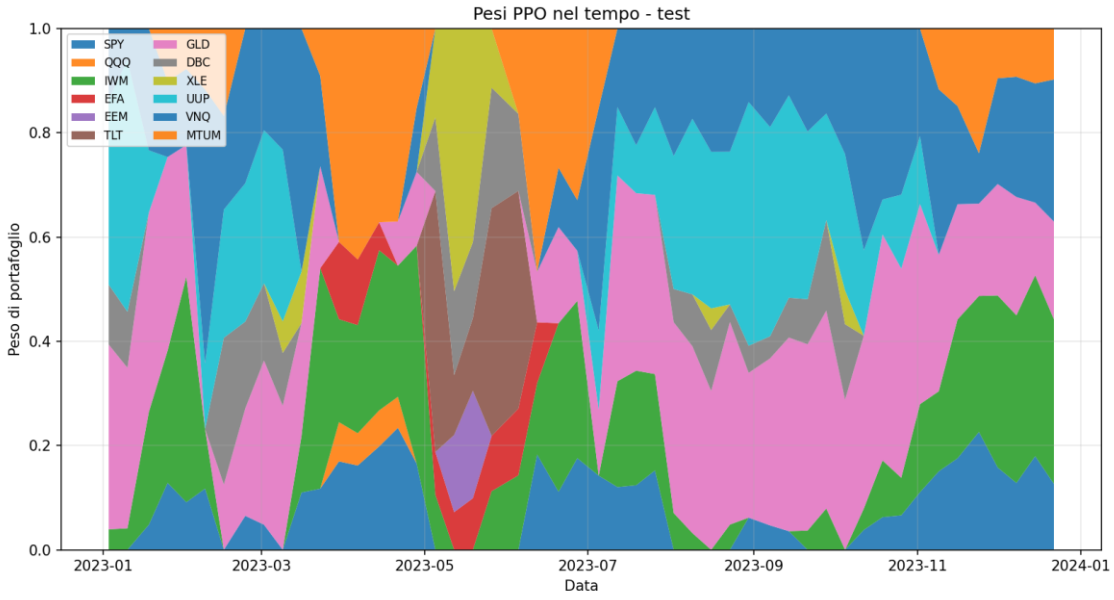

Figura 6.4: pesi di portafoglio in fase di test

Per verificare se effettivamente l'introduzione del sentiment porti un vantaggio al modello baseline è utile considerare la variabilità delle cinque inizializzazioni casuali, in modo da verificare se le migliori performance date dal nuovo modello dipendano dal modello selezionato o siano consistenti in altri test. Confrontando i risultati ottenuti dagli stessi seed, il modello con FinBERT supera il corrispondente baseline in termini di rendimento cumulato netto in quattro casi su cinque, ciò suggerisce come il vantaggio informativo di

FinBERT non dipenda esclusivamente dal singolo modello selezionato ma risulti consolidato nelle diverse simulazioni.

Nel complesso, il test 2023 fornisce evidenza favorevole all'integrazione delle informazioni testuali, soprattutto in termini di crescita del rendimento e nel miglioramento delle metriche corrette per il rischio, a fronte di un aumento delle attività di negoziazione e di un drawdown sostanzialmente invariato.

6.8. Valutazione del modello

I risultati complessivi mostrano come l'integrazione delle informazioni testuali mediante FinBERT migliori complessivamente le prestazioni del modello rispetto alla configurazione baseline, sia sul validation set che sul test set. In particolare, si può notare un chiaro aumento nel rendimento cumulato e nelle metriche di rendimento corretto per il rischio in entrambi set, mentre rimane ambiguo il massimo drawdown che migliora ampiamente in fase di validation ma peggiora marginalmente in fase di test. L'aggiunta del sentiment sembra quindi produrre soprattutto un incremento del rischio e una migliore gestione della volatilità negativa, senza però modificare in modo significativo il rischio complessivo del portafoglio.

Da evidenziare come il miglioramento non risulti limitato al singolo modello selezionato, bensì nel confronto tra le cinque inizializzazioni casuali in fase di test, la configurazione con FinBERT supera la corrispondete baseline in termini di rendimento cumulato in quattro casi su cinque, suggerendo come il contributo delle informazioni testuali non dipenda esclusivamente da una particolare inizializzazione.

Il confronto con i restanti benchmark suggerisce tuttavia una valutazione più moderata, in particolare nel periodo di test, seppur il periodo di test migliori i risultati in termini di rendimento rispetto al modello baseline, non riesce a superare il rendimento cumulato dell'investimento passivo in SPY. Tuttavia, come già analizzato per il modello baseline, la scelta di un periodo particolarmente favorevole alla crescita del mercato azionario statunitense non favorisce il confronto con una strategia multi-asset vincolata alla diversificazione. D'altra parte, il modello con FinBERT presenta una volatilità annualizzata inferiore rispetto a SPY, un massimo drawdown più contenuto e soprattutto un Sortino ratio superiore. Ciò indica che nonostante il modello non superi SPY in rendimento assoluto, questo riesce comunque ad offrire un profilo maggiormente difensivo e una migliore efficienza rispetto alla volatilità negativa.

Il confronto risulta invece più favorevole rispetto ai benchmark diversificati: la configurazione con FinBERT supera in termini di rendimento cumulato tutte le strategie multi-asset considerate e presenta contemporaneamente migliori valori di Sharpe ratio, Sortino ratio e massimo drawdown. Questi risultati mettono in evidenza il contributo della sentiment analysis quando il modello viene confrontato con strategie caratterizzate da un livello di diversificazione simile.

Il miglioramento delle prestazioni è tuttavia accompagnato da una maggiore attività di negoziazione rispetto al modello baseline, che porta a quasi raddoppiare il turnover medio e i costi di transazione. Nonostante ciò, il vantaggio rispetto alla baseline e ai benchmark diversificati non viene completamente assorbito dalla maggiore negoziazione, ma ciò non elimina il rischio che, in un contesto reale, costi di transazione maggiori possano ridurre ulteriormente il beneficio osservato.

L'esperimento dimostra quindi come, mantenendo invariato l'algoritmo PPO e il resto della struttura sperimentale, l'aggiunta di informazioni testuali, anche se limitate al solo titolo, può migliorare la configurazione baseline e produrre risultati competitivi rispetto a semplici strategie diversificate. Tuttavia, l'esperimento non dimostra che il modello sia in grado di superare sistematicamente il mercato e i risultati ottenuti richiedono verifiche su periodi più lunghi e differenti condizioni di mercato.

CONCLUSIONI:

Il presente elaborato ha analizzato la possibilità di applicare le tecniche di reinforcement learning alla gestione dinamica di portafoglio, concentrandosi in particolare sugli algoritmi di Proximal Policy Optimization. Il punto di partenza dell'analisi è stata la considerazione che i mercati finanziari non siano necessariamente ambienti caratterizzati da relazioni stabili nel tempo, bensì queste possono modificarsi in funzione delle condizioni del mercato e del comportamento degli investitori, rendendo particolarmente complessa l'applicazione di strategie statiche di allocazione del capitale.

In questo senso, le Adaptive Markets Hypothesis hanno costituito il principale riferimento teorico alla base dell'elaborato, nel quale i mercati vengono rappresentati come sistemi nei quali strategie ed operatori sono costretti a adattarsi progressivamente all'ambiente circostante. Questa rappresentazione dei mercati trova una naturale corrispondenza nel funzionamento del reinforcement learning, all'interno del quale un agente modifica il proprio comportamento attraverso l'interazione continua con l'ambiente e sulla base dei risultati ottenuti. L'obbiettivo dell'analisi empirica non è stato pertanto quello di individuare una regola di investimento universalmente ottimale, bensì di verificare se un algoritmo caratterizzato da capacità di apprendimento e adattamento potesse apprendere una politica di allocazione del capitale competitiva in presenza di condizioni di mercato differenti.

La prima parte dell'elaborato ha riguardato lo sviluppo di un modello baseline PPO-MLP applicato ad un universo multi-asset composto da dodici ETF in un ambiente caratterizzato da elementi puramente quantitativi. In questo contesto l'agente opera esclusivamente attraverso posizioni long, mediante ribilanciamenti a frequenza settimanale e sostenendo costi di transazione proporzionali alla variazione dei pesi, seguendo una funzione di ricompensa che combina il rendimento netto con misure di performance corrette per il rischio, come Sharpe Ratio e Sortino ratio, per evirare che l'apprendimento venga orientato esclusivamente alla massimizzazione del rendimento assoluto.

I risultati ottenuti da questo primo modello evidenziano tanto le potenzialità quanto i limiti di questo approccio. Nel periodo di validation 2021-2022 la strategia PPO ottiene dei risultati particolarmente favorevoli, registrando valori complessivamente migliori

rispetto ai benchmark considerati, dimostrando come in questa fase la politica appresa sia stata in grado di reagire efficacemente anche ad un periodo caratterizzato da condizioni di mercato complesse, soprattutto durante il ribasso del 2022. Per quanto riguarda invece la valutazione sul test set, questa risulta più articolata e complessa e rappresenta proprio per questo motivo uno degli elementi più rilevanti dell'analisi. Nel periodo compreso tra il 2023 e il 2026 il modello registra rendimenti cumulati superiori ai benchmark equally weighted e al portafoglio 60/40, ma non riesce a replicare la crescita dello SPY; inoltre, il modello non riesce a superare sistematicamente il modello equal weight ribilanciato nelle principali metriche corrette per il rischio. D'altra parte, la volatilità e il massimo drawdown del modello risultano comunque inferiori rispetto allo SPY, evidenziando una maggiore prudenza della strategia multi-asset.

Questo risultato suggerisce una prima conclusione importante: la maggiore complessità di un modello di reinforcement learing non determina automaticamente una superiorità rispetto a semplici strategie passive tradizionali. La PPO dimostra di poter apprendere una politica economicamente interpretabile e capace di modificare dinamicamente e con profitto la composizione del portafoglio, ma il beneficio dell'adattività deve essere confrontato tenendo conto dei costi operativi estremamente inferiori delle strategie più tradizionali. Tale osservazione assume particolare rilievo considerando il turnover medio prodotto dal modello, pari al 30,16% durante il periodo di tesi, che determina costi di transazione pari al 5,19% del capitale. Sebbene tali costi non annullino la performance positiva del modello, evidenziano comunque come l'efficienza operativa rappresenti uno dei principali aspetti da migliorare in applicazioni reali.

La seconda parte dell'analisi ha cercato di verificare se le prestazioni del modello potessero essere migliorate ampliando l'informazione disponibile per l'agente al momento della decisione. A tale scopo è stato implementato l'applicativo FinBERT, utilizzato per trasformare notizie finanziarie pubbliche in misure quantitative di sentiment. L'algoritmo PPO, la rete MLP, la funzione di ricompensa e i principali parametri dell'esperimento sono stati mantenuti invariati, rendendo possibile il confronto tra i due modelli.

I risultati ottenuti forniscono indicazioni favorevoli al contributo del sentiment. Nel validation set, l'aggiunta delle informazioni prodotte dalle notizie finanziarie porta ad un aumento del rendimento cumulato, che passa dal 14,89% al 26,80%, riducendo

contemporaneamente la volatilità e il massimo drawdown e migliorando sensibilmente le misure corrette per il rischio. Particolare rilevanza assume però la verifica effettuata sul test 2023, dove anche in questo caso la configurazione con FinBERT migliora il risultato del modello: il rendimento cumulato passa dal 13,02% al 16,37%, le misure corrette per il rischio aumentano sensibilmente e il massimo drawdown diminuisce leggermente.

Tuttavia, anche in questo caso i risultati devono essere interpretati con prudenza. La migliore capacità di adattamento prodotta dalle informazioni aggiuntive è accompagnata da un significativo incremento delle attività di negoziazione. In particolare, nel test 2023 il turnover medio passa dal 36,8% del modello baseline al 53% della configurazione con FinBERT, mentre i costi complessivi aumentano dall'1,84% al 2,65%. Il vantaggio ottenuto dal nuovo modello rimane positivo anche la netto di tali costi, ma l'incremento evidenzia un trade-off particolarmente importante tra capacità di reazione alle nuove informazioni e stabilità nell'allocazione del capitale.

Il confronto con i benchmark porta, infine, ad una valutazione più completa dei risultati ottenuti. Nemmeno la configurazione integrata con FinBERT dimostra di poter superare lo SPU in termini di rendimento assoluto, ma probabilmente questo è dovuto ad un periodo particolarmente favorevole al mercato azionario statunitense. D'altra parte, il modello presenta un profilo di rischio più difensivo e risultati particolarmente favorevoli se confrontati con strategie tradizionali multi-asset, suggerendo come il valore del modello emerga maggiormente quando questo viene confrontato con strategie e portafogli caratterizzati da analoghi obbiettivi di diversificazione.

Rimangono comunque da sottolineare alcuni inevitabili limiti operativi: in primo luogo, l'utilizzo di un universo composto da dodici ETF, seppur diversificati, costituisce una semplificazione rispetto alla varietà di strumenti effettivamente disponibili sul mercato; il modello non contempla posizioni corte, leva finanziaria o componenti di liquidità, inoltre limita l'allocazione ad un numero massimo di strumenti in contemporanea; i costi di negoziazione sono rappresentati mediante un coefficiente costante semplificativo e non incorporano integralmente tutti i possibili costi di una reale transazione nel mercato; infine, per quanto riguarda l'applicativo FinBERT, la disponibilità del dataset di notizie limita il test fino al 2023, inoltre l'analisi è stata effettuata sui titoli degli articoli e l'associazione delle notizia agli ETF è necessariamente basata su criteri approssimativi.

Possibili sviluppi futuri potrebbero riguardare l'estensione dell'analisi ad un orizzonte temporale più lungo, l'utilizzo di più asset investibili e la verifica della strategia in diversi regimi di mercato. Particolare attenzione potrebbe essere dedicata alla riduzione del turnover attraverso vincoli più stringenti o alla formulazione della funzione di ricompensa maggiormente sensibili ai reali costi operativi. Per quanto riguarda l'integrazione del sentiment informativo, sarebbe inoltre possibile utilizzare il contenuto completo delle notizie, introdurre ulteriori fonti testuali e sviluppare procedure che aumentino la precisione nel collegamento delle notizie alle esposizioni effettivamente contenute negli ETF presenti nell'universo investibile.

In conclusione, le evidenze ottenute consentono di rispondere positivamente, seppur con le dovute precisazioni, agli interrogativi posti all'inizio del lavoro. La PPO si dimostra applicabile ad un problema di gestione dinamica di portafoglio e consente di apprendere politiche di allocazione dinamica, economicamente interpretabili e con risultati competitivi rispetto alle strategie di investimento più tradizionali. Allo stesso tempo, i risultati dimostrano come l'adattività non costituisce di per sé una garanzia di superiorità e che la maggiore complessità del modello deve poter essere giustificata da benefici sufficientemente elevati da compensarne i maggiori costi di negoziazione. Diversamente, l'integrazione delle notizie finanziarie attraverso FinBERT dimostra sicuramente il risultato più significativo all'interno dell'elaborato, dove l'ampiamento dello stato informativo migliora in modo consistente la configurazione baseline, soprattutto nella qualità dei rendimenti e nella gestione della volatilità negativa. Più che dimostrare la possibilità di superare in modo sistematico il mercato, il lavoro evidenzia come la combinazione tra apprendimento adattivo e informazioni eterogenee possa rappresentare una direzione promettente per lo sviluppo di modelli di gestione di portafoglio, sempre più capaci di reagire ad un ambiente finanziario complesso, dinamico ed in continua evoluzione.

Bibliografia consultata

Araci, D. (2019). finbert: financial sentiment analysis with pre-trained language models. *arXiv:1908.10063 [cs]*. [online] Available at: https://arxiv.org/abs/1908.10063.

Bertoluzzo, F. and Corazza, M. (2012). Testing Different Reinforcement Learning Configurations for Financial Trading: Introduction and Applications. *Procedia Economics and Finance*, 3, pp.68–77. doi:10.1016/S22125671(12)001220.

Carter, C.F. and Samuelson, P.A. (1950). Foundations of Economic Analysis. *The Economic Journal*, 60(238), p.351. doi:10.2307/2227059.

Chadi, M. and Mousannif, H. (2023). *Understanding Reinforcement Learning Algorithms: The Progress from Basic Qlearning to Proximal Policy Optimization*. doi:10.48550/arXiv.2304.00026.

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2018). *bert: pre-training of deep bidirectional transformers for language understanding*. arXiv.

Dong, Z., Fan, X. and Peng, Z. (2024). *fnspid: a comprehensive financial news dataset in time series*. [online] *arXiv.org*. Available at: https://arxiv.org/abs/2402.06698.

Fama, E. (1970). Efficient Capital markets: a Review of Theory and Empirical Work. *The Journal of Finance*, 25(2), pp.383–417. doi:https://doi.org/10.2307/2325486.

Gao, H., Kou, G., Liang, H., Zhang, H., Chao, X., congcong and Dong, Y. (2024). Machine learning in business and finance: a literature review and research opportunities. *Financial Innovation*, 10. doi:10.1186/s4085402400629z.

Goodfellow, I., Bengio, Y. and Courville, A. (2016). *Deep learning - Chapter 6 "Deep Feedforward Networks."* [online] Deeplearningbook.org. Available at: https://www.deeplearningbook.org/.

Grossman, S. and Stiglitz, J. (1980). On The Impossibility of Informationally Efficient Markets. *American Economic Review*, 70, pp.393–408.

Kabbani, T. and Duman, E. (2022). Deep Reinforcement Learning Approach for Trading Automation in The Stock Market. *IEEE Access*, PP, pp.1–1. doi:10.1109/ACCESS.2022.3203697.

Kahneman, D. and Tversky, A. (1979). Prospect theory: an Analysis of Decision Under Risk. *Econometrica*, 47(2), pp.263–292. doi:https://doi.org/10.2307/1914185.

Lo, A.W. (2004). *Adaptive markets hypothesis : market efficiency from an evolutionary perspective*. Cambridge, Ma: Mit Sloan School Of Management.

Markowitz, H. (1952). Portfolio Selection. *The Journal of Finance*, 7(1), pp.77–91. doi:https://doi.org/10.2307/2975974.

Schulman, J., Levine, S., Moritz, P., Jordan, M.I. and Abbeel, P. (2015). *Trust Region Policy Optimization*. [online] arXiv.org. Available at: https://arxiv.org/abs/1502.05477.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017). *Proximal Policy Optimization Algorithms*. [online] arXiv.org. Available at: https://arxiv.org/abs/1707.06347.

Shleifer, A. and Vishny, R.W. (1997). The Limits of Arbitrage. *The Journal of Finance*, 52(1), pp.35–55. doi:10.2307/2329555.

Simon, H.A. (1955). A Behavioral Model of Rational Choice. *The Quarterly Journal of Economics*, [online] 69(1), pp.99–118. doi:https://doi.org/10.2307/1884852.

Sutton, R.S. and Barto, A. (2018). *Reinforcement learning: An introduction*. 2nd ed. Cambridge, Ma ; London: The Mit Press.

Terven, J. (2025). Deep Reinforcement Learning: A Chronological Overview and Methods. *AI*, 6(3), p.46. doi:https://doi.org/10.3390/ai6030046.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L. and Polosukhin, I. (2017). *Attention Is All You Need*. [online] *Cornell University*. Available at: https://arxiv.org/abs/1706.03762.

Williams, R.J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4), pp.229–256. doi:https://doi.org/10.1007/BF00992696.

Appendice

Di seguito sono riportati alcuni dei passaggi di maggiore rilievo del codice Python, sviluppato per l'implementazione del modello.

A.1 – Costruzione delle feature di mercato

Costruzione delle variabili che compongono lo stato osservato dall'agente.

```
return_lag = config.return_lag_days
technical_window = config.technical_window_days
feature_blocks: dict[str, pd.DataFrame] = {}
# Rendimento giornaliero
feature_blocks[f"return_lag_{return_lag}d"] = daily_returns.shift(
    return_lag - 1)
# Volatilità giornaliera
high_low_ratio = (highs / lows).replace([np.inf, -np.inf], np.nan)
feature_blocks["volatility_1d"] = np.sqrt(
    (np.log(high_low_ratio) ** 2) / (4.0 * np.log(2.0)))
# Momentum a 21 sedute
feature_blocks[f"momentum_{technical_window}d"] = adjusted_close.pct_change(
    technical_window)
# Rapporto prezzo / media mobile a 21 sedute
moving_average = adjusted_close.rolling(technical_window).mean()
feature_blocks[f"ma_ratio_{technical_window}d"] = (
    adjusted_close / moving_average - 1.0)
# Drawdown rolling a 21 sedute
rolling_max = adjusted_close.rolling(technical_window).max()
```

```
feature_blocks[f"drawdown_{technical_window}d"] = (
    adjusted_close / rolling_max - 1.0)
# Volume giornaliero
volumes = daily_prices.loc[:, pd.IndexSlice[list(config.tickers), "Volume"]]
volumes = volumes.droplevel(1, axis=1).sort_index(axis=1)
feature_blocks["volume_1d"] = volumes
# Correlazioni rolling a 21 sedute tra coppie di ETF
lookback = config.technical_window_days
tickers = list(config.tickers)
correlation_features: dict[str, pd.Series] = {}
for left_index, left_ticker in enumerate(tickers):
    for right_ticker in tickers[left_index + 1 :]:
        feature_name = f"corr_{left_ticker}_{right_ticker}_{lookback}d"
        correlation_features[feature_name] = daily_returns[left_ticker].rolling(
            lookback).corr(daily_returns[right_ticker])
```

A.2 – Trasformazione dell'azione in pesi di portafoglio

Trasformazione degli score prodotti dall'agente in pesi di portafoglio attraverso funzione softmax.

```
def softmax_weights(
    action: np.ndarray,
    top_k: int | None = None,
    previous_weights: np.ndarray | None = None,
    min_score_improvement: float = 0.0,) -> np.ndarray:
    """Trasforma un vettore reale in pesi long-only con somma pari a 1"""
    action = np.asarray(action, dtype=np.float64)
```

```
    if action.ndim != 1:
        raise ValueError("L'azione deve essere un vettore monodimensionale.")
    n_assets = action.shape[0]
    if top_k is None or top_k >= n_assets:
        selected_indices = np.arange(n_assets)
    else:
        selected_indices = select_top_k_with_inertia(
            action=action,
            top_k=top_k,
            previous_weights=previous_weights,
            min_score_improvement=min_score_improvement )
    selected_action = action[selected_indices]
    shifted_action = selected_action - np.max(selected_action)
    exp_action = np.exp(shifted_action)
    selected_weights = exp_action / exp_action.sum()
    weights = np.zeros(n_assets, dtype=np.float64)
    weights[selected_indices] = selected_weights
    return weights.astype(np.float32)
```

A.3 – Calcolo del rendimento e della funzione di ricompensa

A partire dai pesi di portafoglio, vengono calcolate: metriche rolling, rendimento, turnover, costi di transazione e funzione di ricompensa utilizzata per l'addestramento dell'agente.

```
# Calcolo di Sharpe e Sortino rolling
def _calculate_rolling_reward_metrics(
    self,
```

```
    latest_net_return: float,
) -> tuple[float, float]:
    """ Calcola Sharpe e Sortino rolling usati nella reward"""
    returns = [*self.net_return_history, latest_net_return]
    window = self.config.reward_metric_window_rebalances
    if window > 0:
        returns = returns[-window:]
    if len(returns) < 2:
        return 0.0, 0.0
    returns_array = np.asarray(returns, dtype=np.float64)
    std = returns_array.std(ddof=1)
    sharpe = 0.0 if std == 0.0 else float(returns_array.mean() / std)
    downside_returns = returns_array[returns_array < 0.0]
    if len(downside_returns) < 2:
        sortino = 0.0
    else:
        downside_std = downside_returns.std(ddof=1)
        sortino = (
            0.0
            if downside_std == 0.0
            else float(returns_array.mean() / downside_std))
    return sharpe, sortino
# Conversione dell'azione continua del modello in pesi investibili
pre_rebalance_weights = self.current_weights.copy()
```

```
new_weights = softmax_weights(
    action,
    top_k=self.top_k_assets,
    previous_weights=pre_rebalance_weights,
    min_score_improvement=self.top_k_switch_min_score_improvement,)
# Applicazione dei rendimenti forward
asset_returns = self.forward_returns.iloc[self.current_step].to_numpy(
    dtype=np.float32)
portfolio_return = float(np.dot(new_weights, asset_returns))
turnover = float(np.abs(new_weights - pre_rebalance_weights).sum())
transaction_cost = self.config.transaction_cost_rate * turnover
net_return = portfolio_return - transaction_cost
rolling_sharpe, rolling_sortino = self._calculate_rolling_reward_metrics(
    net_return)
sharpe_reward = self.config.reward_sharpe_coef * rolling_sharpe
sortino_reward = self.config.reward_sortino_coef * rolling_sortino
reward = net_return + sharpe_reward + sortino_reward
```

A.4 - Configurazione del modello PPO

Configurazione dell'agente PPO utilizzato nel modello baseline, specificando la politica MLP e i principali iperparametri che regolano il processo di apprendimento.

```
def create_ppo_model(
    env: PortfolioAllocationEnv,
    config: ProjectConfig,
    verbose: int = 1,
) -> Any: """Crea un modello PPO con "MlpPolicy" """
```

```
    monitored_env = Monitor(env)
    model = PPO(
        policy="MlpPolicy",
        env=monitored_env,
        learning_rate=3e-4,
        n_steps=64,
        batch_size=64,
        gamma=0.99,
        gae_lambda=0.95,
        clip_range=0.2,
        ent_coef=config.ppo_entropy_coef,
        verbose=verbose,
        seed=config.random_seed)
    return model
```

A.5 - Addestramento multi-seed e selezione del modello

Addestramento della PPO su più seed e selezione del miglior checkpoint sulla base delle prestazioni ottenute nel validation set.

```
for seed in seeds:
    seed_results_dir = original_results_dir / f"multiseed_seed_{seed}"
    seed_config = replace(
        config,
        random_seed=int(seed),
        ppo_model_path=seed_results_dir / "ppo_mlp_portfolio.zip",
        results_dir=seed_results_dir,)
    print(f"\n=== Training multi-seed PPO | seed={seed} ===")
```

```
train_ppo_agent_with_validation_checkpoints(
    seed_config,
    total_timesteps=total_timesteps,
    eval_frequency=eval_frequency,
    selection_metric=selection_metric,
    force_rebuild_data=force_rebuild_data,
    force_download=force_download,
    model_verbose=0, )
checkpoint_metrics_path = (
    seed_config.results_dir / "ppo_mlp_validation_checkpoints.csv" )
checkpoint_metrics = pd.read_csv(checkpoint_metrics_path)
best_row = checkpoint_metrics.loc[
    checkpoint_metrics["selection_score"].idxmax() ]
summary_rows.append(
    {"seed": int(seed),
        "best_model_path": str(
            seed_config.results_dir / "ppo_mlp_best_validation.zip"),
        "best_timesteps": int(best_row["timesteps"]),
        "selection_metric": selection_metric,
        "selection_score": float(best_row["selection_score"]),
        "cumulative_return": float(best_row["cumulative_return"]),
        "annualized_volatility": float(best_row["annualized_volatility"]),
        "sharpe_ratio": float(best_row["sharpe_ratio"]),
        "sortino_ratio": float(best_row["sortino_ratio"]),
```

```
        "max_drawdown": float(best_row["max_drawdown"]),
        "average_turnover": float(best_row["average_turnover"]),
        "total_transaction_cost": float(best_row["total_transaction_cost"]), } )
summary = pd.DataFrame(summary_rows)
best_summary_row = summary.loc[summary["selection_score"].idxmax()]
best_source_path = Path(str(best_summary_row["best_model_path"]))
best_model = PPO.load(best_source_path)
best_model.save(get_best_multiseed_model_path(config))
# Determinazione del migliore checkpoint
validation_results = evaluate_policy_on_env(
   model,
   validation_env,
   deterministic=True,)
metrics = calculate_ppo_validation_metrics(validation_results)
score = get_metric_score(metrics, selection_metric)
if score > best_score:
   best_score = score
   best_step = completed_timesteps
   model.save(best_model_path)
```

A.6 – Valutazione delle performance del modello

Calcolo delle metriche utilizzate per la valutazione delle performance del modello, come: rendimento cumulato, volatilità annualizzata, Sharpe Ratio, Sortino Ratio e massimo drawdown.

```
def calculate_max_drawdown(portfolio_values: pd.Series) -> float:
   """Calcola il massimo drawdown di una curva di valore portafoglio."""
```

```
    running_max = portfolio_values.cummax()
    drawdowns = portfolio_values / running_max - 1.0
    return float(drawdowns.min())
def calculate_financial_metrics(
    period_returns: pd.Series,
    portfolio_values: pd.Series,
    periods_per_year: int = 52,) -> dict[str, float]:
    """Calcola metriche finanziarie standard da rendimenti per ribilanciamento."""
    clean_returns = period_returns.dropna()
    if clean_returns.empty:
        raise ValueError("Impossibile calcolare metriche: rendimenti vuoti.")
    cumulative_return = float((1.0 + clean_returns).prod() - 1.0)
    volatility = float(clean_returns.std() * np.sqrt(periods_per_year))
    if clean_returns.std() == 0:
        sharpe = np.nan
    else:
        sharpe = float(
            clean_returns.mean() / clean_returns.std() * np.sqrt(periods_per_year))
    downside_returns = clean_returns[clean_returns < 0]
    downside_std = downside_returns.std()
    if downside_returns.empty or downside_std == 0:
        sortino = np.nan
    else:
        sortino = float(
```

```
            clean_returns.mean() / downside_std * np.sqrt(periods_per_year))
    return { "cumulative_return": cumulative_return,
        "annualized_volatility": volatility,
        "sharpe_ratio": sharpe,
        "sortino_ratio": sortino,
        "max_drawdown": calculate_max_drawdown(portfolio_values),}
def calculate_ppo_validation_metrics(
    evaluation_results: dict[str, pd.DataFrame],
) -> dict[str, float]:
    """Calcola metriche validation del PPO usando rendimenti netti dopo i costi."""
    portfolio_history = evaluation_results["portfolio_history"]
    metrics = calculate_financial_metrics(
        period_returns=portfolio_history["net_return"],
        portfolio_values=portfolio_history["portfolio_value"], )
    metrics["average_turnover"] = float(portfolio_history["turnover"].mean())
    metrics["total_transaction_cost"] = float(
        portfolio_history["transaction_cost"].sum())
    metrics["total_reward"] = float(portfolio_history["reward"].sum())
    metrics["average_reward"] = float(portfolio_history["reward"].mean())
    return metrics
```

A.7 - Estrazione del sentiment tramite FinBERT

Applicazione di FinBERT ai titoli delle notizie finanziarie e ottenimento delle probabilità associate alle classi: positiva, negativa e neutrale per la costruzione del score di sentiment.

```
frame = batch.to_pandas()
positive, neutral, negative = _score_titles(
```

```
    frame["title"].fillna("").astype(str).tolist(),
    tokenizer=tokenizer,
    model=model,
    label_indices=label_indices,
    batch_size=config.batch_size,
    max_length=config.max_length,
    device=device,)
probabilities = np.column_stack((positive, neutral, negative))
if not np.isfinite(probabilities).all():
    raise RuntimeError(
        f"Non-finite FinBERT probabilities found for year {year}.")
maximum_sum_error = np.max(
    np.abs(probabilities.sum(axis=1) - 1.0))
if maximum_sum_error > 1e-4:
    raise RuntimeError(
        "FinBERT probabilities do not sum to one: "
        f"max_error={maximum_sum_error}" )
output_frame = frame.drop(columns=["title"]).copy()
output_frame["positive_probability"] = positive.astype(np.float32)
output_frame["neutral_probability"] = neutral.astype(np.float32)
output_frame["negative_probability"] = negative.astype(np.float32)
output_frame["sentiment_score"] = (
    positive – negative
).astype(np.float32)
```

A.8 - Allineamento temporale delle notizie e associazione agli ETF

Assegnazione di ciascuna notizia alla prima seduta di mercato utilizzabile e associazione agli ETF rilevanti tramite ticker e parole chiave.

```
def assign_available_trading_dates(
    published_at: pd.Series,
    trading_dates: pd.DatetimeIndex,) -> pd.Series:
    if trading_dates.empty:
        raise ValueError("trading_dates must not be empty.")
    clean_trading_dates = pd.DatetimeIndex(trading_dates).tz_localize(None).normalize()
    clean_trading_dates = clean_trading_dates.sort_values().unique()
    timestamps_utc = pd.to_datetime(published_at, utc=True)
    local = timestamps_utc.dt.tz_convert(NEW_YORK)
    local_dates = local.dt.tz_localize(None).dt.normalize()
    utc_dates = timestamps_utc.dt.tz_localize(None).dt.normalize()
    midnight_utc = ( (timestamps_utc.dt.hour == 0)
        & (timestamps_utc.dt.minute == 0)
        & (timestamps_utc.dt.second == 0) )
    before_close = local.dt.time <= time(16, 0)
    is_trading_date = local_dates.isin(clean_trading_dates)
    same_day = (~midnight_utc) & before_close & is_trading_date
    base_dates = local_dates.where(~midnight_utc, utc_dates)
    insertion_side = np.where(same_day, "left", "right")
    result = np.empty(len(base_dates), dtype="datetime64[ns]")
    for side in ("left", "right"):
```

```
        mask = insertion_side == side
        positions = clean_trading_dates.searchsorted(base_dates[mask], side=side)
        if np.any(positions >= len(clean_trading_dates)):
            raise ValueError("News timestamps extend beyond the supplied trading calendar.")
        result[mask] = clean_trading_dates.to_numpy()[positions]
    return pd.Series(result, index=published_at.index, name="available_trading_date")

def map_article_etf_exposures(
    titles: pd.Series,
    article_tickers: pd.Series,
    etfs: Iterable[str],
    rules: Mapping[str, Mapping[str, Iterable[str]]] = DEFAULT_ETF_EXPOSURE_RULES,
) -> pd.DataFrame:
    """Map each article to economically exposed ETFs with auditable rules."""
    clean_titles = titles.fillna("").astype(str).str.lower()
    clean_ticker_sets = article_tickers.map(
        lambda values: {str(value).strip().upper()
            for value in (values if isinstance(values, (list, tuple, np.ndarray)) else [])
            if str(value).strip()} )
    exposures = pd.DataFrame(False, index=titles.index, columns=list(etfs))
    for etf in exposures.columns:
        rule = rules.get(etf, {})
        relevant_tickers = { str(value).strip().upper()
            for value in rule.get("tickers", ())}
```

```
        relevant_tickers.add(etf.upper())
        ticker_match = clean_ticker_sets.map(
            lambda values: not values.isdisjoint(relevant_tickers))
        keywords = tuple(
            str(value).strip().lower()
            for value in rule.get("keywords", ())
            if str(value).strip())
        if keywords:
            alternatives = "|".join(
                sorted((re.escape(value) for value in keywords), key=len, reverse=True) )
            pattern = rf"(?<!\w)(?:{alternatives})(?!\w)"
            keyword_match = clean_titles.str.contains(
                pattern,
                regex=True,
                na=False,)
        else:
            keyword_match = pd.Series(False, index=titles.index)
        exposures[etf] = ticker_match | keyword_match
    return exposures
```