



Università  
Ca' Foscari  
Venezia

**UNIVERSITA' CA' FOSCARI VENEZIA**

**DIPARTIMENTO DI ECONOMIA**

**Corso di Laurea Magistrale in Economia e Finanza**

## **Machine learning ed investimenti finanziari**

Studio ed elaborazione di trading system  
automatici basati su reinforcement learning ed  
analisi tecnica

**Relatore**

**Ch. Prof.** Claudio Pizzi

**Laureando**

Nicolò Facchini

Matricola 852185

**Anno Accademico 2018/2019**



*Ai miei genitori,  
con affetto e gratitudine*



# **INDICE**

<b>INTRODUZIONE</b>	<b>1</b>
<b>CAPITOLO I. MACHINE LEARNING</b>	<b>5</b>
1.1 <i>COS'È IL MACHINE LEARNING E COME SI È SVILUPPATO     FINO AI GIORNI D'OGGI</i>	5
1.1.1 <i>Definizioni e terminologia</i>	5
1.1.2 <i>Storia</i>	8
1.2 <i>I MODELLI DI APPRENDIMENTO AUTOMATICO</i>	13
1.2.1 <i>Supervised learning</i>	13
1.2.2 <i>Unsupervised learning</i>	17
1.2.3 <i>Reinforcement learning</i>	19
1.2.4 <i>Artificial neural network e deep learning</i>	23
1.2.5 <i>Forme ibride di machine learning</i>	25
1.2.6 <i>Forme alternative di machine learning</i>	27
1.3 <i>DATA MINING E MACHINE LEARNING</i>	33
1.4 <i>APPLICAZIONI</i>	36
<b>CAPITOLO II. TRADING SYSTEM, ANALISI TECNICA E     MACHINE LEARNING</b>	<b>41</b>
2.1 <i>INTRODUZIONE AI TRADING SYSTEM</i>	41
2.2 <i>CENNI DI ANALISI TECNICA</i>	43
2.2.1 <i>Medie mobili</i>	45
2.2.2 <i>Bande di oscillazione, bollinger bands e indicatori collegati</i>	49
2.2.3 <i>Momentum</i>	53
2.2.4 <i>Moving Average Convergence/Divergence (MACD)</i>	57
2.2.5 <i>Relative Strenght Index (RSI)</i>	58
2.2.6 <i>Stochastics oscillator</i>	61
2.2.7 <i>Average Directional Movement Index (ADX)</i>	65
2.3 <i>STATO DELL'ARTE IN LETTERATURA</i>	68

<b>CAPITOLO III. APPLICAZIONE DI TRADING SYSTEM AUTOMATICI BASATI SU REINFORCEMENT LEARNING</b>	<b>75</b>
3.1 <i>IL CASO DI STUDIO</i>	75
3.1.1 <i>Ipotesi e finalità del progetto</i>	75
3.1.2 <i>Strumenti informatici</i>	78
3.2 <i>FASE DI ALLENAMENTO</i>	79
3.2.1 <i>Preparazione dei dati e analisi tecnica</i>	79
3.2.2 <i>Training con apprendimento per rinforzo</i>	86
3.3 <i>FASE DI TEST</i>	87
<b>CAPITOLO IV. RISULTATI FINALI</b>	<b>91</b>
4.1 <i>RISULTATI OTTENUTI</i>	91
4.1.1 <i>Scelta dei parametri</i>	91
4.1.2 <i>Analisi dei rendimenti</i>	93
4.1.3 <i>Analisi dell'operatività</i>	96
4.2 <i>CONSIDERAZIONI</i>	97
4.2.1 <i>Implementazioni future</i>	98
<b>CONCLUSIONI</b>	<b>101</b>
<b>INDICE DELLE FIGURE</b>	<b>105</b>
<i>Indice dei grafici</i>	105
<i>Indice delle tabelle</i>	106
<b>BIBLIOGRAFIA</b>	<b>107</b>
<b>SITOGRAFIA</b>	<b>113</b>
<b>APPENDICE</b>	<b>115</b>
<i>CODICE</i>	115
<i>FASE DI TRAINING</i>	115
<i>Preparazione dei dati e analisi tecnica</i>	115
<i>Reinforcement learning</i>	119
<i>FASE DI TEST</i>	120
<b>RINGRAZIAMENTI</b>	







## INTRODUZIONE

La presente tesi di laurea nasce da un interesse personale verso l'analisi dei mercati finanziari e lo sviluppo di trading system automatici. Questo aspetto, insieme alla curiosità nei confronti del tema dell'intelligenza artificiale, ha posto le basi per la nascita di questa tesi di ricerca.

I progressi che sta vivendo la tecnologia mondiale consentono oggi risultati inimmaginabili prima: dalle automobili a guida autonoma, agli assistenti vocali presenti oramai in tutti i device elettronici di nuova commercializzazione, non si contano più le applicazioni dell'intelligenza artificiale nella vita quotidiana. La società d'oggi, d'altronde, è abituata ad essere circondata da ogni possibile forma di tecnologia. Tuttavia, ciò che quasi mai ci si chiede è: "quale strumento permette di ottenere questi risultati?". La risposta nella maggior parte dei casi è una: il *machine learning*.

Noto in italiano come apprendimento automatico, il *machine learning* è una branca dell'intelligenza artificiale che studia e sviluppa la capacità dei sistemi automatici di imparare un compito autonomamente, senza il bisogno di specifiche indicazioni da parte dello sviluppatore umano. L'utilizzo di algoritmi di questo tipo rappresenta il futuro della scienza. E non si parla solo del campo dell'informatica: le potenzialità di queste metodologie appaiono trasversali all'intero novero delle discipline scientifiche, dall'automazione industriale all'ingegneria biomedica, dal marketing alla chirurgia medica. L'unico vero impedimento al pieno utilizzo di strumenti in grado di imparare da soli è il timore da parte dell'uomo che le macchine possano diventare troppo intelligenti, sottraendogli libertà e facoltà di scelta. Una paura che, come afferma il professore Pedro Domingos dell'Università di Washington, luminare di *machine learning* e *data mining*, non esiste: "la gente ha paura che i computer diventino troppo intelligenti e dominino il mondo, ma il vero problema è che, pur essendo ancora troppo stupidi, lo hanno già conquistato." Lo sviluppo di nuove tecnologie, come ad esempio le reti di connessione di quinta generazione (5G) o i computer quantistici, permettono inoltre l'utilizzo in tempo reale di algoritmi ad altissima capacità computazionale, in grado di trattare moli di dati immense. Tutto ciò, pertanto, apre la strada a diverse nuove applicazioni.

Nel campo dell'economia, i maggiori utilizzi delle tecniche di *machine learning* rientrano nell'area del marketing e nell'utilizzo di social network e piattaforme online. Tuttavia, questi sono solamente i primi sentieri che sono stati percorsi finora. Come detto, le possibili applicazioni sono veramente infinite ed una tra queste è proprio l'oggetto di studio del lavoro di ricerca che viene qui presentato: l'utilizzo di forme di intelligenza artificiale per la composizione di trading system automatici per l'investimento in mercati finanziari.

La definizione più consona di "mercato finanziario" è "sistema complesso intrecciato" (dal latino *complexus* – abbracciato), tanto che, la caratteristica peculiare è proprio l'evidente influenza reciproca che assume ogni sua componente: ciascuna è legata alle altre secondo modalità e pesi via via differenti. Non è possibile intervenire su una componente senza influenzare contemporaneamente, in qualche misura, gli altri fattori del sistema. Si considerino, ad esempio, gli influssi sugli indici azionari da parte di fattori macroeconomici, scelte politiche, fattori psicologici degli investitori. A questo punto, si potrebbe presumere che i mercati finanziari siano avulsi da qualsivoglia previsione, rendendo inutili i tentativi di costruzione di modelli statistici che provino a descriverli. In realtà, spesso i mercati si muovono secondo strutture di comportamento che tendono a ripetersi nel tempo, anche se in forme diverse, tramite trend e cicli. Questa sorta di regolarità è ciò che consente agli investitori di fare previsioni e di prendere le proprie scelte d'investimento. Facendo un ulteriore step, è ciò che permette l'esistenza dei cosiddetti trading system, insiemi di regole stabilite dall'investitore per regolare il proprio processo d'investimento. Un esempio banalissimo potrebbe essere "compra ogni volta che il prezzo dell'azione supera i 100 dollari e vendi se supera i 120". Ciò che fa la differenza tra guadagno e perdita, ovviamente, è la solidità delle regole poste in essere. Ma come stabilire le regole migliori per guadagnare? Se esistesse una risposta universale, probabilmente la presente tesi non avrebbe motivo di esistere, così come anche i mercati finanziari perderebbero d'utilità.

Nel corso dei decenni, sono stati sviluppati studi, strategie, analisi sul processo d'investimento nei mercati finanziari. Fra le correnti più note vi è senza dubbio l'analisi tecnica. Con questo concetto si intende in economia lo studio dell'andamento delle serie storiche dei prezzi dei titoli finanziari, al fine di poter prevedere le tendenze future tramite analisi grafiche, quantitative e qualitative. In particolare, l'analisi tecnica quantitativa ha fornito nel corso degli anni molteplici indicatori utili per l'osservazione di segnali, al verificarsi dei quali è probabile un aumento, o un calo, del prezzo del titolo in esame.

Alla luce di quanto detto, la finalità del presente lavoro di ricerca è stata quella di coniugare una specifica forma di *machine learning*, il *reinforcement learning*, con l'utilizzo dei suddetti indicatori di analisi tecnica, il tutto all'interno di un trading system finanziario automatico.

L'ideazione del progetto ha avuto origine proprio dall'analisi della letteratura scientifica in merito al tema dell'apprendimento automatico, di quello per rinforzo ed infine delle loro applicazioni in ambito finanziario.

Per quanto riguarda la presente esposizione, dopo aver descritto le caratteristiche dei modelli di *machine learning* e dei trading system finanziari a tutto tondo (con particolare riguardo, rispettivamente, al *reinforcement learning* ed all'analisi tecnica), la trattazione verterà sulla personale elaborazione di sistemi d'investimento basati su tali concetti:

- nel primo capitolo il focus è sul tema del *machine learning*, analizzato nelle sue diverse varianti ed applicazioni, ovviamente con grande attenzione al *reinforcement learning*;
- il secondo, invece, si concentra sulla descrizione dei trading system proposti nel corso degli anni, in particolare, con l'utilizzo di indicatori di analisi tecnica;
- nel terzo capitolo vengono descritte la creazione ed applicazione di trading system automatici che combinano *reinforcement learning*, analisi tecnica e diverse soluzioni per l'implementazione dei modelli già proposti in letteratura;
- con il quarto capitolo, infine, si espongono i risultati ottenuti, analizzando sia i rendimenti sia l'operatività del trading system.

Solo dopo aver trattato queste quattro fasi di approfondimento, verranno proposti in conclusione una valutazione sull'applicazione dei suddetti modelli ed eventuali implementazioni future.



# CAPITOLO I. MACHINE LEARNING: DEFINIZIONI, TIPOLOGIE, ALGORITMI E APPLICAZIONI

## 1.1 COS'È IL MACHINE LEARNING E COME SI È SVILUPPATO FINO AI GIORNI D'OGGI

### 1.1.1 Definizioni e terminologia

Il *machine learning*, noto in italiano come apprendimento automatico, è la branca dell'*Artificial Intelligence* (AI) che sviluppa la capacità delle "macchine" (i sistemi informatici) di imparare qualcosa senza che siano date loro esplicitamente le indicazioni per farlo. In particolare, ci si riferisce ad algoritmi che, basati su studi di probabilità e statistica, sviluppano autonomamente le loro conoscenze grazie ai pattern di dati ricevuti, senza il bisogno di avere degli input iniziali specifici da parte dello sviluppatore umano. Questo significa accantonare la classica programmazione esplicita, sistema in cui la componente umana programma un modello in base a comandi del tipo "*if-then*", a favore di un metodo in cui la macchina è in grado di stabilire da sola gli schemi da seguire per ottenere il risultato desiderato. Pertanto, il vero fattore che distingue l'intelligenza artificiale è l'autonomia.

Volendo generalizzare, è possibile scindere un processo qualunque di *machine learning* in tre fasi principali:

- l'apprendimento dei dati, nelle diverse forme in cui si possono manifestare;
- la valutazione dei dati stessi, nella quale il sistema informatico ipotizza dei modelli statistici che descrivano la realtà osservata;
- l'ottimizzazione dei modelli stimati e la conseguente formulazione di una strategia di risposta/azione in base ai feedback raccolti con l'esperienza.

È all'interno di questa prima partizione che risulta importante introdurre la terminologia che si ritroverà nel corso del presente capitolo. Innanzitutto, si prenda in esame il concetto di dati, elementi con cui il sistema di apprendimento automatico viene a contatto. Essi non sono tutti uguali, e di certo non hanno tutti la stessa valenza. Nella prima fase del processo di apprendimento, il sistema riceve un insieme di dati necessari all'addestramento (per questo chiamato *training dataset*), momento in cui il computer stima delle relazioni tra i

dati di input e quelli di output. Questi nessi individuati descrivono i parametri (o pesi) del modello stimato dal sistema. Talvolta i dati di ingresso (chiamati anche input o esempi) vengono forniti con i rispettivi dati di uscita (output), cioè il sistema riceve dei dati iniziali ai quali è data una precisa realizzazione futura determinata dalle azioni prese. In altre parole, ciò che il computer vede sono dei dati già “etichettati”<sup>1</sup> (o classificati). La presenza o meno di etichette nei dati iniziali è ciò che distingue alcune metodologie di machine learning (apprendimento con o senza supervisione) che verranno analizzate in seguito. Più precisamente, il *label* in machine learning non è altro che un target, ciò che in statistica viene definito variabile dipendente. Rimanendo nel parallelo con la statistica, quella che solitamente viene definita variabile esplicativa, nell’apprendimento automatico è chiamata feature (letteralmente “attributo”).

Una volta che il sistema ha ricevuto i *training data*, ciò che va a fare è una stima delle strutture ricorrenti e delle regole logiche che osserva negli esempi forniti, che altro non è che la determinazione di modelli e parametri che descrivono la realtà. Volendo nuovamente ricorrere alla statistica, questi *pattern* (modelli, schemi) di dati tentano di descrivere il cosiddetto *Data Generating Process (DGP)*<sup>2</sup> del set di dati. Il modello stimato dal computer non per forza deve essere statico, o determinato a posteriori, ma anzi è in continua evoluzione: dopo un primo addestramento, la macchina inizia ad applicare la strategia di azioni che ritiene ottimale per poter raggiungere l’obiettivo prefissato; questa sequenza di operazioni può rivelarsi efficace come no, e questo viene descritto da una funzione di valore che restituisce alla macchina un feedback sul quale può basare le sue mosse future. Con il progredire dell’addestramento, e dunque dei feedback ricevuti, il computer migliora le proprie mosse in risposta alle situazioni particolari che deve affrontare. Al termine dell’analisi del *dataset* di addestramento, il sistema, in base all’esperienza accumulata, possiederà “un’idea” su come comportarsi nel momento in cui affronterà dei dati nuovi ma simili a quelli passati; tuttavia, ciò che interessa ai soggetti umani è valutare l’efficienza su informazioni future che possono essere diverse da quelle già assorbite dal sistema, e ricevere previsioni future. In quest’ottica, si inserisce

---

<sup>1</sup> Dall’inglese “labels”, etichette.

<sup>2</sup> Il DGP (Data Generating Process) descrive come ciascuna osservazione all’interno dell’insieme di dati è stata prodotta, ovvero in base a quali relazioni fra i parametri è risultato quello specifico risultato.

un'ulteriore categoria di dati chiamata "*test data*" (ovvero dati di esame), che serve per l'appunto a testare i risultati che la "macchina" restituisce dopo la fase di apprendimento sui *training data*.

L'insieme di regole, formule, azioni ed operazioni logiche viste permettono di definire il concetto di algoritmo: tramite l'esperienza, le macchine (così come gli umani) creano delle regole generali che convogliano in modelli di apprendimento, i quali a loro volta permettono di costruire algoritmi di apprendimento necessari a risolvere un determinato problema. L'algoritmo indica al sistema le azioni/operazioni che può eseguire in un determinato contesto. Se il sistema è in grado di compiere tale operazione, utilizzerà questa informazione per i compiti successivi.

Sebbene gli studi sugli algoritmi di machine learning siano in continuo sviluppo, la teoria dell'apprendimento non fornisce alcuna garanzia sull'effettiva prestazione degli algoritmi. La finitezza dei dati e la mancanza di conoscenza sull'evoluzione futura del modello d'altronde rappresentano di per sé un limite. Affinché la modellizzazione fornisca le migliori prestazioni possibili, la complessità dell'ipotesi induttiva deve essere pari a quella del modello sottostante il *training dataset*. Problemi di *underfitting* e *overfitting* nascono proprio da questa disparità: in particolare, il problema dell'*overfitting* nasce quando un modello statistico complesso si "sovra-adatta" alle osservazioni poiché ha un numero eccessivo di parametri rispetto al numero di dati campionari; viceversa vale invece per l'*underfitting*. Vista sotto un altro punto di vista, l'*overfitting* si riscontra quando un modello allenato su un *training set* non è poi in grado di generalizzare il *pattern* appreso su un insieme di dati più ampio: è molto preciso per quanto riguarda i dati di addestramento, ma non risulta utile su grandi serie di informazioni sconosciute. L'*underfitting*, invece, si riscontra nei casi in cui non vi sono sufficienti dati per stimare un adeguato numero di parametri e dunque il modello risulta poco aderente al reale DGP. Per evitare questi problemi, è importante eseguire l'addestramento utilizzando anche dati imprevisti, oppure utilizzando espedienti che verranno presentati nel corso della parte applicata della presente tesi. In ogni caso, ad oggi, non esiste una soluzione univoca a problemi di questo tipo: ogni applicazione, ogni task da risolvere, richiede modelli e algoritmi diversi. È anche per questo motivo che negli ultimi vent'anni sono state sviluppate diverse proposte di machine learning, frutto di sperimentazione e studio. Nel corso del presente capitolo verrà data una visione quanto più ampia su di esse.

### 1.1.2 Origini ed evoluzione

Il termine *machine learning* venne coniato per la prima volta nel 1959 da Arthur Lee Samuel, ricercatore americano pioniere negli studi sull'intelligenza artificiale; tuttavia, la definizione più celebre di *machine learning* si deve ad un altro grande scienziato americano, Tom Michael Mitchell:

“Si dice che un programma apprende dall'esperienza  $E$ , con riferimento ad alcune classi di compiti  $T$  e con misurazione della performance  $P$ , se le sue performance nel compito  $T$ , come misurato da  $P$ , migliorano con l'esperienza  $E$ .”<sup>3</sup>

In parole semplici, tramite l'esperienza, e dunque il tempo (intesi non nell'accezione “umana”, ma in riferimento alla mole di dati esaminati), la macchina migliora esponenzialmente le proprie risposte alla realtà che, di volta in volta, si trova ad affrontare: i metodi matematico-computazionali che sono alla base degli algoritmi di *machine learning* permettono al sistema informatico di implementare le proprie prestazioni in modo “adattivo” man mano che gli “esempi” da cui imparare aumentano. La definizione di Mitchell è rilevante perché fornisce una codifica operativa e non più una formulazione in semplici termini cognitivi. Di fatto, con questa formulazione, Mitchell seguì la linea proposta dal celebre matematico britannico Alan Turing, che, nel suo articolo “Computing Machinery and Intelligence”, alla domanda “Le macchine possono pensare?” fornì un diverso punto di vista, rispondendo con un'altra domanda: “Le macchine possono fare quello che noi (in quanto entità pensanti) possiamo fare?”. Sempre alla mente di Turing si deve un popolare test sviluppato nel 1950, e proposto nello stesso articolo appena citato. È un criterio che ha l'obiettivo di valutare l'abilità di un computer di riprodurre il comportamento umano: prendendo spunto dal “gioco dell'imitazione”<sup>4</sup>, il test prevede che un esaminatore intrattenga una conversazione sia con un essere umano, sia con una macchina, e che debba riuscire a identificare le rispettive identità. Un'applicazione molto diffusa del test di Turing è, ad esempio, quella che quotidianamente si incontra online

---

<sup>3</sup> Tratto da Mitchell, T. M. (1997), *Machine Learning*, McGraw-Hill

<sup>4</sup> È un gioco a tre partecipanti: un uomo, una donna ed una terza persona. Quest'ultima è separata dagli altri due e tramite delle domande deve stabilire chi dei due è l'uomo, e chi la donna. Allo stesso tempo, uno tra la donna e l'uomo deve cercare di aiutare l'esaminatore, mentre l'altro tenta di portarlo sulla falsa strada.



durante le registrazioni ai siti web: i CAPTCHA<sup>5</sup> che chiedono di riconoscere immagini piuttosto che numeri o testi. In sintesi, se il sistema informatico è in grado di sostituirsi alla figura umana, allora si può considerare la macchina intelligente, cioè in grado di concatenare idee e di esprimerle. Sebbene sia indiscussa la grandezza innovativa del pensiero di Turing, il suo test ha ricevuto nei decenni successivi diverse critiche poiché non è ritenuto prova sufficiente per valutare l'effettiva intelligenza di una macchina. A semplice titolo esemplificativo, si veda la cosiddetta stanza cinese<sup>6</sup>, esperimento mentale ideato da John Searle che dimostra l'inattendibilità del test di Turing, così come il programma ELIZA<sup>7</sup>, che soddisfa i criteri del test nonostante sia evidentemente un ente non pensante per costruzione.

A tal proposito, risulta esplicita la distinzione tra AI forte (o *Artificial General Intelligence, AGI*) e AI debole (o *Artificial Narrow Intelligence, ANI*): mentre la prima si basa sull'idea che le macchine possano effettivamente essere considerate intelligenti, ovvero possano riprodurre qualsiasi attività cerebrale svolta da un umano, l'intelligenza artificiale debole non pretende che il sistema comprenda i processi cognitivi umani, ma solo che si occupi della risoluzione di problemi specifici. Come immaginabile, tutte le applicazioni odierne di intelligenza artificiale rientrano nell'ANI. Questo slittamento di paradigma proposto da Turing, fu alla base dei primi studi sulla possibilità di creare delle forme di intelligenza artificiale, che da lì a pochi anni avrebbe avuto origine. Anche se la prima espressione di apprendimento automatico si può far risalire al 1943, anno in cui McCulloch & Pitts introdussero il primo modello matematico che riproduce una rete neurale artificiale<sup>8</sup>, il termine "*Artificial Intelligence*" venne coniato ed utilizzato per la prima volta solo nel 1956 nel corso della Dartmouth Conference, tenuta da Marvin Minsky (Harvard

---

<sup>5</sup> Acronimo di "Completely Automated Public Turing-test-to-tell Computers and Human Apart", un CAPTCHA è nell'ambito dell'informatica un test a domande per scoprire se l'utente è un umano oppure un bot.

<sup>6</sup> Presentato in: Searle, John. R. (1980), "Minds, brains, and programs", *Behavioral and Brain Sciences*, 3, pp. 417-457.

<sup>7</sup> ELIZA è un bot per chat scritto nel 1966 dall'informatico tedesco Joseph Weizenbaum. Il programma simula il comportamento di un terapeuta Rogersiano (dal fondatore della terapia non direttiva, Carl Ramson Rogers), che è solito rispondere con riformulazioni delle affermazioni del paziente stesso.

<sup>8</sup> Warren S. McCulloch, Walter Pitts (1943), "A logical calculus of the ideas immanent in nervous activity", *The Bulletin of Mathematical Biophysics*, Volume 5, Issue 4, pp 115-133.

University), Claude Shannon (Bell Telephone Lab), Nathan Rochester (IBM) e John McCarthy (Dartmouth College). Quest'ultimo, in particolare, è riconosciuto come uno dei padri dell'intelligenza artificiale, definita da lui stesso come "la scienza e l'ingegneria della fabbricazione di macchine intelligenti". Ricercatori come Marvin Minsky, Arthur Samuel e Frank Rosenblatt, contribuirono alla fine degli anni '50 alla definizione di intelligenza artificiale, sia attraverso metodi formali, sia grazie ad approcci probabilistici, sia con l'applicazione delle cd. reti neurali, allora formate da singoli percettroni<sup>9</sup> e da modelli derivanti dal modello lineare generalizzato (GLM) della statistica. Fu intorno a quegli anni, ad esempio, che vennero avviate le prime simulazioni del cervello umano con le *neural networks* (1957) e che nacque il primo software di *machine learning* (1959): Arthur Samuel sviluppò, infatti, per conto di IBM, sul primo calcolatore commercializzato dall'azienda statunitense (l'IBM 701), il primo gioco di dama "intelligente", nel quale i computer miglioravano le proprie mosse via via che interagivano con i giocatori umani. Grazie all'analisi probabilistica delle posizioni raggiungibili e della possibilità di vittoria per ogni mossa praticabile e casella occupata, il programma decideva i propri spostamenti in base ad una strategia di ottimizzazione della funzione di guadagno (assumendo che l'avversario ragionasse allo stesso modo). Inoltre, tramite l'apprendimento mnemonico, il programma era in grado di memorizzare tutte le posizioni già viste e dunque poteva basarsi sulle esperienze passate per sapere quali mosse erano più efficaci. Facendo poi confrontare il computer con giocatori professionisti, e facendolo anche giocare contro sé stesso per centinaia di volte, esso divenne infine, nel giro di più di dieci anni, in grado di sfidare giocatori di alto livello.

Nonostante questi primi traguardi, lo studio dell'AI verteva in quegli anni su approcci logici *knowledge-based*, determinando perciò una distanza importante tra l'analisi dell'intelligenza artificiale ed il suo sottoinsieme rappresentato dall'apprendimento automatico. Quest'ultimo, che deve per forza di cose basarsi su un approccio probabilistico, riscontrava allora diversi problemi applicativi, non ultime l'acquisizione e la rappresentazione di grandi moli di dati che la scienza non era ancora in grado di trattare.

---

<sup>9</sup> Un percettrone è, nell'ambito del machine learning, un particolare classificatore binario che mappa un vettore di dati di ingresso  $x$  con uno scalare di output  $f(x)$ . Fu proposto da Frank Rosenblatt nel 1958.

Inoltre, la ricerca sulle *neural networks* vide un importante rallentamento a causa della pubblicazione degli studi di Minsky e Papert<sup>10</sup>, nei quali venivano descritte delle importanti limitazioni intrinseche all'utilizzo di perceptron e reti neurali. Negli anni settanta e ottanta, l'analisi dell'intelligenza artificiale era completamente concentrata sull'implementazione di sistemi esperti<sup>11</sup>, tralasciando completamente quelli su base statistica. La ricerca sull'apprendimento automatico venne portata avanti principalmente al di fuori del campo dell'intelligenza artificiale, ovvero da ricercatori di altre discipline che ottennero, dapprima, diversi risultati nel riconoscimento di *pattern* e nell'*information retrieval*<sup>12</sup> e, successivamente, nello studio di algoritmi come la *backpropagation*<sup>13</sup> e la *self-organization*<sup>14</sup>. L'apprendimento automatico, sviluppatosi pertanto come disciplina distinta dall'intelligenza artificiale classica, iniziò a prendere piede seriamente solo negli anni '90. L'obiettivo non era più quello di creare una forma di intelligenza artificiale, concepita sulla falsariga di quella umana, ma piuttosto quello di risolvere specifici problemi pratici. Questo diverso approccio, insieme ad un diverso interesse verso modelli statistici presi dalla teoria della probabilità, a discapito delle metodologie simboliche ereditate dall'AI, ha permesso un forte sviluppo della disciplina del *machine learning*. In sostanza, la svolta si è manifestata con il passaggio dall'approccio *knowledge-based* ad uno *data-based*.

Celebri sono stati i risultati ottenuti da IBM alla fine degli anni '90 con Deep Blue, primo calcolatore di sempre a vincere una partita a scacchi contro l'allora campione del mondo, il russo Garry Kasparov, con tempistiche regolamentari rispettate: la prima vittoria risale al 10 febbraio 1996, occasione in cui, tuttavia, Kasparov riuscì a vincere per 4 incontri a 2; in seguito aggiornato, Deep Blue riuscì invece a conquistare la rivincita l'11 maggio 1997,

---

<sup>10</sup> Minsky, M. e Papert S. (1969), *Perceptrons. An Introduction to Computational Geometry*, M.I.T. Press, Cambridge, Massachusetts.

<sup>11</sup> Si definisce sistema esperto un programma di AI che tenta di riprodurre le azioni di una persona esperta in uno specifico campo di attività.

<sup>12</sup> In italiano "recupero di informazioni", è definita come l'insieme di tecniche per la rappresentazione, organizzazione e gestione di oggetti contenenti informazioni (ad esempio oggetti multimediali, database, pagine web ecc.).

<sup>13</sup> La retropropagazione dell'errore è un algoritmo di apprendimento supervisionato utilizzato per l'implementazione delle artificial neural networks, spesso utilizzato assieme ad un metodo di ottimizzazione, quale ad esempio la discesa stocastica del gradiente.

<sup>14</sup> L'auto-organizzazione, anche nota nella teoria dei sistemi come ordine spontaneo, è una forma di sviluppo del sistema tramite forze ordinanti e limitative originate dalle componenti medesime che formano l'ambiente oggetto di analisi. Questi sistemi possiedono le caratteristiche di complessità, autoreferenza, ridondanza e autonomia.

nonostante diversi sospetti, successivamente confermati, circa delle modifiche al programma da parte dei creatori tra una sfida e l'altra, elemento questo che consentiva un'intelligenza e creatività così profonde da non poter essere comprese dall'avversario umano. Ciononostante, la vittoria di Deep Blue costituì una pietra miliare nell'ascesa del *machine learning*, ed aprì la strada a diverse applicazioni ed implementazioni, nonché, ebbe il grande merito di accrescere ancor più l'interesse verso le scienze sull'intelligenza artificiale.

La nascita e diffusione di Internet, infine, hanno sancito la definitiva affermazione degli studi in materia, poiché ha permesso un'accelerazione esponenziale delle applicazioni digitali, nonché della reperibilità e distribuzione dell'informazione (e dunque dei dati da poter utilizzare come *sample*). Nell'apprendimento automatico i dati assumono un ruolo fondamentale, spesso anche più importante della tecnologia stessa, in quanto essi rappresentano la vera fonte di conoscenza che il computer può utilizzare per imparare: è questo il motivo principale per il quale il *machine learning* ha visto la propria diffusione solamente negli ultimi quindici anni, nonostante le metodologie alla base risalgano al secolo scorso. La quantità di dati prodotta dall'uomo è cresciuta esponenzialmente nell'ultimo ventennio, grazie soprattutto al passaggio al web 2.0 ed alla diffusione dei social media: ciò ha aperto la strada alla creazione di quelli che oggi vengono definiti "*Big Data*", ovvero moli giganti di dati, ed allo sviluppo della disciplina del *data mining*<sup>15</sup>, ovvero l'insieme di tecniche che permettono l'estrazione di informazioni utili proprio da queste grandi quantità di dati, di qualunque forma essi siano. Questi due fattori assieme hanno permesso l'introduzione di molte nuove tecnologie che, allo stato dell'arte, entrano a far parte della vita di tutti i giorni. Dagli assistenti vocali al riconoscimento facciale, dalle *smart house* domotiche alle pubblicità mirate, dai veicoli senza pilota alla robotica, sono innumerevoli le applicazioni che, ad oggi, attraversano trasversalmente tantissimi settori industriali e non<sup>16</sup>.

Infine, la recente introduzione di connessioni 5G e la creazione di computer quantistici, da parte di Google e IBM, preannunciano una capacità di raccolta dei dati e una potenza di

---

<sup>15</sup> Nel corso del capitolo si vedrà che il data mining si applica limitatamente alle cd. forme di apprendimento non supervisionato.

<sup>16</sup> Per una maggior comprensione, l'analisi approfondita delle principali applicazioni odierne del machine learning viene rimandata al paragrafo 1.4, dopo l'esame delle tipologie e degli algoritmi di apprendimento automatico.

calcolo inimmaginabili prima. Si pensi, a semplice titolo esemplificativo, che uno smartphone di fascia media prodotto nel 2019 è milioni di volte più potente dei migliori sistemi informatici al mondo posseduti dalla NASA negli anni '60. Tutto questo, presumibilmente, contribuirà, in misura ancor maggiore, alla diffusione capillare dell'utilizzo del *machine learning* in ogni aspetto della vita umana, aprendo nuove strade applicative tutte da scoprire.

## 1.2 I MODELLI DI APPRENDIMENTO AUTOMATICO

Come visto in precedenza, i momenti del processo di *machine learning* possono avere attuazione in modalità molto diversa, a seconda dei compiti richiesti alla macchina o dei dati che è possibile raccogliere. Le funzioni dell'apprendimento automatico sono tipicamente classificate in tre principali macrocategorie, che si distinguono per la natura degli input su cui si basa l'apprendimento o della funzione di valore con la quale il sistema giudica le azioni prese. Questi sottoinsiemi del *machine learning*, anche detti paradigmi, sono il *supervised learning*, l'*unsupervised learning* ed il *reinforcement learning*. Parallelamente a questi, ha ormai un ruolo fondamentale il *deep learning*, metodo che nasce dallo studio delle reti neurali e che, ad oggi, sembra essere la forza trainante dello sviluppo della scienza del *machine learning*, viste le sue grandi applicazioni. In aggiunta a queste quattro principali forme, sono sorte negli anni sia delle forme ibride, punto di contatto tra i paradigmi del *machine learning*, sia forme alternative che trovano origine in diversi approcci scientifici. Scopo del presente sottocapitolo sarà pertanto l'esame delle suddette categorie.

### 1.2.1 Supervised learning

Nel *machine learning* supervisionato (*Supervised Learning, SL*), al modello vengono forniti sia dei dataset di input sia le informazioni dei relativi output, con l'obiettivo che il computer identifichi *pattern* e regole generali che associno le informazioni iniziali  $x$  con i risultati finali  $y$ . In tal modo dovrebbe essere possibile riutilizzare tale nesso logico utilizzato anche per altri *task* simili, ovvero per nuovi input non presenti nel *training set*. In particolare, l'obiettivo del *supervised learning* è quello di fornire un'ipotesi induttiva, una funzione che

sia in grado di imparare dalle coppie input-output fornite nella fase di formazione, per poi ottenere i risultati desiderati in un test dataset nuovo.

Ciascun dato di input  $x_i$  rappresenta un *feature*, e può essere ad esempio un attributo numerico, testuale oppure un oggetto più sofisticato, come un'immagine o un file audio. L'output  $y_i$  invece rappresenta il risultato, la variabile di risposta, e può presentare qualunque forma. Proprio quest'ultima è ciò che distingue i diversi approcci applicativi del *supervised learning*. Dal punto di vista logico, l'apprendimento supervisionato parte dal presupposto che, se il sistema possiede un numero adeguato di informazioni, esso possiederà un'esperienza  $E$  tale da permettergli di determinare una funzione (l'ipotesi induttiva) in grado di approssimare la funzione obiettivo. Visto il parallelo tra la funzione stimata e quella obiettivo (ove gli output rappresentano le risposte desiderate dal progettista), quando verranno proposti dei dati di input non contenuti in  $E$ , l'ipotesi induttiva dovrebbe comunque riuscire, con sufficiente precisione, a fornire delle risposte soddisfacenti. A tal fine, il sistema sfrutta solitamente i principi di distribuzione matematica e di funzione di verosimiglianza. Ogni qualvolta il modello identifica la funzione che lega le variazioni di input alle variazioni di output, sceglie i pesi migliori ed esprime dunque la funzione corretta, o di massima verosimiglianza. Molti degli algoritmi supervisionati funzionano bene in contesti lineari. Tuttavia, non sempre vi sono le condizioni per una simile approssimazione. La stima della funzione di massima verosimiglianza serve proprio per ridurre le problematiche derivanti dall'applicazione in contesti non lineari.

Altro fattore delicato è l'esperienza: se questa è ridotta, l'algoritmo potrebbe restituire una funzione poco efficiente, mentre se è eccessiva, l'ipotesi induttiva potrebbe divenire talmente complessa da creare rallentamenti computazionali. Inoltre, nel caso supervisionato, vi è una forte sensibilità al rumore: anche pochi *outliers* sbagliati possono condurre il sistema a decisioni scorrette, e rendere dunque il modello inaffidabile. Una soluzione proposta negli anni può essere quella di associare agli algoritmi dei controlli basati sulla cosiddetta logica *fuzzy*<sup>17</sup>.

---

<sup>17</sup> Anche chiamata logica sfocata, è una logica polivalente in cui si attribuisce un grado di verità diverso compreso tra 0 e 1 a ciascuna proposizione utilizzata.

Generalmente, si possono distinguere due tipologie principali di problemi di apprendimento supervisionato: la classificazione e la regressione<sup>18</sup>, rispettivamente per valori qualitativi e quantitativi.

La classificazione è il processo con cui una macchina riesce a riconoscere ed etichettare oggetti visivi e dimensionali all'interno di un set di dati. L'attività di classificazione, pertanto, parte dalla fase di training, testa il modello addestrato su un insieme di dati le cui etichette non sono note, ed infine, dopo eventuali modifiche, effettua la previsione sui dati da classificare. Come anticipa il nome, l'obiettivo della classificazione è quello di formare un modello che sia in grado di ordinare i dati di input in gruppi. Un problema di classificazione potrebbe essere, per esempio, quello di associare a delle foto di numeri scritti a mano delle etichette (output) che identifichino le cifre rappresentate nei dati di input, in forma di pixel. Con uguale ragionamento, altre applicazioni potrebbero essere quelle di distinguere delle immagini in foto di cani, gatti e altri animali, oppure di classificare le email in "spam" e "non spam".

Allo stesso modo, un problema di regressione potrebbe essere rappresentato dal determinare il prezzo di alcune case partendo da variabili di input che descrivono il quartiere. Come si può intuire, l'approccio alla base di classificazione e regressione è simile, la differenza risiede però nel fatto che quest'ultima si applica su previsioni di variabili quantitative, ovvero output appartenenti all'insieme dei numeri reali. La base dei dati su cui lavorare sarà in questo caso continua, e non discreta come per la classificazione. Infine, ciò che guida la previsione è lo scarto di errore che viene generato dal modello rispetto alla funzione obiettivo: è insomma un'applicazione delle regressioni statistiche al mondo dell'informatica. Ad ogni modo, in qualunque forma esso sia, il *supervised learning* è comunemente usato in situazioni in cui i dati sono in grado di predire possibili scenari futuri.

---

<sup>18</sup> Entrambi i problemi possono avere una o più variabili di ingresso, così come i dati possono avere diverse forme (numerica, categorica ecc.).

Benché la gamma di questi algoritmi sia molto vasta, tra le tradizionali forme applicative del *supervised learning* si annoverano l'albero di decisione, le *Support Vector Machine* (SVM), i sistemi esperti e gli algoritmi genetici<sup>1920</sup>:

- l'albero di decisione (o *decision tree*) è un metodo di approssimazione, composto da una ramificazione di decisioni e delle possibili conseguenze, che ha l'obiettivo di formare una strategia decisionale (*plan*) mirata ad un fine (*goal*). In particolare, nel *machine learning*, rappresenta un modello predittivo dove ogni nodo costituisce una variabile, ed ogni nodo "figlio" una realizzazione possibile del suddetto fattore. Ogni particolare sequenza di decisioni si delinea dunque come il cammino (*path*) dal nodo radice (*rooth*) al nodo foglia, ovvero il punto d'arresto del processo. A tal riguardo, il programmatore deve porre un criterio di arresto (*halting*), oppure un criterio di potatura (*pruning*). Al di là del fatto che non avrebbe senso un albero infinito, bisogna sottolineare come la profondità massima del grafo non determina per forza la bontà del modello: la crescita eccessiva potrebbe determinare una complessità computazionale non accompagnata da benefici di uguale entità in termini di accuratezza;
- le *Support Vector Machine* (SVM) rappresentano una classe di metodi per la classificazione e regressione in cui, dato un training data set, l'algoritmo stesso costruisce un modello che assegna i nuovi dati ad una classe, ottenendo dunque un classificatore lineare non probabilistico. Formalmente, una SVM costruisce un iperpiano che può essere utilizzato per i problemi di apprendimento supervisionato<sup>21</sup>: in pratica, il modello rappresenta gli esempi come punti nello spazio, posti in modo tale che tra le diverse classi si formi un intervallo vuoto che le separi il più possibile. I nuovi dati verranno pertanto mappati in base a questa suddivisione dello spazio. L'applicazione delle SVM non si esaurisce con il solo caso

---

<sup>19</sup> Nonostante l'elenco non abbia la pretesa di citare tutti gli algoritmi sviluppati negli anni, una delle principali applicazioni riguarda l'utilizzo delle reti neurali artificiali, qui non citate perché rimandate ad una trattazione specifica al sottoparagrafo *1.2.4 Artificial neural networks e deep learning*. Si noti, in ogni caso, che lo studio delle neural network è avulso dalle classificazioni in esame perché trasversale all'intera disciplina del machine learning.

<sup>20</sup> Si vedano inoltre i già citati concetti di backpropagation e self-organization.

<sup>21</sup> Ad onor del vero, esistono implementazioni dell'algoritmo applicabili anche in contesti non supervisionati; tuttavia, il loro utilizzo avviene solamente dopo che sono stati applicati metodi come il clustering.



lineare, ma anzi sono possibili utilizzi in situazioni non lineari grazie all'uso dei metodi *kernel*;

- come visto in precedenza, un sistema esperto è un modello che tenta di riprodurre le azioni di una persona esperta in uno specifico campo di attività. Rappresenta, tradizionalmente, una delle prime applicazioni di apprendimento supervisionato;
- infine, gli algoritmi genetici descrivono un approccio ispirato all'evoluzione simulata della specie. La ricerca di soluzioni prende avvio con la creazione, o valutazione, di una popolazione di strategie iniziali. Grazie a processi come la mutazione casuale ed il *crossover*, la popolazione attuale dà luogo alle generazioni successive. Ad ogni passo, le diverse soluzioni sono valutate in base ad una funzione obiettivo, in modo tale che quelle migliori vengano selezionate probabilisticamente come semi per la popolazione futura.

### 1.2.2 *Unsupervised learning*

Un'altra tipologia di modelli di *machine learning* è l'apprendimento non supervisionato (*Unsupervised Learning, UL*). In maniera opposta alla versione supervisionata, il sistema in questo caso riceve solo i set di dati di input senza alcuna indicazione sugli output da ottenere. Il fine di questo metodo di apprendimento è quello di risalire a *pattern* nascosti, ovvero identificare nei dati di ingresso una struttura logica senza che questi siano stati preventivamente classificati. Questi algoritmi rappresentano la soluzione migliore nei casi in cui il problema insiste su dati di tipo numerico, visto che possono utilizzare le tecniche proprie della statistica e degli studi di distribuzione. Viceversa, sono metodi poco efficaci per quanto riguarda l'analisi di dati non numerici.

I principali problemi affrontati da sistemi di apprendimento non supervisionato sono il *clustering*, ovvero la ricerca di gruppi e classi all'interno dei dati, le regole di associazione, la stima della distribuzione dei dati, la visualizzazione dei dati in grafici o tabelle e la riduzione del volume di dati in rappresentazioni riassuntive.

Specificatamente al *clustering*<sup>22</sup>, esso è un algoritmo non supervisionato che riceve in input un set di dati non etichettato e restituisce gli stessi dati divisi per classi in base a similarità tra i dati di input. Le tecniche di *clustering* dunque si fondano sull'analisi della

---

<sup>22</sup> In italiano raggruppamento, dal termine inglese *cluster*, gruppo.

somiglianza tra gli esempi: somiglianza che può anche essere vista come misura di distanza in uno spazio multidimensionale, ovvero l'appartenenza ad una classe è determinata dalla vicinanza o meno dell'elemento rispetto all'insieme. Il *clustering* può essere di tipo *top-down* (divisivo), quando ha origine da un gruppo che viene suddiviso ricorsivamente in due o più gruppi, oppure *bottom-up* (aggregativo), quando ciascun elemento del campione di dati inizialmente è considerato un *cluster* a sé stante che viene poi unito a quelli con caratteristiche simili. In ogni caso, per guidare il sistema all'obiettivo, è necessario che l'individuo umano stabilisca il numero di classi che intende ottenere che rappresenta pertanto anche la condizione di arresto dell'algoritmo. Al di là di questa prima classificazione, le tecniche di *clustering* possono anche essere distinte in base alla possibilità o meno che un esempio venga assegnato a diversi *cluster*: nel primo caso avremo un *clustering* non-esclusivo (anche detto *soft-clustering* o *fuzzy clustering*); nel secondo un *clustering* esclusivo (*hard clustering*). Un'altra suddivisione infine riguarda la tipologia di algoritmo di divisione dello spazio:

- il *clustering* partizionale (non gerarchico, o *k-clustering*) è una classe di metodi che derivano dal più noto algoritmo di raggruppamento, quello delle *k-means*: ideato da MacQueen nel 1967, questo processo iterativo punta alla determinazione di K gruppi in cui suddividere gli oggetti, utilizzando la distanza di ciascun elemento dal cosiddetto centroide<sup>23</sup> del *cluster*. Con il progredire dell'algoritmo, i punti convergono sempre più ai centroidi dei nuovi *cluster*, andando a perfezionare le prestazioni di classificazione;
- il *clustering* gerarchico invece prevede una gerarchia di divisioni nella quale il numero di classi cresce (o decresce).

Le applicazioni più diffuse della tecnica del *clustering* sono quelle nel campo del marketing, per esempio per la profilazione dei clienti al fine di progettare campagne mirate in base alla tipologia di clientela. Ciononostante, i campi di utilizzo del *clustering* sono molteplici, dalle ricerche di mercato alla rielaborazione di immagini.

Un'altra tecnica di apprendimento supervisionato, il cui obiettivo è la modellizzazione di correlazioni tra i dati, sono le cosiddette regole di associazione. Esse rappresentano un buono strumento per determinare le relazioni di dipendenza fra i feature nel set di dati,

---

<sup>23</sup> Termine col quale si definisce il punto medio di un cluster.

senza avere alcuna informazione a priori. In base alla tipologia di attributi esaminati, si distinguono diverse regole di associazione: tradizionali, quando si fa riferimento a feature categorici; quantitative quando si utilizzano valori continui per gli attributi; ed infine regole multilivello, definite in base ad una scala gerarchica di generalizzazioni. Anche qui, esistono numerosi ambiti di applicazione per le regole di associazione, uno fra tutti i sistemi di raccomandazione utilizzati nel marketing.

### 1.2.3 Reinforcement learning

Terzo ed ultimo paradigma del *machine learning* è l'apprendimento per rinforzo (*Reinforcement Learning, RL*<sup>24</sup>): il software, qui, interagisce con un ambiente dinamico e con un sistema di "ricompense" e "punizioni" che lo guida verso il miglioramento delle proprie prestazioni. Una volta ottenuti i dati di ingresso, il computer deve raggiungere un obiettivo che viene stabilito in base ad una funzione di valore che gli dice solamente se lo scopo è stato raggiunto oppure no. In questo modo, grazie ai feedback, il comportamento del sistema progredisce tramite un processo di tipo *trial and error* (tentativi ed errori). È sostanzialmente lo stesso processo che l'uomo affronta quando deve imparare ad andare in bici ad esempio: all'inizio si cade spesso, finché, un po' alla volta, non si apprende quali sono le mosse sbagliate, come bilanciare i pesi, quale la posizione da mantenere. Con il passare del tempo, la valutazione di ciò che ha funzionato e ciò che invece porta a cadere, consente di apprendere ad andare in bici. Lo stesso vale per i computer che utilizzano algoritmi di *reinforcement learning*: provano diverse azioni, valutano quali di queste hanno fornito il miglior risultato, rinforzano quelle che hanno funzionato e via dicendo. Un semplice esempio di applicazione dell'apprendimento per rinforzo sono i robot che imparano a camminare: prima il robot compie un passo molto grande e cade; il risultato della caduta dovuta ad un passo così grande è un dato che il sistema memorizza; il tentativo successivo vedrà pertanto un passo più piccolo come aggiustamento dell'errore commesso; il robot così impara ad avanzare camminando.

A differenza dei due precedenti modelli di *machine learning*, l'apprendimento per rinforzo non necessita un dataset iniziale da cui partire, ma può semplicemente prendere le sue prime scelte in modo assolutamente casuale: i risultati che otterrà man mano

---

<sup>24</sup> Anche noto come programmazione neuro-dinamica.

guideranno le sue scelte future. Per questo stesso motivo, tale paradigma si occupa di processi sequenziali, in cui l'operazione da compiere dipende dallo stato attuale del sistema e determina la situazione futura. È un processo di interazione con l'ambiente, e non più una relazione con un campione di dati già fornito in partenza.

Nel *reinforcement learning*, vengono definiti rispettivamente:

- un insieme di stati  $s_t \in S$ ;
- un insieme di azioni  $a_t(s_t)$  che possono essere intraprese in ragione dello stato attuale;
- le probabilità di transizione  $\mathbb{P}_a(s_t, s_{t+1})$  che definiscono la probabilità che si raggiunga lo stato successivo  $s_{t+1}$ , dato lo stato iniziale  $s_t$  ed un'azione  $a$ ;
- il valore atteso della ricompensa  $R_a(s_t, s_{t+1})$ ;
- il fattore di sconto  $\gamma \in [0,1]$ , che rappresenta la differenza fra *present rewards* e *future rewards*;
- la funzione di valore  $V^\pi(s)$ , che dovendo essere massimizzata, guida la strategia intrapresa dal sistema.

Ad ogni passo, l'agente riceve una ricompensa come conseguenza dell'azione nel passo precedente e dello stato nuovo in cui si trova. La ricompensa rappresenta dunque un valore numerico del benessere o della soddisfazione dell'agente. L'obiettivo di quest'ultimo, però, non è la semplice ricompensa, ma piuttosto la massimizzazione del rendimento globale, inteso come funzione della ricompensa attuale e di quelle future scontate:

$$R(s_t) = r(s_t, a_t, s_{t+1}) + \gamma r(s_{t+1}, a_{t+1}, s_{t+2}) + \gamma^2 r(s_{t+2}, a_{t+2}, s_{t+3})$$

Questi processi lavorano grazie ad una funzione, chiamata *policy*,  $\pi(s_t) = a_t$ , che, per ogni stato  $s$ , identifica l'azione  $a$  da attuare. Tipicamente, questa funzione utilizza il valore atteso di una somma di ricompense future scontate. Grazie alla proprietà di Markov, che permette di considerare un processo stocastico esclusivamente in funzione dell'ultimo stato e non di quelli passati, la politica ottimale  $\pi^*$  (intesa come la politica che massimizza la funzione di valore) può essere considerata funzione del solo stato  $s$ .

In particolare, il valore di un'azione  $a_t = a$  in uno stato  $s_t = s$  sotto la *policy*  $\pi(\cdot)$  può essere espresso come:

$$Q^\pi(s, a) = E[R^\pi(s_t) | s_t = s, a_t = a]^{25}$$

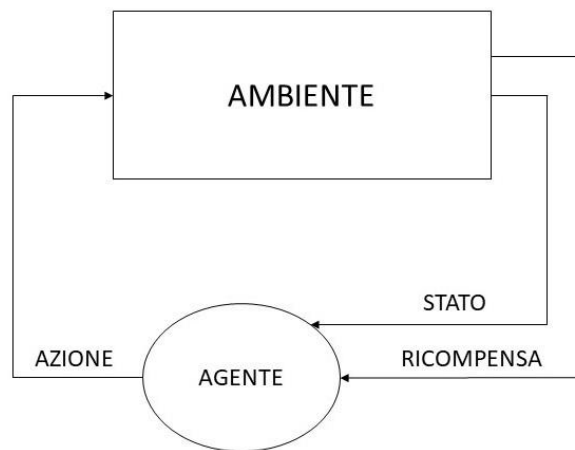
che, grazie alle relazioni ricorsive viste, equivale a:

$$V^\pi(s) = E[r(s_t, \pi(s_t), s_{t+1}) + \gamma V^\pi(s_{t+1}) | s_t = s]$$

Questa equazione è nota come equazione di Bellman, dal nome del matematico statunitense che l'ha formulata per la prima volta nel 1957.

Generalmente il *reinforcement learning* è modellizzato tramite i processi decisionali di Markov<sup>26</sup> (MDP), così rappresentabili:

Grafico 1. Markov Decision Process



Tra i più popolari algoritmi di reinforcement learning vi sono i metodi mediante differenza temporale (*Temporal Difference methods, TDms*), tra i quali il più conosciuto è il *Q-learning* e l'algoritmo SARSA. Prima di procedere alla loro trattazione però, serve introdurre la distinzione fra approcci *on-policy* e *off-policy*. Un agente, o allievo, *off-policy* aggiorna la propria funzione di valore utilizzando il rendimento legato allo stato successivo e l'azione ottimale per esso: in altre parole, valuta il rendimento di una coppia stato-azione assumendo che sia stata seguita la politica ottimale, nonostante il fatto che magari non si stia seguendo la miglior politica possibile. In sostanza, significa che si stanno utilizzando due politiche: una per stimare la funzione di valore ed un'altra per controllare il processo di miglioramento. Un allievo *on-policy*, invece, utilizza come base della propria funzione di

<sup>25</sup> Dove la Q sta per Q-value.

<sup>26</sup> Dal nome del matematico russo Andrej Andreevic Markov, il quale per primo ne ha dato formulazione agli inizi del '900.

valore lo stato successivo e l'azione data dalla politica corrente: stima, cioè, il rendimento della coppia stato-azione assumendo che la politica continui ad essere seguita.

Grafico 2. Algoritmi Q-learning e SARSA

---

<p><b>Algorithm</b> The off-policy <math>Q(\lambda)</math> algorithm.</p> <hr/> <p>Initialize <math>Q(s, a)</math> randomly and <math>e(s,a)=0</math> for all <math>s,a</math>  Repeat (for each episode):      <math>t \leftarrow 0</math>      Initialize <math>s_t</math>      Choose <math>a_t</math> from <math>s_t</math> using policy derived from <math>Q(\epsilon\text{-greedy})</math>      Repeat (for each step of episode)          Take action <math>a_t</math>, observe <math>r_t</math> and <math>s_{t+1}</math>          Choose <math>a_{t+1}</math> from <math>s_{t+1}</math> using policy derived from <math>Q(\epsilon\text{-greedy})</math>          <math>a^* \leftarrow \underset{b}{\arg \max} Q(s_{t+1}, b)</math> (if <math>a_{t+1}</math> ties for the max, then <math>a^* \leftarrow a_{t+1}</math>)          <math>\delta \leftarrow r_t + \gamma Q(s_{t+1}, a^*) - Q(s_t, a_t)</math>          <math>e(s_t, a_t) \leftarrow 1</math>          For all <math>s,a</math>:              <math>Q(s, a) \leftarrow Q(s, a) + \alpha \delta e(s, a)</math>              If <math>a_{t+1} = a^*</math>, then <math>e(s, a) \leftarrow \gamma \lambda e(s, a)</math>              else <math>e(s, a) \leftarrow 0</math>          End For          <math>s_t \leftarrow s_{t+1}; a_t \leftarrow a_{t+1};</math>          <math>t \leftarrow t + 1</math>      until <math>s_t</math> is terminal state</p> <hr/>	<p><b>Algorithm</b> The on-policy SARSA(<math>\lambda</math>) algorithm.</p> <hr/> <p>Initialize <math>Q(s, a)</math> randomly and <math>e(s,a)=0</math> for all <math>s,a</math>  Repeat (for each episode):      <math>t \leftarrow 0</math>      Initialize <math>s_t</math>      Choose <math>a_t</math> from <math>s_t</math> using policy derived from <math>Q(\epsilon\text{-greedy})</math>      Repeat (for each step of episode)          Take action <math>a_t</math>, observe <math>r_t</math> and <math>s_{t+1}</math>          Choose <math>a_{t+1}</math> from <math>s_{t+1}</math> using policy derived from <math>Q(\epsilon\text{-greedy})</math>          <math>\delta \leftarrow r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)</math>          <math>e(s_t, a_t) \leftarrow 1</math>          For all <math>s,a</math>:              <math>Q(s, a) \leftarrow Q(s, a) + \alpha \delta e(s, a)</math>              <math>e(s, a) \leftarrow \gamma \lambda e(s, a)</math>          End For          <math>s_t \leftarrow s_{t+1}; a_t \leftarrow a_{t+1};</math>          <math>t \leftarrow t + 1</math>      until <math>s_t</math> is terminal state</p> <hr/>
--	---

Fonte: Pendharkar, P. C., & Cusatis, P. (2018), "Trading financial indices with reinforcement learning agents", *Expert Systems with Applications*, 103, pp. 1-13

Il *Q-learning* è un algoritmo *off-policy* di *reinforcement learning* che si basa sull'equazione di Bellman, e pertanto l'obiettivo di questo modello è massimizzare il *Q-value*:

$$Q^*(s, a) = E_{s_{t+1}} [r(s_t, \pi(s_t), s_{t+1}) + \gamma \max Q^\pi(s_{t+1}, a_{t+1}) | s, a]$$

L'algoritmo SARSA (acronimo di *State-Action-Reward-State-Action*) è un modello molto simile al *Q-learning*, con la peculiare differenza che presenta un approccio *on-policy*. L'acronimo sta ad indicare che la funzione di aggiornamento dei *Q-value* dipende esclusivamente dalla quintupla  $(s_t, a_t, r_t, s_{t+1}, a_{t+1})$ .

In Grafico 2 si presentano gli pseudocodici di entrambi gli algoritmi<sup>27</sup>.

Per quanto riguarda l'utilizzo del *reinforcement learning*, grazie a questi modelli le macchine sono in grado di imparare a svolgere diverse attività complesse, come battere un avversario in un gioco o guidare autonomamente un veicolo: l'obiettivo è quello di massimizzare il valore della ricompensa e, con il tempo, ovvero con l'accumularsi degli errori commessi, la macchina perfeziona il proprio comportamento ottimizzando le prestazioni. In molti settori complessi, il *reinforcement learning* è l'unica possibilità di ottenere un programma con prestazioni ad alti livelli: per esempio, nell'utilizzo dell'intelligenza artificiale nei giochi, è di notevole difficoltà per un essere umano fornire accurate e rilevanti valutazioni di un vasto numero di posizioni, necessarie per sviluppare una funzione di valore direttamente dagli esempi forniti. Invece, se semplicemente si dice al sistema quando ha vinto e quando ha perso, esso può usare questo feedback per stimare una funzione che restituisca con ragionevole accuratezza la probabilità di vittoria per ogni possibile mossa e posizione. È il caso questo del risultato ottenuto dal software AlphaGo, sviluppato dai ricercatori di Deep Mind di Google, che è riuscito a sconfiggere per la prima volta nella storia un campione umano del gioco del go<sup>28</sup>, missione ben più complessa di quella di Deep Blue con il gioco degli scacchi. Come appare evidente, per costruzione, il *reinforcement learning*, necessita di tempo prima di ottenere buoni risultati, ma la velocità di miglioramento delle prestazioni è esponenziale.

#### 1.2.4 Artificial neural networks e deep learning

Non si può parlare di *deep learning* (DL) senza parlare di reti neurali artificiali (*artificial neural networks*, ANN o semplicemente NN): i due concetti sono infatti strettamente connessi e spesso vengono considerati interscambiabili. Tuttavia, è bene ricordare che le

---

<sup>27</sup> Il parametro  $\alpha$  indica il tasso di apprendimento (compreso tra 0 e 1), che solitamente viene fissato a 0,1.

<sup>28</sup> Nell'ottobre del 2015 il software AlphaGo sconfisse con il punteggio di 5-0 il campione europeo Fan Hui.

reti neurali rappresentano una forma di algoritmo di apprendimento automatico che non si applica solamente al caso “profondo”, ma può benissimo essere utilizzato anche in modelli con o senza supervisione o per rinforzo. Allo stesso tempo il *deep learning* non rappresenta una categoria distinta dai paradigmi visti, ma piuttosto una classe trasversale di modelli di *machine learning*. Si consideri, ad esempio, la tecnica del *deep reinforcement learning*, mistione appunto di apprendimento profondo e di rinforzo.

Come anticipa il nome, un’ANN è un modello computazionale che si ispira vagamente<sup>29</sup> ai circuiti di neuroni biologici presenti nel cervello animale. Sono infatti formate da unità, chiamate “neuroni artificiali”, connesse fra loro tramite “sinapsi” (qui detti *edges*) che consentono la trasmissione di segnali da un neurone all’altro. Ciò che si viene a formare è un reticolo in cui i neuroni rappresentano i nodi della rete e gli *edges* i collegamenti tra quest’ultimi: il neurone ricevente (detto postsinaptico) elabora i segnali e li trasmette ai neuroni a valle collegati ad esso, e così per l’intera rete. Solitamente questi segnali presentano uno stato numerico che oscilla tra 0 e 1; l’output che esce da ogni neurone viene ottenuto tramite qualche funzione non-lineare che “somma” i diversi dati di input che riceve dai neuroni a monte. Neuroni e sinapsi di norma presentano pesi diversi e che possono variare nel tempo, ovvero, con il procedere dell’apprendimento, possono aumentare o diminuire la forza del segnale che inviano a valle. Inoltre, possono essere presenti anche delle soglie (*threshold*), tali che, se il segnale che arriva al neurone non raggiunge un determinato valore, questo non viene trasmesso ai neuroni successivi.

Tipicamente, i neuroni vengono organizzati in strati (*layers*). Diversi strati possono eseguire differenti tipologie di trasformazioni sui dati di input. I segnali dunque viaggiano dal primo strato d’ingresso all’ultimo strato di uscita, talvolta dopo aver attraversato gli strati intermedi più volte. In questi casi le ANN vengono definite *deep neural network* (DNN) proprio per sottolineare la presenza di diversi “piani” di apprendimento, esattamente ciò che avviene nel deep learning.

Ulteriori elementi, poi, che influiscono sul funzionamento delle reti neurali, sono: il tasso di apprendimento (iperparametro già introdotto nel caso del *reinforcement learning*); la funzione di costo, o valore, che valuta i risultati e fornisce i feedback; ed infine la retropropagazione, metodo che regola e determina i pesi delle connessioni per compensare

---

<sup>29</sup> L’avverbio “vagammente” vuole sottolineare come una reale comprensione delle reti neurali biologiche richiederebbe una complessità ben maggiore.



gli errori riscontrati durante l'apprendimento, e che funziona tramite la discesa stocastica del gradiente relativo ad ogni stato precedente.

Nel corso degli anni, sono state sviluppate diverse forme di reti neurali. Tra queste, hanno riscontrato grande successo le reti neurali convoluzionali (*Convolutional Neural Network*, CNN) e le reti antagoniste generative (*Generative Adversarial Network*, GAN). Senza voler entrare troppo nel merito, le prime sono una classe di reti neurali profonde contraddistinte dal fatto che ogni neurone è connesso a tutti i nodi che compongono lo strato successivo. Intuitivamente, questa peculiarità consente una velocità di apprendimento enorme. Non per altro, esse sono largamente utilizzate nell'analisi visiva di immagini e testi, dal campo medico a quello del marketing. Ad oggi, rappresentano una delle maggiori applicazioni in assoluto del machine learning. Per quanto riguarda invece le reti GAN, esse sfruttano una struttura che pone in competizione diverse reti neurali all'interno dello stesso *framework*.

### 1.2.5 Forme ibride di machine learning

Oltre a queste macrocategorie, esistono dei metodi di apprendimento automatico cosiddetti ibridi, poiché rappresentano delle vie di mezzo rispetto agli approcci poc'anzi esposti. La linea di demarcazione fra *supervised* e *unsupervised learning* non è netta, anzi è all'interno di questo spazio che si inseriscono campi di studio come il *semi-supervised learning*, il *self-supervised learning* ed il *multi-instance learning*.

L'apprendimento semi-supervisionato è esattamente una via di mezzo tra il *supervised learning* e l'*unsupervised learning*. In particolare, al sistema viene fornito per l'apprendimento un *training set* di dati "incompleto": la maggior parte delle informazioni di ingresso è priva dei rispettivi esempi di output (come nell'apprendimento senza supervisione), mentre i rimanenti dati sono accompagnati dai risultati d'uscita (come nell'apprendimento con supervisione). Lo scopo, tuttavia, rimane identico: identificare schemi e regole per la risoluzione di problemi, nonché *pattern* utili al raggiungimento di particolari fini. Questa applicazione del *machine learning* trova particolare utilità in quelle situazioni del mondo reale in cui la difficoltà progettuale e la potenza computazionale richieste per una procedura classica di *supervised learning*, potrebbero rivelarsi troppo costose, anche in termini di tempo. Per esempio, classificare delle fotografie, secondo un processo supervisionato, richiederebbe un vasto dataset di immagini già etichettate da

soggetti umani, ed è immaginabile che, per ottenere buoni risultati, il sistema debba avere una quantità di dati tale da richiedere uno sforzo troppo elevato ai programmatori. Inoltre, in diverse applicazioni pratiche, i dati etichettati risultano scarsi, mentre i volumi dei dati non classificati appaiono elevati. È per questi motivi che tale forma ibrida di *machine learning* trova grande realizzazione, per esempio, in campi come la visione artificiale<sup>30</sup> (dati sotto forma di immagini), l'elaborazione del linguaggio naturale<sup>31</sup> (dati di testo) ed il riconoscimento automatico della voce (dati audio).

L'apprendimento auto-supervisionato (più comunemente chiamato *self-supervised learning*) è una tecnica di *machine learning* relativamente recente che si distingue per la classificazione autonoma (o automatica) dei *training data* di input. È definibile come una forma di apprendimento supervisionato, tuttavia i set di dati non necessitano di essere classificati da un umano, ma possono, per esempio, essere etichettati grazie all'estrazione da parte del sistema stesso di relazioni fra i differenti segnali di input (ovvero informazioni di ingresso provenienti da diversi sensori). Più propriamente, sarebbe una forma di apprendimento non supervisionato dove i dati stessi conferiscono la supervisione: è uno slittamento di paradigma non indifferente. Una conseguenza naturale, ed un vantaggio, del *self-supervised learning* è pertanto che può essere eseguito più facilmente (rispetto alle altre forme già viste) in un ambiente online, ove i dati vengono raccolti ed etichettati istantaneamente dal sistema senza l'intervento umano, e dove i modelli possono essere addestrati completamente da zero. Dunque, risulta molto adatto anche ad applicazioni con ambienti e dati in continuo mutamento. In particolare, ha riscontrato successo nella stima della profondità relativa dei campi visivi sottoposti ad esame, grazie all'utilizzo di tecniche di segmentazione del movimento che sfruttano dei vincoli geometrici tra lo spazio d'azione della scena e quello della telecamera. A livello pratico, il *self-supervised learning* viene utilizzato principalmente nel campo della robotica, specie nella chirurgia robotica e nella

---

<sup>30</sup> Dall'inglese *computer vision*, è il campo scientifico interdisciplinare che cerca di riprodurre nei computer il sistema visivo umano, con ciò intese l'acquisizione, elaborazione e analisi di immagini digitali. Un esempio recente dell'applicazione del semi-supervised learning in questa branca sono le fotocamere in grado di riconoscere il volto delle persone.

<sup>31</sup> Detta anche NLP (Natural Language Processing), è un campo che trae origine dalla linguistica, la computer science, l'ingegneria dell'informazione e l'AI, e che tratta l'interazione tra i sistemi informatici ed il linguaggio umano, in modo tale da processare ed analizzare grandi volumi di dati di testo. Le applicazioni vanno dal riconoscimento di testi alla generazione di risposte automatiche da parte di assistenti virtuali.

stima di ostacoli e asperità nei veicoli a guida autonoma, che utilizzano per l'appunto sistemi con apprendimento auto-supervisionato quali telecamere monoculari e LIDAR<sup>32</sup>.

Infine, il *multi-instance learning* (MIL) è una forma di apprendimento supervisionato nel quale, al posto di ricevere un set di esempi singolarmente classificati, il sistema parte da un gruppo di cd. *bags* classificate, ciascuna contenente diverse realizzazioni<sup>33</sup>. Di seguito un semplice esempio di MIL<sup>34</sup>: si immagina di avere una moltitudine di persone, ciascuna delle quali possiede un mazzo con diverse chiavi. Alcune di queste persone sono in grado di aprire una stanza, mentre altre no. L'obiettivo è quello di stabilire quando una determinata chiave o un determinato mazzo di chiavi consente di entrare nella stanza. Per risolvere la questione serve determinare quale chiave è comune a tutti i mazzi di chiavi grazie ai quali è possibile aprire la porta. Se è possibile identificare correttamente la chiave, è anche possibile classificare tutti i mazzi di chiavi (positivo se contiene la suddetta chiave, negativo in caso contrario). Ovviamente, dopo l'iniziale periodo di addestramento del sistema, il compito del software rimane quello di predire la classificazione delle nuove bags non ancora esaminate. Questo tipo di apprendimento è utilizzato, specialmente, nella classificazione di immagini, testi e documenti, di conseguenza, nello studio dell'attività molecolare e nel supporto all'analisi di immagini mediche strumentali.

### 1.2.6 Forme alternative di machine learning

I paradigmi visti hanno il grande pregio di permettere una prima classificazione delle diverse attuazioni del machine learning nel mondo applicato. Ciononostante, il novero delle modalità di apprendimento automatico non si esaurisce con essi, né con le forme ibride precedentemente esposte: esistono, infatti, diversi punti di vista con i quali approcciare la materia, considerando tra l'altro che questa appare oggi una scienza in velocissima evoluzione, con mistioni in molte discipline parallele. La grande fruibilità e adattamento dell'apprendimento automatico consente l'utilizzo di forme miste a seconda del risultato ricercato, cosicché la catalogazione delle modalità di machine learning diventa meno netta

---

<sup>32</sup> Acronimo di Laser Imaging Detection And Ranging, rappresenta un'attività di telerilevamento che consente di valutare la distanza di oggetti e superfici tramite impulsi laser.

<sup>33</sup> Per necessità di scorrevolezza, realizzazioni ed esempi vengono utilizzati, qui, come sinonimi.

<sup>34</sup> Tratto da: Babenko, B. (2008) "Multiple instance learning: algorithms and applications", Dept. of Computer Science and Engineering, University of California, San Diego.

e più complessa. È così che, ad esempio, è possibile studiare forme di apprendimento che discendono dall'inferenza statistica o dalle scienze dell'educazione e dell'informazione, che in ogni caso non hanno la pretesa di essere avulse dalle precedenti forme, ma piuttosto di rappresentare un diverso punto di vista.

Con inferenza statistica si suole indicare un processo di generalizzazione dei risultati, ottenuti a partire da una rilevazione parziale, all'intera popolazione da cui deriva il campione. Nel *machine learning*, l'adattamento dei modelli e la stima di previsioni sono il risultato di un processo inferenziale. Esistono diversi paradigmi di inferenza che possono aiutare a comprendere gli algoritmi e i problemi di *machine learning*. I principali paradigmi sono: l'apprendimento induttivo, l'apprendimento deduttivo e l'apprendimento trasduttivo.

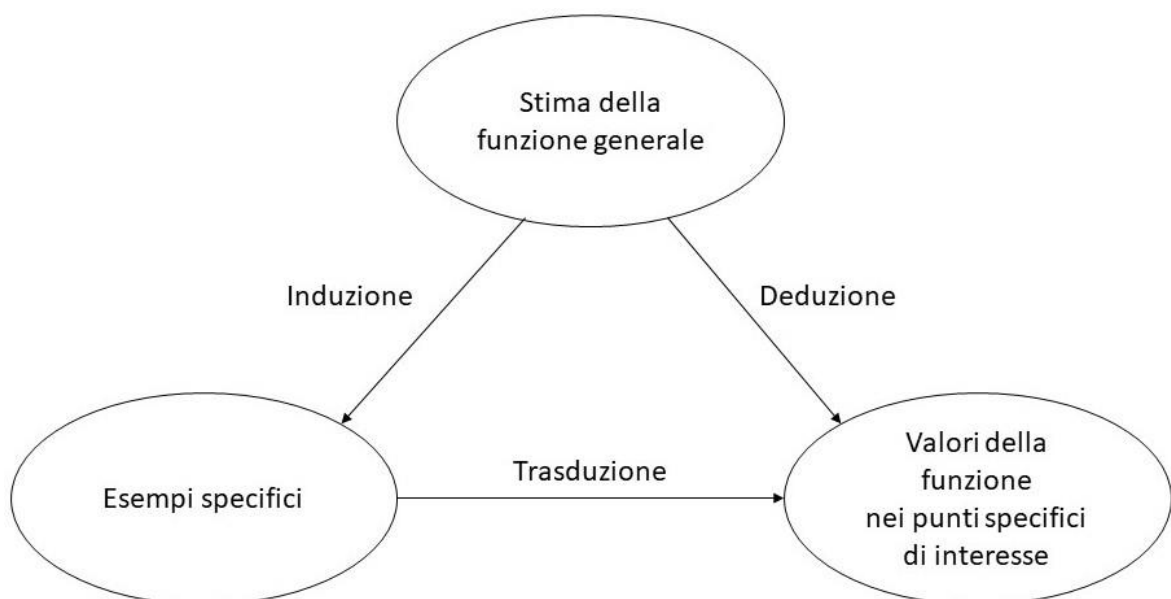
Il ragionamento induttivo si riferisce all'utilizzo di esempi specifici per determinare risultati generali, ovvero utilizza un approccio *specific-to-general*. La maggior parte dei modelli di *machine learning* apprende grazie all'inferenza induttiva dove le regole generali (il modello) sono tratte da specifici esempi storici (i dati di ingresso del campione). Sostanzialmente, è il problema di come ottenere conclusioni generali sul futuro, partendo però da osservazioni del passato. Il modello viene ricavato dai dati di addestramento, ma gli viene richiesto di funzionare su nuovi dati non ancora visti. In mancanza di ulteriori informazioni, l'ipotesi è che l'opzione migliore, per quanto riguarda i casi non ancora visti, rimanga quella che meglio si adatta ai dati osservati nel passato. Questo è il presupposto fondamentale dell'apprendimento induttivo ed è ciò che troviamo alla base della programmazione logica induttiva (dall'inglese *Inductive Logic Programming*, ILP, spesso utilizzato come sinonimo di apprendimento relazionale), sottoarea del *machine learning* che funge da punto di contatto con la programmazione logica. In particolare, per il ragionamento induttivo, si fa ricorso alla logica di primo ordine, linguaggio formale utilizzato per gestire meccanicamente ragionamenti che includono connettivi logici, relazioni e quantificatori.

L'inferenza deduttiva si occupa di determinare risultati specifici partendo da regole generali. Logicamente, il concetto di deduzione è l'esatto opposto dell'induzione: la prima è una forma di ragionamento di tipo *top-down* che cerca di soddisfare tutte le premesse prima di indicare la conclusione, mentre la seconda è un ragionamento di tipo *bottom-up* che usa i dati a disposizione come prova di un risultato. Nel campo del *machine learning*,

una volta che si utilizza l'apprendimento induttivo per adattare il modello sui dati di formazione, il modello ottenuto può essere usato per formulare previsioni. Quest'ultima applicazione è una forma di deduzione, o inferenza deduttiva. Tali metodi deduttivi sono tipicamente usati per velocizzare sistemi di *problem solving* fornendo conoscenze che possono essere ricavate deduttivamente dall'informazione esistente, permettendo di trovare soluzioni in modo più rapido.

Infine, la trasduzione viene utilizzata nel ramo dell'apprendimento statistico con riferimento alle previsioni di specifici esempi, dati altri specifici esempi da un determinato dominio. In altre parole, la trasduzione coglie la funzione sconosciuta per i punti di interesse, partendo direttamente da altri dati raccolti. A differenza dell'induzione, pertanto, non è richiesta alcuna generalizzazione: gli esempi specifici vengono qui usati direttamente (è un approccio del tipo *specific-to-specific*, vedi Grafico 1). Questo paradigma può consentire un processo più semplice per risolvere un problema. È un nuovo concetto di inferenza che risulta molto utile nelle situazioni in cui si vuole ottenere il miglior risultato possibile a partire da una quantità ristretta di informazioni. Esempio classico di inferenza trasduttiva applicata all'apprendimento automatico è l'algoritmo *k-Nearest Neighbors* che, come suggerisce il nome, utilizza le caratteristiche degli oggetti vicini a quelli in oggetto, il tutto con lo scopo di riconoscere i pattern necessari alla classificazione, ma senza il bisogno di alcuna modellizzazione generale.

Grafico 3. I tre approcci di inferenza



Per quanto concerne gli approcci che discendono dalle scienze dell'apprendimento, è possibile riconoscere diverse tecniche:

- il *multi-task learning* è una forma di apprendimento supervisionato che prevede l'adattamento di un modello su un *dataset* che affronta più problemi correlati. L'ideazione di uno schema che viene addestrato al fine di risolvere più problemi consente prestazioni migliori rispetto al training per un singolo compito. L'apprendimento *multi-tasking* è una via per implementare la generalizzazione mettendo in comune gli esempi derivanti dalle diverse funzioni. È, pertanto, un approccio che si dimostra utile in quei casi in cui vi è abbondanza di dati di ingresso classificati per un compito, ma che allo stesso tempo possono essere condivisi con un altro compito avente molti meno dati etichettati. In tale configurazione, ogni output può essere ottenuto da una diversa parte del modello di input, consentendo così al sistema comune di generalizzare gli stessi esempi attraverso ciascun compito richiesto. Allo stesso modo in cui i *training data* aggiuntivi modificano i parametri del modello ottenendo una migliore generalizzazione, quando un *pattern* (o parte di esso) è condiviso tra i compiti, questo modello contiene dei coefficienti più vicini al DGP reale (supposto che la condivisione abbia una giustificazione logica). Un esempio popolare di apprendimento multitasking è quello in cui una stessa parola viene utilizzata per imparare una distribuzione logica di parole nel testo, la quale a sua volta viene poi condivisa in differenti compiti di elaborazione del linguaggio naturale. In questo caso la macchina da un lato impara a cogliere l'utilizzo di una determinata parola all'interno di frasi e testi, e dall'altro utilizza questa conoscenza per compiti di elaborazione di testi, risposte, traduzioni ecc.;
- l'*active learning* è una tecnica in cui il sistema è in grado di interrogare l'operatore umano durante il processo di apprendimento per sciogliere le ambiguità sorte durante il procedimento stesso. In altre parole, l'allievo raccoglie in maniera adattiva o interattiva (da qui la formula apprendimento attivo) gli esempi necessari alla sua formazione, solitamente chiedendo alla componente umana le etichette dei nuovi punti non conosciuti. Si connota dunque come una forma di apprendimento supervisionato, che tenta di ottenere migliori prestazioni rispetto al cosiddetto apprendimento "passivo" dal momento che dovrebbe rivelarsi più efficiente: l'idea alla base dell'apprendimento attivo, infatti, è che un algoritmo di

*machine learning* può raggiungere una maggior precisione, pur con meno coppie input-output in fase di addestramento, se gli è permesso scegliere i dati da cui imparare. Sotto questo punto di vista, l'*active learning* può anche esser visto come un approccio di apprendimento semi-sorvegliato, o una soluzione alternativa agli stessi tipi di problemi. “[...] si osservi che l'*active learning* ed il *semi-supervised learning* affrontano lo stesso problema da direzioni opposte. Mentre i metodi di apprendimento semi-supervisionati sfruttano ciò che l'allievo pensa di sapere sui dati non etichettati, i metodi attivi tentano di esplorare gli aspetti sconosciuti. È dunque naturale immaginare di poter combinare i due approcci.”<sup>35</sup> Ciò è particolarmente utile quando non ci sono molti dati disponibili e la raccolta/classificazione di dati può rivelarsi costosa, per esempio come nel campo della biologia computazionale. L'utilizzo dell'apprendimento attivo consente, insomma, di orientare il campionamento in modo tale da minimizzare il numero di esempi da utilizzare per l'addestramento e massimizzare l'efficacia del modello;

- l'*online learning* comporta l'utilizzo di dati a disposizione e l'implementazione del modello direttamente prima che sia necessaria una previsione, oppure subito dopo che è stata raccolta l'ultima osservazione. È un metodo appropriato per quei problemi in cui le osservazioni sono fornite nel corso del tempo ed in cui si prevede che la stessa distribuzione di probabilità dei dati possa cambiare nel tempo. Pertanto, ci si aspetta che il modello cambi altrettanto frequentemente al fine di cogliere e sfruttare tali mutamenti. Tradizionalmente il *machine learning* ha sempre avuto applicazione *offline*, ovvero, una volta ricevuto un set di dati definiti, si è sempre tentato di ottimizzare un'equazione che descrivesse i pattern presenti. Tuttavia, negli ultimi anni, grazie ai progressi tecnologici in precedenza citati, si è iniziato ad analizzare i dati in streaming, in modo tale da poter aggiornare le stime passo per passo, piuttosto che doverlo fare a posteriori. Questo è di particolare utilità nelle situazioni in cui i dati possono cambiare rapidamente nel tempo, oppure nei casi in cui i cambiamenti sono gradualmente ma sia richiesta una raccolta di dati su grande scala ed in costante crescita. In generale, ciò che guida l'apprendimento

---

<sup>35</sup> Tratto e tradotto da: Settles, B. (2009), “Active Learning Literature Survey”, *University of Wisconsin-Madison Department of Computer Sciences*.

online è il concetto di “rimpianto”, inteso come la differenza tra quanto bene il modello ha funzionato rispetto a quanto avrebbe potuto funzionare se le informazioni disponibili fossero state raccolte in blocco. Porre come obiettivo l’ottenimento del minor “rimpianto” possibile significa cercare di minimizzare la perdita media subita rispetto all’ipotetica situazione che si sarebbe potuta ottenere utilizzando un unico valore fisso per il parametro (che tuttavia si può ricavare solo a posteriori da un unico modello che presenta il miglior coefficiente possibile per quei dati). Un esempio di apprendimento online si trova nella cosiddetta discesa stocastica, o discesa a gradiente, algoritmo utilizzato per adattare in tempo reale una rete neurale artificiale. Il vantaggio di utilizzo di questo algoritmo viene evidenziato proprio nelle applicazioni dell’*online learning*;

- il *transfer learning* è una modalità di apprendimento in cui il sistema viene prima addestrato su un compito per poi essere applicato, in toto o in parte, come punto di partenza per un compito correlato. In altre parole, all’allievo è richiesto di eseguire due o più compiti differenti, assumendo che molti dei fattori che incidono sulle variazioni del parametro A siano rilevanti per le variazioni di un secondo parametro B. Si tratta di un approccio utile in problemi in cui vi è un compito principale ed un compito correlato, il quale possiede una gran quantità di dati. Potrebbe sembrare uguale all’apprendimento *multi-task*, tuttavia non lo è: qui i compiti vengono appresi in sequenza, mentre nel *multitasking* il processo cerca di ottenere buone prestazioni contemporaneamente in tutti i compiti considerati. Un esempio di questo approccio è la classificazione delle immagini, ove un modello predittivo, come per esempio una rete neurale artificiale, può essere allenato su una grande quantità di immagini generali, mentre i pesi del modello stesso possono essere utilizzati come punto di partenza per l’addestramento su un campione inferiore di immagini: le caratteristiche apprese dal modello sul compito più ampio, come l’estrazione di linee e pattern ricorrenti, saranno applicate al nuovo compito correlato. Se nella prima fase generale si utilizzano molti dati, l’apprendimento nella seconda fase sarà molto più veloce a generalizzare partendo solo da pochi esempi. Come detto, il *transfer learning* viene utilizzato in modelli formati in modo incrementale, e un modello esistente può essere punto di partenza per la



formazione continua, cosa che avviene, ad esempio, nel caso delle reti neurali alla base del *deep learning*;

- *l'ensemble learning*, o apprendimento di gruppo, è un approccio in cui due o più modelli si adattano agli stessi dati e dunque le previsioni di ciascun modello vengono combinate. Esistono diversi modi di combinare le previsioni, tra questi la ponderazione uniforme e la scelta di pesi basati su un set di validazione. Ovviamente, essendo l'obiettivo quello di ottenere prestazioni migliori con l'insieme dei modelli rispetto all'utilizzo di un modello unico, diventa influente sia la decisione su come creare i diversi modelli, sia le modalità con cui si intende combinare le previsioni. Pertanto, l'apprendimento di gruppo può essere scisso in due compiti: lo sviluppo, prima, di una popolazione di allievi di base partendo dai dati di formazione, e la combinazione poi per formare il predittore composito. L'utilità di questo metodo risiede nella possibilità di migliorare l'abilità predittiva su domini problematici, ove risulta necessario ridurre la varianza degli algoritmi stocastici. Alcuni esempi popolari di algoritmi utilizzati in questo campo sono il *boosting*, il *bagging*, lo *stacking* e il *random forest*. Questi approcci sono stati sviluppati negli ultimi due decenni e le loro prestazioni si sono spesso dimostrate sorprendentemente buone.

Conclusa la descrizione dei paradigmi del *machine learning*, delle sue forme ibride e alternative, si introduce ora una parentesi sul significato di *data mining*, e sulle relazioni tra quest'ultimo e l'apprendimento automatico.

### 1.3 DATA MINING E MACHINE LEARNING

Per prima cosa bisogna introdurre il concetto fondamentale di *big data*: con questo termine si indicano raccolte di dati talmente elevate da necessitare specifiche tecnologie e modellazioni, creati ad hoc per estrapolarne del contenuto informativo.

Riprendendo il modello delle "3V" dell'analista Doug Laney, successivamente esteso a "5V", vi sono cinque aspetti fondamentali da considerare:

- Volume: la quantità di dati generati ogni secondo da sorgenti eterogenee;
- Varietà: la differente tipologia di dati creati, accumulati ed utilizzati. Le informazioni ricavabili possono essere dunque sotto forma di dati strutturati (ad esempio

database), non strutturati (come immagini, dati GPS, informazioni provenienti da social network ecc.) o semi-strutturati. Prima dell'introduzione dei *big data*, venivano presi in considerazione solo forme strutturate, mentre oggi è possibile analizzarle tutte;

- Velocità: si riferisce alla rapidità con cui vengono generati nuovi dati, ma anche alla necessità di una gestione in *real-time* degli stessi;
- Veridicità: considerate la varietà e velocità con le quali i dati sono generati, diventa rilevante garantire la stessa qualità di input data dai sistemi utilizzati in precedenza. Se i dati che formano le analisi sono poco accurati è evidente che i risultati non saranno migliori, e pertanto l'implementazione di velocità e varietà non fornirà un miglioramento anche in termini assoluti;
- Valore: indica la capacità di trasformare i dati in informazioni con un valore aggiunto. Dal momento che investire nell'analisi dei *big data* richiede grandi risorse, finanziarie e non, è importante verificare che questo studio apporti dei benefici reali.

Per quanto riguarda la caratteristica peculiare di questi dati, ovvero il volume, la mole in questione è nell'ordine dei zettabyte<sup>36</sup>: per dare un termine di paragone, si pensi che la misura dei contenuti digitali nel mondo nel 2010<sup>37</sup> non superava il singolo zettabyte, mentre nel 2013 ammontava già a 4.4 ZB e, secondo recenti studi<sup>38</sup>, è previsto un volume di 44 ZB nel 2020, con un tasso medio di crescita per la prossima decade del 40% l'anno<sup>39</sup>. Tuttavia, queste stime potrebbero rivelarsi addirittura riduttive, perché i sistemi informatici sono sempre più rapidi (si pensi alla recente introduzione di computer quantistici) e la quantità di dati cresce ad una velocità ancor maggiore, grazie all'affermazione dell'*Internet*

---

<sup>36</sup> Uno zettabyte (ZB) corrisponde ad un miliardo di terabyte, ovvero mille miliardi di gigabyte, nonché  $10^{21}$  byte.

<sup>37</sup> Secondo quanto riferito dall'allora Product Manager di Google, Jonathan Effrat, nel corso della presentazione del sistema Google Instant.

<sup>38</sup> Tratto dallo studio IDC "Executive Summary: Data Growth, Business Opportunities, and the IT Imperatives | The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things", su [www.emc.com](http://www.emc.com).

<sup>39</sup> Si veda parallelamente l'enunciato della prima legge di Moore: "La complessità di un microcircuito [...] raddoppia ogni 18 mesi (e quadruplica quindi ogni 3 anni)". È interessante sottolineare come la crescita della mole di dati raccolti deve essere necessariamente accompagnata dall'evoluzione dei sistemi informatici incaricati della loro analisi.

of Things<sup>40</sup> e dell'introduzione delle connessioni 5G. Che piaccia o meno, ogni accesso ad Internet, per qualsivoglia scopo (dalle transazioni finanziarie agli acquisti online, dalla visione di un film all'utilizzo dei social, piuttosto che per attività illecite nel *dark web*<sup>41</sup>), lascia delle tracce sotto forma di dati.

Il termine *data mining*, in italiano "estrazione di dati", indica l'insieme di metodologie che hanno lo scopo di trarre informazioni utili da grandi quantità di dati, per l'appunto i *big data*. Il concetto è simile a quello di statistica, intesa come "estrapolazione di informazioni da un set di dati". Tuttavia, vi è una sostanziale differenza: mentre quest'ultima elabora informazioni generali su una popolazione, il *data mining* ricerca le possibili correlazioni presenti tra diverse variabili riguardanti singoli individui. Tipicamente, questa estrazione di dati è preceduta da processi di trasformazione e filtraggio dei dati, in modo tale che la loro analisi massimizzi i risultati cercati.

Divenuto popolare alla fine degli anni Novanta, il *data mining* oggi ha un duplice significato:

- quello di estrazione di informazione implicita da dati già strutturati;
- e quello di esplorazione, solitamente automatica, di grandi dati al fine di determinare regolarità significative applicabili per predire il futuro.

In ogni caso, informazione e significato dipendono strettamente dall'applicazione che si vuole fare del *data mining*: dati che potrebbero essere fondamentali nel campo della ricerca scientifica, potrebbero ad esempio essere ininfluenti in ricerche di mercato.

Lo sviluppo del *data mining* nell'ultimo ventennio è fortemente correlato a tre fattori cardini come il grande accumulo di dati elettronici, la diminuzione dei costi per gli strumenti di *storage* e infine l'implementazione delle tecniche di *machine learning*. Proprio quest'ultimo spesso e volentieri viene associato al *data mining*, talvolta erroneamente fino a sovrapporsene. Infatti, sebbene di frequente i confini tra i due settori siano molto labili, è rilevante sottolineare che essi non sono sinonimi, ma anzi presentano una differenza di

---

<sup>40</sup> Coniato dal direttore esecutivo del consorzio di ricerca Auto-ID Center del MIT di Boston, Kevin Ashton, l'IoT è un neologismo inglese che si riferisce all'estensione di Internet al mondo degli oggetti. In particolare, descrive l'utilizzo odierno di una moltitudine di apparecchiature tecnologiche connesse ad internet (elettrodomestici, smartphone, PC, televisori, macchinari industriali ecc.) e che pertanto determinano una continua produzione di dati in rete.

<sup>41</sup> Con dark web ci si riferisce ad una piccola parte del deep web, raggiungibile tramite specifici software, e che viene utilizzata specialmente a scopi illegali.

fondo: mentre il *machine learning* si focalizza sulla previsione grazie a modelli noti ma appresi dai dati, il *data mining* rappresenta un'analisi esplorativa con lo scopo di disegnare proprietà prima sconosciute. Se gli strumenti sono spesso simili, non lo sono dunque gli scopi: il *data mining* ambisce a migliorare grazie a conoscenze sempre nuove; il *machine learning* punta ad implementare il sistema informatico con un apprendimento sempre più profondo e prestazioni sempre più specifiche.

## 1.4 APPLICAZIONI

Ad oggi le applicazioni dell'intelligenza artificiale sono innumerevoli e i progressi che la tecnologia sta manifestando consentono una proliferazione di utilizzi inimmaginabile prima. Spesso l'uso di tecniche di *machine learning* non è molto evidente agli occhi degli utenti, tuttavia vi sono tanti campi ove queste tecnologie trovano realizzazione. Gli esempi più celebri riguardano sicuramente i colossi mondiali della tecnologia come Google, Facebook e Amazon, che grazie all'analisi dei *big data* sono in grado di sviluppare sistemi sempre più efficienti e redditizi che riescono ad adattarsi di volta in volta all'utente con cui si stanno rapportando. In particolare, al di là delle applicazioni già menzionate, Google negli anni ha creato diversi team e progetti al fine di implementare l'utilizzo dell'intelligenza artificiale: tra questi va senza dubbio menzionato "*Google Brain Project*", un progetto nato nel 2011 e che ha dato il via ai successivi studi in tema di *deep learning*, come anche "*Rankbrain*" (solo per voler citare uno dei tanti strumenti sviluppati dall'azienda statunitense), algoritmo di ranking che implementa l'accuratezza dei risultati forniti dal motore di ricerca grazie all'apprendimento delle connessioni logiche e delle espressioni linguistiche utilizzate dagli umani nei diversi contesti. Ad oggi la maggior parte dei motori di ricerca al mondo utilizzano queste tecnologie per migliorare l'efficacia dei loro sistemi nell'individuare le SERP<sup>42</sup> delle quali l'utente ha bisogno.

Rimanendo sempre nel contesto del riconoscimento testuale da parte della "macchina", si pensi ai sistemi di *optical character recognition* (OCR), programmi in grado di tramutare in file di testo digitali delle immagini di documenti, una sorta di scanner evoluto. Peraltro, anche i siti e software di traduzione utilizzano sempre più degli algoritmi di apprendimento

---

<sup>42</sup> Acronimo di Search Engine Result Page, rappresentano le pagine di risultati ottenute con un motore di ricerca online.

automatico: tra questi si veda ad esempio DeepL, traduttore online che utilizza un sistema di *deep learning* che gli permette di migliorare continuamente grazie al confronto con milioni di utenti che si rapportano in molte lingue diverse; trattando sempre di sistemi di traduzione ma riferendosi ora al campo del riconoscimento vocale, ha fatto notizia l'introduzione nel 2018 di un software utilizzato da Skype, il noto programma di videochiamata di Microsoft, in grado di tradurre istantaneamente la voce di un utente nella lingua desiderata.

Sicuramente di attuale interesse, sono poi le applicazioni del *machine learning* nell'ambito del cosiddetto marketing mirato: l'utilizzo, da parte di grandi aziende, dei dati personali e di navigazione degli utenti, consente di poter individuare più rapidamente la fascia dei possibili clienti a cui offrire i propri prodotti. All'interno di questo approccio rientrano le pubblicità traccianti, le *sentiment analysis*, nonché le raccomandazioni fornite dai siti in base ai dati raccolti sull'utente: è così che servizi come Google, Amazon, Netflix ecc. offrono a chi ne usufruisce dei consigli mirati che migliorano l'esperienza di utilizzo, nonostante ciò produca spesso dei dibattiti in merito al rispetto della privacy.

Devono far fronte a queste discussioni anche altri sistemi che utilizzano il *machine learning*, in particolare i software di riconoscimento vocale e facciale. Per quanto riguarda il primo, i cosiddetti *digital assistants* sono ad oggi uno strumento di utilizzo quotidiano in ogni smartphone, pc ecc. Tra i più famosi si annoverano Siri di Apple, Cortana di Microsoft, Alexa di Amazon e soprattutto Google Now, considerato dagli esperti il più efficiente in assoluto, sia in tema di dettatura che di interazione con l'utenza. Sono sistemi che non solo agevolano tutto ciò che è possibile compiere attraverso degli apparecchi tecnologici, ma che possono anche interfacciarsi con il cosiddetto *Internet of Things*. Sono infatti in grado di gestire diversi aspetti delle moderne *smart home*, in cui la domotica, e la robotica, fanno oggi da padrone. Come non citare poi, in tema di assistenza, i *virtual assistant* presenti ormai in tantissimi siti web di aziende di servizi, in cui, all'interno di *chatbot*, l'utente si relaziona con un assistente in grado di fornire risposte e soluzioni come se fosse un umano.

Passando al riconoscimento facciale, esistono molteplici applicazioni a tutt'oggi. La prima, se non altro per notorietà, si deve ad un team del social network statunitense Facebook, che ha sviluppato il sistema Deep Face, in grado di riconoscere il viso di un soggetto all'interno di enormi set di immagini. Al di là di questo utilizzo a scopo commerciale, che ha senza dubbio il merito di aver dato il via a questi studi, il riconoscimento facciale viene

anche sfruttato nell'ambito della sicurezza: esistono infatti diversi programmi in tutto il mondo che vengono utilizzati dalle forze di polizia e di sicurezza, in particolare in aeroporti, stazioni e grandi siti turistici, che riconoscono i volti dei passanti e li associano a quelli già schedati presenti nel database. L'applicazione del *machine learning*, pertanto, si dimostra utile anche nella lotta al terrorismo ed alla criminalità.

In tema di sicurezza, vanno poi menzionati i diversi sistemi di protezione online: dai filtri spam presenti principalmente nei software di gestione delle mail, ai programmi di prevenzione delle frodi e dei furti di dati, sono innumerevoli le applicazioni "silenziose" dell'apprendimento automatico che garantiscono un utilizzo sicuro del web. Su questa via, si trovano anche i maggiori utilizzi del *machine learning* in ambito finanziario: sempre più banche e compagnie di assicurazioni utilizzano sistemi come quelli visti per l'analisi del credit score dei clienti, per la prevenzione delle frodi finanziarie, per la gestione del rischio ed infine anche per la valutazione di opportunità di investimento. A ragion del vero, negli ultimi anni stanno prendendo piede anche applicazioni come quella della presente trattazione, tuttavia l'utilizzo di trading system automatici governati da algoritmi di reinforcement learning appare ancora incostante ed in fase di sviluppo.

Altri settori dove si stanno accumulando studi e applicazioni del *machine learning* sono quelli dell'automazione industriale, della robotica e dei trasporti. In particolare, sono oggetto di sviluppo da diversi anni i veicoli a guida autonoma, i quali, grazie a sensori radar, laser e videocamere, riescono a muoversi rispettando le comuni regole della strada. Gli algoritmi di *reinforcement learning* e *deep learning* riescono in questo caso a migliorare sensibilmente il rapporto dei veicoli senza piloti con la componente umana circostante, che spesso si dimostra imprevedibile: se in una fase embrionale, il *machine learning* viene utilizzato per allenare l'algoritmo a riconoscere le componenti della strada e ad individuare le azioni corrette da compiere, in un secondo momento il focus degli ingegneri si concentra proprio sul rendere i veicoli affidabili in ogni circostanza. È questa la sfida attuale nel campo delle auto a guida autonoma, riuscire a gestire le situazioni imprevedibili che la mente umana affronta d'istinto in una frazione di secondo. Rimanendo sempre nel campo dell'automobile, vi sono diverse case automobilistiche che utilizzano oggi sistemi di intelligenza artificiale per l'assistenza alla guida, sia sotto forma di assistenti vocali, sia sotto forma di computer di bordo che regolano le impostazioni di guida in base al comportamento del guidatore.

Nel settore della robotica, l'utilizzo del *machine learning* sta permettendo di raggiungere grandi risultati nei più diversi campi di applicazione: esistono oggi stabilimenti industriali che, ad esempio, sono formati interamente da catene di produzione autonome comandate da algoritmi di apprendimento automatico; nel campo della medicina, sono sempre più gli strumenti avanzati che, tramite il *deep learning*, forniscono supporto negli interventi chirurgici e nell'analisi dei parametri vitali dei pazienti. In particolare, si stanno creando algoritmi che si interfacciano con database di pazienti sotto costante monitoraggio e che sono in grado di segnalare al medico curante gli individui che potrebbero essere soggetti a particolari malattie o disturbi.

Infine, anche nel campo videoludico, il *machine learning* è alla base di grandi miglioramenti: al di là dei casi già citati di macchine intelligenti in grado di giocare a dama, scacchi o al gioco del go, sono tantissimi i videogiochi che utilizzano algoritmi di apprendimento automatico per migliorare l'esperienza di gioco, e di conseguenza l'adattabilità della difficoltà e/o realistica in base alle abilità del giocatore.





## CAPITOLO II. TRADING SYSTEM, ANALISI TECNICA E MACHINE LEARNING

Il presente capitolo verte sul concetto di trading system, con particolare riferimento ai cosiddetti *automatic financial trading system* (FTS). Nello specifico, dopo un'introduzione sul tema ed una rapida trattazione sulla dottrina dell'analisi tecnica, ove vengono citati solo alcuni indicatori, utili al fine della presente tesi, il focus si sposta sull'analisi degli articoli scientifici in materia. Quest'ultimi rappresentano i maggiori risultati disponibili in letteratura in materia di ricerca su FTS che utilizzano *reinforcement learning*, e costituiscono, pertanto, la base sulla quale è stata ideata l'applicazione esposta al capitolo seguente.

### 2.1 INTRODUZIONE AI TRADING SYSTEM

Un trading system è letteralmente un sistema di investimento, ovvero un insieme di regole stabilite dall'investitore al fine di determinare le operazioni di mercato da compiere. Più precisamente, costituiscono uno schema di segnali che di volta in volta indicano se entrare, uscire o restare nel mercato, cioè se inviare rispettivamente un ordine di acquisto, di vendita oppure nessun ordine. Il fine della costituzione di questi insieme di regole è quello di dare una logica razionale al processo di investimento, in modo tale da poter controllare l'emozionalità umana: senza volersi addentrare nel campo della finanza comportamentale e/o della psicologia, quando si decide di investire non è semplice controllare i diversi istinti che si scatenano, come ad esempio la paura, l'avarizia, l'euforia o la disperazione. È per questo motivo che, non solo si stabiliscono degli indicatori che forniscano segnali di acquisto o vendita, ma si introducono spesso anche delle soglie oltre le quali si decide di chiudere la posizione in essere: è il caso dei cosiddetti *stop-loss* e *take-profit*. I primi sono limiti inferiori al raggiungimento dei quali la posizione viene chiusa per evitare una perdita eccessiva; i secondi, viceversa, rappresentano dei limiti superiori, toccati i quali la posizione viene chiusa per evitare che il profitto accumulato possa successivamente ridursi. Queste due soglie possono essere statiche o dinamiche: statiche se sono rappresentate da un valore fisso stabilito a priori; dinamiche se vengono determinate in base a percentuali della posizione aperta, in modo tale da consentire di

seguire la maggior parte del trend, pur tuttavia tutelando parte dei guadagni ottenuti. Ovviamente la prima soluzione è più sicura, ma può dimostrarsi troppo restrittiva; viceversa, la seconda consente di ottenere meno chiusure della posizione, ma ha una minor tutela dell'investimento.

Le regole che formano un trading system possono avere origine in mille modi diversi e potenzialmente l'investitore potrebbe decidere come fonte di segnali d'investimento qualsiasi cosa, dagli indicatori di analisi tecnica, alle previsioni metereologiche di giornata. Senza dubbio, se si vogliono ottenere dei risultati, il trading system deve avere delle fondamenta logiche che esprimano delle relazioni tra l'andamento del titolo oggetto di investimento e gli indicatori selezionati. Una volta formato il proprio sistema di regole è bene testarlo sui dati passati, possibilmente andando a vedere i risultati su un ipotetico futuro virtuale<sup>43</sup>. Stabilito il trading system, è dunque possibile iniziare ad utilizzarlo per investire nei mercati. Questo processo può avvenire manualmente, per mezzo dell'investitore che valuta i segnali ricevuti e decide se tramutarli in ordini di mercato, oppure automaticamente. Con l'espressione *automatic trading system*, ci si riferisce infatti a sistemi di investimento gestiti direttamente da un computer, o software, che tramite gli algoritmi stabiliti dal trader, invia in maniera automatica gli ordini di mercato.

Per anni lo studio di sistemi d'investimento automatici si è concentrato sui più disparati indicatori economici, da quelli di natura aziendale a quelli di analisi tecnica, andando talvolta ad ottenere risultati molto redditizi. Bisogna però considerare che il mercato, proprio perché abitato da fattori come l'avarizia e l'istinto, non rispecchia sempre una logica razionale, specialmente nel breve periodo. Pertanto, nonostante i trading system classici possano restituire ottimi risultati, l'utilizzo di nuove metodologie come il *reinforcement learning* apre delle nuove strade, ove la capacità di analisi e di calcolo di una macchina può rilevarsi più adatta, rispetto alla mente umana, all'interazione con un ente apparentemente irrazionale come il mercato finanziario.

Grazie al progresso delle capacità computazionali ed al potenziale espresso dalle metodologie di apprendimento autonomo, la costruzione di FTS automatici è materia di ricerca di grande interesse, sia in ambito accademico che finanziario.

---

<sup>43</sup> Con futuro virtuale si intende un set di dati effettivamente realizzatisi, ma che si suppone sconosciuto al fine di valutare la bontà di una stima previsiva basata sui dati precedenti.

## 2.2 CENNI DI ANALISI TECNICA

Per analisi tecnica si intende in economia lo studio dell'andamento delle serie storiche dei prezzi dei titoli finanziari, al fine di poter prevedere le tendenze future tramite analisi grafiche, quantitative e qualitative. Si contrappone storicamente all'analisi fondamentale, altro strumento per la definizione delle strategie operative di investimento finanziario, che invece cerca di quantificare il vero valore economico di un'impresa, società o istituzione, in modo tale da poter stabilire se il titolo è sovrastimato o sottostimato.

Dal momento che è impossibile prevedere l'esatto momento in cui il prezzo toccherà un minimo o massimo relativo, l'analisi tecnica non ha la pretesa di essere la sfera di cristallo che permette di entrare ed uscire nel mercato nei momenti perfetti. Essa invece ha come obiettivo quello di individuare i trend in atto e di anticipare le possibili inversioni future. Questo processo di studio dell'andamento dei prezzi si basa, innanzitutto, su alcuni assunti fondamentali:

- il prezzo di mercato sconta tutti i fattori che incidono su domanda e offerta: con questa affermazione perciò si sostiene che i prezzi non rappresentano dei semplici grafici, ma sintetizzano la psicologia rialzista o ribassista del mercato;
- la storia si ripete: l'andamento oscillatorio dei prezzi, all'apparenza completamente aleatorio, compie dei cicli che si ripetono. In particolare, si può notare spesso come si presentino ripetuti livelli di massimi o minimi relativi, ovvero casi in cui i prezzi invertono la tendenza sempre allo stesso livello: questo dimostrerebbe come esistono punti critici che mantengono importanza lungo la storia di un titolo;
- infine, quando il mercato evidenzia una tendenza definita dei prezzi, questa è ritenuta valida finché non appaiono segnali di inversione.

Fatte queste premesse, lo studio del movimento dei mercati finanziari prende in esame principalmente il prezzo, i volumi di scambio e l'*open interest* di un titolo<sup>44</sup>. Nel fare questo, l'analisi tecnica si distingue principalmente in analisi grafica, qualitativa, quantitativa e strutturale. I metodi qualitativi prevedono lo studio del comportamento dei volumi e dunque della forza che ha un trend, mentre i metodi strutturali cercano di individuare

---

<sup>44</sup> Nonostante l'analisi tecnica nasca originariamente nel mercato azionario, qui ci si riferisce al mondo dei titoli finanziari in generale, poiché ad oggi rappresenta una metodologia d'investimento anche nei mercati obbligazionari, valutari e delle commodities.

l'eventuale presenza di ciclicità degli andamenti dei prezzi. Sono entrambe applicazioni che trovano realizzazione in una visione di più ampio spettro in cui si traggono considerazioni generali sulle opportunità che offre il titolo in quel preciso momento ed in quel determinato mercato.

Per quanto riguarda invece i metodi grafici e quantitativi, essi rappresentano i sentieri tradizionali dell'analisi tecnica. I primi si fondano sullo studio dei grafici, ed hanno il fine di individuare due differenti situazioni di mercato: una chiamata *trending market*, in cui vi è una tendenza definita; l'altra denominata *trading market*, in cui non vi è un trend definito, ma una cosiddetta fase laterale<sup>45</sup>. In quest'ultimo caso, l'applicazione dell'analisi grafica appare difficoltosa perché non è in grado di distinguere dei trend su cui basare le decisioni di investimento. Ad ogni modo, i principali tipi di grafici sono il grafico a barre (*bar chart*), il grafico dei prezzi di chiusura (*close only chart*, semplice grafico linea dei prezzi nel tempo) ed il grafico a candele (*candlesticks chart*). In tutte le diverse rappresentazioni, l'obiettivo dell'analisi grafica resta quello di individuare i trend, le resistenze, i supporti ed eventuali modelli di inversione di consolidamento. Quest'ultimi sono particolari configurazioni o figure grafiche che vengono a formarsi, e che starebbero ad indicare dei momenti particolari nell'andamento del prezzo: esempio classico di modello di inversione è il "testa e spalle", mentre tra i segnali di consolidamento del trend è frequente la configurazione "a triangolo". Resistenze e supporti, invece, sono livelli di prezzo in corrispondenza dei quali le tendenze, rispettivamente rialzista e ribassista, si interrompono. Spesso, questi livelli sono significativi perché possono mantenersi anche per un lungo periodo di tempo, talvolta trasformandosi da resistenza a supporto (e viceversa), e, in ogni caso, quando vengono "rotti" (ovvero oltrepassati dal prezzo) forniscono un forte segnale a favore del trend in atto.

Ovviamente, questo approccio di analisi tecnica non può essere sfruttato in un'applicazione automatica perché richiede l'esperienza e l'abilità umana. I metodi quantitativi invece si basano sullo studio di indicatori e/o misure precise, e dunque si prestano all'utilizzo in trading system automatici. Spesso questi indicatori presentano una rappresentazione grafica, tuttavia è bene distinguerli dalle forme di analisi grafica poc'anzi viste, come *trendline*, resistenze, supporti ecc. perché essi sono frutto di formule

---

<sup>45</sup> Fase laterale indicherebbe proprio il grafico dei prezzi che non descrive andamenti positivi o negativi, ma un semplice spostamento laterale nel tempo.

matematiche e possono essere riprodotti e gestiti da un calcolatore elettronico. In generale, è possibile distinguere l'analisi quantitativa in modelli per l'individuazione della tendenza in corso (*Trend calculation* e *Trend system*) ed in modelli per la valutazione della forza del trend (*Momentum* e oscillatori). A titolo d'esempio, nella prima categoria rientrano le diverse forme di media mobile (semplice, geometrica, esponenziale, pesata ecc.) e di bande di oscillazione, come le celebri *Bollinger Bands*. Tra le misure di momentum più utilizzate, invece, v'è il MACD (*Moving Average Convergence Divergence*), mentre tra gli oscillatori i più famosi sono l'RSI (*Relative Strength Index*), l'ADX (*Average Directional movement index*) e lo stocastico.

Volendo essere generalisti, ciò che accomuna tutti gli strumenti di analisi tecnica, è la rappresentazione grafica, ovvero la possibilità di osservare i segnali di trading direttamente sul grafico grazie agli incroci tra le diverse linee, che, rispettivamente, formano l'indicatore e rappresentano il prezzo.

Nel corso delle seguenti pagine vengono esposti i diversi indicatori di analisi tecnica utilizzabili in trading system automatici, cercando, ove possibile, di darne anche un'interpretazione e rappresentazione grafica. Quest'ultima è stata ottenuta attraverso l'applicazione degli indicatori nel software R (quando erano presenti i comandi corrispondenti), per poi tracciarne il grafico. Nei casi ciò non sia stato attuabile, si è provveduto all'utilizzo di software o esempi online. In generale, è stata presa ad esempio, al solo fine espositivo, la serie storica dei prezzi (contenente prezzi di apertura, chiusura, massimi, minimi, volumi e volatilità) del FTSE Mib.

### 2.2.1 *Medie mobili*

La famiglia degli indicatori tecnici più utilizzata e conosciuta è sicuramente quello delle medie mobili, una serie di utili strumenti matematici che consentono in particolare di smussare le fluttuazioni erratiche dei prezzi. Una media mobile è una media di una determinata quantità di dati che considera soltanto le ultime rilevazioni in ordine di tempo. Tale strumento è infatti definito "mobile" perché il numero degli elementi considerati (i prezzi) è fisso, ma l'intervallo di tempo avanza. I vecchi dati devono essere rimossi quando ne vengono aggiunti di nuovi: questo fa sì che la media si muova in progressione con

l'andamento del prezzo del titolo oggetto del calcolo. Una media mobile a 200 giorni, ad esempio, prende in esame i valori delle ultime 200 sedute e viene ricalcolata ogni giorno, aggiungendo l'ultimo dato ed escludendo dal calcolo il più remoto, in modo che siano sempre e solo 200 i prezzi considerati.

Solitamente vengono considerati i prezzi di chiusura di giornata (*adjusted*<sup>46</sup> o semplici), ma possono essere utilizzati anche i massimi, i minimi, o i valori di apertura delle contrattazioni. Ovviamente questo vale nel caso in cui si stia studiando un grafico con dati giornalieri, ma si potrebbero impiegare "chiusure orarie", settimanali, mensili od annuali se l'analisi dovesse avere orizzonti temporali di brevissimo termine (grafici orari) o più lunghi (grafici settimanali, mensili od annuali).

Esistono vari tipi di medie mobili: semplice, ponderata ed esponenziale. Ognuna di queste ha un differente metodo di calcolo.

La media mobile semplice, detta anche aritmetica, è la più utilizzata in virtù della sua più facile costruzione. Vengono presi i dati di un determinato periodo e ne viene calcolata la media sommandoli fra loro e dividendo per il numero totale di valori:

$$MA_t = \frac{\sum_{i=1}^n P_{t-i+1}}{n}$$

La maggiore semplicità di calcolo viene pagata con un grosso difetto: tale algoritmo assegna la stessa rilevanza ad ogni singolo input; l'ultimo valore (magari anche molto lontano nel tempo) ha la stessa importanza del primo.

La media mobile ponderata cerca di risolvere questo problema, attribuendo un peso maggiore ai dati più recenti e minore a quelli più lontani nel tempo:

$$WMA_t = \frac{\sum_{i=1}^n w_i P_{t-i+1}}{\sum_{i=1}^n w_i}$$

Prendendo ad esempio una media a 10 periodi, la chiusura del decimo ed ultimo giorno viene moltiplicata per 10, quella del giorno precedente (nono) per nove, quella dell'ottavo giorno per otto e così via. Il totale verrà poi diviso per la somma dei multipli: nel nostro

---

<sup>46</sup> I prezzi di chiusura *adjusted* sono così chiamati perché aggiustati in modo tale da considerare eventuali dividendi distribuiti.

caso  $1+2+3+\dots+10=55$ . La media è quindi molto più reattiva agli eventi recenti, mentre “smorza” le oscillazioni passate.

Una media mobile semplice a 10 giorni pesa il prezzo più recente esattamente come quello più lontano, per il 10% sul totale; la media ponderata, invece, assegna all’ultimo dato una ponderazione del 18%, mentre a quello più lontano dell’1.8%.

Ancora più complesso è il sistema di calcolo della media mobile esponenziale, un indicatore che, per essere costruito, richiede appropriati software di calcolo. Vengono considerati tutti i dati disponibili - non solo quelli relativi al periodo preso in esame - anche se i prezzi del dominio, in particolare i più recenti, hanno un peso maggiore. Tale media è costruita prendendo in considerazione tutti gli elementi della serie ma con una ponderazione esponenzialmente decrescente.

$$EMA_t = \frac{P_t + \alpha_1 P_{t-1} + \alpha_2 P_{t-2} + \dots + \alpha_{n-1} P_{t-n+1}}{1 + \alpha + \alpha^2 + \dots + \alpha^{n-1}}$$

La ponderazione dei primi dati della serie sarà minore col passare del tempo fino a diventare infinitesimale, ma rientrerà sempre nel calcolo. In sostanza, tutta la storia del titolo è tenuta in considerazione.

L’esempio seguente, relativo al FTSE Mib per l’anno 2017, mostra le medie a 25 giorni calcolate in modo aritmetico (linea rossa), esponenziale (gialla) e ponderato (blu). Quest’ultima risulta la più volatile e reattiva rispetto, invece, alla media semplice che “smorza” maggiormente le oscillazioni.

Grafico 4. FTSE Mib e medie mobili, a 25 giorni, semplice, ponderata ed esponenziale (2017)



Dal momento che la loro funzione è proprio quella di seguire il movimento dei prezzi in atto, le medie mobili sono l'indicatore per eccellenza nei cosiddetti "trading system" di tipo "trend following" (cioè sistemi di trading che seguono il trend); altro fattore molto vantaggioso relativo a questi indicatori sono la facilità di costruzione e l'eliminazione parziale del fattore di soggettività.

Risulta comunque evidente che i loro segnali non possono essere generati se non con un discreto ritardo.

Il metodo più semplice di utilizzo consiste nel considerarle alla pari di *trendline*, considerandole come vere e proprie resistenze/supporti dinamici e di prendere decisioni alla rottura delle stesse da parte dei prezzi.

Questo tipo di funzione tende a smorzare l'erraticità dei prezzi e l'effetto è tanto più evidente quanto maggiore è il numero di eventi inclusi nel calcolo della media mobile. Una media mobile, infatti, graficamente è una curva che segue la linea dei prezzi, posizionandosene al di sotto, in caso di trend al rialzo, o al di sopra in caso di trend al ribasso. Il momento in cui la media "taglia" i prezzi solitamente indica il momento di cambiamento del trend per l'orizzonte temporale definito dal dominio. Più precisamente si avrà un segnale di acquisto quando i prezzi saliranno al di sopra della media mobile, mentre quando scenderanno al di sotto sarà il momento di vendere. Va tenuto presente che il periodo utilizzato influisce notevolmente sulle indicazioni generate: se si sceglierà un dominio minore si otterrà una media mobile molto più vicina ai prezzi che avrà come conseguenza una maggiore tempestività, ma un maggior numero di falsi segnali; al contrario un periodo maggiore darà origine a una linea più smussata che fornirà suggerimenti più affidabili, ma più in ritardo rispetto all'altra.

Il problema viene parzialmente superato utilizzando una combinazione di due medie, una lenta ed una veloce. Tale sistema è il più utilizzato dagli analisti. I segnali operativi vengono forniti all'incrocio tra le due medie: quando quella più veloce, ovvero generata da un periodo minore, incrocerà al rialzo quella più lenta verrà generato un suggerimento di acquisto; viceversa quando quella lenta sarà tagliata al ribasso da quella più veloce si avrà la vendita.

Per ridurre ulteriormente i falsi segnali si considera: segnale di acquisto quando il prezzo sale sopra entrambe le medie mobili; segnale di chiusura delle posizioni lunghe quando il prezzo scende sotto la media mobile più veloce; segnale di vendita, o apertura di posizioni



*short*, quando il prezzo scende al di sotto di entrambe le medie mobili; infine segnale di chiusura delle posizioni corte quando il prezzo sale sopra la media mobile più veloce.

Grafico 5. FTSE Mib e medie mobili semplici a 15 (linea viola) e 25 (linea rossa) passi (2017)



### 2.2.2 Bande di oscillazione, bollinger bands e indicatori collegati

Una banda di oscillazione è un'area attorno ad una linea di tendenza che viene utilizzata dal trader come anticipazione del cambiamento nell'andamento dei prezzi. Solitamente, la banda è ottenuta aggiungendo e sottraendo una stessa quantità in modo tale da risultare simmetrica rispetto alla linea di tendenza, mediana dell'area in questione. Quest'ultima è comunemente rappresentata da una media dei prezzi. Esempio base di una banda di oscillazione è dunque:

$$B = \mu_t \pm \alpha \mu_t ,$$

con  $\mu_t$  media dei prezzi al tempo t e  $0 \leq \alpha \leq 1$ .

Per calibrare la banda si utilizza un valore multiplo chiamato *scaling factor*, S:

$$B = \mu_t \pm S(\alpha \mu_t) .$$

Tale costante può moltiplicare diversi fattori a seconda della costruzione che si vuole dare alla banda di oscillazione. Per esempio, si possono utilizzare:

- la quotazione P al tempo t:  $B = \mu_t \pm S(\alpha P_t) ;$

- un'oscillazione di prezzo:  $B = \mu_t \pm S(\Delta_p)$  ;
- il differenziale tra minimo e massimo di giornata:  $B = \mu_t \pm S(H_t - L_t)$  ;
- la volatilità del prezzo o della media dei prezzi:  $B = \mu_t \pm S(\sigma_t)$  .

Le bande di oscillazione più conosciute ed utilizzate sono le *Bollinger Bands*. Prendendo spunto dalla teoria del ricercatore John Hurst, John Bollinger creò queste bande negli anni '90 e ne registrò il marchio nel 2011.

Le bande di Bollinger sono formate da tre linee:

- la linea mediana (MB), rappresentata da una media mobile semplice di n prezzi consecutivi, generalmente con n=20; viene scelta la media mobile semplice in quanto altri tipi di media mobile possono portare a incongruenze tra il calcolo di quest'ultima e quello della deviazione standard, fattore utilizzato per l'incremento e/o decremento necessari a formare la banda superiore e/o inferiore;
- la banda superiore (UB), somma della mediana e k deviazioni standard, con k=2 nella formulazione originale; essa si trova, ovviamente, sopra la media mobile ed è pari a:

$$UB_t = MB_t + k\sigma_{p,t}$$

- la banda inferiore (LB), differenza tra la mediana e k deviazioni standard, anche qui, originariamente, con k pari a 2; si trova sotto la media mobile e corrisponde dunque a:

$$LB_t = MB_t - k\sigma_{p,t} .$$

L'andamento del prezzo del titolo è solitamente compreso tra la banda superiore e la banda inferiore e tende a tornare verso la mediana , nel caso classico la media mobile semplice a 20 passi. In base a come si relazionano le diverse linee si possono trarre dei segnali di trading: in particolare, quando il prezzo arriva in prossimità della banda superiore e la buca, il mercato può essere considerato in fase di ipercomprato, a significare un'imminente inversione ribassista del trend. In tal caso sarà opportuno aprire una posizione di vendita qualora il prezzo rientri prontamente all'interno della banda superiore; viceversa, quando il prezzo arriva in prossimità della banda inferiore e la buca, il mercato può essere considerato in fase di ipervenduto, a significare una imminente inversione rialzista del trend. In tal caso sarà opportuno aprire una posizione di acquisto qualora il

prezzo rientri prontamente all'interno della banda inferiore.

Grafico 6. FTSE Mib e Bollinger Bands (2017)



Un problema delle bande di Bollinger è che sono poco resilienti nei periodi di volatilità e che non sempre alle rotture seguono poi continuazioni del movimento. Ecco perché questo indicatore, per essere compreso e interpretato appieno, deve essere confrontato con altri indicatori. Due strumenti direttamente connessi alle bande di Bollinger sono il BBW e il %B, utili ad aumentarne la comprensione e l'efficacia.

L'ampiezza della banda (*Bollinger Bands Width*, BBW), ossia la distanza tra la linea superiore e quella inferiore, è una misura di volatilità relativa utile a determinare eventuali inversioni di trend: essa è infatti elevata quando c'è molta volatilità, ad indicare che il trend corrente potrebbe terminare; è ridotta quando c'è poca volatilità, anticipando la possibilità che si verifichi l'inizio di un nuovo trend. La formulazione dell'indicatore BBW è intuitiva, in quanto differenziale tra l'*upper band* e la *lower band*; questa differenza viene dunque rapportata alla media mobile semplice a 20 giorni, in modo da agevolarne i confronti su base relativa:

$$BBW_t = \frac{2k\sigma_{P,t}}{MB_t}.$$

I momenti di restringimento delle bande si possono sfruttare per individuare il momento migliore per piazzare un ordine di *breakout*. In questo caso, che viene detto *squeeze*, si possono avere due differenti situazioni:

- le candele rompono la banda superiore e ciò indica una fase di trend rialzista, tanto che dopo la rottura della banda solitamente il prezzo continua la salita;
- le candele rompono la banda inferiore e ciò indica l'arrivo di un trend ribassista, pertanto dopo la rottura i prezzi continuano a scendere.

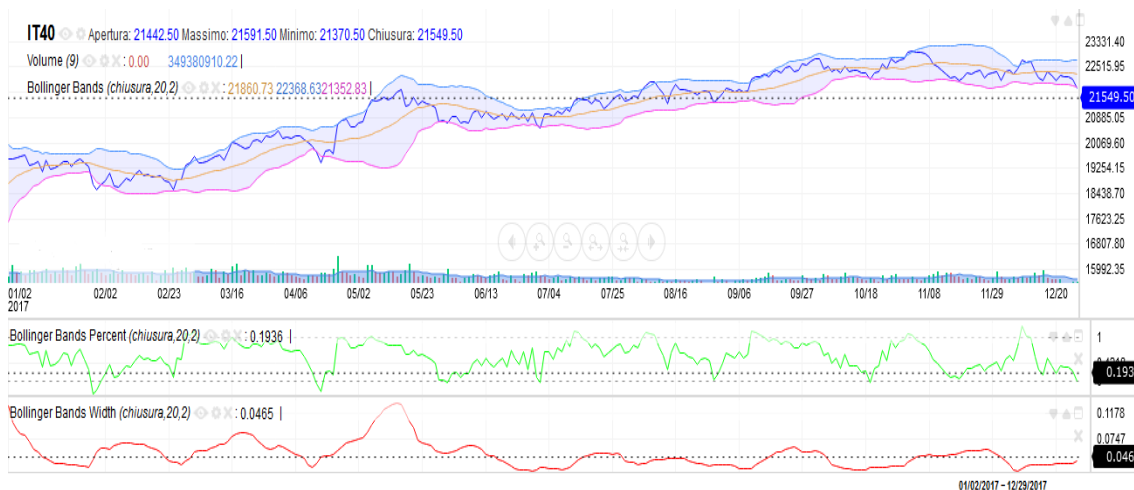
Il %B (*Percentage Bollinger*) è invece un indicatore che permette di risolvere l'incapacità delle BB di offrire visivamente la posizione del prezzo in relazione alle bande stesse: spesso i prezzi corrono per molto tempo lungo la banda inferiore o superiore, prima di staccarsene e prendere una direzione netta e contraria. Il trader che vendesse al raggiungimento dell'*upper band* o comprasse al raggiungimento della *lower band*, probabilmente uscirebbe prematuramente dal mercato. Il %B subentra in aiuto dando un'idea, ed un valore, di come è posizionato il prezzo nell'ambito delle bande. Esso corrisponde infatti al rapporto avente al numeratore la differenza fra il prezzo e la banda inferiore, ed al denominatore la differenza fra le due bande:

$$\%B = \frac{P_t - LB_t}{UB_t - LB_t} .$$

L'indicatore, così ottenuto, tipicamente oscilla tra 0 e 100, e individua situazioni di ipercomprato, ipervenduto e divergenze, informazioni importanti, pertanto, a completare il quadro fornito dalle semplici *Bollinger bands*.

L'utilizzo congiunto delle bande e di questi due indicatori poc'anzi esposti, consente di mettere a punto delle strategie di trading più precise di quanto possa suggerire l'utilizzo delle sole bande: ad esempio, si potrebbero considerare interessanti i titoli finanziari che presentano un basso BBW e un %B verso lo zero (ipoteticamente inferiore a 10), mentre i titoli da vendere presenterebbero ugualmente un BBW a livelli depressi ma un %B in ipercomprato attorno a cento (ipoteticamente superiore a 90).

Grafico 7. FTSE Mib, Bollinger Bands, %B e BBW (2017)



Fonte: it.investing.com

### 2.2.3 Momentum

Il concetto di momentum può essere compreso attraverso un semplice esempio. Quando un oggetto, sia esso una palla, un proiettile o altro, viene lanciato in aria, esso inizia la sua traiettoria a forte velocità, cioè possiede un elevato momento. Gradualmente, la velocità con la quale la palla viaggia verso l'alto diminuisce, finché non giunge ad un temporaneo arresto prima che la forza di gravità gli faccia invertire la direzione. Questo processo di rallentamento, conosciuto come perdita di momento verso l'alto, è anche un fenomeno dei mercati finanziari: la velocità di avanzamento del prezzo diminuisce notevolmente prima che venga raggiunto l'ultimo prezzo massimo (così come prima di raggiungere l'ultimo prezzo minimo in una fase discendente).

D'altro canto, se l'oggetto viene lanciato verso l'alto contro un soffitto mentre il suo momento è ancora in crescita, l'oggetto e il momento stesso invertono la direzione allo stesso istante: purtroppo i prezzi di mercato si comportano in modo simile, poiché talvolta il momento ed il prezzo raggiungono un massimo simultaneamente, sia per la forza improvvisa e repentina delle vendite che per la netta debolezza degli acquisti. Sebbene in tali evenienze il momento non sia un indicatore particolarmente efficace per determinare l'andamento del mercato, generalmente fornisce informazioni corrette e tempestive in merito a situazioni di cambiamento dei movimenti dei prezzi.

L'analisi tecnica fondata sugli oscillatori trova come base applicativa proprio il concetto

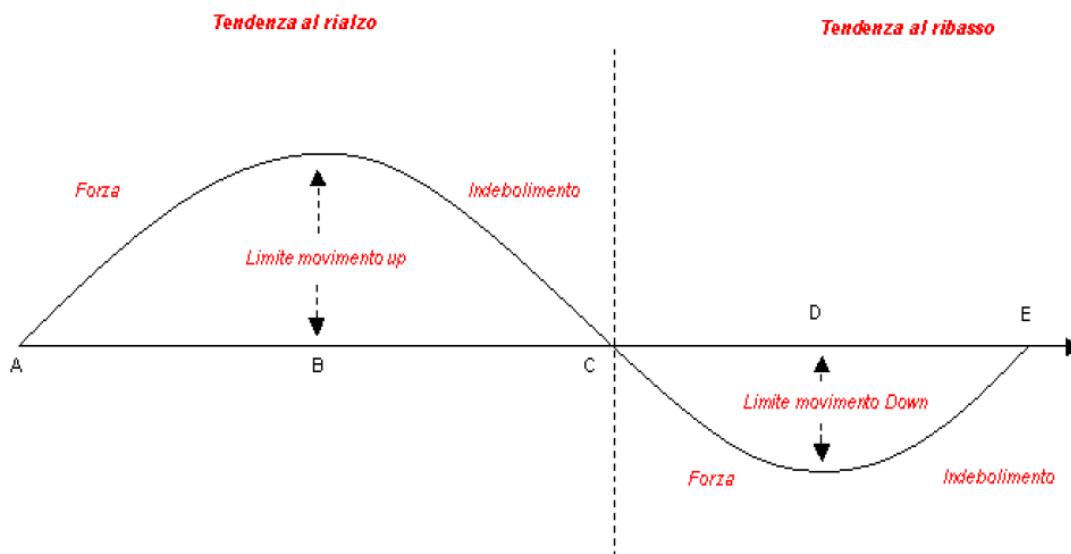
di momentum, considerando che la maggior parte di questi indicatori sono appunto assimilabili, per costruzione e informazione fornita, al principio di momentum. Con quest'ultimo si intende, in sostanza, una misura del grado di accelerazione che un movimento di prezzo, sia esso al ribasso o al rialzo, esprime nell'arco del suo evolversi.

Passando all'aspetto analitico, il momentum può essere interpretato come una misura di velocità media del prezzo e pertanto è così calcolata:

$$M_t = \frac{(P_{t1} - P_{t2})}{t_2 - t_1}$$

Intuitivamente, valori positivi e crescenti dimostrano un'accelerazione della tendenza in atto. Un ritorno verso lo zero, pur in presenza di prezzi ancora al rialzo, indica che il trend sta perdendo forza e velocità. La sua valenza operativa, dunque, è interpretabile come da grafico seguente:

Grafico 8. Interpretazione del momentum



Fonte: Prof. Luca Cappellina, materiali didattici del corso "Laboratorio di Finanza" (2019)

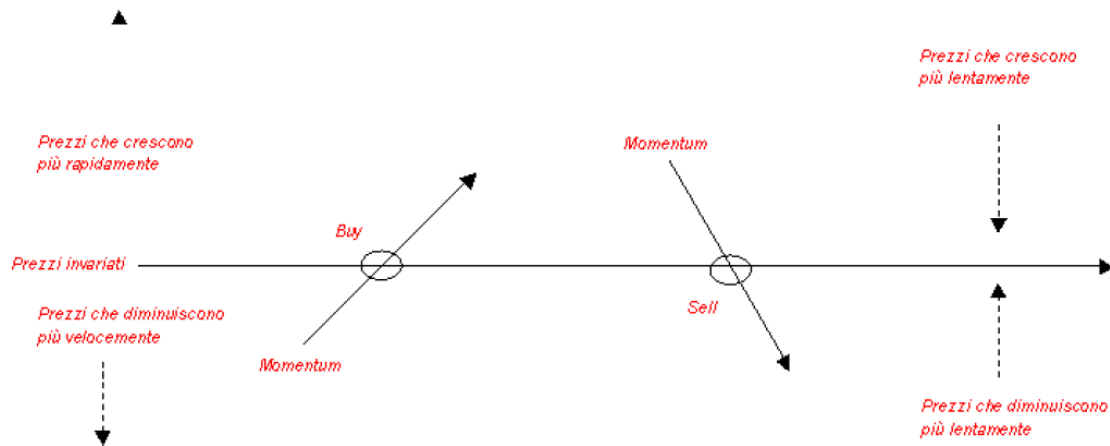
Precisamente:

- valori del momentum maggiori di zero indicano una fase espansiva del prezzo; ciononostante bisogna distinguere tra valori crescenti, che indicherebbero un rafforzamento dell'espansione corrente, e valori decrescenti, che indicherebbero un indebolimento della stessa;
- valori del momentum minori di zero indicano una fase ribassista del prezzo; anche

in questo caso si distingue in valori decrescenti, caratterizzanti una forte intensità del ribasso, e crescenti, denotanti un indebolimento della fase decrescente.

I segnali che possono formare un trend system vengono presentati al Grafico 9.

Grafico 9. *Trading signals* forniti dall'analisi del momentum

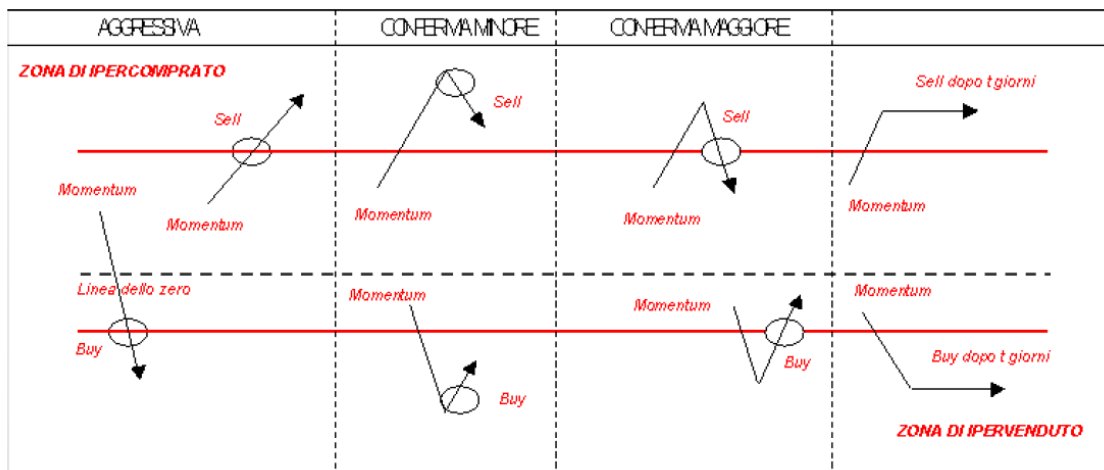


Fonte: Prof. Luca Cappellina, materiali didattici del corso "Laboratorio di Finanza" (2019)

Al fine di ridurre i falsi segnali, gli indicatori di momentum, in generale, hanno bisogno di essere posti in relazione alla ciclicità dei movimenti, problema che si può risolvere esaminando i massimi e minimi significativi, e dunque applicando al calcolo una media del numero dei giorni di un ciclo.

In merito ai diversi indicatori di momentum, ed ai diversi oscillatori, è possibile individuare un livello di ipercomprato (e di ipervenduto), il quale viene a verificarsi quando l'indicatore in esame oltrepassa, al rialzo (ribasso), un determinato livello detto, appunto, di ipercomprato (ipervenduto). L'*overbought zone* e l'*oversold zone* possono essere determinate sia graficamente che analiticamente, in quest'ultimo caso si possono determinare o come percentuale del valore massimo del momentum o in base ad un multiplo della deviazione standard. Un mercato si definisce in ipercomprato quando non può sostenere a lungo la forza della tendenza al rialzo corrente e ci si attende un'imminente correzione al ribasso; viceversa, un mercato è detto in ipervenduto quando la forza della tendenza al ribasso corrente è in calo e ci si aspetta una correzione al rialzo. Pertanto, al livello di ipercomprato è associato un segnale di vendita, mentre al livello di ipervenduto è associato un segnale di acquisto.

Grafico 10. *Trading signals 'scremati' forniti dall'analisi del momentum*



Fonte: Prof. Luca Cappellina, materiali didattici del corso "Laboratorio di Finanza" (2019)

Naturalmente, il concetto di momentum può essere applicato anche ad altri valori che non siano prezzi: si può calcolare il momentum di medie mobili, o di altri indicatori, basta sostituire ai valori dei prezzi le osservazioni della variabile desiderata in due successive unità temporali.

Delle varianti del momentum, ad esempio, possono essere:

- il *rate of change* ( $ROC_t$ ), ovvero una sorta di momentum in termini percentuali:

$$ROC_t = \frac{P_t - P_{t-n}}{P_{t-n}}$$

- una misura del momentum che nasce dalla differenza del prezzo con la media mobile (price and trend difference):

$$M_t = \frac{(P_t - MA_t)}{t_2 - t_1}$$

- il MACD (Moving Average Convergence/Divergence), noto indicatore di analisi tecnica, che può essere descritto come momentum basato sulla differenza tra due medie mobili di lunghezza differente.



## 2.2.4 Moving Average Convergence/Divergence (MACD)

Il celebre indicatore di momentum *Moving Average Convergence / Divergence*, comunemente noto con l'acronimo MACD, si riferisce all'eventuale convergenza o divergenza tra più medie mobili. È un oscillatore apprezzato per la facilità nell'utilizzo e l'applicabilità, infatti è probabilmente il più utilizzato perché attuabile su qualsiasi serie univariata (e non soltanto per il prezzo). Raramente i suoi segnali operativi si rivelano errati o contraddittori, di conseguenza il MACD permette di ottenere segnali tempestivi, in particolar modo riducendo il numero di periodi presi in considerazione. Affinché il MACD sia più efficace è meglio associarlo ad un altro indicatore, come ad esempio all'RSI<sup>47</sup> oppure alle bande di Bollinger. La costruzione dell'indice è basata su diverse medie mobili esponenziali:

- una media mobile esponenziale dei prezzi di 12 periodi (la più veloce)
- una media mobile esponenziale dei prezzi di 26 periodi (la più lenta)
- una media mobile esponenziale dei prezzi di 9 periodi sulla differenza delle prime due medie mobili.

L'indicatore MACD è dunque composto da due curve: il MACD vero e proprio e la relativa *signal line*. Il MACD è il risultato della differenza<sup>48</sup> tra la media mobile esponenziale dei prezzi a 12 giorni e la media mobile esponenziale dei prezzi a 26 giorni<sup>49</sup>:

$$MACD_t = EMA_{t,12} - EMA_{t,26}$$

La *signal line* corrisponde, invece, ad una media mobile esponenziale a 9 giorni, calcolata non sui prezzi ma sul MACD:

$$SL_t = EMA_{t,9}(MACD_t)$$

Questo indicatore, pertanto, fornisce tre tipologie di segnali:

- MACD e *signal line*: quando il MACD taglia verso l'alto la *signal line*, il segnale è rialzista (conviene comprare); se il MACD taglia verso il basso la *signal line*, il segnale

---

<sup>47</sup> Relative Strength Index, analizzato nel capitolo seguente.

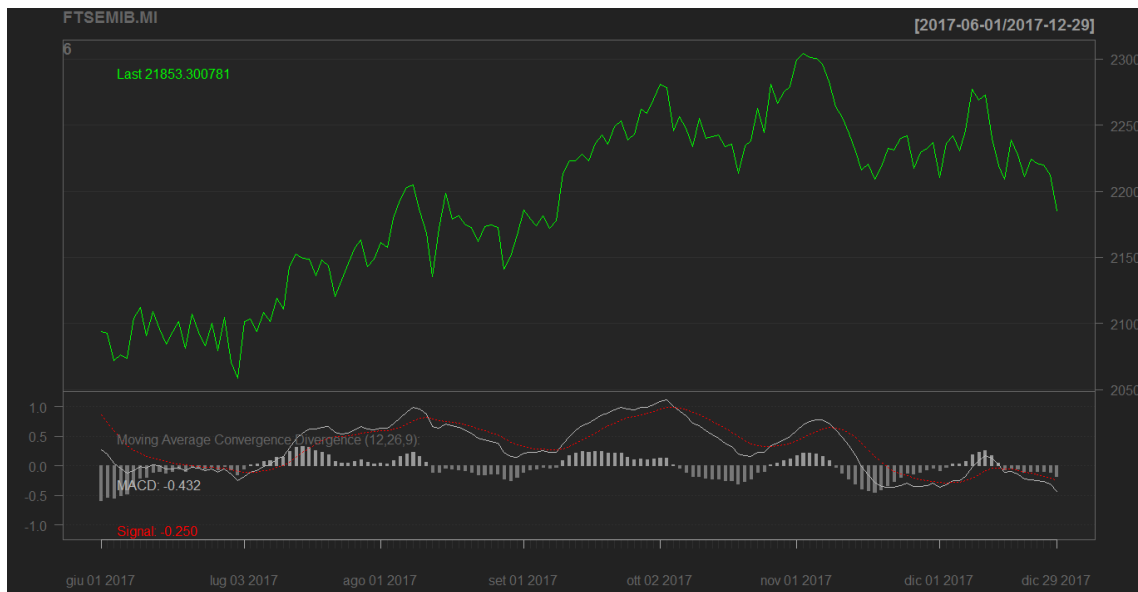
<sup>48</sup> Per questo motivo la MACD è definita *differential line*.

<sup>49</sup> I periodi di tempo per la MACD sono spesso indicati come 26 e 12, ma la formula originale usava come costanti esponenziali 0.075 e 0.15, che corrispondono a 25.66667 e 12.33333 periodi.

è ribassista (conviene vendere);

- prezzo e MACD: a prescindere dalla direzione, la divergenza tra il prezzo e il MACD suggerisce che il trend sta perdendo forza e il mercato è vicino a un'inversione di rotta;
- *central line* (o livello zero) e MACD: quando il MACD taglia dall'alto verso il basso la *central line* (ovvero diventa inferiore a zero, negativo) il segnale è ribassista. Quando il MACD taglia dal basso verso l'alto la *central line*, il segnale è rialzista.

Grafico 11. FTSE Mib, MACD e *Signal Line* (2017)



### 2.2.5 Relative Strength Index (RSI)

Il Relative Strength Index, meglio noto con l'acronimo RSI, è un popolarissimo indicatore utilizzato per individuare situazioni di ipercomprato/ipervenduto. Ideato da John Welles Wilder jr.<sup>50</sup>, l'RSI, nonostante quanto il suo nome possa suggerire, non ha il fine di misurare la forza relativa del titolo rispetto ad una media di mercato, bensì quello di quantificare il suo momentum. La formulazione relativamente semplice e la facilità di lettura fanno dell'RSI l'indicatore più osservato e seguito dagli analisti tecnici.

Di seguito la sua formulazione:

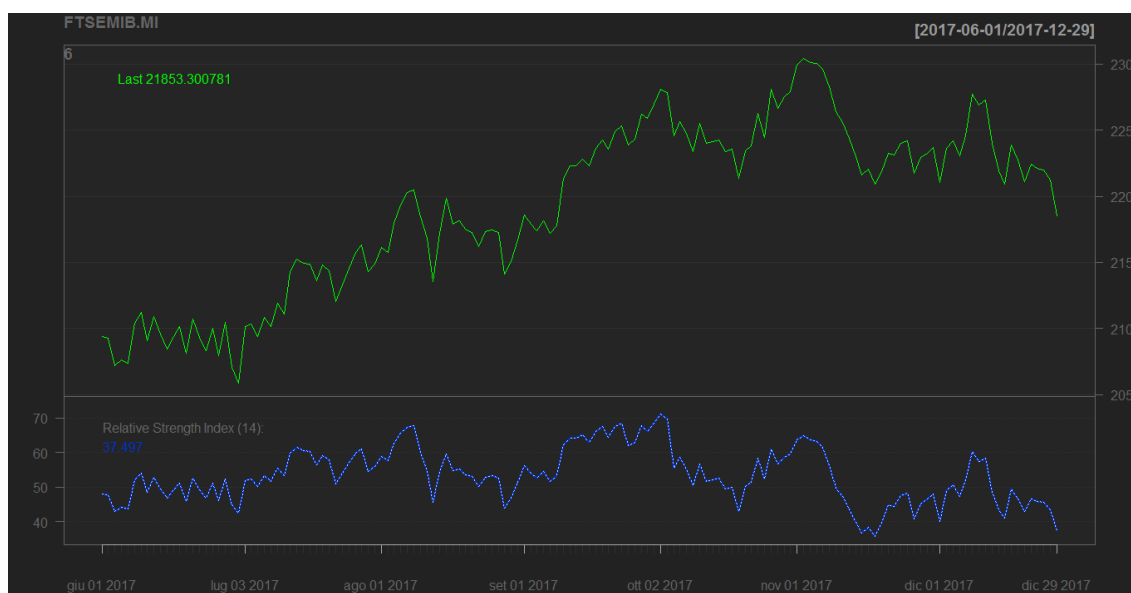
<sup>50</sup> L'RSI venne presentato per la prima volta in: Wilder, J. W. Jr. (1978), *New concepts in technical trading systems*, Trend Research.

$$RSI_t = 100 - \frac{100}{RS_t} = 100 \left( \frac{RS_t}{1 + RS_t} \right)$$

con  $RS_t = AU_t/AD_t$ , ovvero rapporto tra la media degli incrementi di prezzo degli ultimi n giorni e la media dei decrementi di prezzo degli ultimi n giorni.

Qui sorgono i primi problemi applicativi dell'RSI, dal momento che occorre definire un orizzonte temporale di riferimento: Wilder nella sua formulazione originale ha suggerito un periodo di 14 giorni, ovvero metà di un ciclo lunare, ritenuto influenzante il mercato, quantomeno nel breve periodo. Per tale motivo la forma standard di questo indicatore prevede il calcolo su 14 giorni. Ciononostante, diversi studiosi e professionisti, sottolineando come la grande diffusione dell'RSI a 14 giorni avesse portato a scontare direttamente le sue conclusioni operative nei prezzi, impoverendo l'indicatore stesso, si sono spesi per isolare il ciclo dominante il mercato e prendere come giorni di riferimento la metà di esso. Non essendoci, a priori, una versione dell'indicatore migliore delle altre, sta nel singolo trader decidere come applicarlo, tanto che, infatti, si sono diffuse diverse varianti dell'RSI originale.

Grafico 12. FTSE Mib e RSI (2017)



Una volta formato il grafico dell'RSI, vi sono diversi modi per interpretarlo e trarre segnali di trading:

- massimi e minimi dell'oscillatore: sebbene l'esatto livello da utilizzare sia oggetto

di discussione<sup>51</sup>, Wilder, nella formulazione originale, prevede l'utilizzo dei livelli di 70 e di 30, fuori dai quali si verificherebbe una situazione, rispettivamente, di ipercomprato e ipervenduto. Secondo questo ragionamento, se l'RSI aumenta oltre 70, è probabile il raggiungimento di un nuovo massimo; se l'RSI diminuisce sotto il valore di 30, esiste una buona probabilità che si verifichi un nuovo minimo. Tuttavia, negli anni seguenti, si è capito che l'impiego delle zone 30/70, per stabilire i segnali operativi, è sconsigliato poiché i due valori sarebbero troppo vicini per selezionare effettivamente la presenza di massimi o minimi e troppo spesso nelle zone di ipercomprato e ipervenduto si preparano le basi di un robusto trend. Per tale motivo, le *overbought* e *oversold zone* non andrebbero mai considerate come uniche fonti di segnali operativi;

- lettura delle figure generate dall'RSI: tale pratica si fonda sui concetti propri dell'analisi grafica, poiché figure come, ad esempio, la conformazione a triangolo, la testa e spalle, minimi e massimi, ecc. possono essere viste in anticipo nell'RSI, con le relative considerazioni;
- *failure swings*: nei prezzi massimi, si verifica quando l'indicatore aumenta sopra il livello della zona di ipercomprato (per convenzione posto a 70), scende ad un punto inferiore (detto punto di fallimento), aumenta nuovamente senza riuscire tuttavia a raggiungere la linea di 70, e infine riscende al di sotto del punto di fallimento. Questa conformazione del grafico indicherebbe che è il momento di vendere. Vale ovviamente la stessa considerazione per quanto riguarda il processo inverso nella zona di ipervenduto;
- supporti e resistenze: all'interno del grafico dell'RSI si possono facilmente individuare una serie di massimi o minimi relativi, che sono collegabili tra loro in modo tale da formare resistenze o supporti. La rottura di quest'ultimi da parte dell'RSI può fornire gli stessi segnali operativi che si colgono nel caso di supporti e resistenze dei grafici a barre dei prezzi;
- divergenza tra il grafico del prezzo e l'andamento dell'RSI: questa situazione indica

---

<sup>51</sup> Nel corso degli anni, diversi studi sulla distribuzione dell'RSI hanno mostrato che il valore medio di un massimo o minimo RSI è consistentemente raggruppato vicino 72 e 32 e che il 50% di tutti i valori del RSI sta fra 32 e 72. Ciò dovrebbe suggerire che i livelli 30-70 sono troppo vicini per agire come selezionatori di zone di ipercomprato/ipervenduto.

in modo chiaro che i prezzi stanno per cambiare il loro trend. In particolare, se i prezzi stanno aumentando o sono stabili, mentre l'RSI è in diminuzione, occorre prestare attenzione ad un probabile cambiamento di direzione dei prezzi al ribasso; se, viceversa, i prezzi diminuiscono e l'RSI aumenta, ci si aspetta un probabile aumento del valore del titolo. Nonostante quanto appena esposto sia un segnale che si verifica realmente, spesso i segnali che si ottengono con questo metodo sono falsi, e scremare le divergenze che preludono un'inversione del trend non è compito semplice.

In generale bisogna prestare attenzione al tempo che passa tra due minimi o massimi relativi dell'RSI: tempi brevi e reazioni veloci sono solitamente molto meno affidabili di divergenze che si formano tra due punti distanti almeno 20 giorni. In ogni caso una divergenza su un RSI a 14 giorni va interpretata diversamente da una divergenza formata su un RSI a 50 giorni, più breve è il periodo e più limitato sarà il trend seguente. Il modo migliore per interpretare questi segnali è quello di considerarli come campanelli di allarme, dei segnali di possibile inversione o quantomeno di indebolimento della tendenza in atto. Le divergenze più affidabili sono quelle che nascono da grafici settimanali e hanno almeno 20-30 giorni come periodo di formazione.

Possibile variante dell'RSI può essere, oltre ai già citati casi in cui si decide di utilizzare un arco temporale differente dai 14 giorni canonici della formulazione di Wilder, il *Net Momentum Oscillator*,

$$NMO_t = 100 \left( \frac{\sum_{i=1}^n U_i - \sum_{i=1}^n D_i}{\sum_{i=1}^n U_i + \sum_{i=1}^n D_i} \right)$$

dove al numeratore troviamo la differenza delle sommatorie, rispettivamente, degli incrementi e decrementi degli ultimi n giorni, e al denominatore la loro somma.

### 2.2.6 Stochastics oscillator

L'oscillatore stocastico (o indicatore stocastico) è, anch'esso, uno degli indicatori tecnici più utilizzati nell'ambito dell'analisi tecnica e nello studio dei grafici di prezzo. Fu creato da George Lane nel 1970, grazie all'intuizione che normalmente le chiusure di giornata vicine al loro massimo sono tipiche di tendenze al rialzo e, viceversa, le chiusure verso il minimo

sono indice di un trend al ribasso. Esso misura infatti quanto le chiusure delle candele siano vicine al massimo o al minimo giornaliero del prezzo. Questo strumento serve ad anticipare una possibile inversione di tendenza del prezzo, identificando le situazioni di ipercomprato o di ipervenduto, e fornendo quindi dei segnali di entrata o di uscita su uno strumento finanziario. L'indicatore confronta il valore di chiusura attuale e lo rapporta ai valori massimi e minimi del prezzo registrati in un certo periodo di tempo.

L'oscillatore stocastico è un indicatore di momentum normalizzato delimitato dai limiti di 0 e 100, ed indica i massimi e minimi di mercato, la zona in cui ci troviamo e come muoverci. Il calcolo di questo oscillatore si basa sul rapporto tra valori minimi e massimi di prezzo di un periodo di tempo. È costituito infatti da due linee:

- *%K line* (o *fast line*), rappresenta la percentuale del prezzo di chiusura nel periodo misurato:

$$\%K_t = 100 * \frac{[C_t - \min(L_{t-n+1}, \dots, L_t)]}{[\max(H_{t-n+1}, \dots, H_t) - \min(L_{t-n+1}, \dots, L_t)]}$$

dove  $C_t$  rappresenta il prezzo di chiusura odierno,  $\min(L_{t-n+1}, \dots, L_t)$  il prezzo minimo degli ultimi n giorni e  $\max(H_{t-n+1}, \dots, H_t)$  il prezzo massimo degli ultimi n giorni.

La formula fornisce un valore percentuale, indica infatti dove si trova attualmente il prezzo rispetto al range massimo-minimo degli ultimi n giorni;

- *%D line* (o *slow line*), rappresenta la media mobile (calcolata su tre periodi) della linea %K. Serve a filtrare i movimenti erratici della fast line e generare un effetto di *crossover* con la linea %K, generando i segnali di vendita e acquisto.

Esistono differenti tipi di oscillatore stocastico e ciascuno mostra segnali in modo diverso. A variare è il periodo di tempo preso in esame e il modo in cui opera l'oscillatore. Rimane uguale invece la modalità di calcolo dei minimi e massimi di mercato. Vi sono 3 tipi di oscillatore stocastico: veloce (*fast*), lento (*slow*) e completo (*full*):

- la versione veloce dell'oscillatore è l'idea originale di Lane, che mette in rapporto massimi e minimi di prezzo di un periodo di tempo, tracciando così il valore %K e il valore %D. Questa versione non è molto utilizzata a causa dell'eccessiva reattività dell'indicatore e della difficile applicazione ed interpretazione (falsi segnali generati);

- la versione lenta è stata creata per cercare di ovviare a questo problema. L'upgrade permette una migliore generazione dei segnali, in quanto la nuova linea del valore %K (chiamata ora *K Slow*, %KS) viene disegnata come media mobile a 3 periodi di quella %K dello stocastico veloce:

$$\%KS(t) = \frac{\sum_{i=t-2}^t \%K_i}{3}$$

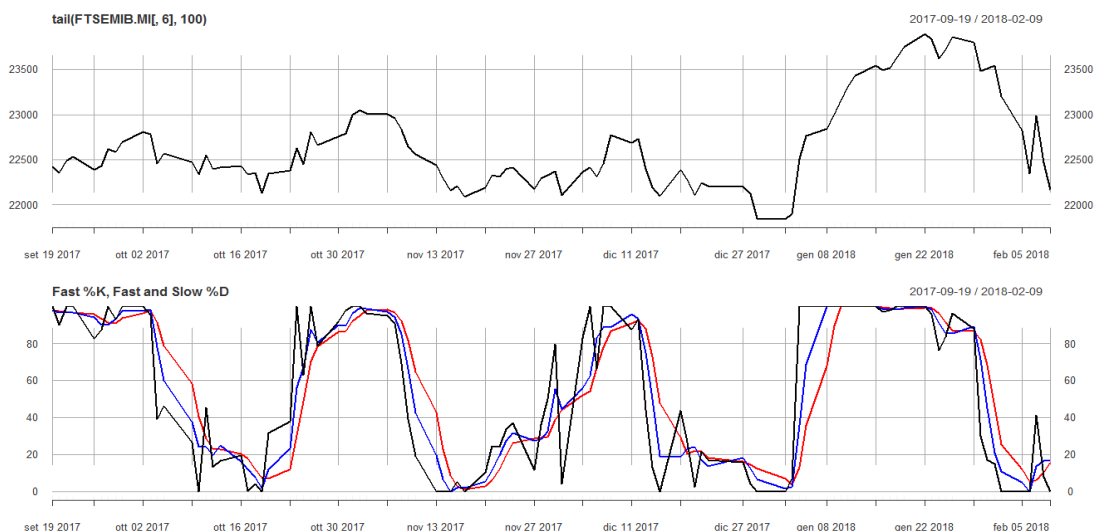
In questo caso la linea %D (chiamata ora *D Slow*, %DS) è la media mobile a 3 periodi della nuova linea %KS:

$$\%DS(t) = \frac{\sum_{i=t-2}^t \%KS_i}{3}$$

Questa versione è preferibile a quella veloce, in quanto restituisce segnali meno sensibili alle variazioni di prezzo. Si ha un'operatività minore, ma anche molti meno falsi segnali, riducendo gli errori;

- la versione completa è la versione più utilizzata, in quanto permette la personalizzazione sia per il numero di periodi da prendere in considerazione, sia per il numero di periodi per %K e %D.

Grafico 13. FTSE Mib e Stochastics Oscillator (settembre 2017-febbraio 2018)



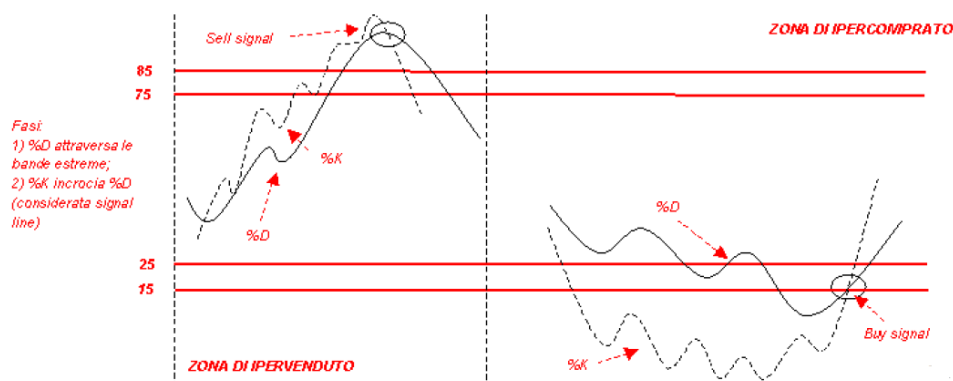
I segnali operativi maggiormente significativi riguardano due aspetti:

- l'interazione tra i due indicatori %K e %D: la linea %K viene utilizzata come campanello di allarme per identificare possibili momenti di entrata e uscita dal mercato, mentre la linea %D viene utilizzata per generare il segnale vero e proprio nel momento in cui incrocia la linea più lenta. Quando %K attraversa al rialzo o al

ribasso %D, un mutamento nel trend di mercato è imminente. Tuttavia, per evitare i falsi segnali, questa condizione non è sufficiente: perché il segnale sia consistente, l'incrocio delle due linee deve avvenire nelle zone di ipercomprato e ipervenduto, determinate grazie a livelli prestabiliti. I valori dello stocastico possono variare tra 0 e 100: intorno a quota 20, solitamente, si traccia la linea di ipervenduto (situazione che potrebbe dar luogo ad un'inversione rialzista), mentre a quota 80, generalmente, si pone la linea di ipercomprato (situazione che potrebbe dar luogo ad un'inversione ribassista). In generale, se il prezzo è ipercomprato è un ottimo segnale per posizionarsi short, se il prezzo è in condizioni di ipervenduto vi sono buone opportunità per posizionarsi long. Va sottolineato che, anche se i prezzi si trovano nelle zone di ipercomprato o ipervenduto, non è detto che ci sarà un'inversione di tendenza, o che comunque sarà immediata. Per questo motivo è utile valutare assieme il posizionamento in queste zone, con l'incrocio delle linee dello stocastico ed eventualmente anche altri indicatori o segnali;

- eventuali divergenze tra l'andamento del prezzo e quello dell'indicatore: sono i casi in cui il prezzo continua a segnare nuovi massimi (o minimi), mentre lo stocastico si muove in direzione opposta. Si parla di divergenza rialzista quando il prezzo effettua nuovi minimi verso il basso e l'oscillatore effettua minimi crescenti. Viceversa, siamo di fronte ad una divergenza ribassista quando il prezzo effettua massimi sempre più alti mentre lo stocastico effettua massimi decrescenti. Queste divergenze segnalano un cedimento del trend, e quindi un forte presupposto di inversione in tempi molto brevi.

Grafico 14. Trading signals forniti dallo stocastico



Fonte: Prof. Luca Cappellina, materiali didattici del corso "Laboratorio di Finanza" (2019)



Lo stocastico può essere utilizzato sia quando il mercato è in una fase di trend, sia quando è in una fase laterale o di trend non ben definito. Questo tipo di indicatore non è sempre facile da utilizzare, per tale ragione molti investitori non lo usano in modo ottimale e vengono ingannati dai falsi segnali generati. In conclusione, l'indicatore stocastico è capace di generare un alto numero di segnali. Tuttavia, come la maggior parte degli indicatori tecnici, è sempre meglio utilizzare un altro oscillatore in affiancamento allo stocastico, che può dare un maggiore aiuto nel capire la forza del trend in atto.

### 2.2.7 Average Directional Movement Index (ADX)

L'*Average Directional Movement Index (ADX)*, il *Minus Directional Indicator* e il *Plus Directional Indicator* rappresentano un gruppo di indicatori di tendenza sviluppati nel 1978 dal già noto J. Welles Wilder. In particolar modo, questi indici permettono di individuare l'andamento del trend partendo dalla serie storica dei prezzi dello strumento finanziario (stocks, mutual funds, exchange-traded funds e futures). L'ADX misura la forza di tale trend senza però fornire indicazioni sulla sua direzione. Il +DI (*Plus Directional Indicator*) e il -DI (*Minus Directional Indicator*) completano la funzione dell'ADX definendo anche la direzione del trend. Per cui, utilizzando insieme i tre indicatori, è possibile determinare sia la direzione del trend che la sua forza.

L'ADX viene derivato dalla media mobile delle differenze tra +DI e -DI. Per poter ottenere i valori +DI e -DI, e quindi di ADX, sono indispensabili i prezzi massimi, minimi e di chiusura di ogni periodo (normalmente si effettua tale analisi utilizzando osservazioni giornaliere) ed è necessario calcolare il *Plus Directional Movement (+DM)*, il *Minus Directional Movement (-DM)* e l'*Average True Range (TR)*. Il +DM è dato da:

$$+DM_t = \begin{cases} (H_t - H_{t-1}) & , se (H_t - H_{t-1}) > (L_{t-1} - L_t) \text{ e } (H_t - H_{t-1}) > 0 \\ 0 & , altrimenti \end{cases}$$

mentre -DM è dato da:

$$-DM_t = \begin{cases} (L_{t-1} - L_t) & , se (L_{t-1} - L_t) > (H_t - H_{t-1}) \text{ e } (L_{t-1} - L_t) > 0 \\ 0 & , altrimenti \end{cases}$$

e l'*Average True Range* è dato da:

$$ATR_t = \frac{ATR_{t-1} * (n - 1) + TR_t}{n}$$

dove  $TR_t = \max [(H_t - L_t), |H_t - C_{t-1}|, |L_t - C_{t-1}|]$ .

Dopo aver fissato il numero di periodi per il calcolo (Wilder suggerisce di utilizzarne 14), si può procedere alla misurazione di +DI e -DI. Per ottenere il *Plus Directional Indicator* e il *Minus Directional Indicator*, +DM e -DM vengono divisi per l'*Average True Range*:

$$+DI_t = \frac{+DM_t}{ATR_t}$$

$$-DI_t = \frac{-DM_t}{ATR_t}$$

Infine, l'algoritmo per il calcolo del *Average Directional Movement Index* è:

$$ADX_t = \frac{1}{k} \sum_{i=0}^{k-1} \frac{DI_{t-1}^+ - DI_{t-1}^-}{DI_{t-1}^+ + DI_{t-1}^-} * 100$$

Si possono utilizzare vari tipi di media mobile, tra cui la media mobile pesata o la media mobile adattiva. In ogni caso, sono necessari almeno 150 periodi per poter assorbire le *smoothing techniques*, ed ottenere dei risultati attendibili.

Secondo Wilder c'è la presenza di un forte trend in un titolo quando l'indicatore ADX supera il valore di 25, mentre il trend è debole o assente quando l'indicatore si trova sotto il valore di 20. Nel caso in cui si voglia analizzare il mercato italiano, si può accettare un valore di ADX pari a 20 come soglia critica, superata la quale il titolo è da considerarsi in trend, questo perchè i titoli italiani sono in linea generale meno volatili di quelli americani. Nonostante le diverse fasce di forza del trend in base al valore dell'ADX siano a discrezione del singolo trader, si possono considerare, a titolo esemplificativo, le seguenti:

Tabella 1. Valori dell'ADX e forza del trend

ADX VALUE	TREND STRENGHT
0-25	Absent or weak trend
25-50	Strong trend
50-75	Very strong trend
75-100	Extremely strong trend

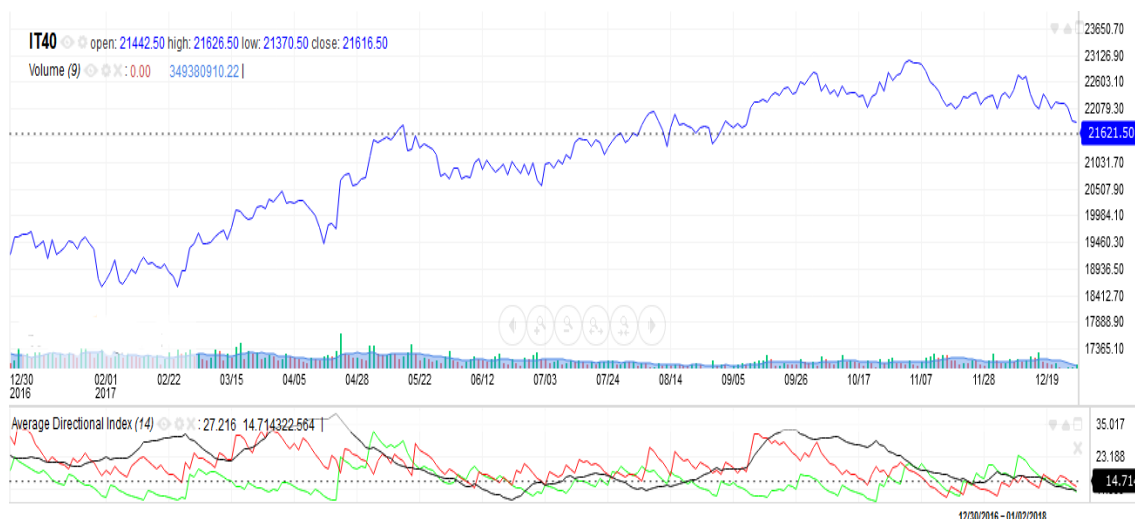
Va ricordato che l'ADX è un indicatore di ritardo (*lagging indicator*): il trend deve essersi già sviluppato prima che l'indice ADX generi un segnale che identifica la presenza di una tendenza.

L'ADX, insieme al +DI e -DI, fornisce la possibilità di determinare il momento più adeguato per entrare o uscire da una determinata negoziazione. In particolar modo, esistono tre regole fondamentali che indirizzano le decisioni, basate, per l'appunto, su questi indici. Una prima indicazione viene fornita dalla *Crossover Rule*: si dovrà adottare una posizione lunga ogni qualvolta +DI incroci e oltrepassi -DI. Viceversa, quando il -DI supera +DI, si adotterà una posizione corta.

Una seconda regola, l'*Extreme Point Rule*, permette di individuare il momento migliore in cui effettuare un'operazione. I segnali di acquisto o vendita vengono attivati quando le curve di +DI e -DI si incrociano. Quando questo accade, si dovrà procedere con l'individuazione del punto estremo registrato in quella giornata: il punto estremo è il prezzo più alto se +DI supera -DI, mentre è il prezzo più basso della sessione quando -DI supera +DI. Per esempio, dopo avere avuto un segnale d'acquisto (il +DI supera il -DI), bisogna attendere fino a che il prezzo del titolo superi il punto estremo (il prezzo maggiore della seduta in cui le linee del +DI e del -DI si sono intersecate) prima di comprare. Se il prezzo fallisce il tentativo di superare il punto estremo, occorre mantenere la posizione short.

Infine, la *Turning Point Rule* afferma che quando la curva ADX scende sotto le curve DI (*Plus and Minus*) ci potrebbe essere un'inversione di tendenza nel mercato. Potrebbe essere quindi un segnale di avvertimento che il mercato sta cambiando la propria direzione.

Grafico 15. FTSE Mib, +DI (rosso), -DI (verde) e ADX (nero) - (2017)



## 2.3 STATO DELL'ARTE IN LETTERATURA

Nel corso degli ultimi vent'anni, la ricerca scientifica mondiale ha sviluppato e pubblicato molteplici studi sulle applicazioni dell'intelligenza artificiale nel campo degli investimenti finanziari. Cercare di predire l'andamento di un mercato finanziario è d'altronde da sempre una delle più grandi sfide dell'economia; pensare di poterlo fare con l'utilizzo di tecnologie di machine learning è, senza dubbio, una rivoluzione di pensiero.

All'origine della formulazione di trading system automatici basati sul *reinforcement learning*, descritta nel seguito della presente tesi, vi è lo studio di diversi articoli scientifici internazionali<sup>52</sup>. Ciascuno di questi ha permesso di trarre delle considerazioni utili sia in fase di sviluppo del sistema, sia nel momento della valutazione dei limiti e delle possibili implementazioni future. Vengono dunque descritti ora i principali contributi e risultati delle suddette ricerche scientifiche.

I primi studi sull'efficacia dei trading system risalgono alla metà del Novecento. In particolare, l'utilizzo di cosiddette regole di filtraggio dei prezzi, che determinassero quando comprare, vendere o mantenere, si dimostrava poco efficiente in confronto alla semplice strategia *buy and hold*<sup>53</sup> (Alexander, 1961; Fama & Blume, 1966). Altri studi, invece, si concentravano sulle strategie di momentum, le quali prevedono che i vincenti (intesi come i titoli che guadagnano) continuino ad essere vincenti, mentre i perdenti continuino ad essere perdenti. Nelle ricerche di Jegadeesh e Titman (1993,2001) viene dimostrato come strategie di questo tipo possano restituire profitti elevati. Altri approcci di trading sono le cosiddette strategie contrarie, così chiamate perché cercano di sfruttare le reazioni eccessive dei mercati comprando i titoli "bearish" e vendendo quelli "bullish". Un caso particolare di strategia contraria è il *pairs trading*, che prevede di selezionare coppie di titoli strettamente sostituibili e dunque di comprare quello in calo e vendere quello in aumento. Do e Faff (2010, 2012) e Gatev, Goetzmann e Geert Rouwenhorst (2006) hanno ottenuto guadagni rilevanti utilizzando strategie di questo tipo.

---

<sup>52</sup> Per un preciso riferimento testuale degli scritti menzionati nel presente sottocapitolo, si rimanda alla bibliografia conclusiva.

<sup>53</sup> Letteralmente "compra e mantieni", è la strategia classica del cassetista che prevede l'acquisto del titolo ed il mantenimento sino a scadenza del periodo d'investimento.

In aggiunta alle strategie di trading, molte ricerche si concentrano sull'utilizzo di sistemi intelligenti in problemi di controllo dinamico, come, per l'appunto, il processo d'investimento finanziario. Queste decisioni d'investimento dinamico sono spesso modellizzate da processi decisionali di Markov (MDP), stretti parenti degli algoritmi di *reinforcement learning* utilizzati nel presente studio. Gli MDP sono stati largamente applicati in finanza nel corso degli anni. Tuttavia, gli MDP sono criticati perché raramente i mercati finanziari moderni rispettano le ipotesi Markoviane di perfetta efficienza (Nevmyvaka, Feng & Kearns, 2006). Nel trading di titoli finanziari, infatti, è difficile che vi sia conoscenza a priori di un modello stocastico che descriva la transizione da stato a stato; inoltre, non è facile intuire una funzione di ricompensa universale. In questi casi, utilizzare modelli MDP pone diversi problemi, perché si cerca di approcciare un ambiente parzialmente osservabile (quello finanziario reale) come se fosse pienamente osservabile. Questo limite può essere superato scindendo un modello MDP in due fasi: una prima fase in cui l'algoritmo impara il modello di transizione e la funzione di ricompensa dell'ambiente in cui deve essere applicato; ed una seconda fase in cui il processo decisionale di Markov impara la *policy* ottimale. In alternativa, la politica migliore può anche essere appresa attraverso un processo *trial-and-error*, proprio degli approcci *model-free* di *reinforcement learning*. Nel corso degli ultimi vent'anni, molte ricerche si sono dedicate all'applicazione dell'apprendimento per rinforzo nell'esecuzione del trading finanziario. Portato in auge dagli studi di Sutton & Barto (1998), il forte interesse verso le applicazioni del RL in finanza è dovuto alla possibilità di creare agenti automatici in grado di processare in tempo reale grandi moli di dati ad alta frequenza al fine di eseguire ordini di mercato. Tra le prime applicazioni finanziarie rilevanti, gli studi di Bertsimas & Lo (1998) descrissero l'utilizzo del RL per la gestione di grandi blocchi di azioni (propria degli investitori istituzionali) lungo un periodo d'investimento fisso, scelto al fine di minimizzare i costi di transazione attesi in modo tale da non doverli considerare. Il risultato dello studio mostrò come la scelta di modificare i pesi delle azioni rispetto al portafoglio totale, permetteva una riduzione dei costi di esecuzione fino al 40% rispetto ad una semplice strategia che divide in parti uguali il numero di azioni in portafoglio. Ciononostante, vi era il grande limite di assumere che il volume di ogni transazione fosse sempre sufficientemente elevato da poter influenzare il prezzo di mercato. In più, secondo altri studi, come quello di Hildenbrand & Kirman (1998), il prezzo di titoli contrattati in mercati di grandi dimensioni è fuori dal controllo di ciascun

investitore, anche se questo è un investitore istituzionale: questo dato sottolineava la scarsa applicabilità degli studi di *Bertsimas & Lo*.

*Nevmyvaka et al. (2006)* proposero un algoritmo di RL per ottimizzare l'esecuzione di investimenti ad alta frequenza, con l'utilizzo inoltre di "limit order", ovvero livelli fissati dall'investitore, al raggiungimento dei quali in automatico si compra (o vende). I risultati ottenuti furono decisamente interessanti, di gran lunga migliori di una classica strategia *submit-and-leave*.

Diversi studi, poi, hanno introdotto il concetto di *Sharpe ratio*<sup>54</sup> come misura di performance da utilizzare nella funzione di *reward* che guida l'apprendimento per rinforzo (*Almahdi & Yang, 2017*). Maggiori livelli dell'indice di Sharpe indicherebbero allocazioni di portafoglio migliori, in linea con le teorie di Markowitz: per questo motivo utilizzare questo rapporto come funzione di ricompensa da massimizzare può essere più efficace che utilizzare una semplice misura di profitto. Oltre alla forma canonica dello *Sharpe ratio* spesso viene utilizzata la versione differenziale dell'indice, specialmente nei casi in cui sia richiesta un'ottimizzazione delle performance in tempo reale.

Gli studi di *Moody, Wu, Liao e Saffell (1998)* e di *Moody e Saffell (2001)* proposero l'utilizzo di algoritmi di *reinforcement learning* per la definizione dell'*asset allocation* di un portafoglio: in particolare, le loro analisi avevano il fine di determinare un sistema automatico che imparasse a gestire i pesi interni delle componenti di un portafoglio, quella azionaria composta dall'indice statunitense S&P500, e quella obbligazionaria rappresentata dagli *U.S. Treasury bills*. Al di là dell'apporto scientifico innovativo fornito da questa diversa applicazione di RL, questi studi sottolinearono i potenziali limiti nell'utilizzo dello *Sharpe ratio*<sup>55</sup>: questo, infatti, penalizza i ritorni di grandi dimensioni, dà maggior risalto ai rendimenti recenti rispetto a quelli passati, ed infine rappresenta una misura di utilità adattiva e non assoluta, in altre parole instabile. Inoltre, l'indice di Sharpe non fa distinzione tra la crescita potenziale positiva e quella negativa, quando in realtà è noto che

---

<sup>54</sup> L'indice di Sharpe, dal nome del premio nobel per l'economia William Sharpe, è una misura di performance del portafoglio definita dal rapporto tra la differenza fra il rendimento del portafoglio ed il tasso free-risk, e la deviazione standard del portafoglio stesso:  $\frac{r_P - r_f}{\sigma_P}$ .

<sup>55</sup> Seppur al singolare, si fa riferimento ad entrambe le forme di indice di Sharpe, originale e differenziale.

la funzione di utilità di un investitore non è simmetrica: perdere è percepito in misura maggiore rispetto a vincere.

*Bertoluzzo e Corazza (2012)* hanno sviluppato degli algoritmi per il trading finanziario automatico basati su due forme di apprendimento per rinforzo, il *Q-learning* ed il *Kernel-based*. Gli autori hanno deciso di utilizzare la forma classica dell'indice di Sharpe e di calcolarla su un arco temporale di una settimana finanziaria: cinque giorni compongono dunque un singolo stato. Le azioni che l'agente di RL può prendere, sono state suddivise in *buy, sell e hold*. Utilizzando sia dati simulati che dati reali di tre differenti titoli azionari italiani, sono state paragonate le performance dei due algoritmi sviluppati: in generale l'approccio di *Q-learning* si è rivelato più performante, mentre è stata riconosciuta una certa influenza dell'inizializzazione randomica sui risultati finali, in particolar modo nel caso delle serie finanziarie reali. Inoltre, la scelta di parametri come il tasso di apprendimento  $\alpha$  o  $\epsilon$  si è rivelata molto importante, anche in relazione ai diversi dati analizzati. Infine, la scelta di definire gli stati come performance su base settimanale si è dimostrata semplicistica, a maggior ragione perché, anche in questo caso, vengono sottolineati i già menzionati limiti dello *Sharpe ratio*.

In *Pendharkar & Cusatis (2018)* viene per la prima volta investigato l'uso di agenti intelligenti, basati su algoritmi di *reinforcement learning*, nella gestione di portafogli personali pensionistici composti da due soli asset, uno azionario (l'S&P500) ed uno obbligazionario (i *T-note* a 10 anni). L'ottica analizzata è pertanto quella di investimenti di lungo periodo, con operazioni poco frequenti. Gli autori hanno preso in esame sia un metodo *on-policy* (SARSA), sia uno *off-policy* (*Q-learning*); in entrambi i casi hanno proceduto a massimizzare il rendimento del portafoglio oppure il *differential Sharpe ratio*. Le azioni che l'agente può prendere si riferiscono ai pesi dell'allocazione di portafoglio: sono state distinte due casistiche, le azioni a dominio discreto e le azioni a dominio continuo. I risultati dei test hanno sottolineato come la possibilità per l'agente di prendere azioni adattive (dunque in un dominio continuo) sia molto più performante. Inoltre, la frequenza annuale dei dati ha permesso di ottenere risultati migliori rispetto all'analisi con dati semi-annuali o trimestrali. Infine, è stato osservato che a *training dataset* di dimensioni maggiori non corrispondono sempre performance migliori.

Prospettiva completamente diversa, invece, quella analizzata da *Carapuço, Neves e Horta (2018)*, i quali hanno ideato un nuovo sistema di speculazione di brevissimo periodo nel

*foreign exchange market* (Forex), basato su una rete neurale a tre livelli allenata tramite un algoritmo di *Q-learning*. L'utilizzo di una rete neurale in fase di training ha permesso una maggior qualità dell'apprendimento ed una migliore accuratezza in fase di test.

Quest'ultimo studio non è l'unico caso di utilizzo combinato di reti neurali e algoritmi di *reinforcement learning*, ma appare piuttosto come l'ultima tendenza di studio in ambito di ricerca scientifica. Jeong & Young Kim (2018) hanno sviluppato un modello di *deep Q-learning* che affianca alla componente di apprendimento per rinforzo, una rete neurale profonda allo scopo di predire il numero ottimale di azioni da investire in ogni singola operazione aperta. Inoltre, hanno introdotto diverse strategie che utilizzano il *Q-value* di ogni possibile azione per analizzare quale strategia sia ottimale in situazioni di mercato altamente volatile. Infine, hanno proposto un approccio di *transfer learning* per ovviare al problema dell'*overfitting*. Queste tre misure, testate su 4 differenti indici internazionali, hanno fornito risultati strabilianti.

Lei, Zhang et al. (2019) hanno proposto invece un sistema che utilizza congiuntamente un modello di *deep learning* per la fase di apprendimento dell'ambiente, ed uno di *reinforcement learning* per quanto riguarda il compito di *decision making*. In particolare, gli autori dimostrano come l'apprendimento profondo sia migliore di quello per rinforzo nella rappresentazione e percezione dell'ambiente, mentre sia decisamente limitato nell'adattamento richiesto per il processo decisionale. Oltre alla mistione di *deep learning* e *reinforcement learning*, lo studio introduce sia un "Gate" alla rete neurale che permette di selezionare in maniera adattiva le caratteristiche delle serie storiche ritenute dal sistema influenti (come i segnali degli indicatori tecnici, le variazioni di prezzo ecc.), sia una *Gated Recurrent Unit* (GRU) che consente di catturare le dipendenze di lungo periodo. Questi espedienti nascono come soluzione a limiti come la scarsa efficienza della selezione delle caratteristiche influenti sul prezzo ed il rumore nelle serie storiche che non permette di cogliere i trend di lungo periodo. I test del modello sono stati applicati su titoli con trend diversi, ed i risultati hanno sottolineato la grande robustezza e profittabilità del sistema proposto.

Sono stati creati negli anni anche sistemi che si basano sul *sentiment* di mercato. In particolare, ottimi risultati sono stati ottenuti da Yang, Yu e Almahdi (2018), i quali hanno sviluppato un modello in grado di estrarre dei segnali, nel *sentiment* degli investitori, che generano movimenti positivi o negativi nel mercato. Questo sistema che estrae di volta in



volta il *reward* collegato al segnale osservato, non si basa solamente sul semplice rendimento di mercato ma anche sulla sua volatilità. I risultati dei test, oltre ad essere superiori ad altre forme classiche di investimento, hanno mostrato come un sistema di trading basato sul sentiment di mercato, funzioni molto meglio in situazioni di mercato volatile, anche se ciò lo espone al problema dei costi di transazione, dovendo fronteggiare un numero di operazioni ben maggiore.

Infine, sono ancora pochi gli studi che riescono a tener conto dei costi di transazione, spesso assunti nulli per ipotesi. Il problema dei costi di transazione è trasversale all'intera economia, poiché di frequente non considerati nelle modellizzazioni teoriche. La loro influenza, tuttavia, è rilevante, specialmente negli studi sui sistemi di investimento. In *García-Galicia, Carsteanu e Clempner (2019)* viene, ad esempio, proposta una forma di penalizzazione in base al tempo che consente di limitare l'incidenza dei costi di transazione sulle performance di portafoglio.



## CAPITOLO III. APPLICAZIONE DI TRADING SYSTEM AUTOMATICI BASATI SU REINFORCEMENT LEARNING

Nelle pagine seguenti viene esposto e discusso il lavoro di ricerca svolto. In particolare, dopo la disamina della letteratura affine, lo studio ha preso avvio con la decisione delle finalità e delle ipotesi alla base del modello da sviluppare. Solo dopo aver posto questi primi vincoli, si è iniziato a costruire un'architettura che permettesse, dapprima di preparare i dati di input per svolgere la fase di training, e successivamente di analizzare i risultati ottenuti tramite l'applicazione dell'algoritmo ideato.

### 3.1 IL CASO DI STUDIO

#### 3.1.1 Ipotesi e finalità del progetto

Lo studio svolto nasce con l'intenzione di sviluppare un modello in grado di coniugare l'utilizzo di indicatori di analisi tecnica, a tutt'oggi molto usati all'interno dei trading system, con strumenti informatici avanzati, quali, ad esempio, gli algoritmi di intelligenza artificiale riconducibili alla sfera del *reinforcement learning*.

La finalità del progetto è stata quella di raccogliere dei segnali derivanti dall'utilizzo di diversi indicatori tecnici in un *dataset* da sottoporre ad un algoritmo di apprendimento per rinforzo, che potesse dunque imparare e prevedere le situazioni in cui il titolo è in aumento o calo, per trarne infine un profitto. Per ottenere questo risultato, si è iniziato con la fase di preparazione dei dati, al cui interno si è calcolato i diversi indicatori tecnici e, di conseguenza, i segnali derivanti; successivamente, si è passati alla fase di training con l'applicazione dell'algoritmo di apprendimento per rinforzo; infine, è stata messa in pratica, all'interno della fase di test, la *policy* ottimale appresa in precedenza.

Il sistema creato, in particolare, è stato allenato e testato su 9 diversi indici finanziari internazionali, così scelti sia per la dimensione elevata dei volumi, sia per poter analizzare un'eventuale applicabilità universale, non limitata ad un singolo Paese. In questo modo, considerando l'indice di Borsa come riflesso dell'andamento dell'economia nazionale, si è potuto esaminare diversi scenari e diversi trend di lungo periodo. Sotto questo punto di vista, la scelta di utilizzare come titoli finanziari degli indici azionari, ha dovuto considerare,

tramite qualche aggiustamento che verrà descritto nel seguito del capitolo, il naturale trend crescente di lungo periodo.

Di seguito i titoli finanziari analizzati:

- Standard & Poor 500: noto semplicemente come S&P500, è il maggiore indice di Borsa statunitense. È stato creato nel 1957 dalla celebre società di rating da cui prende il nome e raccoglie un paniere azionario formato dalle 500 società statunitensi a maggior capitalizzazione;
- Dow Jones Industrial Average: è l'indice più conosciuto del *New York Stock Exchange* e racchiude le 30 maggiori *blue chip*<sup>56</sup> degli Stati Uniti. A differenza della maggior parte degli indici azionari del mondo, il Dow Jones non tiene conto della capitalizzazione di ciascuna società, ma compie solo una media semplice dei prezzi;
- NASDAQ Composite Index: acronimo di *National Association of Securities Dealers Automated Quotation*, il NASDAQ nasce nel 1971 come primo mercato borsistico elettronico al mondo. Esso racchiude i maggiori titoli tecnologici della borsa americana;
- Hang Seng Index: nato nel 1969, è il maggior indice azionario della borsa di Hong Kong. Racchiude le 50 maggiori società, le quali assieme costituiscono circa il 60% della capitalizzazione totale della suddetta borsa asiatica. È quotato in dollari di Hong Kong;
- Nikkei 225: creato nel 1950, rappresenta il maggior indice azionario giapponese. Segmento della borsa di Tokyo (il *Tokyo Stock Exchange*), racchiude le 225 maggiori compagnie quotate. È quotato nella valuta giapponese, lo Yen;
- SSE Composite Index: è il maggiore indice azionario della borsa cinese di Shanghai. Istituito nel 1990, è quotato nella valuta cinese, il renminbi;
- FTSE 100: rappresenta il più importante indice azionario della Borsa inglese, il *London Stock Exchange*. Acronimo di *Financial Times Stock Exchange*, esso racchiude al suo interno le 100 società più capitalizzate quotate al LSE. È valutato in sterline inglesi;
- CAC 40: è il principale indice azionario della Borsa di Parigi. Racchiude le 40 maggiori società per capitalizzazione ed è quotato in euro;

---

<sup>56</sup> Con "blue chip" si indicano comunemente le società ad alta capitalizzazione azionaria.

- DAX Performance Index: è il maggiore indice azionario della Borsa di Francoforte. Contiene i 30 maggiori titoli tedeschi per capitalizzazione. È quotato in euro.

Nel corso del lavoro di ricerca, si è assunto un ruolo ininfluente dell'inflazione sull'andamento dei prezzi di mercato. Questa condizione potrebbe apparire limitante, tuttavia si è preferito concentrarsi esclusivamente sui valori nominali di mercato e sulle loro fluttuazioni. Considerare un fattore che tenga conto dell'inflazione presente nel Paese potrebbe, ad esempio, essere una possibile implementazione futura.

Per simulare l'operato di un trading system con operatività continua, si è deciso di utilizzare dati a frequenza giornaliera. L'entità dei dati a disposizione ha potuto dunque raggiungere quantità sufficientemente elevate da produrre un buon numero di combinazioni di segnali derivanti dagli indicatori tecnici. L'orizzonte temporale considerato ha dovuto tener conto dei dati reperibili: in diversi casi, nonostante gli indici fossero stati creati già da diversi anni, veniva fornito un valore unico per massimo, minimo, prezzo di apertura, prezzo di chiusura e prezzo di chiusura aggiustato per i dividendi. Essendo questi dati fondamentali per il calcolo di alcuni indicatori di analisi tecnica utilizzati, si è dovuto considerare le serie storiche a partire dalla data in cui i suddetti valori erano distinti. In ogni caso si è deciso di considerare come training dataset, tutti i valori utili sino al 31/12/2009, mentre per quanto riguarda la fase di test è stato considerato un periodo temporale uguale per tutti gli indici finanziari considerati, e cioè pari a dieci anni (dal 2010 al 2019, compresi). In tabella 2, una sintesi dei dati utilizzati.

Tabella 2. Dati di input del modello

INDICI	VALUTA	TRAINING SET	TEST SET
S&P500	Dollaro americano (USD)	01/01/1962-31/12/2009	01/01/2010-31/12/2019
DOW JONES	Dollaro americano (USD)	01/02/1985-31/12/2009	01/01/2010-31/12/2019
NASDAQ	Dollaro americano (USD)	01/03/1971-31/12/2009	01/01/2010-31/12/2019
HANG SENG	Dollaro di Hong Kong (HKD)	11/10/1989-31/12/2009	01/01/2010-31/12/2019
NIKKEI 225	Yen giapponese (JPY)	08/04/1988-31/12/2009	01/01/2010-31/12/2019
SSE COMPOSITE	Renminbi cinese (CNY)	02/07/1997-31/12/2009	01/01/2010-31/12/2019
FTSE 100	Sterlina inglese (GBP)	01/01/1984-31/12/2009	01/01/2010-31/12/2019
CAC 40	Euro (EUR)	01/03/1990-31/12/2009	01/01/2010-31/12/2019
DAX 30	Euro (EUR)	14/12/1993-31/12/2009	01/01/2010-31/12/2019

Ad ogni modo, si è stabilito, nel corso dell'intero studio, di considerare un'operatività esclusivamente di fine giornata: questa ipotesi prevede che si venda o compri solamente una volta al giorno, ed esclusivamente un minuto prima della chiusura di mercato. È stato

così possibile utilizzare il prezzo di chiusura aggiustato come unico prezzo da considerare ai fini di investimento.

Nell'ideazione del modello creato, si è assunto per ipotesi che ogni operazione di mercato venisse compiuta su una singola azione del titolo esaminato, dunque senza tener conto di un capitale d'investimento iniziale. In altre parole, il sistema ideato ha voluto considerare il profitto e rendimento derivanti dal possesso di una sola azione, qualunque fosse il capitale necessario per l'acquisto iniziale.

Inoltre, si è assunto che la situazione dell'investitore non fosse mai neutrale, ovvero esterna al mercato, ma sempre con una posizione aperta. Quest'ultima si è divisa in due tipologie: una posizione lunga ed una corta. Ovviamente, per il caso *short* si è assunta la possibilità di vendere allo scoperto, delineando dunque l'operazione come una semplice scommessa al ribasso.

Per quanto riguarda le azioni attuabili, si è deciso di considerare, oltre ai classici ordini di mercato "*buy*" e "*sell*", anche la possibilità di mantenere stabile la posizione già aperta con l'azione "*hold*". In questo modo, si è tentato di ridurre l'instabilità del modello nelle situazioni in cui esso non sia sicuro di quale sia la mossa migliore da compiere.

Infine, la scelta degli indicatori di analisi tecnica da utilizzare è ricaduta su:

- l'Average Directional Movement Index, considerato sia per i segnali sulla forza del trend forniti dall'ADX vero e proprio, sia per i segnali sulla direzione del trend forniti dall'incrocio tra +DI e -DI;
- il Moving Average Convergence Divergence (MACD);
- il Relative Strength Index (RSI).

### 3.1.2 Strumenti informatici

Per la creazione del modello si è utilizzato il software di programmazione R, ambiente di sviluppo specifico per l'analisi statistica dei dati. Tutti gli script che vengono riportati nel seguito della tesi, sono pertanto relativi allo specifico linguaggio di R<sup>57</sup>.

---

<sup>57</sup> Per un'analisi più approfondita è possibile consultare in Appendice parte del codice sviluppato.

L'utilizzo di questo strumento open source ha permesso di utilizzare alcuni pacchetti e funzioni già create da altri sviluppatori. In particolare, oltre alle comuni librerie scaricabili da CRAN<sup>58</sup>, si vuole sottolineare l'uso de:

- il pacchetto "*quantmod*", attraverso il quale è stato possibile utilizzare la funzione "*getSymbols*", la quale permette di scaricare direttamente le serie storiche dei titoli finanziari all'interno di R: i dati sono stati presi di default da *Yahoo Finance* e sono composti di prezzo di apertura, chiusura, chiusura aggiustata, minimo, massimo e volume;
- il pacchetto "*TTR*", all'interno del quale si trovano le funzioni per calcolare diversi indicatori di analisi tecnica partendo direttamente dalla serie storica scaricata;
- il pacchetto "*ReinforcementLearning*": frutto del lavoro del professore tedesco Nicolas Pröllochs che ne ha condiviso i risultati nel 2019, è ad oggi l'unica libreria in R ove è possibile reperire funzioni di *reinforcement learning* di tipo *model-free*. In particolare, l'utilizzo della funzione "*ReinforcementLearning*" ha permesso di utilizzare un algoritmo di *Q-Learning* che, attraverso il calcolo del *Q-value*, restituisce la strategia migliore per ogni stato analizzato. L'applicazione di questi strumenti ha richiesto un lavoro accurato in fase di formulazione dell'ambiente e dei dati di input (tra cui la definizione della funzione di *reward* e di transizione da stato a stato), ma è indubbio il contributo che ha dato l'utilizzo del pacchetto in menzione.

## 3.2 FASE DI ALLENAMENTO

Vengono di seguito descritte le operazioni svolte nella prima fase dello studio: l'allenamento dell'algoritmo di apprendimento per rinforzo.

### 3.2.1 Preparazione dei dati e analisi tecnica

La primissima fase del lavoro di ricerca ha previsto la preparazione dei dati da utilizzare per il training. Lo scopo di questa fase è stato quello di ottenere una matrice di dati,

---

<sup>58</sup> Acronimo di *Comprehensive R Archive Network*, è una piattaforma online libera ove sono disponibili moduli e pacchetti aggiuntivi al software statistico R.

necessaria per poter utilizzare la funzione “ReinforcementLearning” già menzionata. Questa matrice deve essere composta da 4 vettori colonna che rappresentano:

- lo stato  $s_t$ : indica lo stato attuale in cui ci troviamo;
- l'azione  $a_t$ ;
- lo stato  $s_{t+1}$ : rappresenta lo stato successivo, determinato dallo stato precedente  $s_t$  e dall'azione  $a_t$  presa;
- la ricompensa  $r_t$ , determinata dalle precedenti quantità.

Innanzitutto, si è proceduto a calcolare i quattro indicatori di analisi tecnica scelti, grazie alle rispettive funzioni presenti in R. Ottenuti ADX (all'interno del quale è presente sia l'ADX che il DX, differenza tra +DI e -DI), MACD ed RSI, si è creata una funzione per il calcolo del rendimento giornaliero, da utilizzare come quinto fattore descrittivo dello stato. A questo punto, sono state sviluppate cinque diverse funzioni con lo scopo di sintetizzare i valori numerici degli indicatori e del rendimento giornaliero in un dominio discreto e non più continuo. Sono state stabilite, dunque, diverse etichette, ciascuna di due lettere, per suddividere i valori degli indicatori in classi. L'output delle funzioni è pertanto un vettore, dove al posto del valore dell'indicatore, in ogni istante temporale, si trova un'etichetta di due lettere che descrive il segnale fornito.

L'ADX è un indicatore che oscilla tra 0 e 100 ed indica la forza del trend; più è elevato, più il trend sarà forte. Per questo motivo è stato suddiviso in 4 stati diversi, ognuno indicante una diversa forza del trend:

Tabella 3. Etichette ADX

ADX	ETICHETTA	SIGNIFICATO
<20	WT	Weak Trend
>20 & <50	ST	Strong Trend
>50 & <75	VS	Very Strong trend
>75	ES	Extremely Strong trend

Il valore DX, invece, descrive la differenza fra il +DI e il -DI. Sapendo che l'incrocio tra le due curve rappresenta un segnale preciso di acquisto o vendita, si è stabilito di dare risalto alle situazioni in cui il DX passa da positivo a negativo, e viceversa<sup>59</sup>. Per fare ciò, la funzione ideata restituisce un'etichetta di acquisto/vendita nei giorni in cui è avvenuto l'incrocio,

<sup>59</sup> Questo equivale a dire che le curve +DI e -DI si sono incrociate.



mentre negli altri fornisce una coppia di lettere che indica il mantenimento della posizione *long* o *short*:

Tabella 4. Etichette DX

DX	ETICHETTA	SIGNIFICATO
$DX[t]>0 \ \& \ DX[t-1]<0$	LL	Segnale di acquisto (Long)
$DX[t]<0 \ \& \ DX[t-1]>0$	SS	Segnale di vendita (Short)
$DX[t]>0 \ \& \ DX[t-1]>0$	HL	Mantenimento posizione lunga
$DX[t]<0 \ \& \ DX[t-1]<0$	HS	Mantenimento posizione corta

Stesso ragionamento è valso per il MACD: quando la *differential line* incrocia dal basso verso l'alto la *signal line* si presenta un segnale di acquisto, viceversa un segnale di vendita. Negli altri casi, specialmente quando le curve sono molto distanti, conviene mantenersi nella posizione presa. La suddivisione in classi è stata la seguente:

Tabella 5. Etichette MACD

MACD	ETICHETTA	SIGNIFICATO
$MACD[t]-SL[t]>0 \ \& \ MACD[t-1]-SL[t-1]<0$	LL	Segnale di acquisto (Long)
$MACD[t]-SL[t]<0 \ \& \ MACD[t-1]-SL[t-1]>0$	SS	Segnale di vendita (Short)
$MACD[t]-SL[t]>0 \ \& \ MACD[t-1]-SL[t-1]>0$	HL	Mantenimento posizione lunga
$MACD[t]-SL[t]<0 \ \& \ MACD[t-1]-SL[t-1]<0$	HS	Mantenimento posizione corta

Per quanto riguarda l'RSI, si è deciso di dividere i valori dell'indicatore in 5 sottoclassi. È noto che valori dell'RSI maggiori di 70 indicano una situazione di ipercomprato, mentre valori inferiori a 30 una situazione di ipervenduto. Tuttavia, diversi studi hanno dimostrato che utilizzare queste due soglie fornisce molti falsi segnali e sarebbe preferibile restringere le zone di *overbought* e *oversold* ai valori superiori a 80 e inferiori a 20. In questo caso, si è optato per il mantenimento delle fasce originali dell'RSI, pur tuttavia scindendo ciascuna in due sottocategorie. È l'algoritmo di reinforcement learning a dover stabilire la significatività dei segnali dell'RSI. Le etichette individuate sono le seguenti:

Tabella 6. Etichette RSI

RSI	ETICHETTA	SIGNIFICATO
<15	MS	"Mega"-Sold
>15 & <30	OS	OverSold
>30 & <70	NN	Nessuna indicazione
>70 & <85	OB	OverBought
>85	MB	"Mega"-Bought

Infine, con riferimento ai rendimenti giornalieri, è stato deciso di distinguere 7 fasce. L'intenzione di inserire il rendimento del giorno precedente, all'interno delle etichette da analizzare, deriva dalla volontà di trovare un espediente che riesca a far cogliere all'agente di *reinforcement learning* la presenza di eventuali trend. Questo, insieme ad un fattore correttivo inserito nella funzione di *reward*, dovrebbe evitare che il singolo stato osservato sia avulso dall'intero andamento della serie storica. Di seguito le etichette scelte per i rendimenti giornalieri:

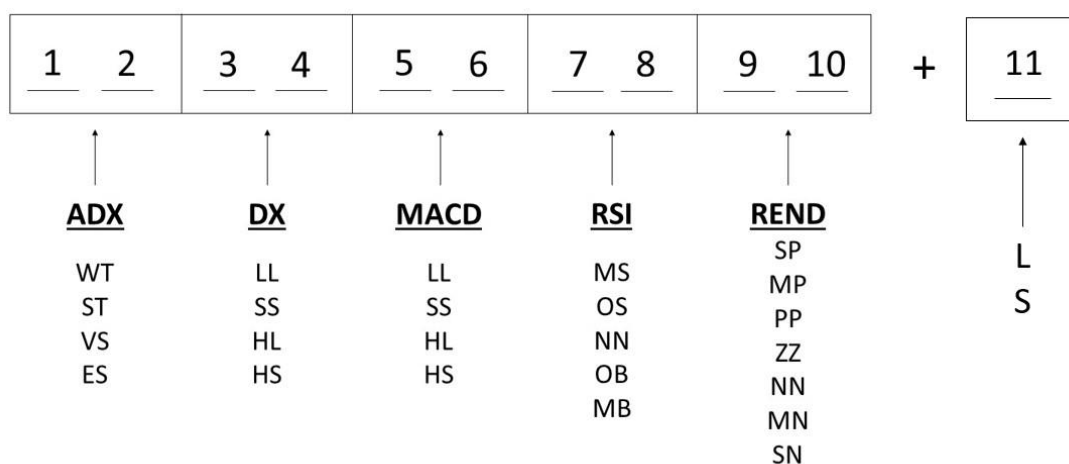
Tabella 7. Etichette per i rendimenti giornalieri

RENDIMENTI	ETICHETTA	SIGNIFICATO
>3%	SP	Super Positivo
>1% & ≤3%	MP	Molto Positivo
>0% & ≤1%	PP	Positivo
0	ZZ	Nulla
>(-1%) & <0	NN	Negativo
>(-3%) & ≤ (-1%)	MN	Molto Negativo
≤(-3%)	SN	Super Negativo

Una volta creati i vettori di etichette degli indicatori, si è provveduto ad unirli in modo tale da ottenere, per ogni giorno finanziario, una stringa di 10 caratteri (2 lettere per ognuno dei 5 indicatori) che descrivono i diversi segnali osservabili.

A questa stringa è stato infine aggiunto un undicesimo carattere, incaricato di descrivere la posizione in cui si trova il portafoglio: potrà insomma essere rappresentato dalla lettera "L", se portafoglio in posizione lunga, oppure dalla lettera "S", qualora il portafoglio sia short. In questo modo, ogni stato è descritto sia dai segnali forniti dagli indicatori, sia da una componente che descrive la posizione al rialzo o ribasso dell'investitore. Quest'ultimo carattere è stato ideato per poter costruire una funzione di transizione da stato a stato, in modo tale da poter legare lo stato attuale, l'azione presa e lo stato successivo.

Grafico 16. Stringa dei segnali degli indicatori



Il vettore delle azioni, invece, è stato creato in maniera casuale: grazie alla funzione “*sample*” di R è stata formata una serie di azioni casuali, scelte tra “B” (*buy*), “S” (*sell*) e “H” (*hold*). La scelta dell’inizializzazione randomica delle azioni è dettata semplicemente dal tipo di funzione utilizzata per lanciare l’algoritmo di *reinforcement learning*: è infatti il sistema ad imparare stato dopo stato quale azione è meglio prendere, in base alla massimizzazione del *Q-value* associato a ciascuna di esse.

Per determinare il valore dello stato successivo, si è stabilito che esso riportasse i primi 10 caratteri dello stato attuale, e che si andasse eventualmente a modificare solamente la posizione del portafoglio, in base all’azione presa. In questo modo l’azione va ad incidere solamente sullo stato del portafoglio e non sui segnali degli indicatori (né sui prezzi), così come avviene nella realtà. Ciò che deve guidare la scelta dell’azione è dunque la serie di caratteristiche presenti in quel momento, ma ciò che determina l’azione stessa ha effetto solo sulla posizione dell’investitore. Ovviamente, da giorno a giorno, le etichette possono variare in qualsiasi modo, a seconda del comportamento degli indicatori e della serie finanziaria. Pertanto, i primi dieci caratteri della stringa degli stati (attuale e successivo) sono indipendenti da riga a riga (ovvero da un tempo all’altro). Quello che lega lo “stato successivo” di una riga con lo “stato attuale” della riga seguente è semplicemente l’undicesimo carattere, la lettera che descrive il portafoglio: questo, ovviamente, perché se è vero che i segnali possono cambiare da un giorno all’altro, non vale lo stesso per lo stato del portafoglio, il quale può essere modificato solamente alla chiusura di mercato in base all’azione scelta.

Questo meccanismo di concatenazione degli stati è stato ottenuto tramite la creazione di una funzione di transizione che serve a descrivere le regole che governano l'environment su cui deve operare l'algoritmo di reinforcement learning. Al grafico 17 è possibile osservare con degli esempi pratici quanto finora esposto.

Grafico 17. Transizione da stato a stato e da riga a riga

STATO ATTUALE	AZIONE	STATO SUCCESSIVO
ES HL HL OB MP <b>S</b>	<b>B</b>	ES HL HL OB MP <b>L</b>

Il portafoglio all'inizio è in posizione short. In base ai segnali osservati, l'azione scelta è buy; il portafoglio pertanto diventerà long.

ST SS HL NN PP <b>L</b>	<b>S</b>	ST SS HL NN PP <b>S</b>
VS HS SS OS MN <b>S</b>	<b>H</b>	VS HS SS OS MN <b>S</b>

Come accennato in precedenza, l'ultima colonna della matrice dei dati da creare riguarda la ricompensa che il sistema riceve in base all'azione presa. Rielaborando una funzione di ricompensa proposta negli studi di Wang et al. (2017), si è deciso di utilizzare la seguente formula di reward:

$$r_t = \pm \left( \frac{p_{t+1} - p_t}{p_t} \right) * \left( \frac{p_t}{p_{t-200}} \right)$$

dove  $p$  indica il prezzo di chiusura.

Come si può notare, la ricompensa è composta dal rendimento giornaliero, calcolato dal momento dell'azione presa sino al giorno dopo, moltiplicato per un fattore correttivo che tiene conto di un eventuale trend di lungo periodo<sup>60</sup>. In questo modo, si è cercato di indirizzare l'apprendimento, non solo verso la scelta della migliore azione possibile per quel

<sup>60</sup> Come si potrà notare nel codice riportato in Appendice, l'introduzione di questo fattore che utilizza il prezzo di 200 periodi prima, ha richiesto un adattamento dell'intero set di dati. Per tale motivo, è stato necessario partire con gli indicatori al tempo  $t=201$ , in modo tale da potervi affiancare un reward calcolato in questo modo.

singolo giorno, ma anche verso la comprensione di eventuali andamenti di medio-lungo periodo. I test effettuati, con e senza il fattore correttivo, hanno confermato l'utilità di questo rapporto all'interno della funzione di *reward*.

Il segno della ricompensa è condizionale alla posizione del portafoglio: se la posizione è lunga, allora il segno sarà positivo, viceversa in caso di posizione corta. Il senso è di immediata comprensione: se compro ed il prezzo scende, il rendimento sarà negativo e dunque il *reward* dovrà essere anch'esso negativo; tuttavia se vendo allo scoperto ed il prezzo scende, il rendimento dell'operazione dovrà essere positivo, dunque serve invertire il segno. Inoltre, è stata aggiunta una componente penalizzante da applicare nei momenti in cui si invia degli ordini di mercato, dunque nelle situazioni in cui si cambia la posizione del portafoglio. Questa scelta nasce sia con l'intento di considerare e limitare i costi di commissione applicati dai broker al momento dell'esecuzione degli ordini di mercato, sia, soprattutto, per evitare l'eccessiva operatività del sistema: infatti, si è potuto constatare nel corso delle diverse prove, che, quando il trading system entra ed esce dal mercato in modo spasmodico, si ottengono sempre perdite consistenti. Questo fatto si spiega considerando che, probabilmente, quando il sistema non è certo dell'azione da prendere, i segnali di analisi tecnica diventano chiari solamente in ritardo rispetto all'effettiva variazione dei prezzi. È uno dei limiti che viene imputato agli strumenti di analisi tecnica, e con l'espedito della penalità si è appunto cercato di limitarne gli effetti. In particolare, poi, la penalità è stata distinta in penalità *long* e penalità *short*. Il motivo di questa scelta è legato ai titoli presi in esame: gli indici finanziari, nel medio-lungo periodo, si distinguono per il loro trend positivo, per cui, dovendo considerare come orizzonte temporale dieci anni solari, si è voluto differenziare il caso *long* da quello *short*. Penalizzando leggermente di più le operazioni *short*, si è visto che i rendimenti finali miglioravano, e dunque si è mantenuto questo aspetto. Il *reward* ottenuto quando si passa da una posizione corta ad una lunga è dato da:

$$r_t = \left[ \frac{p_{t+1} * (1 - \text{penalità long}) - p_t}{p_t} \right] * \left( \frac{p_t}{p_{t-200}} \right)$$

Viceversa, la ricompensa ricevuta quando si passa da una posizione lunga ad una corta è:

$$r_t = - \left[ \frac{p_{t+1} * (1 + \text{penalità short}) - p_t}{p_t} \right] * \left( \frac{p_t}{p_{t-200}} \right)$$

Definita anche la colonna dei *reward*, la matrice dei dati è pronta per essere utilizzata nella fase di apprendimento vero e proprio.

### 3.2.2 Training con apprendimento per rinforzo

Una volta creata la matrice dei dati da sottoporre all'apprendimento, è stato possibile lanciare la funzione di *reinforcement learning*. Quest'ultima, oltre alla suddetta matrice, necessita dell'inserimento dei parametri, esposti nel corso del primo capitolo, che regolano il processo di apprendimento per rinforzo: il *learning rate*  $\alpha$ , il tasso di sconto  $\gamma$  ed infine l' $\epsilon$ -greedy<sup>61</sup>. Stabiliti i parametri ed il numero di iterazioni da far compiere al sistema, si è proceduti al calcolo del modello. La risultante del processo di *reinforcement learning* è un insieme di dati all'interno del quale è possibile trovare diverse informazioni utili, prima fra tutti la *policy* ottimale stimata. Inoltre, la funzione restituisce l'elenco di tutti gli stati osservati, il *Q-value* associato ad ogni possibile azione in ogni possibile stato, la serie dei *reward* ottenuti durante l'apprendimento ed infine la deviazione standard rispetto alle diverse iterazioni compiute. Sono informazioni molto utili, che possono guidare la ricerca verso la strada migliore. In questo caso, ciò che è stato utilizzato è la *policy* ottimale imparata; lo sfruttamento dei dati contenuti nella serie dei *Q-value*, tuttavia, rappresenta una possibile implementazione futura che verrà proposta nel corso del seguente capitolo. Dalla *policy* imparata è stata infine creata una matrice a due colonne, una con gli stati ed una con le relative azioni migliori da prendere. Questo passaggio si è rilevato necessario solo per una questione di programmazione, dal momento che non vi è stata alcuna rielaborazione effettiva tra l'output della funzione di *reinforcement learning* e la matrice appena menzionata.

A questo punto il modello allenato sui dati di un singolo titolo finanziario è pronto. Tuttavia, si è deciso di sviluppare ulteriormente le conoscenze dell'agente di *reinforcement learning*.

A tal fine, è stato possibile aggiornare e migliorare il modello creato rilanciando la funzione già utilizzata, con la specifica condizione che non si ripartisse da zero come prima, ma piuttosto dalle informazioni già apprese. Seguendo quest'ottica, sono stati considerati

---

<sup>61</sup> Rappresenta la probabilità che il sistema di RL compia un'azione casuale piuttosto che scegliere l'azione migliore appresa. Indica il tasso di esplorazione alternativa alla *policy* ottimale imparata.

due percorsi. Una prima via ha previsto la reiterazione del modello per quattro volte utilizzando di volta in volta una matrice di dati diversa, distinta dalle altre solamente per la differente inizializzazione randomica delle azioni. In altre parole, sono stati creati quattro diversi vettori di azioni da utilizzare per creare le matrici con i meccanismi in precedenza descritti. I risultati ottenuti con questo metodo si sono rivelati insoddisfacenti, perciò si è deciso di non proseguire oltre. Dietro questo insuccesso, probabilmente, vi è un problema di sovra-adattamento che fa sì che la strategia imparata sia troppo dipendente dai dati del particolare allenamento. Riutilizzare una sequenza identica di stati, ove le ricompense insistono sugli stessi prezzi, porta con ogni probabilità il modello a “perdere la bussola” e a non cogliere più la strada migliore in contesti diversi da quelli già osservati.

La seconda via esplorata ha previsto, invece, la reiterazione del modello non più su matrici di dati dello stesso titolo finanziario, bensì derivanti da indici finanziari diversi. Quello che si è realizzato, perciò, è stato un processo di allenamento a catena, o, più propriamente, a *step*. Dapprima il modello è stato allenato sui dati dell’S&P500; successivamente è stato aggiornato utilizzando il set di segnali del Dow Jones; poi con quelli del Nasdaq, e così via per tutti e nove gli indici in esame. In questo modo, sono stati sottoposti al modello in totale 51.476 giorni finanziari. Il risultato è stato un sistema in grado di riconoscere un numero maggiore di combinazioni di etichette degli indicatori, e soprattutto di prendere le proprie decisioni con maggior accuratezza.

### **3.3 FASE DI TEST**

Ottenuta la politica migliore grazie all’allenamento sulle nove serie di segnali, è stata creata una struttura di funzioni per l’ottenimento di diversi risultati, necessari al calcolo finale del rendimento e del profitto totale ottenuti nel periodo di test, ma altresì utili alla valutazione di valori intermedi come, ad esempio, il profitto di ogni singola operazione di mercato compiuta.

Il primo passaggio è stato quello di replicare la preparazione dei dati compiuta nella fase di apprendimento: scaricata la serie finanziaria per il periodo 2010-2019, sono stati calcolati i diversi indicatori di analisi tecnica e successivamente le etichette, grazie alle funzioni create in precedenza. Il risultato è stato dunque un nuovo set di segnali sul quale poter applicare la politica ottimale stimata.

La prima funzione creata è servita da un lato a riprodurre il funzionamento di transizione dell'ambiente già visto, in modo tale da poter legare gli stati con le azioni intraprese, e dall'altro a calcolare il profitto uniperiodale derivante dalla posizione in essere del portafoglio. Questa quantità è stata definita semplicemente come la variazione di prezzo giornaliera in seguito all'azione intrapresa<sup>62</sup>. Il risultato è stato dunque una prima matrice composta da 4 colonne: lo stato attuale nel test dataset, l'azione ottimale da prendere, lo stato successivo derivante dallo stato iniziale e dall'azione assunta, ed infine il profitto giornaliero. In questa fase del lavoro, è stato deciso che il modello dovesse assumere come azione quella imparata nel corso del training: tuttavia, è possibile che il sistema venga a contatto con stati mai osservati prima; per questo motivo, è stato espressamente previsto che venisse presa un'azione solo qualora la stringa di segnali fosse già conosciuta, ed in caso contrario che si mantenesse la posizione invariata. Un possibile *upgrade* futuro potrebbe essere quello di riallenare il modello sul set di dati utilizzati per il test ed analizzare eventuali miglioramenti nelle performance.

Di default è stato assunta una prima posizione del portafoglio lunga, che poi verrà eventualmente corretta dal modello alla prima azione possibile, qualora fosse discorde da ciò che richiederebbero i segnali degli indicatori.

La seconda funzione creata, a partire dagli stati del test, dal profitto uniperiodale ottenuto e dai prezzi di chiusura giornalieri, è andata a calcolare tre vettori colonna: il prezzo di acquisto per ogni operazione aperta, il numero progressivo delle operazioni compiute ed infine quello che è stato definito il "profitto cumulato per operazione". Quest'ultimo valore corrisponde alla cumulata dei profitti giornalieri all'interno di una stessa operazione: il meccanismo utilizzato prevede dunque che venga sommato, ogni giorno, il profitto di giornata alla somma del giorno precedente e, qualora si passi ad una posizione nuova, il profitto cumulato riparta da zero. Questo risultato intermedio ha permesso di analizzare l'andamento delle performance operazione per operazione, così da discutere quali potessero essere dei limiti del modello.

A questo punto, è stata ideata un'ultima funzione allo scopo di calcolare un vettore colonna dove venisse riportato semplicemente il profitto finale di ogni operazione: molto

---

<sup>62</sup> Ovviamente, si è provveduto ad invertire il segno nei casi di operazioni short.



semplicemente il sistema è andato a riportare i profitti cumulati alla chiusura di ogni operazione.

In conclusione, è stata creata una matrice di sintesi contenente tutti i diversi aspetti calcolati fino a questo momento nella fase di test (si veda tabella 8).

Tabella 8. Esempio di dati di sintesi della fase di test

	StateTest	OptAction	NextStateTest	Prof_Unip	Px_Acq	N_operaz	Prof_Cumulato	Prof_operazione
72	STHLHSNNMPL	H	STHLHSNNMPL	14.130005	NA	1	5.409912	NA
73	STHLHSNNMPL	H	STHLHSNNMPL	-6.039917	NA	1	-0.630005	NA
74	STHLHSNNNML	H	STHLHSNNNML	-34.170044	NA	1	-34.800049	-34.800049
75	STHLHSNNSNL	S	STHLHSNNSNS	-3.079956	1063.11	2	-3.079956	-3.079956
76	STHLHSNNPPS	B	STHLHSNNPPL	-9.449951	1066.19	3	-9.449951	NA
77	STHLHSNNNML	H	STHLHSNNNML	13.780030	NA	3	4.330079	NA
78	STHLHSNNMPL	H	STHLHSNNMPL	-2.390015	NA	3	1.940064	NA
79	STHLHSNNNML	H	STHLHSNNNML	10.339966	NA	3	12.280030	NA
80	WTHLHSNNPPL	H	WTHLHSNNPPL	-2.959961	NA	3	9.320069	NA
81	WTHLHSNNNML	H	WTHLHSNNNML	19.359985	NA	3	28.680054	NA
82	WTHLLLNNMPL	H	WTHLLLNNMPL	4.640015	NA	3	33.320069	NA
83	WTHLHLNPPPL	B	WTHLHLNPPPL	7.239990	NA	3	40.560059	NA
84	WTHLHLNPPPL	B	WTHLHLNPPPL	2.420044	NA	3	42.980103	NA
85	WTHLHLNPPPL	B	WTHLHLNPPPL	-1.160034	NA	3	41.820069	NA
86	WTHLHLNPNML	H	WTHLHLNPNML	-13.410034	NA	3	28.410035	28.410035
87	WTHLHLNPNML	S	WTHLHLNPNMS	-10.640014	1094.60	4	-10.640014	NA
88	WTHLHLNPPPS	S	WTHLHLNPPPS	2.300049	NA	4	-8.339965	-8.339965
89	STHLHLNPNNS	B	STHLHLNPNML	1.550049	1102.94	5	1.550049	NA

Ultimo passaggio è stato quello di calcolare i profitti e rendimenti corretti per i costi di transazione, ovvero le commissioni applicate ad ogni ingresso nel mercato. Quest'ultime sono state calcolate in forma percentuale sul prezzo d'acquisto (o vendita nel caso di operazioni *short*). Sono stati così ottenuti:

- i rendimenti per operazione:

$$r = \frac{\text{profitto operazione} - (\text{prezzo d'acquisto} * \text{commissione})}{\text{prezzo d'acquisto}};$$

- i profitti netti per operazione:

$$\text{profitto netto} = \text{profitto operazione} - (\text{prezzo d'acquisto} * \text{commissione});$$

- il profitto totale del test:

$$\text{profitto totale netto} = \sum \text{profitti netti per operazione};$$

- il rendimento totale del test:

$$\text{rendimento test} = \frac{\text{profitto totale netto}}{\text{prezzo d'acquisto al tempo } 1};$$

- la cumulata del profitto all'aumentare del numero di operazioni, ottenuta grazie ad una funzione scritta ad hoc.

La procedura fin qui descritta è stata applicata in ugual modo a tutti e nove gli indici finanziari per poter valutare le performance del modello stimato.

## CAPITOLO IV. RISULTATI FINALI

In questo capitolo vengono esposti ed analizzati i risultati ottenuti con l'applicazione del sistema ideato. La trattazione delle performance prevede la descrizione dei parametri scelti, l'analisi dei rendimenti e dell'operatività del trading system creato, le osservazioni ed infine la proposta di alcune possibili implementazioni future.

### 4.1 RISULTATI OTTENUTI

#### 4.1.1 Scelta dei parametri

Prima di passare alla descrizione dei risultati veri e propri, è importante aprire una parentesi sulla scelta dei parametri del modello. Nel corso del lavoro di ricerca, sono state svolte molteplici prove di training con altrettante combinazioni di parametri diversi. In generale, si è andato a modificare sia i parametri propri del modello di *reinforcement learning* ( $\alpha$ ,  $\gamma$ ,  $\epsilon$ ), sia i valori delle penalità relative ai cambi di posizione. I risultati che vengono esposti nel corso del presente capitolo, si riferiscono solamente ai parametri migliori, ovvero quelli che hanno permesso di ottenere buone performance. Al fine di rendere la trattazione snella ed efficace, vengono dunque omessi i risultati derivanti dall'applicazione di parametri poco performanti. Bisogna in ogni caso sottolineare che la modifica dei parametri, anche minima, può provocare grandi differenze in termini di performance finale. In particolare, la scelta del tasso di apprendimento si è dimostrata fondamentale: avendo a disposizione una mole di dati consistente, un valore  $\alpha$  troppo elevato rischia di creare problemi di *overfitting* nel modello; quello che si è notato, dunque, è che risulta più proficuo utilizzare un *learning rate* contenuto, che anteponga l'accuratezza delle strategie all'ampiezza delle azioni conosciute. Per questo motivo, l'applicazione del valore standard pari a 0,10, si è rivelata inadatta e si è preferito utilizzare  $\alpha=0,05$ .

Il tasso di sconto  $\gamma$  è stato fissato al livello dello 0,9. Sono stati svolti diversi esperimenti con parametri inferiori e, a parità di condizioni, le performance migliori sono state ottenute con il valore menzionato. In ogni caso, non si è mai ritenuto di porre il parametro sotto lo 0,5 perché l'intenzione dichiarata è stata quella di far cogliere al sistema eventuali relazioni

di lungo periodo: stabilire un tasso di sconto elevato è stata una scelta votata a conferire un maggior peso ai risultati futuri attesi.

Infine, si è scelto un valore di  $\epsilon$  pari a 0,1. Ad onor del vero, stabiliti i precedenti parametri, si è notata scarsa differenza di performance fra l'utilizzo di  $\epsilon=0,1$  e  $\epsilon=0,05$ . Ciò, tuttavia, non vale se si pone  $\epsilon=0,01$ : è un tasso di esplorazione casuale troppo basso per produrre effetti positivi nell'abilità previsiva del modello. Di contro, non è stata presa in considerazione la scelta di un  $\epsilon$  maggiore al 10% perché ritenuto esagerato: chiedere al sistema di prendere un'azione casuale, al posto di quella ottimale imparata, in più del 10% dei casi appare incoerente con il processo di apprendimento per rinforzo ideato.

Prima di poter utilizzare il modello, si è dovuto stabilire anche il valore delle penalità per l'ingresso nel mercato da utilizzare nella funzione di *reward*. Il valore della penalità nel caso *short* è rimasto sempre stabile a 0,01, mentre è cambiato di volta in volta la penalità *long*, andando, rispettivamente, ad ampliare o restringere il divario tra i due casi. In particolare, la scelta finale è stata quella di utilizzare un valore dello 0,009. Si è potuto notare come l'utilizzo di due penalità con differenza maggiore provochi dei miglioramenti di performance in quei titoli che presentano maggior volatilità ed un trend crescente meno forte e netto degli altri titoli; quest'ultimi, invece, manifestano rendimenti inferiori. Al contrario, l'utilizzo di penalità quasi equivalenti per l'ingresso in posizione *long* o *short*, determina dei risultati talvolta addirittura negativi nei titoli con maggior volatilità. Ciò è imputabile al fatto che tali indici hanno subito, nel corso dei dieci anni di test, diversi movimenti di prezzo inattesi: non distinguere le penalità equivale, in questi casi, a non dare al modello "una protezione" all'alta volatilità presente nel mercato; quando il sistema sa che deve prediligere le azioni long sta sostanzialmente evitando di aprire molte più posizioni che lo porterebbero in perdita. Per questo stesso motivo tali titoli evidenziano una posizione netta del trading system automatico che diventa positiva solo dopo molte operazioni, anche utilizzando i migliori parametri trovati. Questo comportamento si ipotizza che sia dovuto anche alla presenza di fasi laterali nell'andamento del prezzo, le quali non permettono di determinare delle azioni certe da prendere.

La scelta del numero di iterazioni da far compiere all'algorithmo di *reinforcement learning* è in linea con quanto fin qui esposto. Visti i problemi di sovra-adattamento ai *dataset* di training, si è deciso di utilizzare un'unica iterazione dei dati per ogni nuova matrice

sottoposta all'apprendimento. Il modello finale, dunque, è il risultato di 9 iterazioni, una per ogni indice utilizzato.

Infine, per quanto riguarda la commissione da considerare al momento dell'apertura di una posizione, è stato scelto un tasso dello 0,05% sul prezzo iniziale.

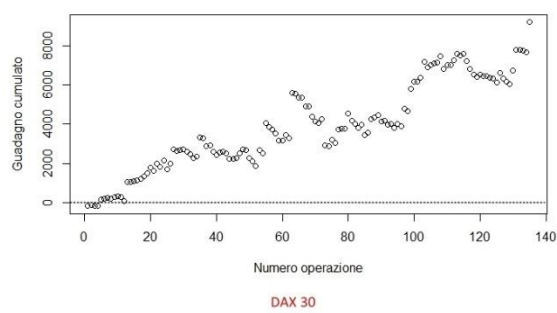
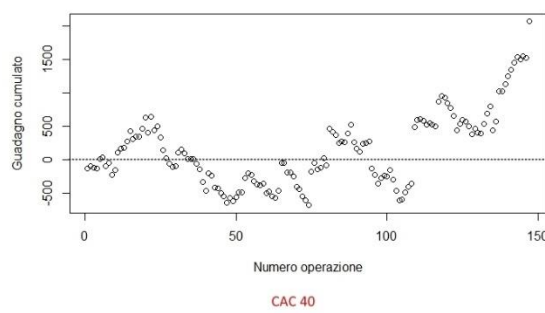
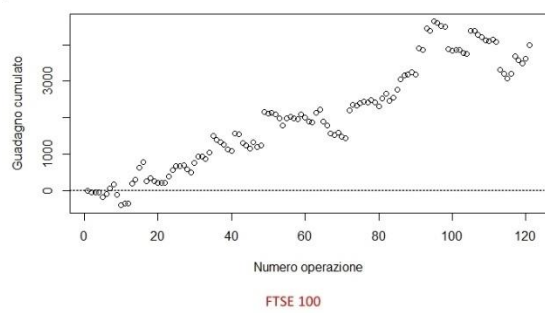
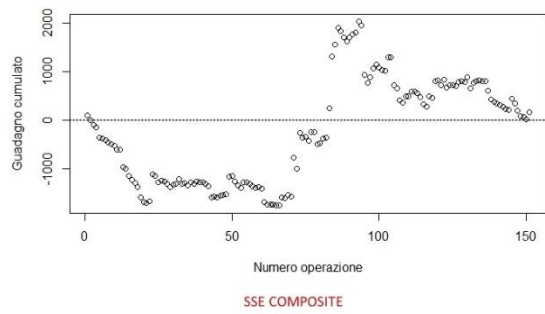
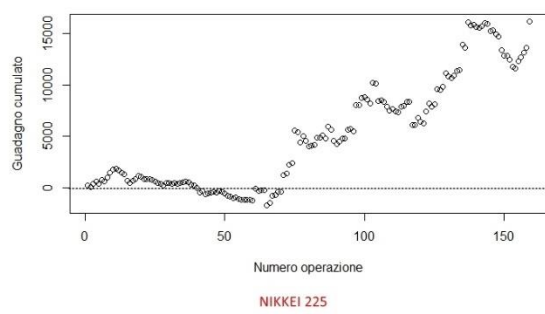
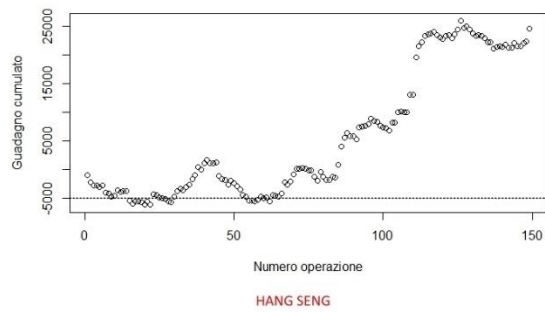
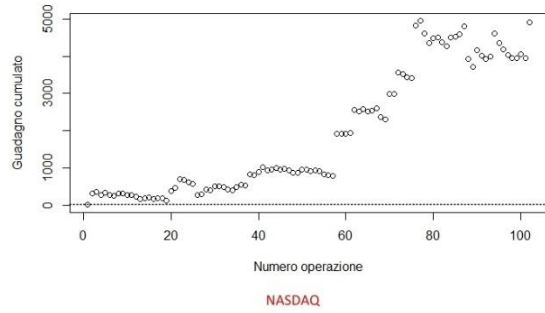
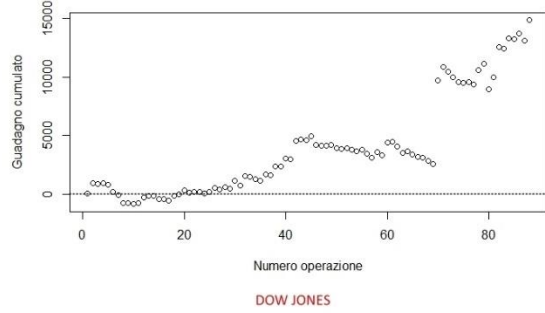
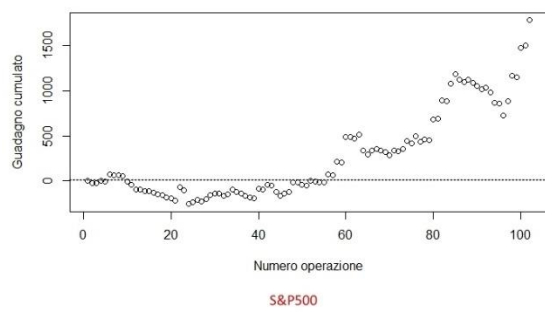
#### *4.1.2 Analisi dei rendimenti*

Il modello creato ha permesso di ottenere buone performance di investimento, specialmente con i titoli che non hanno subito variazioni di prezzo improvvise, nette e di grandi dimensioni: in generale, i rendimenti migliori si sono registrati nei mercati statunitensi, mentre in quello cinese di Shanghai e in quello di Parigi le performance lungo il periodo di test non sono state sempre ottime. Quel che risalta osservando l'andamento del profitto cumulato nel corso delle operazioni (si veda il grafico 18) è che, nella maggior parte dei casi, ci vogliono diverse operazioni prima che la funzione di guadagno cumulata riesca a staccarsi positivamente dal livello zero. Tutto ciò indica che è necessario considerare l'evenienza di eventuali ricapitalizzazioni iniziali del portafoglio, perché è presente la possibilità di perdite.

In due casi (SSE Composite Index e CAC40) le performance sono da considerarsi negative: nonostante il risultato finale sia comunque positivo, lungo i 10 anni di test buona parte del tempo il portafoglio è rimasto in una situazione di forte perdita. Questo risultato è emerso in tutte le prove attuate; l'utilizzo di parametri differenti non determina un miglioramento delle performance e questo ha spinto a chiedersi il motivo di tale comportamento andando a studiare l'andamento degli indici finanziari nel corso degli anni. Si è ipotizzato che le performance negative siano da imputarsi principalmente alle diverse fasi laterali attraversate dagli indici, le quali, evidentemente, producono molteplici falsi segnali che traggono in inganno il trading system creato.

In tre casi (NASDAQ, FTSE100 e DAX30) la posizione netta non è quasi mai andata in rosso, ma ci sono state solo operazioni profittevoli. È da considerarsi un ottimo risultato: in particolare, questa performance molto positiva non si è riscontrata utilizzando altri parametri diversi da quelli menzionati, il che sottolinea la bontà della scelta presa.

Grafico 18. Profitto cumulato dei nove indici durante il test ( $\alpha=0.05$ ,  $\gamma=0.9$ ,  $\epsilon=0.1$ )



Esaminando i grafici, è possibile osservare come spesso vi siano dei salti di livello rilevanti da operazione a operazione: questi sono casi in cui i segnali degli indicatori sono talmente chiari che il sistema decide di rimanere a lungo nella stessa posizione riuscendo a cogliere molto bene il trend in atto ed ottenendo ottimi risultati.

In Tabella 9 è possibile osservare i rendimenti finali ottenuti dal trading system applicato sui diversi titoli analizzati.

Tabella 9. Rendimenti totali ( $\alpha=0.05$ ,  $\gamma=0.9$ ,  $\epsilon=0.1$ )

INDICE	RENDIMENTO FINALE
S&P500	162%
DOW JONES	147%
NASDAQ	225%
HANGSENG	111%
NIKKEI225	163%
SSE COMPOSITE	5,6%
FTSE100	77%
CAC40	53%
DAX	135%

Per completezza, vengono riportati i risultati ottenuti con penalità differenti da quelle stabilite in precedenza (Tabella 10).

Tabella 10. Rendimenti totali con penalità diverse

INDICI	RENDIMENTI TOTALI			
	Penalità Long = 0,0095	Penalità Short = 0,01	Penalità Long = 0,0085	Penalità Short = 0,01
S&P500	176%		105%	
DOW JONES	146,00%		142,00%	
NASDAQ	135%		179%	
HANGSENG	68%		63%	
NIKKEI225	103%		26%	
SSE	63,0%		19,6%	
FTSE100	20%		103%	
CAC40	-53%		34%	
DAX	47%		132%	

Talvolta, i rendimenti finali sono stati maggiori di quelli già visti; tuttavia, bisogna sottolineare che quest'ultimi non sono l'unico fattore da considerare per la valutazione delle performance. Infatti, è importante andare a vedere come si comportano i rendimenti nel corso del periodo testato e quale sia la volatilità propria della serie del profitto cumulato. Analizzati questi aspetti, si è confermata la scelta di una penalità *long* pari a 0,009 ed una *short* uguale a 0,01.

### 4.1.3 Analisi dell'operatività

Nel corso del test, il sistema si è distinto per un'operatività abbastanza lenta, probabilmente a causa dei diversi fattori utilizzati per catturare le relazioni di lungo periodo. Dunque, l'applicazione del trading system ideato non si addice ad utilizzi ad alta frequenza, come per esempio gli investimenti *intraday*, anche perché, in effetti, la composizione dei dati è semplicemente giornaliera. La media dei dati utilizzati per i test di ciascun indice è intorno ai 2550 giorni finanziari. Il fatto che le serie non abbiano lo stesso identico numero di giorni finanziari, pur avendo origine dallo stesso periodo temporale di dieci anni, trova spiegazione nel diverso numero di giornate in cui gli indici sono rimasti chiusi, sia questo per festività diverse da Paese a Paese, sia perché mantenuti chiusi dalle autorità di gestione della Borsa per turbolenze eccessive nei mercati.

Come è possibile osservare in Tabella 11, il numero medio di operazioni compiute nel corso dei dieci anni si attesta intorno a 130, di cui circa il 60% sono posizioni lunghe. La durata media di ciascuna operazione è di circa 20 giorni finanziari, pressoché un mese solare. Si è potuto constatare che il numero di operazioni è stato maggiore nelle applicazioni su indici finanziari che presentano dei trend meno decisi: il sistema in questi casi cambia posizione più frequentemente perché il *Q-value* delle scelte non è sufficientemente elevato da determinare una strategia certa. Di conseguenza, la durata media delle operazioni si riduce, assestandosi intorno ai 16 giorni finanziari.

Tabella 11. Operatività media del trading system  
( $\alpha = 0.05$ ,  $\gamma = 0.9$ ,  $\varepsilon = 0.1$ , penalità Long = 0.009, penalità Short = 0.01)

INDICI	DATI TOTALI	NUMERO OPERAZIONI	DURATA MEDIA PER OPERAZIONE
S&P500	2567	102	25,17
DOW JONES	2567	88	29,17
NASDAQ	2567	102	25,17
HANG SENG	2504	149	16,81
NIKKEI 225	2491	159	15,67
SSE COMPOSITE	2473	151	16,38
FTSE 100	2570	121	21,24
CAC 40	2610	147	17,76
DAX 30	2585	157	16,46
<b>Valori medi:</b>	<b>2548,22</b>	<b>130,67</b>	<b>20,42</b>



In Tabella 12 vengono riportati anche i risultati ottenuti con penalità diverse da quelle prescelte. Si noti che, quando le penalità vanno ad avvicinarsi determinando una quasi parità di trattamento tra operazioni lunghe e corte, il numero di operazioni aumenta considerevolmente. La spiegazione è intuitiva: una penalizzazione minore, in termini relativi, per l'apertura di posizioni *short* consente al sistema di aumentare la propria operatività. Applicando una penalità long dello 0.0095, ad esempio, il numero di operazioni aumenta dell'85% rispetto al caso precedente, mentre la durata media delle operazioni si dimezza.

Utilizzando, invece, una penalità *long* inferiore, ovvero aumentando la disparità tra casi lunghi e corti, l'operatività non cambia molto: il numero medio di operazioni compiute è di circa 135, mentre la durata media per operazione rimane intorno ai 20 giorni finanziari.

Tabella 12. Operatività media dei trading system alternativi

INDICI	PENALITA' LONG = 0,0095		PENALITA' LONG = 0,0096	
	NUMERO OPERAZIONI	DURATA MEDIA PER OPERAZIONE	NUMERO OPERAZIONI	DURATA MEDIA PER OPERAZIONE
S&P500	193	13,30	101	25,42
DOW JONES	199	12,90	105	24,45
NASDAQ	185	13,88	111	23,13
HANG SENG	299	8,37	153	16,37
NIKKEI 225	249	10,00	165	15,10
SSE COMPOSITE	277	8,93	169	14,63
FTSE 100	225	11,42	115	22,35
CAC 40	271	9,63	155	16,84
DAX 30	275	9,40	145	17,83
<b>Valori medi:</b>	<b>241,44</b>	<b>10,87</b>	<b>135,44</b>	<b>19,57</b>

## 4.2 CONSIDERAZIONI

Quanto finora visto stimola alcune riflessioni. In primis, è importante sottolineare i diversi andamenti di prezzo degli indici. Queste differenze determinano delle performance diverse da parte del trading system ideato, poiché riesce a adattarsi, e dunque a prevedere, meglio le serie storiche con decisi trend crescenti di lungo periodo. Questo aspetto è sicuramente frutto delle ipotesi assunte e della scelta dei parametri da utilizzare, altresì selezionati proprio per assecondare la tendenza comune degli indici finanziari a crescere di valore nel lungo periodo. Il set di parametri composto da  $\alpha=0.05$ ,  $\gamma=0.9$ ,  $\epsilon=0.1$ , penalità\_Long=0.009 e penalità\_Short=0.01, ha permesso di ottenere i risultati migliori in termini di affidabilità

del trading system: nella maggior parte dei casi, infatti, il portafoglio assume una posizione netta positiva per quasi tutto il periodo di test. Tuttavia, come si è visto, alcuni dei titoli esaminati hanno ottenuto rendimenti finali migliori con l'utilizzo di penalità diverse. Ciò fa pensare che l'utilizzo di un set di parametri universale può risultare limitante per le performance finali. Una valida soluzione potrebbe essere quella di calcolare diverse policy ottimali, tramite altrettanti training aventi parametri differenti, da applicare poi ai titoli finanziari in base alle situazioni di mercato (fasi laterali, alta volatilità, difficoltà di affermazione di un trend di lungo periodo ecc.).

#### 4.2.1 Implementazioni future

Visti i risultati e viste le osservazioni poste, esistono alcune implementazioni future che potrebbero migliorare il sistema oggetto di studio.

Prima fra tutti, si potrebbe modificare la fase di apprendimento andando ad inserire un fattore di correzione dell'inflazione: ripulire le serie finanziarie in modo tale da avere solo valori reali potrebbe determinare risultati diversi.

In secondo luogo, per ovviare al problema dell'incertezza del sistema nel prendere alcune decisioni, le quali poi portano a cambiare velocemente più posizioni (e ad ottenere quasi sempre delle perdite), si potrebbe stabilire una regola di trading alternativa che preveda di cambiar posizione solamente in seguito ad un doppio segnale di acquisto o vendita. In sostanza, un sistema che apre una posizione *long* solamente quando per due stati consecutivi l'azione ottimale è "*buy*", e viceversa una posizione *short* quando osserva due azioni ottimali "*sell*" consecutive. L'introduzione di questa misura prevederebbe anche il rifiuto dell'ipotesi, qui assunta, che l'investitore sia sempre all'interno del mercato, consentendo anche momenti di neutralità e attesa. Questa nuova operatività necessita di diversi test e prove, perché il rischio è quello di ritardare ancor di più il tempismo del sistema nell'entrare nel mercato al momento giusto. In più, troppe misure conservative potrebbero alla fin fine determinare il risultato opposto a quello sperato.

Una misura che nasce con lo stesso scopo potrebbe essere quella di accantonare l'idea di contrattare sempre e solo un'azione del titolo a favore di un sistema che, in base al *Q-value* associato ad ogni azione ottimale, determini il numero di azioni da sottoscrivere in ogni

operazione. Quest'idea, ripresa dallo studio di *Jeong & Kim (2018)*, potrebbe rivelarsi molto interessante e redditizia.

Infine, potrebbe determinare delle performance migliori anche l'applicazione di un semplice limite "*stop loss*", sia in fase di apprendimento, che di test, il quale, da un lato limita le perdite nei momenti in cui le scelte del sistema non sono buone, e dall'altro guida il sistema stesso a cambiare posizione quando sta subendo perdite eccessive.



## CONCLUSIONI

L'obiettivo di questa tesi è stato in primo luogo analizzare la dottrina del *machine learning*, estranea agli studi finanziari finora condotti, e, successivamente, di ideare un sistema che potesse sfruttare le potenzialità dell'intelligenza artificiale a fini d'investimento.

In particolare, si è partiti con un'attenta disamina della scienza dell'apprendimento automatico, andando a coglierne le diverse procedure, tipologie e applicazioni. Solo dopo aver studiato l'esatto funzionamento di queste metodologie, con particolare attenzione al *reinforcement learning*, è stato proposto un personale sistema automatico d'investimento finanziario. Si è osservato che, specialmente negli ultimi dieci anni, la ricerca scientifica ha fornito molteplici studi sull'utilizzo di tecniche di intelligenza artificiale al servizio del mondo finanziario. L'applicazione dei metodi per rinforzo, oltre che delle reti neurali, mostra potenzialità giganti per l'analisi dei mercati finanziari perché permette di catturare e modellizzare connessioni che la mente umana non è in grado di cogliere.

L'utilizzo del *reinforcement learning* si dimostra molto performante nei compiti di *decision making*, ma forse un po' meno in quella di apprendimento delle caratteristiche dell'*environment*. Per tale motivo, volendo in questo lavoro di ricerca concentrarsi solo sulle funzionalità di codesta classe di algoritmi, è stato ideato un meccanismo di etichettatura dei dati provenienti dagli indicatori tecnici con il fine di agevolare il successivo apprendimento. In altre parole, è stato sfruttato un approccio supervisionato nella fase d'inizializzazione di un processo che, tuttavia, è di apprendimento per rinforzo.

I risultati hanno nel complesso soddisfatto le attese, mettendo in luce pro e contro delle scelte prese in fase di progettazione. Di positivo si è osservata la capacità del trading system di stimare con discreta accuratezza la presenza di trend di medio periodo, compito non semplice in sistemi di questo tipo, come si è potuto notare nei diversi lavori presenti in letteratura. Si è altresì visto come l'introduzione di alcuni fattori quali le penalità diverse per l'apertura di operazioni lunghe o corte, piuttosto che una funzione di *reward* pesata per un rapporto di trend di lungo periodo, ha determinato gli effetti sperati. In particolare, è stato rispettato l'obiettivo di includere l'effetto dei costi di transazione, spesso non considerati nella letteratura economica sebbene siano influenti sui risultati finali.

In generale, il sistema creato ha avuto un'operatività di medio periodo, a metà fra trading system ad alta frequenza e FTS per portafogli pensionistici, con una durata media per operazione intorno ai 20 giorni. Questo dato è stato frutto dell'architettura progettata, poiché con diversi parametri è possibile ottenere performance differenti.

Tuttavia, si è notato che, con le assunzioni prese, la scelta migliore è rappresentata dal set di valori utilizzati. Ciò può essere visto come un limite del modello creato, il quale necessita di adattamenti strutturali a seconda della tipologia di titoli finanziari sul quale si vuole investire. È stato osservato, per esempio, che i titoli che non presentano trend di lungo periodo ben definiti, o che subiscono nel corso degli anni variazioni repentine e di grandi dimensioni, restituiscono performance molto più scarse rispetto agli altri. Inoltre, la volatilità che manifestano nell'andamento della serie dei prezzi non viene mitigata dal trading system, ma anzi si riflette nelle relative performance. Sono state dunque fornite delle possibili implementazioni future utili all'applicazione del sistema in situazioni diverse. Sono state suggerite, peraltro, alcune misure tratte da diversi articoli proposti nella comunità scientifica che potrebbero ampliare la gamma degli aspetti considerati, e di conseguenza restituire dei rendimenti ancor più consistenti.

La conclusione alla quale si è giunti è che probabilmente le performance riflettono direttamente pregi e difetti degli indicatori di analisi tecnica, i quali sono ottimi strumenti per l'individuazione dei trend, ma che tuttavia peccano spesso di mancanza di tempismo dei segnali. L'applicazione migliore che si possa fare di questi indicatori, all'interno di un trading system, sembrerebbe pertanto quella attuata nel corso del presente lavoro: la scelta di un'ottica temporale non troppo breve permette di sfruttare le variazioni determinate da un trend piuttosto che le oscillazioni continue di mercato, difficilmente prevedibili con un trading system così ideato.







## INDICE DELLE FIGURE

### *Indice dei grafici*

Grafico 1. Markov Decision Process

Grafico 2. Algoritmi Q-learning e SARSA

Grafico 3. I tre approcci di inferenza

Grafico 4. FTSE Mib e medie mobili, a 25 giorni, semplice, ponderata ed esponenziale (2017)

Grafico 5. FTSE Mib e medie mobili semplici a 15 (linea viola) e 25 (linea rossa) passi (2017)

Grafico 6. FTSE Mib e Bollinger Bands (2017)

Grafico 7. FTSE Mib, Bollinger Bands, %B e BBW (2017)

Grafico 8. Interpretazione del momentum

Grafico 9. *Trading signals* forniti dall'analisi del momentum

Grafico 10. *Trading signals* 'scremati' forniti dall'analisi del momentum

Grafico 11. FTSE Mib, MACD e *Signal Line* (2017)

Grafico 12. FTSE Mib e RSI (2017)

Grafico 13. FTSE Mib e Stochastics Oscillator (settembre 2017-febbraio 2018)

Grafico 14. *Trading signals* forniti dallo stocastico

Grafico 15. FTSE Mib, +DI (rosso), -DI (verde) e ADX (nero) - (2017)

Grafico 16. Stringa dei segnali degli indicatori

Grafico 17. Transizione da stato a stato e da riga a riga

Grafico 18. Profitto cumulato dei nove indici durante il test ( $\alpha=0.05$ ,  $\gamma=0.9$ ,  $\epsilon=0.1$ )

## *Indice delle tabelle*

Tabella 1. Valori dell'ADX e forza del trend

Tabella 2. Dati di input del modello

Tabella 3. Etichette ADX

Tabella 4. Etichette DX

Tabella 5. Etichette MACD

Tabella 6. Etichette RSI

Tabella 7. Etichette per i rendimenti giornalieri

Tabella 8. Esempio di dati di sintesi della fase di test

Tabella 9. Rendimenti totali ( $\alpha=0.05$ ,  $\gamma=0.9$ ,  $\varepsilon=0.1$ )

Tabella 10. Rendimenti totali con penalità diverse

Tabella 11. Operatività media del trading system

( $\alpha = 0.05$ ,  $\gamma = 0.9$ ,  $\varepsilon = 0.1$ , penalità Long = 0.009, penalità Short = 0.01)

Tabella 12. Operatività media dei trading system alternativi

## BIBLIOGRAFIA

Alexander, S. (1961), "Price movements in speculative markets: Trends or random walks", *Industrial Management Review*, 5, 7–26.

Alimoradi, Muhammad Reza, e Ali Husseinzadeh Kashan (2018), *A League Championship Algorithm Equipped with Network Structure and Backward Q-Learning for Extracting Stock Trading Rules*, *Applied Soft Computing*, 68 (luglio): 478–93.

Almahdi, S., & Yang, S. Y. (2017), *An adaptive portfolio trading system: A risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown*, *Expert Systems with Applications*, 87, 267–279.

Aboussalah, Amine Mohamed, e Chi-Guhn Lee (2020), *Continuous Control with Stacked Deep Dynamic Recurrent Reinforcement Learning for Portfolio Optimization*, *Expert Systems with Applications*, 140 (febbraio): 112891.

Babenko, B. (2008) "Multiple instance learning: algorithms and applications", Dept. of Computer Science and Engineering, University of California, San Diego.

Barto A.G., Sutton R.S. (1998), *Reinforcement Learning: An Introduction. Adaptive Computation and Machine Learning*, The MIT Press.

Bellman, R. (1957), "A Markovian decision process", *Journal of Mathematics And Mechanics*.

Bertoluzzo, F., & Corazza, M. (2012), *Reinforcement learning for automatic financial trading: Introduction and some applications*, Department of Economics, Ca' Foscari University of Venice Working paper, 33.

Bertsimas, D., & Lo, A. W. (1998), "Optimal control of execution costs", *Journal of Financial Markets*, 1, 1–50.

Bishop, Christopher M. (2006), *Pattern Recognition And Machine Learning*, New York: Springer Verlag.

Carapuço, João, Rui Neves, e Nuno Horta (2018), *Reinforcement Learning Applied to Forex Trading*, *Applied Soft Computing*, 73 (dicembre): 783–94.

Do, B., & Faff, R. (2010), *Does simple pairs trading still work*, *Financial Analysts Journal*, 66 (4), 83–95.

Do, B., & Faff, R. (2012), *Are pairs trading profits robust to trading costs*, *The Journal of Financial Research*, 35 (2), 261–287.

Fama, E., & Blume, M. (1966), *Filter rules and stock market trading profits*, *Journal of Business*, 39, 226–241.

Gatev, E., Goetzmann, W. N., & Geert Rouwenhorst, K. (2006), *Pairs trading: Performance of a relative-value arbitrage rule*, *The Review of Financial Studies*, 19 (3), 797.

Hildenbrand, W., & Kirman, A. P. (1988), *Equilibrium analysis*, The Netherlands: Elsevier Science Publishers, Amsterdam.

Jegadeesh, N., & Titman, S. (1993), *Returns to buying winners and selling losers: Implications for stock market efficiency*, *The Journal of Finance*, 48 (1), 65.

Jegadeesh, N., & Titman, S. (2001), *Profitability of momentum strategies: An evaluation of alternative explanations*, *The Journal of Finance*, 56 (2), 699–720.

Jeong, Gyeon, e Ha Young Kim (2019), *Improving Financial Trading Decisions Using Deep Q-Learning: Predicting the Number of Shares, Action Strategies, and Transfer Learning*, *Expert Systems with Applications* 117 (marzo): 125–38.

Laney, D. (2001), “3D Data Management: Controlling Data Volume, Velocity and Variety”, *META Group Research Note*, 6.

Lee, Raymond S. T. (2019), *COSMOS Trader – Chaotic Neuro-Oscillatory Multiagent Financial Prediction and Trading System*, *The Journal of Finance and Data Science* 5 (2): 61–82.

Lei, Kai, Bing Zhang, Yu Li, Min Yang, e Ying Shen (2020), *Time-Driven Feature-Aware Jointly Deep Reinforcement Learning for Financial Signal Representation and Algorithmic Trading*, *Expert Systems with Applications* 140 (febbraio): 112872.

Markowitz, H. (1959), *Portfolio selection: Efficient diversification of investments*, Wiley, New York.

Moody J., Wu L., Liao Y., Saffel M. (1998), *Performance functions and Reinforcement Learning for trading systems and portfolios*, *Journal of Forecasting*, 17, 441-470.

Moody J., Saffel M. (2001), *Learning to trade via Direct Reinforcement*, *IEEE Transactions on Neural Network*, 12, 875-889.

Minsky, M. e Papert S. (1969), *Perceptrons. An Introduction to Computational Geometry*, M.I.T. Press, Cambridge, Massachussets.

Mitchell, Tom M. (1997), *Machine Learning*, McGraw-Hill.

O, Jangmin, Jongwoo Lee, Jae Won Lee, e Byoung-Tak Zhang (2006), *Adaptive Stock Trading with Dynamic Asset Allocation Using Reinforcement Learning*, *Information Sciences* 176 (15): 2121–47.

Nevmyvaka, Y., Feng, Y., & Kearns, M. (2006), *Reinforcement learning for optimized trade execution*, 23rd international conference on machine learning (pp. 1–8).

Pendharkar, P. C. & Cusatis, P. (2018), *Trading Financial Indices with Reinforcement Learning Agents*, *Expert Systems with Applications* 103 (agosto): 1–13.

Rosenblatt, F. (1957), *The perceptron, a perceiving and recognizing automaton*, Cornell Aeronautical Laboratory report, Buffalo, NY: Cornell Aeronautical Laboratory.

Searle, John. R. (1980), "Minds, brains, and programs", *Behavioral and Brain Sciences*, 3, pp. 417-457.

Settles, Burr (2009), "Active Learning Literature Survey", *University of Wisconsin-Madison Department of Computer Sciences*.

Turing, Alan M. (1950), *Computing Machinery and Intelligence*, *Mind* 59 (October): 433–60.

Wang, Y., Wang, D., Zhang, S., Feng, Y., Li, S., & Zhou, Q. (2017), *Deep Q-trading*.

Warren S. McCulloch, Walter Pitts (1943), "A logical calculus of the ideas immanent in nervous activity", *The Bulletin of Mathematical Biophysics*, Volume 5, Issue 4, pp 115-133.

Yang, Steve Y., Yangyang Yu, e Saud Almahdi (2018), *An Investor Sentiment Reward-Based Trading System Using Gaussian Inverse Reinforcement Learning Algorithm*, *Expert Systems with Applications* 114 (dicembre): 388–401.





## SITOGRAFIA

AI4Business, <https://www.ai4business.it/>

Cran.R-project, <https://cran.r-project.org/>

GitHub, <https://github.com/>

Economy Up, <https://www.economyup.it/>

Forbes, <https://www.forbes.com/>

Google AI blog, <https://ai.googleblog.com/>

IONOS Digital guide, <https://www.ionos.it/digitalguide/>

Intelligenza artificiale, <http://www.intelligenzaartificiale.it/machine-learning/>

Investing.com, <https://it.investing.com/>

Investopedia, <http://www.investopedia.com/>

Machine Learning Mistery, <https://machinelearningmastery.com/>

Manolog, <https://www.manolog.it/>

Research Gate, <https://www.researchgate.net/>

Science direct, <https://www.sciencedirect.com/>

Tensor Flow, <https://www.tensorflow.org/>

Towards data science, <https://towardsdatascience.com/>

Wikipedia, <https://www.wikipedia.org/>

Yahoo finance, <https://it.finance.yahoo.com/>

## APPENDICE

Di seguito si presentano delle porzioni del codice sviluppato per il caso di studio. Gli script riportati fanno riferimento ad un solo titolo finanziario poiché la procedura resta, con le dovute sistemazioni, la stessa anche se con indici diversi.

### CODICE

#### FASE DI TRAINING

##### *Preparazione dei dati e analisi tecnica*

```
11 ▾ ##### LEARNING PHASE #####
12 ▾ ##### S&P500 #####
13
14 ## scarico serie
15 getSymbols("AGSPC",from="1962-01-01",to="2009-12-31")
16 table(is.na(GSPC))
17 nGSPC=nrow(GSPC)
18 px.gspc=GSPC[,6]
19 n.px.gspc=nrow(px.gspc)
20
21 ## creo indicatori e li sistemo
22 adx=(ADX(GSPC)[- (1:200)])[-11883]
23 rsi=(RSI(px.gspc)[- (1:200)])[-11883] ##controlla volumi
24 which(is.na(GSPC[,5]))
25 # rsi.em=embed(rsi,2)
26 macd=(MACD(px.gspc)[- (1:200)])[-11883]
27 nrow(macd)
28 nrow(adx)
29 nrow(rsi)
30 varpx=diff(px.gspc)[- (1:200)]
31 px.gspc.2=(px.gspc[- (1:199)])[-11884] ##controlla coerenza numero righe
32 nrow(varpx)
33 nrow(px.gspc.2)
34 # funzione rendimento uniperiodale
35 ▾ rend.periodo=function(diffpx,pxpr){
36 ▾ #####
37 # diffpx = variazione prezzo
38 # pxpr = prezzo precedente
39 ▾ #####
40 n=nrow(diffpx)
41 diffpx=coredata(diffpx)
42 pxpr=coredata(pxpr)
43 rend=vector(length=n)
44 ▾ for(i in 1:n){
45     rend[i]=(diffpx[i]/pxpr[i])
46 }
47     return(rend)
48 }
49 # calcolo rendimento uniperiodale
50 rendperiodo=(rend.periodo(varpx,px.gspc.2))[-11883]
51 ##controlla che rendimenti siano corretti
52 length(rendperiodo)
53
54 ## creo funzione segnali adx
55 ▾ state.adx=function(a.ind){
56 ▾ #####
57 # a.ind = valore adx per forza trend
58 ▾ #####
59 n=nrow(a.ind)
60 adx=coredata(a.ind$ADX)
61 state=vector(length=n)
62 ▾ for(i in 1:n){
63 ▾     if(is.na(adx[i])){
64         state[i]=NA
```

```

65 }else{
66   if(adx[i]<20)
67     state[i]="WT"
68   if(adx[i]>20 & adx[i]<50)
69     state[i]="ST"
70   if(adx[i]>50 & adx[i]<75)
71     state[i]="VS"
72   if(adx[i]>75)
73     state[i]="ES"
74   }
75 }
76 return(state)
77 }
78
79 ## creo funzione segnali dx
80 state.dx=function(d.ind){
81   #####
82   # d.ind = differenza DI+ e DI-
83   #####
84   n=nrow(d.ind)
85   dx=coredata(d.ind$DX)
86   state=vector(length=n)
87   if(dx[1]>=0)
88     state[1]="LL"
89   if(dx[1]<0)
90     state[1]="SS"
91   for(i in 2:n){
92     if(is.na(dx[i])){
93       state[i]=NA
94     }else{
95       if(dx[i]>0 & dx[i-1]<0)
96         state[i]="LL"
97       if(dx[i]<0 & dx[i-1]>0)
98         state[i]="SS"
99       if(dx[i]>0 & dx[i-1]>0)
100        state[i]="HL"
101       if(dx[i]<0 & dx[i-1]<0)
102        state[i]="HS"
103     }
104   }
105   return(state)
106 }
107
108 ## creo funzione segnali macd
109 state.macd=function(m.ind){
110   #####
111   # m.ind = macd DL and SL
112   #####
113   n=nrow(m.ind)
114   macd=coredata(m.ind$macd)
115   sig=coredata(m.ind$signal)
116   state=vector(length=n)
117   if((macd[1]-sig[1])>=0)
118     state[1]="LL"
119   if((macd[1]-sig[1])<0)
120     state[1]="SS"
121   for(i in 2:n){
122     if(is.na(macd[i]) | is.na(sig[i]) | is.na(macd[i-1]) | is.na(sig[i-1])){
123       state[i]=NA
124     }else{
125       if((macd[i]-sig[i])>0 & (macd[i-1]-sig[i-1])<0)
126         state[i]="LL"
127       if((macd[i]-sig[i])<0 & (macd[i-1]-sig[i-1])>0)
128         state[i]="SS"
129       if((macd[i]-sig[i])>0 & (macd[i-1]-sig[i-1])>0)
130         state[i]="HL"
131       if((macd[i]-sig[i])<0 & (macd[i-1]-sig[i-1])<0)
132         state[i]="HS"
133     }
134   }
135   return(state)
136 }
137

```

```

138 ## creo funzione segnali rsi
139 state.rsi=function(mf.ind){
140   #####
141   # mf.ind = rsi
142   #####
143   n=nrow(mf.ind)
144   rsi=coredata(mf.ind)
145   state=vector(length=n)
146   for(i in 1:n){
147     if(is.na(rsi[i])){
148       state[i]=NA
149     }else{
150       if(rsi[i]<15)
151         state[i]="MS"
152       if(rsi[i]>15 & rsi[i]<30)
153         state[i]="OS"
154       if(rsi[i]>30 & rsi[i]<70)
155         state[i]="NN"
156       if(rsi[i]>70 & rsi[i]<85)
157         state[i]="OB"
158       if(rsi[i]>85)
159         state[i]="MB"
160     }
161   }
162   return(state)
163 }
164
165 ## creo funzione segnali rendimenti uniperiodali
166 state.rendunip=function(rendunip.ind){
167   #####
168   # rendunip.ind = rendimenti uniperiodali
169   # (variazione prezzo / prezzo iniziale)
170   #####
171   n=length(rendunip.ind)
172   rend=coredata(rendunip.ind)
173   state=vector(length=n)
174   for(i in 1:n){
175     if(is.na(rend[i])){
176       state[i]=NA
177     }else{
178       if(rend[i]>0.03)
179         state[i]="SP"
180       if(rend[i]>0.01 & rend[i]<=0.03)
181         state[i]="MP"
182       if(rend[i]>0 & rend[i]<=0.01)
183         state[i]="PP"
184       if(rend[i]==0)
185         state[i]="ZZ"
186       if(rend[i]>(-0.01) & rend[i]<0)
187         state[i]="NN"
188       if(rend[i]>(-0.03) & rend[i]<=(-0.01))
189         state[i]="MN"
190       if(rend[i]<=(-0.03))
191         state[i]="SN"
192     }
193   }
194   return(state)
195 }
196
197 ##creo vettore segnali indicatori
198 segnaliadx=state.adx(adx)
199 segnalidx=state.dx(adx)
200 segnalimacd=state.macd(macd)
201 segnalirsi=state.rsi(rsi)
202 segnalirend=state.rendunip(rendperiodo)
203 which(is.na(segnaliadx))
204 which(is.na(segnalidx))
205 which(is.na(segnalimacd))
206 which(is.na(segnalirsi))
207 which(is.na(segnalirend))
208 length(segnaliadx)
209 length(segnalidx)
210 length(segnalimacd)
211 length(segnalirsi)
212 length(segnalirend)
213 #state.na=na.omit(state)
214 n.segnali=length(segnalimacd)
215 state=paste(segnaliadx,segnalidx,segnalimacd,segnalirsi,segnalirend,
216            sep = "") ##controllare

```



```

218 ## creo vettore "action"
219 action=sample(c("B","H","S"),size=n.segnali,replace=TRUE)
220
221 ## creo funzione per transizione, reward e matrice dati
222 nextf=function(state,action,px,penal,penas){
223   #####
224   # state = matrice con segnali dell'indicatore
225   # port = indicatore se portafoglio L o S o O vuoto
226   # px = quotazioni titolo
227   #
228   #####
229   px=coredata(px)
230   reward=double()
231   next_state = state
232   n=length(state)
233   #m=max(which(is.na(state)))
234   state[1]=paste(state[1],sample(c("L","S"),1),sep="")
235   for(i in 1:n){
236     # incrocio dal basso
237     if(substr(state[i],11,11)=="L" & (action[i]=="B" | action[i]=="H"))
238       next_state[i]=state[i]
239     if(substr(state[i],11,11)=="L" & action[i]=="S")
240       next_state[i]=paste(substr(state[i],1,10),"S",sep="")
241     if(substr(state[i],11,11)=="S" & (action[i]=="S" | action[i]=="H"))
242       next_state[i]=state[i]
243     if(substr(state[i],11,11)=="S" & action[i]=="B")
244       next_state[i]=paste(substr(state[i],1,10),"L",sep="")
245     if(i<n)
246       state[i+1]=paste(substr(state[i+1],1,10),substr(next_state[i],11,11),sep="")
247   }
248   if(substr(next_state[1],11,11)=="L")
249     reward[1]=((px[1+201]*(1-penal)-px[1+200])/px[1+200])*(px[1+200]/px[2])
250   if(substr(next_state[1],11,11)=="S")
251     reward[1]=(-(px[1+201]*(1+penas)-px[1+200])/px[1+200])*(px[1+200]/px[2])
252
253   for(i in 2:n){
254     if(substr(next_state[i],11,11)=="L" & substr(next_state[i-1],11,11)=="S"){
255       reward[i]=((px[i+201]*(1-penal)-px[i+200])/px[i+200])*(px[i+200]/px[i+1])
256     }else{
257       if(substr(next_state[i],11,11)=="L")
258         reward[i]=((px[i+201]-px[i+200])/px[i+200])*(px[i+200]/px[i+1])
259       if(substr(next_state[i],11,11)=="S")
260         reward[i]=(-(px[i+201]-px[i+200])/px[i+200])*(px[i+200]/px[i+1])
261     }
262     if(substr(next_state[i],11,11)=="S" & substr(next_state[i-1],11,11)=="L"){
263       reward[i]=(-(px[i+201]*(1+penas)-px[i+200])/px[i+200])*(px[i+200]/px[i+1])
264     }else{
265       if(substr(next_state[i],11,11)=="L")
266         reward[i]=((px[i+201]-px[i+200])/px[i+200])*(px[i+200]/px[i+1])
267       if(substr(next_state[i],11,11)=="S")
268         reward[i]=(-(px[i+201]-px[i+200])/px[i+200])*(px[i+200]/px[i+1])
269     }
270   }
271   out=list(State=state,Action=action,NextState=next_state,Reward=reward)
272   # return(list(State=state,Action=action,NextState=next_state,Reward=reward))
273   return(out)
274 }
275
276 penal=0.009
277 penas=0.01
278 dati=nextf(state,action,px.gspc,penal,penas)
279 head(dati)
280 dati=cbind(data.frame(as.character(dati$State),stringsAsFactors = F),
281             data.frame(as.character(dati$Action),stringsAsFactors = F),
282             data.frame(as.character(dati$NextState),stringsAsFactors = F),
283             dati$Reward)
284 colnames(dati)=c("State","Action","NextState","Reward")

```

## Reinforcement learning

```
286 ## creo parametri modello
287 control<-list(alpha=0.05,gamma=0.9,epsilon=0.1)
288
289 ## lancio funzione Reinforcement Learning
290 model=ReinforcementLearning(data=dati,s="State",a="Action",r="Reward",
291                             s_new = "NextState",control = control,iter=1)
292 summary(model)
293
294 ## valuto se utilizzare un nuovo vettore di azioni casuali porta miglioramenti
295 action_2=sample(c("B","H","S"),size=n.segnali,replace=TRUE)
296 dati_2=nextf(state,action_2,px.gspc,penal,penas)
297 dati_2=cbind(data.frame(as.character(dati_2$State),stringsAsFactors = F),
298               data.frame(as.character(dati_2$Action),stringsAsFactors = F),
299               data.frame(as.character(dati_2$NextState),stringsAsFactors = F),
300               dati_2$Reward)
301 colnames(dati_2)=c("State","Action","NextState","Reward")
302 model_2=ReinforcementLearning(data=dati_2,s="State",a="Action",r="Reward",
303                               s_new = "NextState",control = control,iter=1,
304                               model = model)
305 summary(model_2)
306
307 action_3=sample(c("B","H","S"),size=n.segnali,replace=TRUE)
308 dati_3=nextf(state,action_3,px.gspc,penal,penas)
309 dati_3=cbind(data.frame(as.character(dati_3$State),stringsAsFactors = F),
310               data.frame(as.character(dati_3$Action),stringsAsFactors = F),
311               data.frame(as.character(dati_3$NextState),stringsAsFactors = F),
312               dati_3$Reward)
313 colnames(dati_3)=c("State","Action","NextState","Reward")
314 model_3=ReinforcementLearning(data=dati_3,s="State",a="Action",r="Reward",
315                               s_new = "NextState",control = control,iter=1,
316                               model = model_2)
317 summary(model_3)
318
319 action_4=sample(c("B","H","S"),size=n.segnali,replace=TRUE)
320 dati_4=nextf(state,action_4,px.gspc,penal,penas)
321 dati_4=cbind(data.frame(as.character(dati_4$State),stringsAsFactors = F),
322               data.frame(as.character(dati_4$Action),stringsAsFactors = F),
323               data.frame(as.character(dati_4$NextState),stringsAsFactors = F),
324               dati_4$Reward)
325 colnames(dati_4)=c("State","Action","NextState","Reward")
326 model_4=ReinforcementLearning(data=dati_4,s="State",a="Action",r="Reward",
327                               s_new = "NextState",control = control,iter=1,
328                               model = model_3)
329 summary(model_4)
330
331 ## sistema policy in matrice di due colonne (stati,azioni)
332 policystateaction=cbind(data.frame(as.character(model$States),stringsAsFactors = F),
333                          data.frame(as.character(model$Policy),stringsAsFactors = F))
334 colnames(policystateaction)=c("State","Action")
335
336 policystateaction2=cbind(data.frame(as.character(model_2$States),stringsAsFactors = F),
337                           data.frame(as.character(model_2$Policy),stringsAsFactors = F))
338 colnames(policystateaction2)=c("State","Action")
339
340 policystateaction3=cbind(data.frame(as.character(model_3$States),stringsAsFactors = F),
341                           data.frame(as.character(model_3$Policy),stringsAsFactors = F))
342 colnames(policystateaction3)=c("State","Action")
343
344 policystateaction4=cbind(data.frame(as.character(model_4$States),stringsAsFactors = F),
345                           data.frame(as.character(model_4$Policy),stringsAsFactors = F))
346 colnames(policystateaction4)=c("State","Action")
```



## FASE DI TEST

```
1 - ##### TEST PHASE #####
2
3 model_finale= model.dax
4
5 ## sistema policy in matrice di due colonne (stati,azioni)
6 # !!!! controlla che ci sia modello RL finale !!!!
7 polycystateaction.finale=cbind(
8   data.frame(as.character(model_finale$States),stringsAsFactors = F),
9   data.frame(as.character(model_finale$Policy),stringsAsFactors = F))
10 colnames(polycystateaction.finale)=c("state","Action")
11
12
13
14 - ##### S&P500 #####
15
16 getSymbols("AGSPC",from="2009-01-01",to="2019-12-31")
17 gspc_test=GSPC
18 which(is.na(gspc_test))
19 n_gspc_test=nrow(gspc_test)
20 px.gspc_test=gspc_test[,6]
21 n_px.gspc_test=nrow(px.gspc_test)
22
23 ## indicatori test
24 adx_test=ADX(gspc_test)[-(1:200)]
25 rsi_test=RSI(px.gspc_test)[-(1:200)] ##controlla volumi
26 # rsi.em=embed(rsi,2)
27 macd_test=MACD(px.gspc_test)[-(1:200)]
28 nrow(macd_test)
29 nrow(adx_test)
30 nrow(rsi_test)
31 varpx_test=diff(px.gspc_test)[-(1:200)]
32 px.gspc_test.2=(px.gspc_test[-(1:199)])[-2568]
33 ##controlla coerenza numero righe
34 nrow(varpx_test)
35 nrow(px.gspc_test.2)
36 rendperiodo_test=rend.periodo(varpx_test,px.gspc_test.2)
37 ##controlla che rendimenti siano corretti
38 length(rendperiodo_test)
39
40 ##creo vettore segnali indicatori
41 segnalidx_test=state.adx(adx_test)
42 segnalidx_test=state.dx(adx_test)
43 segnalimacd_test=state.macd(macd_test)
44 segnalirsi_test=state.rsi(rsi_test)
45 segnalirend_test=state.rendunip(rendperiodo_test)
46 which(is.na(segnalidx_test))
47 which(is.na(segnalidx_test))
48 which(is.na(segnalimacd_test))
49 which(is.na(segnalirsi_test))
50 which(is.na(segnalirend_test))
51 length(segnalidx_test)
52 length(segnalidx_test)
53 length(segnalimacd_test)
54 length(segnalirsi_test)
55 length(segnalirend_test)
56 #state.na=na.omit(state)
57 n.segnali_test=length(segnalimacd_test)
58
59 state_test=paste(segnalidx_test,segnalidx_test,segnalimacd_test,
60                 segnalirsi_test,segnalirend_test,sep = "")
61
62 px.gspc_test.prof=(gspc_test[,6])[-(1:200)]
63 nrow(px.gspc_test.prof)
64
```



```

65 ##### funzione profitti uniperiodali
66 ▾ profitto.unip=function(state,optaction,newstate,px,p="L"){
67 ▾ #####
68   # state = stati policy
69   # optaction = azioni policy
70   # newstate = segnali indicatori nel test-dataset
71   # px = prezzi
72 ▾ #####
73   px=coredata(px)
74   m=length(newstate)
75   n=length(state)
76   t_action=vector(length = m)
77   profuni=double()
78   next_state_test = vector(length=m)
79   newstate[1]=paste(newstate[1],p,sep="")
80 ▾ for (t in 1:m) {
81     j=which(state==newstate[t])
82 ▾ if(length(j)==0){
83     t_action[t]="H"
84 ▾ }else{
85     t_action[t]=optaction[j]
86   }
87   if(substr(newstate[t],11,11)=="H" & t_action[t]=="B")
88     next_state_test[t]=paste(substr(newstate[t],1,10),"L",sep="")
89   if(substr(newstate[t],11,11)=="H" & t_action[t]=="S")
90     next_state_test[t]=paste(substr(newstate[t],1,10),"S",sep="")
91   if(substr(newstate[t],11,11)=="H" & t_action[t]=="H")
92     next_state_test[t]=newstate[t]
93   if(substr(newstate[t],11,11)=="L" & (t_action[t]=="B" | t_action[t]=="H"))
94     next_state_test[t]=newstate[t]
95   if(substr(newstate[t],11,11)=="L" & t_action[t]=="S")
96     next_state_test[t]=paste(substr(newstate[t],1,10),"S",sep="")
97   if(substr(newstate[t],11,11)=="S" & (t_action[t]=="S" | t_action[t]=="H"))
98     next_state_test[t]=newstate[t]
99   if(substr(newstate[t],11,11)=="S" & t_action[t]=="B")
100     next_state_test[t]=paste(substr(newstate[t],1,10),"L",sep="")
101   if(t<m)
102     newstate[t+1]=paste(substr(newstate[t+1],1,10),
103                          substr(next_state_test[t],11,11),sep="")
104
105   if(substr(next_state_test[t],11,11)=="L")
106     profuni[t]=px[t+1]-px[t]
107   if(substr(next_state_test[t],11,11)=="S")
108     profuni[t]=(-(px[t+1]-px[t]))
109   }
110   out=list(State_test=newstate,Action_test=t_action,
111           NextState_test=next_state_test,Prof_Unip=profuni)
112   return(out)
113 }
114
115 profitounip=profitto.unip(policystateaction.finale$state,
116                          policystateaction.finale$action,state_test,px.gspc_test.prof)
117 ## toglia ultima riga perchè profit NA (5082)
118 profitounip=cbind(
119   data.frame(as.character(profitounip$State_test),stringsAsFactors = F),
120   data.frame(as.character(profitounip$action_test),stringsAsFactors = F),
121   data.frame(as.character(profitounip$NextState_test),stringsAsFactors = F),
122   profitounip$Prof_Unip)
123 profitounip=profitounip[1:2566,1:4]
124 colnames(profitounip)=c("StateTest","OptAction","NextStateTest","Prof_Unip")
125

```

```

126 ##### funzione px.acq, n'operazioni, profitto cumulato
127 px.acq.operaz=function(next_state_test,profuni,px){
128 #####
129 # next_state_test= nuovi stati ottenuti con la policy
130 # profuni= profitti uniperiodali
131 #####
132 px=coredato(px)
133 m=length(next_state_test)
134 px.acq=double()
135 n.ordini=double()
136 profcumulato=double()
137 profoperazione=double()
138 px.acq[1]=px[1]
139 n.ordini[1]=1
140 profcumulato[1]=profuni[1]
141
142 for (t in 2:m) {
143   if(substr(next_state_test[t],11,11)!=substr(next_state_test[t-1],11,11)){
144     px.acq[t]=px[t]
145   }else{
146     px.acq[t]=NA
147   }
148
149   if(substr(next_state_test[t],11,11)!=substr(next_state_test[t-1],11,11)){
150     n.ordini[t]=(n.ordini[t-1]+1)
151   }else{
152     n.ordini[t]=n.ordini[t-1]
153   }
154
155   if(n.ordini[t]!=n.ordini[t-1]){
156     profcumulato[t]=profuni[t]
157   }else{
158     profcumulato[t]=sum(profcumulato[t-1],profuni[t])
159   }
160 }
161 out=list(NextState_test=next_state_test,Prof_Unip=profuni,Px_Acq=px.acq,
162         N_operaz=n.ordini,Prof_Cumulato=profcumulato)
163 return(out)
164 }
165
166 px.prof.operaz=px.acq.operaz(profittounip$NextStateTest,profittounip$Prof_Unip,
167                             px.gspc_test.prof[-2567])
168
169 px.prof.operaz=cbind(
170   data.frame(as.character(px.prof.operaz$NextState_test),stringsAsFactors = F),
171   px.prof.operaz$Prof_Unip,px.prof.operaz$Px_Acq,px.prof.operaz$N_operaz,
172   px.prof.operaz$Prof_Cumulato)
173 colnames(px.prof.operaz)=c("NextStateTest", "Prof_Unip", "Px_Acq", "N_operaz",
174                             "Prof_Cumulato")
175
176 ##### funzione profitto per operazione
177 profitto.operazione=function(next_state_test,profcumulato){
178 #####
179 # next_state_test= nuovi stati ottenuti con la policy
180 # profcumulato= profitto che si accumula in ciascuna operazione
181 #####
182 m=length(next_state_test)
183 profoperazione=double()
184
185 for (t in 1:m) {
186   if(t<m){
187     if(substr(next_state_test[t],11,11)!=substr(next_state_test[t+1],11,11)){
188       profoperazione[t]=profcumulato[t]
189     }else{
190       profoperazione[t]=NA
191     }
192   }else{
193     profoperazione[t]=profcumulato[t]
194   }
195 }
196 out=list(NextState_test=next_state_test,Prof_Cumulato=profcumulato,
197         Prof_operaz=profoperazione)
198 return(out)
199 }

```

```

200 prof.operaz=profitto.operazione(px.prof.operaz$NextStateTest,
201                                px.prof.operaz$Prof_Cumulato)
202 prof.operaz=cbind(
203   data.frame(as.character(profitto.operaz$NextState_test),stringsAsFactors = F),
204   prof.operaz$Prof_Cumulato,prof.operaz$Prof_operaz)
205 colnames(prof.operaz)=c("NextStateTest","Prof_Cumulato","Prof_operazione")
206
207 ## dati test
208 sintesi.test=cbind(
209   data.frame(as.character(profitto.unip$StateTest),stringsAsFactors = F),
210   data.frame(as.character(profitto.unip$OptAction),stringsAsFactors = F),
211   data.frame(as.character(profitto.unip$NextStateTest),stringsAsFactors = F),
212   profitto.unip$Prof_Unip,px.prof.operaz$Px_Acq,px.prof.operaz$N_operaz,
213   px.prof.operaz$Prof_Cumulato,prof.operaz$Prof_operazione)
214 nrow(sintesi.test)
215 colnames(sintesi.test)=c("StateTest","OptAction","NextStateTest","Prof_Unip",
216                          "Px_Acq","N_operaz","Prof_Cumulato","Prof_operazione")
217
218 table(is.na(sintesi.test$Px_Acq))
219 table(is.na(sintesi.test$Prof_operazione))
220
221 px.acquisto=na.omit(sintesi.test$Px_Acq)
222 length(px.acquisto)
223 profit_operazione=na.omit(sintesi.test$Prof_operazione)
224 length(profit_operazione)
225
226 #calcolo rendimenti singole operazioni con commissione su prezzo di acquisto
227 commissione=0.0005
228 rendimenti.operazioni=(profit_operazione-(px.acquisto*(commissione)))/px.acquisto
229
230 # cumulata del profitto nel tempo, profitto totale e rendimento totale
231 guadagno.operazione=profit_operazione-(px.acquisto*(commissione))
232 prof.tot.test=sum(guadagno.operazione)
233 prof.tot.test
234 rend.tot=(prof.tot.test/px.gspc_test.prof[1])
235 rend.tot
236
237 ▾ guadagno cumulato=function(profittoxoperazione){
238 ▾   #####
239 ▾   # profittoxoperazione = guadagno operazione con commissione
240 ▾   #####
241   n=length(profittoxoperazione)
242   guadagno=vector(length = n)
243   operazioni=c(1:n)
244   guadagno[1]=profittoxoperazione[1]
245
246 ▾   for (i in 2:n) {
247     guadagno[i]=guadagno[i-1]+profittoxoperazione[i]
248   }
249   out=list(N.operazione=operazioni,Guadagno=guadagno)
250   return(out)
251 }
252
253 guadagno nel tempo=guadagno cumulato(guadagno.operazione)
254 guadagno nel tempo=cbind(guadagno nel tempo$N.operazione,guadagno nel tempo$Guadagno)
255 colnames(guadagno nel tempo)=c("Numero operazione","Guadagno cumulato")
256
257 plot(guadagno nel tempo)

```



## RINGRAZIAMENTI

A conclusione di questo elaborato, vorrei dedicare qualche riga a tutti coloro che si sono dimostrati fondamentali per la realizzazione dello stesso.

Un sentito ringraziamento va al mio relatore, Chiarissimo Professor Pizzi, per la sua infinita disponibilità e gentilezza. Senza di lui questo lavoro di tesi non esisterebbe nemmeno.

Ringrazio di cuore i miei genitori per avermi sempre sostenuto ed avermi permesso di inseguire la mia strada. Parte di questo traguardo è senz'altro merito vostro.

Grazie a mia sorella Giulia, da sempre guida e sostegno in ogni mia sfida.

Un pensiero speciale va a Elisa per avermi trasmesso energia e determinazione. Grazie per il tempo ed i suggerimenti che mi hai donato. Grazie perché ci sei sempre stata.

Infine, vorrei ringraziare gli amici e colleghi più cari per aver condiviso con me gioie e difficoltà di questo percorso di studi.

Grazie infinite a tutti voi.