

UNIVERSITÀ DEGLI STUDI DI MILANO

Facoltà di Scienze e Tecnologie

Corso di Laurea in informatica

PROGETTAZIONE E SVILUPPO DI
TEST SUITE AUTOMATICA E
MODULO RACCOLTA DATI PER
INTELLIGENZE ARTIFICIALI CHE SI
OCCUPANO DI CREARE STRATEGIE
DI INVESTIMENTO

Relatore: Prof. Alfio Ferrara

Correlatore: Dr. Andrea Bellacicca

Tesi di:

Garion Musetta

Matricola: 920623

Anno Accademico 2019-2020

Prefazione

Organizzazione della tesi

La tesi è organizzata come segue:

- nel Capitolo 1 è introdotto l'ambito della tesi e gli strumenti finanziari utilizzati, come le strategie di investimento e il modello dei dati.
- il Capitolo 2 è dedicato allo stato dell'arte delle AI per la previsione di serie temporali ed è introdotta Sentyment, la AI da testare oggetto della tesi, inquadrandola nel panorama attuale.
- nel Capitolo 3 si elaborano i dati di trading, dal download alla sorgente fino alla creazione di candele OHLCV e strategie di investimento. Attraverso la breve analisi dei componenti di Sentyment si mostra il workflow implementato dai software di questo tipo.
- nel Capitolo 4 è analizzata la AI in questione, è sviluppato un nuovo strumento intelligente che sceglie i migliori parametri per Sentyment e questa è infine confrontata con altri mezzi simili.
- il Capitolo 5 conclude i test e offre spunti per eventuali futuri miglioramenti.

Introduzione

Lo scopo di questo elaborato è affrontare il tema delle intelligenze artificiali (AI) per il trading illustrandone alcune, applicando delle metodologie di test per analizzarne le prestazioni e descrivere lo sviluppo di nuovi componenti da integrare in quelle già esistenti.

Sono disponibili, attualmente, una grande varietà di algoritmi di AI, nel campo della finanza e in particolare nel campo della previsione del mercato azionario, simili a quella implementata, in grado di produrre ottimi risultati.

Qualsiasi tecnica venga utilizzata, durante lo sviluppo di uno strumento di questo tipo, è indispensabile porsi delle domande circa la qualità dei risultati prodotti facendo un confronto con un modello ottimale di riferimento. Spesso questo modello non è tuttavia disponibile oppure è molto difficile da creare, a volte è perfino incalcolabile. I risultati dovrebbero essere paragonati anche con quelli prodotti da altre applicazioni preesistenti che operano nel campo del trading.

Per la stesura di questo elaborato viene analizzato *Sentyment*, uno fra i numerosi sistemi di AI per trading. Il software è stato sottoposto all'analisi dell'architettura ed a test per verificarne il corretto funzionamento. Sono stati inoltre aggiunti nuovi componenti: un modulo di raccolta dati ed un ulteriore livello di AI in grado di apprendere l'andamento dell'algoritmo ed aggiornare di conseguenza alcuni dei suoi parametri, in automatico.

Indice

Prefazione	ii
Introduzione	iii
1 Financial trading	1
1.1 Criptovalute	2
1.2 Modello dei dati	4
1.3 Guadagnare con il trading	7
1.4 Esempi di strategie	10
1.4.1 Moving Average Crossover	11
1.4.2 Buy-Hold	13
1.5 Ruolo delle intelligenze artificiali e scopo della tesi	15
2 AI per financial prediction e Sentyment	17
2.1 Stato dell'arte delle AI	18
2.1.1 Supporto di informazioni relative agli eventi	20
2.1.2 Punti chiave nelle predizioni finanziarie	21
2.1.3 Metodi di previsione	23
2.2 Sentyment	35
2.2.1 Cenni all'architettura	35
2.2.2 Strumenti e tecnologie utilizzati	39
3 Elaborazione dati di trading	43
3.1 Download dati Kraken	43
3.2 Creazione candele OHLCV	48
3.3 Creazione strategie	52
4 Test per intelligenze artificiali	56
4.1 Test per Sentyment AI	58
4.2 Metriche	59
4.3 Test comparativi e Meta-learner	61

4.3.1	Multi-armed bandit e Meta-learner	66
4.3.2	Scelta dei parametri e adattamento	70
4.4	Test framework workflow e risultati	73
4.4.1	Considerazioni finali su test comparativi	78
4.5	Confronto con altre strategie	81
4.5.1	Maximum profit	88
5	Conclusioni e lavori futuri	96

Capitolo 1

Financial trading

Il dominio considerato per il progetto di tesi è quello della finanza, in particolare l'aspetto legato allo sviluppo di strategie di investimento intelligenti.

L'intelligenza artificiale da testare, *Sentyment*, è un software commerciale che genera diverse strategie di investimento personalizzate per numerosi strumenti finanziari. È anche in grado di piazzare direttamente ordini sul mercato, cioè effettuare operazioni di acquisto e di vendita.

Il mercato in cui opera è quello delle criptovalute. Di queste, Bitcoin è il tipo più famoso ed asset più valevole a livello di borsa valori, insieme ad altre monete come Ethereum, Bitcoin Cash e Ripple, meno famose, ma con le loro particolarità, ed altrettanto scambiate.

L'obiettivo di Sentyment non è strettamente, e solamente, la previsione dell'andamento di un certo titolo azionario. Non è necessario, quindi, predire con precisione assoluta il valore di un titolo in un dato istante di tempo per massimizzare i guadagni. Ciò su cui si focalizza maggiormente è diversificare il portafoglio ed adattarsi alle esigenze degli utenti. Il suo scopo finale è produrre delle *strategie di investimento*, ossia approcci di investimento personalizzati, in funzione della propensione al rischio, degli obiettivi e degli interessi specifici del singolo investitore. Seguendo tale strategia, l'investitore può decidere quali tra i diversi tipi di attività da includere nel proprio portafoglio di investimento.

Una specifica strategia di investimento può essere determinata da una serie di fattori, tra cui la propensione al rischio ed i rendimenti che si vogliono perseguire sugli investimenti, nonché le attività, le regioni ed i settori a cui si è interessati ed il periodo per il quale si intende investire.

Definita una strategia, Sentyment la persegue selezionando quindi, per ogni moneta, un'operazione in ogni "istante di tempo" scegliendo tra: *buy* (comprare il titolo

azionario), *sell* (vendere) o *hold* (mantenere il portafoglio). Lo scopo a lungo termine della strategia è massimizzare il guadagno rispetto a quanto investito, oltre a rispettare i vincoli di rischio ed interessi specifici dell'investitore. La AI è in grado di decidere quali fra queste operazioni effettuare al fine di massimizzare gli obiettivi. L'istante di tempo usato da Sentyment è l'ora. Ogni ora, la piattaforma di trading di riferimento, Kraken¹, produce nuovi dati e li rende disponibili per essere scaricati dai trader, che possono, quindi, compiere le azioni sopra descritte. È anche possibile cambiare l'unità di tempo di riferimento, scegliendo i minuti come slot temporale: 1, 5, 15, 30, 60, 240, 1440, 10080, 21600.

In questo primo capitolo sono spiegati alcuni dei concetti e delle terminologie del campo della finanza utili a comprendere il lavoro svolto dalle AI per trading.

1.1 Criptovalute

Sentyment si concentra sul mercato delle criptovalute, o cryptocurrency. Ci sono alcuni vantaggi che questo mercato offre rispetto allo scambio di valute tradizionali.

La criptovaluta è una rappresentazione digitale di valore basata sulla crittografia. Questo tipo di currency utilizza tecnologie di tipo peer-to-peer², operando su reti i cui nodi risultano costituiti da computer di utenti, situati potenzialmente in tutto il globo, sui quali vengono eseguiti appositi programmi che svolgono funzioni di portamonete. Non c'è, attualmente, alcuna autorità centrale che controlla queste reti pertanto, le transazioni ed il rilascio avvengono collettivamente in rete, previo raggiungimento di un consenso distribuito fra gli utenti.

Essendo prive di autorità centrale, per funzionare e garantire la consistenza delle transazioni che avvengono, le reti devono implementare una forma di consenso distribuito³. Nel caso di Bitcoin, tutti i partecipanti alla rete della Blockchain si accordano accettando la veridicità di un *registro distribuito*, contenente l'elenco di tutte le transazioni effettuate. Nel caso di nuove transazioni, l'elenco è aggiornato e, dopo aver raggiunto il consenso da parte degli altri nodi, la transazione viene validata e, quindi,

¹Kraken Digital Asset Exchange. URL: <https://www.kraken.com/>

²Una rete p2p, ovvero rete paritaria/paritetica, indica un modello di architettura logica di rete in cui i nodi non sono gerarchizzati unicamente sotto forma di client, o server fissi, ma anche sotto forma di nodi equivalenti o 'paritari' (peer), potendo fungere, al contempo, da client e server verso gli altri nodi terminali della rete. Mediante questa configurazione, qualsiasi nodo è in grado di avviare, o completare, una transazione.

³Il problema del consenso distribuito: ottenere l'affidabilità complessiva del sistema in presenza di numerosi processi che possono mostrare comportamenti difettosi. Ciò richiede che i processi concordino su alcuni valori di dati necessari durante il calcolo

riconosciuta da tutti.

Le criptovalute facilitano lo scambio di valore ed informazioni tra due parti e lo fanno in modo sicuro ed affidabile, in modo che non sia necessaria una terza parte che controlli ed approvi la transazione. Dal punto di vista dell'utente, le operazioni di scambio avvengono tra il suo software e la *piattaforma di exchange*. Una piattaforma di exchange (o trading) è un software che permette apertura, chiusura e gestione delle posizioni di mercato attraverso un intermediario finanziario (un broker online). Nel caso di Sentyment, la piattaforma di riferimento è Kraken.

Le criptovalute esistono soltanto all'interno delle strutture a consenso distribuito come Blockchain, si rivelano, quindi, particolarmente adatte all'utilizzo da parte di *bot* o programmi. Per effettuare scambi, non si deve possedere fisicamente la moneta, basta un wallet associato alla piattaforma di exchange e il bot è libero di operare sul mercato.

La differenza principale, rispetto alle valute tradizionali, è l'assenza di un'autorità centrale che controlli il rilascio della moneta il che facilita il lavoro dei trader di cryptocurrency, i quali devono fare riferimento soltanto alla piattaforma di exchange.

La criptovaluta più conosciuta è Bitcoin. Esistono, però, diverse decine di criptovalute e, se si considerano le coppie di scambio (esempio, Bitcoin-Ethereum), sono quasi cento solo quelle trattate da Kraken. Al fine della diversificazione del portafoglio, che consiste in una riduzione della rischiosità del rendimento, legata alla presenza di più attività finanziarie, i cui rendimenti non sono correlati, è opportuno investire in più asset per avere risultati migliori in termini di guadagno.

Alcune fra le monete trattate da Kraken (e Sentyment) sono:

- BTC - Bitcoin, la prima più famosa e col valore di scambio più alto.
- ETH - Ethereum è una piattaforma decentralizzata per la creazione e pubblicazione peer-to-peer di contratti intelligenti creati in un linguaggio di programmazione Turing-completo.
- ZEC - Zcash offre privacy e trasparenza selettiva delle transazioni. I pagamenti sono pubblicati su una blockchain pubblica, ma il mittente, il ricevente e il valore della transazione possono rimanere privati.
- BCH - Bitcoin Cash è un fork di Bitcoin.
- XRP - Ripple

1.2 Modello dei dati

I dati usati dalla AI in questione sono scaricati dalla piattaforma di trading *Kraken*. Ogni secondo, trader da tutto il mondo effettuano azioni di compravendita di titoli, facendo crescere, o diminuire, il valore azionario di ogni asset, esattamente nel modo in cui opera anche Sentyment. Kraken rende disponibile, attraverso delle API (application programming interface), un elenco di record contenenti timestamp e prezzo, che formano lo storico degli scambi effettuati per ognuno degli asset che espone. Questi sono 110 coppie di valute che rappresentano i valori di scambio fra criptovalute e USD / EUR. Si fa distinzione fra dati "*RAW*", ovvero un semplice elenco di prezzi variabili nel tempo, e le "*candele OHLCV*" (*open, high, low, close, volume*).

Le candele sono uno degli strumenti grafici più popolari, in quanto offrono un eccellente riferimento virtuale dei movimenti dei prezzi in un intervallo di tempo: al minuto, all'ora, al giorno, mensile ed altro.

Sono, dunque, la rappresentazione di un dato aggregato secondo l'unità di tempo selezionata.

Le informazioni che racchiudono sono indicate dal loro nome. Prendendo come esempio le candele orarie: la creazione di ognuna di esse parte allo scoccare dell'ora e fissa un prezzo di apertura (*open*: il prezzo di vendita dell'asset in quel momento), per terminare, dopo un'ora, con un prezzo di chiusura (*close*: lo stesso prezzo di vendita dell'asset al momento di chiusura, che sarà ora cambiato rispetto a *open*); si calcola quali sono stati i picchi massimi e minimi di prezzo durante l'ora (*high, low*) e, infine, il volume, che rappresenta l'ammontare totale scambiato nell'ora.

Le candele OHLCV rappresentano, dunque, la storia dell'andamento dei prezzi di un certo asset durante il periodo di tempo fissato: iniziano con un certo prezzo, all'apertura dello slot temporale, che durante il periodo considerato evolve e crea i punti minimo e massimo, per poi terminare, alla chiusura dello slot, con la determinazione del prezzo finale. I prezzi minimo e massimo possono superare, anche di molto, quelli di apertura e chiusura. Esiste la possibilità per cui il prezzo non evolva nel periodo selezionato e che i quattro termini, *open, close, high* e *low*, coincidano.

Volume è invece la somma totale della quantità di titoli comprati e venduti durante l'unità di tempo.

Sentyment lavora quasi esclusivamente su queste candele e, anche se le API di Kraken permettono di scaricare dati già aggregati, la AI scarica soltanto dati di trading raw per poi creare, autonomamente, le sue candele attraverso un modulo dati dedicato.

Ultimo Aggiornamento: 02-05-20 15:06:40 +00:00

Trade recenti (XBT/EUR)

1 - 15 di 250 trade

Ora ▾	Ordine ▾	Prezzo ▾	Volume ▾
15:06:36 +00:00	sell	€8,567.9	0.01946901
15:06:21 +00:00	buy	€8,568.0	0.02432718
15:06:17 +00:00	sell	€8,567.9	0.00453099
15:06:17 +00:00	sell	€8,567.9	0.01546901
15:06:04 +00:00	sell	€8,567.9	0.00288574
15:06:03 +00:00	sell	€8,567.9	0.03500000
15:06:01 +00:00	buy	€8,568.0	0.01082337
15:05:58 +00:00	buy	€8,568.0	0.05478300
15:05:57 +00:00	buy	€8,568.0	0.00673553
15:05:55 +00:00	buy	€8,568.0	0.04054651
15:05:54 +00:00	buy	€8,568.0	0.04081193
15:05:48 +00:00	buy	€8,568.0	0.06738600
15:05:45 +00:00	buy	€8,568.0	0.16574437

Figura 1:

Elenco transazioni Bitcoin/Euro relative ad una finestra di minuti. I record mostrano l'ora in cui è avvenuto il trade, il tipo di operazione (buy / sell), il prezzo di scambio del titolo e la quantità di titoli scambiati. L'elenco dei dati di trade 'raw' è disponibile tramite le API kraken ed è la fonte grezza di dati finanziari utilizzati per calcolare i grafici OHLCV, fondamentali per l'analisi dei mercati. (fonte: <https://www.kraken.com/>)

**Figura 2:**

Grafico OHLCV ricavato dai prezzi in Figura 1. Sulle ascisse è rappresentata l'ora mentre le ordinate sono il prezzo (in Euro) del titolo Bitcoin. Le "barre" orizzontali verdi e rosse sono le candele OHLCV: verdi se il prezzo è in crescita (*open* minore di *close*), rosse se in discesa (*open* maggiore di *close*). Le candele sono di durata 1 ora e, quindi i prezzi elencati nella precedente immagine rientrano soltanto in parte nell'ultima candela (rossa) delle 15:00 - 16:00, ancora aperta e, quindi, in creazione. La candela è rossa perché, come si nota dai prezzi, il valore del titolo è in discesa: partendo da circa 8.568 (probabilmente più alto nei record precedenti) si scende verso 8.567.

Le linee colorate rappresentano le medie mobili dei prezzi di chiusura; usate per analisi tecnica, sono, rispettivamente, le medie a 10 (verde), 21 (azzurro) e 100 (blu) candele. Più candele sono considerate nella media e meno questa cambierà bruscamente, avendo, quindi, la media a 100 candele (cioè 100 ore) molto più lenta delle altre. (fonte: <https://www.kraken.com/>)

Le immagini descrivono l'andamento dei prezzi scambio per scambio e le corrispondenti candele OHLCV come esposte dalla piattaforma di trading. I bordi orizzontali, che compongono i lati superiore ed inferiore della candela, sono il prezzo di apertura e quello di chiusura: se la candela è rossa, il prezzo di chiusura è minore di quello di apertura (significa che il prezzo dell'asset è sceso durante l'unità di tempo) e, quindi, il bordo superiore rappresenta il valore di *open* e quello inferiore *close*, mentre una candela verde indica una crescita del prezzo (*open* è il lato inferiore mentre *close* quello superiore). Le linee verticali della candela si estendono, invece, verso il prezzo minimo e massimo che l'asset ha toccato nel periodo di tempo fra apertura e chiusura della candela.

Le API di Kraken permettono di effettuare diversi tipi di operazioni, come:

- scaricare lo storico dei prezzi di tutte le coppie valuta-criptovaluta disponibili dal 2013
- restare in attesa per ricevere in tempo reale i nuovi dati sugli scambi di titoli
- creare il proprio portafogli sulla piattaforma e piazzare ordini di acquisto e vendita

Tutte funzionalità ben sfruttate da Sentyment, che opera sul mercato in autonomia.

1.3 Gudagnare con il trading

È importante notare che la ricchezza in possesso, calcolata in ogni istante, è data dalla quantità di budget e dalla quantità di titoli posseduti. Gli asset finanziari hanno un valore monetario indicato dal loro prezzo quindi, il possesso di alcuni di questi, anche se il budget da investire è azzerato, comporta, comunque, una ricchezza investita che si può recuperare rivendendo il titolo.

Si prende come esempio l'asset XBT/EUR, valuta di scambio fra Bitcoin e Euro. Partendo da un budget iniziale in Euro, bisogna decidere quando comprare titoli di Bitcoin e quando rivenderli per avere di nuovo Euro. Lo scopo della AI, quindi, non è soltanto massimizzare gli Euro in possesso, ma anche il numero di titoli Bitcoin, o del quantitativo di entrambe le monete da avere in portafoglio, che porti all'aumento del loro valore.

A seconda delle aspettative sul rialzo o ribasso di un titolo è possibile agire in due modi: andare *long* o andare *short*. Andare *long* vuol dire acquistare Bitcoin vendendo Euro, puntando ad un apprezzamento (aumento di valore) della criptovaluta; in questo caso, l'aspettativa è di un aumento del valore del cambio e si avrà, quindi,

una visione rialzista (bullish) del mercato. Andare *short* vuol dire vendere Bitcoin acquistando Euro, puntando, quindi, un apprezzamento dell'Euro nei confronti della criptomoneta. In questo caso, l'aspettativa è di una diminuzione di valore del cambio. Si aprirà, quindi, una posizione che mira ad una flessione del prezzo. La visione che il trader avrà del mercato sarà ribassista (bearish).

Il punto centrale su cui si basano le strategie di investimento è indovinare se la valuta aumenterà o diminuirà di prezzo, per poi andare long o short a seconda della previsione. L'*analisi tecnica* è un metodo riconosciuto e largamente utilizzato per anticipare, o prevedere, l'andamento dei prezzi e si basa sull'aspetto tecnico del mercato, utilizzando grafici e dati storici. In particolare, vengono analizzati prezzi, volumi scambiati e fasce temporali. Si utilizzano indicatori matematici, statistici e grafici. I segnali ed i suggerimenti ricavati da questi strumenti accompagnano i trader nelle loro decisioni in merito all'apertura e chiusura di posizioni di trading. L'*analisi tecnica* è un metodo di previsione dei prezzi basato sui dati storici.

Supponendo che il mercato ad un certo punto cresca e si abbia "indovinato" un buon numero di previsioni, ci si ritroverà a possedere dei titoli che ora valgono un prezzo superiore rispetto al loro valore iniziale. Se i titoli crescono di valore, è bene, quindi, acquistarne finchè salgono, in questo modo si possiede un maggior numero di asset che andranno a valere sempre di più; in caso di perdita di valore del titolo, invece, generalmente, si dovrebbe vendere i titoli, per riacquistare del budget investito che ora stava perdendo valore e poterlo investire in altri titoli che crescono.

Quando si decide di acquistare, si sta effettuando la seguente operazione: dato un certo budget iniziale *budget*, il valore del titolo *price*, e la quantità di titolo acquistato *equity* (Formula (1)) e supponendo di spendere sempre tutto il budget per acquistare il titolo

$$equity = budget / price \quad (1)$$

Mentre, all'occorrenza di un'operazione di vendita, utilizzando equity appena acquistata (Formula (2)):

$$budget = equity * price \quad (2)$$

È possibile, in ogni momento, calcolare il ricavo o quantità di beni in possesso combinando il budget con equity, tenendo presente che, dopo un acquisto, il budget scende a zero e equity assume il valore indicato mentre, dopo una vendita, è equity a scendere a zero e si ritorna in possesso di budget. Lo stesso budget riacquistato verrà usato nuovamente per comprare dei titoli, facendo così crescere di nuovo equity, e così via.

Se si considera un certo budget iniziale *initial_budget*, il valore dei titoli posseduti unito al budget in ogni istante, in questo documento chiamato *guadagno* (*gain*), è dato da:

$$gain = (budget + equity * price) - initial_budget \quad (3)$$

Questa formula viene riutilizzata nel Capitolo 4 nell'ambito dello sviluppo degli strumenti di test. Da notare che il valore complessivo potrebbe scendere sotto il budget iniziale e in questo caso si dovrebbe parlare di *perdita*, ma è comunque usato il termine *guadagno* per semplicità.

A questo punto, è necessario introdurre il concetto di *fee*.

Le varie piattaforme di trading addebitano un costo per ogni singola operazione, di acquisto e di vendita, intestandosi una percentuale di budget speso dai trader come tassa per mantenere la piattaforma stessa.

Nel caso di operazioni di acquisto, la tassa è calcolata come percentuale del budget speso per comprare il titolo, quindi, viene scalata subito ed il risultato è l'acquisto di una quantità leggermente minore di titoli (la restante dei quali viene pagata alla piattaforma). Per le vendite, invece, la fee è tolta dal budget una volta che questo è ricalcolato vendendo equity, risultando in un ricavo leggermente minore di quello che si sarebbe ottenuto vendendo realmente l'intera quantità di titoli.

Nel caso di Kraken, la piattaforma di trading di riferimento su cui opera Sentyment, le fee sono del 0.26%: significa che viene sottratto lo 0.26% di budget che si intende investire, per ogni operazione. Per altri mercati, valute o piattaforme di trading, le fee possono variare e potrebbero non essere basate su una percentuale di acquisto, ma fisse (flat fee vs per share fee). Kraken applica una fee sempre fissa.

La presenza delle fee modifica abbastanza drasticamente il funzionamento del mercato e l'efficacia delle strategie di investimento, che devono essere adattate ad un mercato con fee; è necessario, quindi riscrivere le formule per il guadagno e per le operazioni di acquisto e vendita titoli, che ora comprendono la percentuale di fee.

Per gli acquisti, considerando un certo budget iniziale *budget* (o il budget risultato di una precedente vendita), il valore del titolo *price*, la quantità di titolo acquistato *equity* e una tassa *fee*:

$$\text{equity} = (\text{budget} * (1 - \text{fee})) / \text{price} \quad (4)$$

Per le vendite:

$$\text{budget} = (\text{equity} * \text{price}) * (1 - \text{fee}) \quad (5)$$

Dal momento che le budget e equity sono già calcolati, togliendo le fee per ogni operazione di acquisto e vendita, la formula del guadagno rimane la stessa.

Senza addentrarsi nelle diversità dei titoli e dei mercati, si può affermare che una strategia generale, e semplificata, per guadagnare facendo trading è *compra basso, vendi alto*; nella realtà i segnali di acquisto possono essere dati da altri indicatori più complessi e non sempre è possibile capire quando un titolo sta toccando un prezzo particolarmente alto o basso: è facile analizzare i grafici a posteriori, ma non altrettanto semplice prevedere massimi e minimi locali online. Per questo, si utilizzano delle semplici strategie fondamentali che aiutano a capire l'andamento dei prezzi.

1.4 Esempi di strategie

Come già accennato, l'*analisi tecnica* è lo studio dell'andamento dei prezzi dei mercati finanziari nel tempo, allo scopo di prevederne le tendenze future, mediante, principalmente, metodi grafici e statistici. In senso lato, è quella teoria di analisi secondo cui è possibile prevedere l'andamento futuro del prezzo di un bene quotato, studiando la sua storia passata. Viene utilizzata, assieme all'*analisi fondamentale*, per la definizione delle decisioni di operatività finanziaria.

Volendo distinguere l'*analisi fondamentale* da quella *tecnica*, si può dire che, mentre l'*analisi tecnica* cerca di definire il prezzo futuro di un titolo, basandosi sugli aspetti formali dell'andamento delle quotazioni (derivati da grafici), l'*analisi fondamentale* si occupa di stabilire il prezzo corretto di un titolo in base alle caratteristiche economico-finanziarie intrinseche della società cui fa riferimento.

L'*analisi tecnica* si pone come obiettivo di analizzare, e comprendere, attraverso l'*analisi del grafico*, l'andamento dei prezzi il quale, a sua volta, rispecchia le decisioni degli investitori; si basa, inoltre, sull'*assunto fondamentale* che, poiché il comportamento degli investitori si ripete nel tempo, al verificarsi di certe condizioni grafiche, anche i prezzi si muoveranno di conseguenza. Il compito principale dell'*analisi tecnica* è, quindi, quello

di identificare un cambiamento di tendenza rispetto ad uno stadio iniziale, mantenendo una posizione di investimento fino a quando non vi sia prova che la tendenza stessa si sia di nuovo invertita.

Per individuare un trend si fa uso di indicatori tecnici, basati su prezzo e volume, calcolati a partire dai grafici: questi analizzano i movimenti di prezzo a breve termine e alcuni dei più noti ed usati sono Moving Average Crossover, MACD (moving average convergence / divergence), RSI (relative strength index) e Bollinger Bands.

Analizzando i grafici passati mediante queste strategie, è possibile ricavare i parametri per gli indicatori che meglio si adattano al mercato in analisi e riutilizzare l'indicatore, applicandolo ai dati futuri, per identificare nuovi trend.

1.4.1 Moving Average Crossover

Questa strategia prevede il calcolo di una o più medie mobili a diverso periodo, le cui intersezioni nel grafico definiscono un segnale di buy o sell.

Il moving average *crossover* si verifica quando, nel grafico di due medie mobili, ciascuna basata su diversi periodi, le linee di queste medie mobili si incrociano.

Non prevede la direzione futura ma mostra le tendenze. Questo indicatore utilizza due (o più) medie mobili, una più lenta ed una più veloce. La più veloce è una media mobile a breve termine. Per i mercati azionari di fine giornata, ad esempio, può essere un periodo di 5, 10 o 25 giorni, mentre, la più lenta è la media mobile di medio o lungo termine (ad esempio periodo di 50, 100 o 200 giorni). Una media a breve termine è più veloce perché considera i prezzi solo per un breve periodo di tempo ed è, quindi, più reattiva alle variazioni giornaliere dei prezzi. D'altra parte, una media mobile a lungo termine è considerata più lenta in quanto incapsula i prezzi per un periodo più lungo; tende, tuttavia, ad attenuare i disturbi che si riflettono spesso nelle medie mobili a breve termine.

Una moving average, come una linea a sé stante, viene spesso sovrapposta nei grafici per indicare l'andamento dei prezzi. Un crossover si verifica quando una media mobile più veloce (cioè una media mobile di periodo più breve) attraversa una media mobile più lenta (cioè una media mobile di periodo più lungo). Questo punto di incontro viene utilizzato per comprare o vendere titoli. Quando la media veloce supera quella lenta, si ha un segnale di *buy*, mentre *sell* se viceversa.



Figura 3: Andamento dei prezzi per il titolo TESLA/DOLLARO, accompagnato da due medie mobili a diverso periodo: fast_ma a 20 giorni mentre slow_ma a 100 giorni. Le frecce verdi e rosse indicano i segnali di buy e sell ricavati dalle intersezioni fra le medie.

Esistono diverse declinazioni della strategia a medie mobili, come ad esempio SMA (simple moving average), che utilizza come segnale di buy / sell il crossover fra i prezzi di chiusura e la media mobile semplice; EMA usa una media esponenziale; MACD (moving average convergence / divergence) usa come prima statistica la differenza fra la media veloce e quella lenta e, come seconda statistica, la media di queste differenze.

Calcolati gli indicatori statistici più adatti alla situazione, e scelti i parametri migliori (come i periodi delle medie da usare), si è costruito un modello di trend del titolo e si è in grado di prevedere quando il trend si ripropone. Prendendo l'esempio di una moving average crossover con una media veloce e una lenta, analizzato che per la maggior parte dei casi si riscontra un guadagno vendendo quando la media lenta supera quella veloce, e comprando quando accade l'opposto, allora, si può affermare che sarà molto probabile guadagnare ancora, in futuro, ripetendo le medesime operazioni.

1.4.2 Buy-Hold

Un altro esempio di strategia, molto più semplice di Moving Average Crossover e, in questo ambito, utilizzata più come base per confrontare l'efficacia di altre strategie, che come tecnica vera e propria per massimizzare i guadagni.

Buy and hold, comprare e detenere, chiamato anche trading di posizione, è una strategia di investimento in cui un investitore acquista titoli e li detiene a lungo, con l'obiettivo che i titoli aumentino gradualmente di valore per un lungo periodo di tempo. Ciò si basa sull'idea che, a lungo termine, i mercati finanziari offrano un buon tasso di rendimento, pur tenendo conto di un certo grado di volatilità. Questo punto di vista sostiene che la tempistica del mercato, ovvero il concetto che si può entrare nel mercato ai minimi e vendere ai massimi, non funziona; tentare tale tempismo dà risultati negativi, almeno per gli investitori piccoli o poco sofisticati quindi, è meglio per loro semplicemente acquistare e detenere.

I segnali di buy e sell si riducono, quindi, semplicemente, a uno solo: comprare all'inizio del periodo considerato (appena si vuole utilizzare la strategia) e non vendere *mai*. Supponendo che il valore del titolo continui sempre a crescere, sul lungo periodo, si avrà un aumento delle *equity* acquistate e, quindi, un aumento della ricchezza posseduta. Infatti, facendo riferimento alla formula del guadagno (Formula (3)), se *equity* sale e *budget* rimane sempre a 0, il guadagno sale, essendo direttamente proporzionale al prezzo del titolo posseduto. La strategia, pertanto, segue l'andamento del mercato: se il valore del titolo sale, il guadagno associato sale mentre, in corrispondenza di un calo, anche la ricchezza posseduta nell'investimento scende.

Un vantaggio introdotto da questa strategia è la quasi totale assenza di costi dovuti alle fee. Effettuando una sola operazione di buy iniziale, in cui la fee viene pagata, non è più necessario vendere o comprare ulteriormente e, quindi, sempre nell'ottica di un aumento di valore del titolo a lungo termine, il guadagno aumenta senza dover spendere ulteriore capitale in fee associate ad altre operazioni.

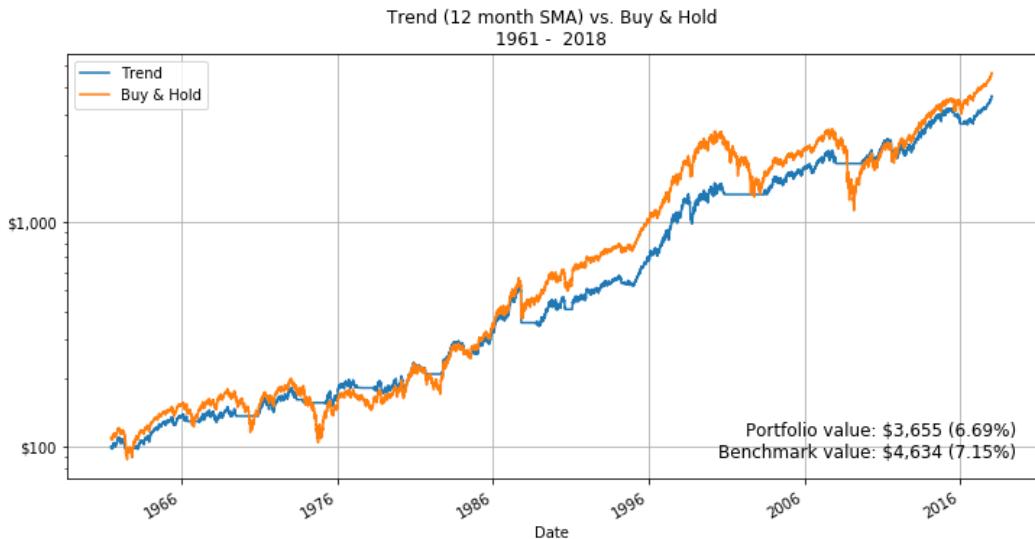


Figura 4:

Esempio dell’andamento di prezzi del titolo S&P 500, rappresentato con una media mobile a 12 mesi, e valore della ricchezza prodotto da una strategia buy-hold. La media ammortizza la variazione di prezzo e rimane più costante, mentre buy-hold si vede assumere tutti i valori di prezzo del titolo. Come si può notare, buy-hold segue esattamente l’andamento del mercato: il valore del portafoglio (guadagno, Formula 3) dipende dal valore del titolo. Analizzando i prezzi dal 1966 al 2016 si nota una lenta salita e quindi buy-hold si rivela comunque efficace.

In questo ambito e nel trading a breve termine, tuttavia, la strategia è utilizzata quasi esclusivamente come *benchmark*, metodo di confronto per le vere strategie che si applicano nel contesto di trading, o si sviluppano ed implementano nei vari algoritmi di AI. Il metodo di confronto di due strategie è trattato nel Capitolo 4 ma, semplificando molto, si può affermare che, dal punto di vista esclusivamente del guadagno totale, considerando due strategie che partono e terminano entrambe nello stesso momento, una è migliore dell’altra se porta ad un guadagno finale maggiore. Nella realtà sono usati diversi indicatori aggiuntivi, oltre al guadagno, e va definito quale ha maggiore importanza per i fini dell’investitore. Una strategia potrebbe rivelarsi migliore di altre se si considera il guadagno, mentre potrebbe risultare la peggiore se la metrica usata è, per esempio, il rischio di perdita.

Ad ogni modo, qualsiasi sia il metro di paragone per il confronto delle strategie, buy-hold è la più semplice che si possa concepire e, nelle giuste condizioni, si rivela

spesso anche quella più redditizia. Se si volesse creare una nuova strategia sarebbe bene confrontarla con i risultati di buy-hold.

1.5 Ruolo delle intelligenze artificiali e scopo della tesi

L'apprendimento automatico e diverse tecniche hanno creato nuovi sistemi per individuare pattern, cosa che il cervello umano non è in grado di fare. Dal momento che la finanza è quantitativa, la AI, nel trading azionario, sta guadagnando terreno. Le società finanziarie hanno investito molto nell'intelligenza artificiale in passato e studiano ed implementano le applicazioni finanziarie di machine learning e del deep learning nelle loro operazioni.

L'analisi tecnica è, dunque, implementata in sistemi automatici, che sono molto più efficienti di un trader umano ed offrono diversi vantaggi. Le AI separano le informazioni prevedibili da qualsiasi "rumore casuale", gli algoritmi apprendono dall'accuratezza delle previsioni precedenti e si adeguano continuamente, abbastanza veloci da prevedere il mutamento della situazione del mercato, anche in un orizzonte di pochi giorni. Possono imparare dai successi e fallimenti e riconfigurare, ogni giorno, le approssimazioni del funzionamento interno del mercato, poiché viene alimentato con nuovi dati. Oltre agli indiscutibili vantaggi computazionali di una macchina rispetto al cervello, ci sono altri aspetti da considerare che hanno portato alla crescita delle AI nel settore. Si può infondere la conoscenza degli esperti dell'ambito in un software che si auto adatta e migliora, senza avere gli svantaggi dell'errore umano; si può estendere per permettergli di gestire diversi ambiti, dal recupero della materia prima (i dati di trading) fino alla vendita del prodotto finito (vendita di strategie di investimento). È possibile costruire sistemi complessi che modellano intere realtà.

Sentyment è uno di questi software descritti. Si occupa di raccogliere dati, elaborarli e, attraverso una AI, li analizza per produrre strategie di investimento. Il modulo di intelligenza artificiale è molto ampio: esistono molte AI che operano producendo costantemente nuovi segnali di buy / sell per le strategie e sono monitorate da una ulteriore, singola AI. Questo supervisore, *meta-learner*, decide quale, fra le molte AI, sta avendo più successo e la promuove come attiva da utilizzare.

Lo scopo della tesi è sviluppare il supervisore ed inserirlo all'interno del complesso sistema di Sentyment, impiegando tecniche di test per intelligenze artificiali che permettano di scegliere un rappresentante migliore fra una serie di agenti che operano indipendentemente.

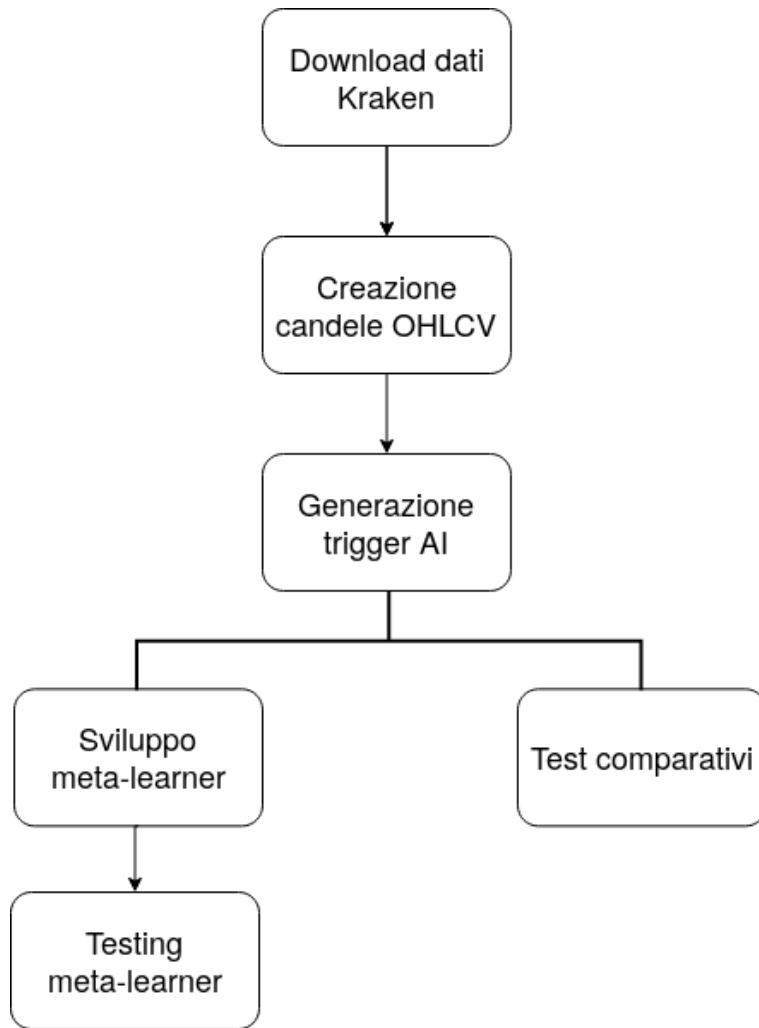
**Figura 5:**

Diagramma delle attività svolte, spiegate nel dettaglio in ogni relativo capitolo. La fase preliminare di download dati comporta la successiva analisi esplorativa degli stessi e fornisce l'input per il resto del sistema. Il passo successivo è la creazione di candele OHLCV, cioè il tipo di dato utilizzato dalle AI. A questo punto la AI Sentyment può mettere in moto i suoi algoritmi per produrre le strategie di investimento, ovvero l'elenco di operazioni buy / sell (*trigger*) per ogni istante di tempo. Da qui si sono svolte due differenti attività: l'analisi dei risultati prodotti dalla AI (test comparativi) e lo sviluppo della AI supervisore (*meta-learner*), con successiva analisi e testing dei suoi risultati.

Capitolo 2

AI per financial prediction e Sentyment

La predizione del mercato azionario è stata una questione importante nel campo della finanza, ingegneria e matematica, a causa del suo potenziale guadagno finanziario. Data la grande quantità di capitale negoziato attraverso il mercato azionario, questo è visto come una grande opportunità ed ha richiamato molti investimenti. La previsione del mercato, inoltre, comporta la sfida di dimostrare se il mercato finanziario sia prevedibile oppure no. Dal momento che non c'è stato consenso sulla validità dell'ipotesi che nel mercato ci sia spazio per la predizione, i ricercatori hanno molto faticato per dimostrarne la prevedibilità[1]. L'*ipotesi del mercato efficiente* (EMH: Efficient Market Hypothesis) è un'ipotesi in economia finanziaria che afferma che i prezzi delle attività riflettono tutte le informazioni disponibili. Una conseguenza diretta è che è impossibile "battere il mercato" in modo coerente basandosi sul rischio poiché i prezzi di mercato dovrebbero reagire solo a nuove informazioni. È una teoria che fa sorgere molte controversie e, come tale, non ancora dimostrata, il che lascia spazio ai tentativi di sviluppo di un algoritmo che sia in grado di prevedere parte dell'andamento dei mercati.

Con l'avvento di computer più veloci e vaste informazioni su Internet, i mercati azionari sono diventati più accessibili agli investitori strategici o al pubblico generale. Internet fornisce la fonte principale di informazioni ed ha un impatto significativo sui mercati azionari, e le tecniche di estrazione ed utilizzo delle informazioni a supporto del processo decisionale hanno raggiunto un'importanza critica. Per prevedere il mercato azionario con precisione, vari algoritmi e modelli di predizione sono stati proposti da molti ricercatori, sia accademici che nell'industria.

2.1 Stato dell'arte delle AI

Da quando il mercato azionario è stato introdotto per la prima volta, molti hanno tentato di prevederne il comportamento utilizzando vari strumenti computazionali come *Linear egression* (LR), *Neural Network* (NNs), *Genetic Algorithm* (GA), *Support Vector Machine* (SVMs), *Case-based Reasoning* (CR) e altri.

Nell'ultimo decennio, Neural Network sono state ampiamente utilizzate, ed in molti casi hanno mostrato prestazioni migliori rispetto ad altri approcci. Una *rete neurale* (NN "Neural Network"), è un modello matematico/informatico di calcolo basato sulle reti neurali biologiche. Tale modello è costituito da un gruppo di interconnessioni di informazioni costituite da neuroni artificiali e processi che utilizzano un approccio di connessionismo di calcolo. Nella maggior parte dei casi, una rete neurale artificiale è un sistema adattivo che cambia la propria struttura in base ad informazioni esterne, o interne, che scorrono attraverso la rete stessa durante la fase di apprendimento.

Il primo modello di previsione del mercato azionario basato su Neural Network è stato implementato da White[2]. Ha usato *Feed Forward Neural Networks* (FFNN)¹ per decodificare regolarità precedentemente non rilevate nei movimenti di prezzo degli asset, come fluttuazioni di prezzo di azioni ordinarie, ed ha mostrato come cercare tali regolarità usando FFNNs. Dal tentativo iniziale di White, un numero di ricercatori ha partecipato allo sviluppo di un modello di previsione del mercato azionario accurato. Phua et al. [3] ha utilizzato Neural Network con Genetic Algorithm per prevedere l'indice di scambio del mercato azionario di Singapore², ottenendo un tasso di accuratezza dell'81%. Kim e Han [4] hanno anche combinato Neural Network con Genetic Algorithm e pronosticato Korea Composite Stock Index 200³. Hanno raggiunto l'82% di accuratezza nel prevedere sia le tendenze in aumento ed in calo del mercato azionario.

Genetic Algorithm è stato studiato e dimostrato essere efficace nell'esplorare uno

¹Una rete neurale feed-forward ("rete neurale con flusso in avanti") o rete feed-forward è una rete neurale artificiale dove le connessioni tra le unità non formano cicli, differenziandosi dalle reti neurali ricorrenti. In questa rete neurale le informazioni si muovono solo in una direzione, avanti, rispetto a nodi d'ingresso, attraverso nodi nascosti (se esistenti) fino ai nodi d'uscita. Nella rete non ci sono cicli. Le reti feed-forward non hanno memoria di input avvenuti a tempi precedenti, per cui l'output è determinato solamente dall'attuale input.

²L'indice FTSE Straits Times (STI) è un indice di borsa ponderato per la capitalizzazione che è considerato l'indice di riferimento per il mercato azionario di Singapore. Tiene traccia delle prestazioni delle prime 30 società quotate alla Borsa di Singapore.

³Korea Composite Stock Index (KOSPI): indice di borsa azionario della Corea del Sud, che ospita più di 780 società quali Samsung, Hyundai e Kia motors

spazio complesso in modo adattivo, guidato dal meccanismo evolutivo di riproduzione, crossover e mutazione.

Un *algoritmo genetico* è un algoritmo euristico⁴ utilizzato per tentare di risolvere problemi di ottimizzazione per i quali non si conoscono altri algoritmi efficienti di complessità lineare o polinomiale. L'aggettivo "genetico", ispirato al principio della selezione naturale ed evoluzione biologica, deriva dal fatto che gli algoritmi genetici attuano dei meccanismi concettualmente simili a quelli dei processi biochimici scoperti dalla biologia.

In sintesi, gli algoritmi genetici consistono in algoritmi che permettono di valutare diverse soluzioni di partenza (come se fossero diversi individui biologici) e che, ricombinandole ed introducendo elementi di disordine (mutazioni genetiche casuali), producono nuove soluzioni che vengono valutate scegliendo le migliori nel tentativo di convergere verso soluzioni "di ottimo". Nonostante questo utilizzo nell'ambito dell'ottimizzazione, data la natura intrinsecamente casuale dell'algoritmo genetico, non vi è modo di sapere, a priori, se sarà effettivamente in grado di trovare una soluzione accettabile al problema considerato. La soluzione trovata sarà frutto di una procedura casuale e, quindi, difficilmente riproducibile.

Di recente, molti ricercatori hanno impiegato Genetic Algorithm per migliorare le prestazioni delle tecniche di AI. Nell'ambito delle Neural Network, Genetic Algorithm è comunemente usato per selezionare la topologia di rete e determinare il numero ottimale di layer nascosti e di elementi di elaborazione.

Molti ricercatori hanno confrontato le Neural Network con approcci statistici per il riconoscimento di pattern [5][6]. Yoon et al. [7] ha sottolineato la superiorità di Neural Network sull'analisi discriminante classica. Garliauskas [5] ha, inoltre, esaminato la previsione del mercato azionario utilizzando Neural Network in combinazione con un approccio alle *funzioni kernel*⁵ ed il metodo dell'errore di previsione ricorsivo⁶. Garliauskas [5] ha concluso che, nel prevedere le serie temporali finanziarie, le Neural

⁴Un algoritmo euristico, o *euristica*, è un particolare tipo di algoritmo progettato per risolvere un problema più velocemente, nel caso in cui i metodi classici siano troppo lenti nel calcolo (ad esempio, in caso di elevata complessità computazionale) o per trovare una soluzione approssimata, nel caso in cui i metodi classici falliscano nel trovare una soluzione esatta.

⁵Le funzioni kernel sono una classe di algoritmi per l'analisi di pattern, schemi, il cui elemento maggiormente conosciuto sono le Support Vector Machine (SVM). Lo scopo generale dell'analisi di pattern è di trovare, e studiare, tipi generici di relazioni. Le funzioni kernel si appropiano al problema mappando i dati in uno spazio di caratteristiche multidimensionale, trasformando i dati in un insieme di punti dello spazio euclideo.

⁶Errore di predizione: è una misura di quanto bene il modello prevede la variabile di risposta, o di come i campioni sono classificati nella categoria corretta.

Network hanno prestazioni migliori rispetto ai metodi statistici classici. In generale, numerosi studi hanno dimostrato che le Neural Network hanno la capacità di prevedere i mercati azionari in modo più accurato rispetto ad altri metodi [7][8][9][10].

2.1.1 Supporto di informazioni relative agli eventi

Sebbene gli studi precedenti fossero limitati ad analisi quantitativa, diversi studi recenti sono stati eseguiti sulla base dell'analisi qualitativa. Molti credevano che l'integrazione di una conoscenza degli eventi nelle Neural Network risultasse promettente per una migliore previsione del mercato azionario [12][13][14][15]. Kohara [14] ha studiato alcune metodologie per migliorare i modelli predittivi multivariati⁷ nella previsione del prezzo delle azioni utilizzando Neural Network e conoscenza pregressa. Ha usato le informazioni dai titoli dei giornali per migliorare la capacità di previsione. I loro risultati sperimentali hanno dimostrato che l'utilizzo della conoscenza degli eventi unita alle Neural Network ha ridotto significativamente il tasso di errore di previsione intorno al livello di significatività del 5% con profitto del 40%.

Inoltre, Hong e Han [12] hanno confrontato le Neural Network con informazioni sugli eventi rispetto al modello *Random Walk* (RW)⁸ e rispetto a Neural Network senza informazioni sull'evento. Neural Network con informazioni sull'evento hanno una media dello 0,527% di errore, che è un errore minore rispetto ad altri modelli. Hanno dimostrato che le previsioni basate su Neural Network sono notevolmente superiori a Random Walk e che l'effetto dell'informazione sugli eventi esiste.

Nella previsione mercato azionario, inoltre, molti ricercatori hanno riconosciuto che fattori qualitativi, quali effetti politici ed eventi internazionali, hanno svolto un ruolo molto importante. Hanno, dunque, proposto che un sistema di previsione del mercato azionario debba incorporare fattori sia quantitativi che qualitativi. I loro studi hanno, infine, dimostrato che le Neural Network che si basano sui fattori sia quantitativi che qualitativi sono di gran lunga superiori a quelle basati solo su fattori quantitativi [16][17].

⁷Il modello multivariato è un comune strumento statistico che utilizza più variabili per prevedere i possibili risultati. Gli analisti della ricerca utilizzano modelli multivariati per prevedere i risultati degli investimenti in diversi scenari, al fine di comprendere l'esposizione che un portafoglio ha a particolari rischi. Ciò consente, ai gestori di portafoglio, di mitigare meglio i rischi identificati attraverso l'analisi della modellazione multivariata. La *simulazione Monte Carlo* è un modello multivariato ampiamente utilizzato che crea una distribuzione di probabilità che aiuta a definire una gamma di possibili risultati di investimento.

⁸In matematica, una passeggiata aleatoria (Random Walk) è la formalizzazione dell'idea di prendere passi successivi in direzioni casuali. Matematicamente parlando, è il processo stocastico più semplice.

2.1.2 Punti chiave nelle predizioni finanziarie

Numerosi ricercatori hanno mostrato il loro ufficiale punto di vista sull'*ipotesi del mercato efficiente* (EMH: efficient market Hypothesis) in ambito accademico [18][19]. EMH afferma che l'attuale prezzo di mercato riflette l'assimilazione di tutte le informazioni disponibili[20]. L'informazione rilevante per un mercato è contenuta nei prezzi e, ogni volta che sorgono nuove informazioni, il mercato corregge se stesso e le assorbe, quindi, il mercato è efficiente e non c'è spazio per la predizione[21].

È in corso un lungo dibattito sulla validità di EMH, ma non è stato raggiunto un consenso. Negli ultimi anni, molti i ricercatori hanno affermato che, sicuramente, EMH deve essere falso[21]. Diversi studi sono stati condotti sui dati dei mercati azionari al fine di dimostrare che il mercato è prevedibile. Se qualche sistema di calcolo avesse la capacità di mostrare previsioni con ragionevole accuratezza, basandosi sui dati storici di mercato, allora la validità di EMH potrebbe essere messa in discussione. Tsibouris e Zeidenberg[22] hanno utilizzato le Neural Network per prevedere il mercato azionario basandosi soltanto sui prezzi passati come input. I loro risultati empirici hanno mostrato un certo livello di capacità predittiva e respinto la forma debole di EMH⁹.

Eyden [23] ha sviluppato, inoltre, un sistema che modella le prestazioni della borsa azionaria di Johannesburg ed il loro sistema ha fornito prove significative dimostrandone la sua capacità di prevedere le direzioni del mercato azionario, quindi, di nuovo, dissentendo da EMH.

Dalla parte opposta, invece, vari studi sono basati sull'*ipotesi di mercato efficiente*. Fung et al. [13] ha indagato sull'immediato impatto delle notizie di giornale sulle serie storiche basate su EMH. Hanno presentato un sistema basato su EMH per prevedere il comportamento futuro del mercato azionario utilizzando informazioni non quantificabili (notizie dai giornali) e le previsioni sono state fatte solo in base al contenuto dell'articolo. Alla fine, il loro risultato empirico ha indicato che l'approccio basato su EMH è più redditizio rispetto alla semplice strategia di trading basata su Buy-Hold. Come è stato discusso, diversi studi hanno concluso di accettare, o confutare, EMH. La questione dell'efficienza del mercato, dunque, non è ancora pienamente indagata e porta alla necessità di ulteriori ricerche.

Un certo numero di ricercatori ha usato la cronologia dei dati numerici delle serie

⁹La forma debole dell'*ipotesi di mercato efficiente* (EMH, efficient market hypothesis) afferma che i prezzi passati, i valori storici e le tendenze non possono prevedere i prezzi futuri. I prezzi delle azioni riflettono soltanto tutte le informazioni attuali

temporali per la previsione dei mercati azionari ed ha raggiunto una ragionevole accuratezza delle previsioni rifiutando l'ipotesi di EMH [23][22].

Tuttavia, ci sono vari fattori che influenzano i prezzi delle azioni, come le prestazioni e la robustezza di un'azienda, le tendenze del mercato, la psicologia degli investitori, il coinvolgimento del governo, i cambiamenti nell'attività economica e così via. Molti ricercatori, quindi, hanno concordato sull'esistenza di una correlazione significativa tra gli eventi che rappresentano i fattori sopra citati ed i mercati azionari [13][14]. Hanno utilizzato le informazioni relative agli eventi in aggiunta a quelle numeriche delle serie temporali e raggiunto un certo livello di precisione nelle previsioni. Poiché, però, ogni evento ha un diverso livello di impatto in termini di intervallo di tempo in cui l'evento stesso può influenzare il mercato, ogni evento dovrebbe essere ponderato in base al suo livello di impatto, che è definito sulla base della conoscenza preliminare costruita in precedenza. In aggiunta, il web è visto come la principale fonte di informazione sugli eventi per la previsione di mercato, contenendo le più aggiornate e preziose informazioni. Sono necessarie, pertanto, tecniche automatiche di *web mining*¹⁰ per ottenere un livello più elevato di accuratezza della previsione in tempi brevi.

¹⁰Il web mining è la raccolta sistematica di dati provenienti dal web, organizzati e visualizzati al fine di far emergere informazioni utili. Prevede l'impiego di tecniche di data mining al fine di estrarre dati da server e pagine web.

2.1.3 Metodi di previsione

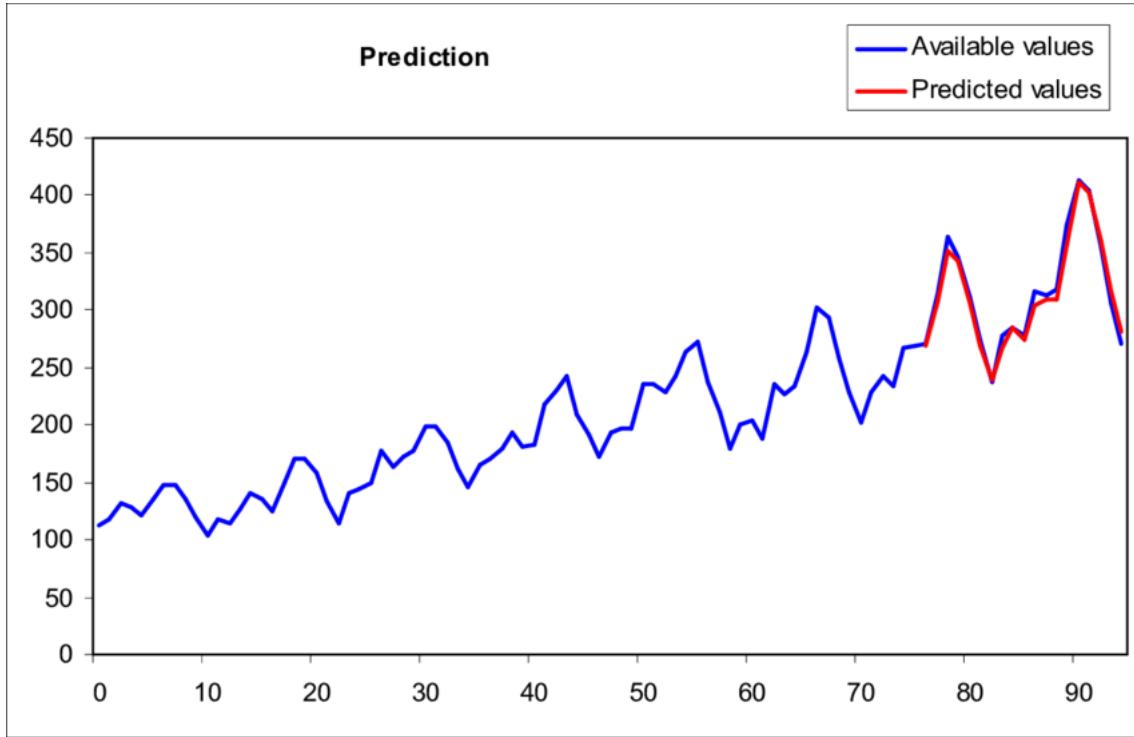


Figura 6:

Esempio di utilizzo di una tecnica per previsione. Si suppone di analizzare il grafico dell’andamento di un titolo (serie storica), in cui le ascisse sono il tempo e le ordinate il prezzo dell’asset. Come si può notare, molto spesso, nelle serie storiche finanziarie, i pattern si ripetono nel tempo e, applicando metodi di regressione o intelligenza artificiale, è possibile prevedere quando questi si ripresenteranno.

Predizione tradizionale di serie storiche

I modelli statistici tradizionali sono ampiamente utilizzati in economia per la previsione di serie storiche¹¹. Questi modelli sono in grado di modellare relazioni lineari tra fattori che influenzano il mercato ed il valore del mercato. In economia, ci sono due tipi di previsione delle serie: univariata (regressione semplice) e multivariata (regressione multivariata) [19]. Un esempio ampiamente usato di modello univariato

¹¹In statistica descrittiva, una serie storica o temporale ("time series") si definisce come un insieme di variabili casuali ordinate rispetto al tempo, ed esprime la dinamica di un certo fenomeno nel tempo. Le serie storiche vengono studiate sia per interpretare un fenomeno, individuando componenti di trend, di ciclicità, di stagionalità e/o di accidentalità, sia per prevedere il suo andamento futuro.

è Box-Jenkins¹², che contiene solo una variabile nell'equazione ricorrente. Mostra un processo complesso di adattamento dei dati ai parametri appropriati del modello. Nelle loro equazioni sono inclusi i valori passati delle medie mobili dei prezzi.

Sebbene Box-Jenkins mostri una buona capacità di previsione a breve termine, richiede una grande quantità di dati per fornire un'alta precisione [24]. I modelli multivariati sono modelli univariati ampliati per "scoprire i fattori causali che incidono sul comportamento dei dati" [25]. Come implica il nome, l'equazione del modello contiene più di una variabile. Un certo numero di studi ha confrontato il modello statistico multivariato con Neural Network. Sebbene i modelli multivariati siano stati ampiamente utilizzati per prevedere i mercati azionari, diverse tecniche di machine learning stanno ora sostituendo i loro ruoli.

In letteratura, molti ricercatori hanno affermato che le Neural Network superano notevolmente i metodi statistici tradizionali [5][6]. Lawrence [1] ha confrontato le Neural Network con tecniche statistiche e di regressione. Ha usato un sistema che utilizza Neural Network con Genetic Algorithm. Il sistema con le AI ha mostrato la sua capacità di prevedere correttamente i movimenti del mercato il 92% delle volte, mentre Box-Jenkins ha mostrato un tasso di precisione soltanto del 60%. Diversi sistemi che utilizzano Neural Network, inoltre, hanno mostrato prestazioni costantemente migliori rispetto a modelli di regressione lineare multipla [26] [27]. Una delle ricerche più peculiari è stata eseguita da Yoon et al. [7]. Il loro sistema basato su Neural Network ha predetto la tendenza del prezzo delle azioni con il 91% di precisione, rispetto al 74% prodotto utilizzando l'*analisi discriminante multipla*¹³. Così, secondo loro, si può concludere che Neural Network hanno costantemente prestazioni migliori rispetto alle tecniche statistiche e di regressione.

¹²Il modello autoregressivo a media mobile, detto anche ARMA, è un tipo di modello matematico lineare che fornisce, istante per istante, un valore di uscita basandosi sui precedenti valori in entrata ed in uscita. A volte denominato modello di Box-Jenkins, dal nome dei suoi inventori, viene utilizzato, in statistica, per lo studio delle serie storiche dei dati.

¹³Multiple Discriminant Analysis (MDA) è una tecnica statistica di riduzione della dimensionalità multivariata. È utilizzata dai pianificatori finanziari per valutare i potenziali investimenti quando devono essere prese in considerazione diverse variabili. Questa tecnica riduce le differenze tra alcune variabili in modo che possano essere confrontate con un'altra variabile. L'analisi discriminante multipla è correlata all'analisi discriminante, che aiuta a classificare un set di dati impostando una regola o selezionando un valore che fornirà la separazione più significativa.

Neural Network

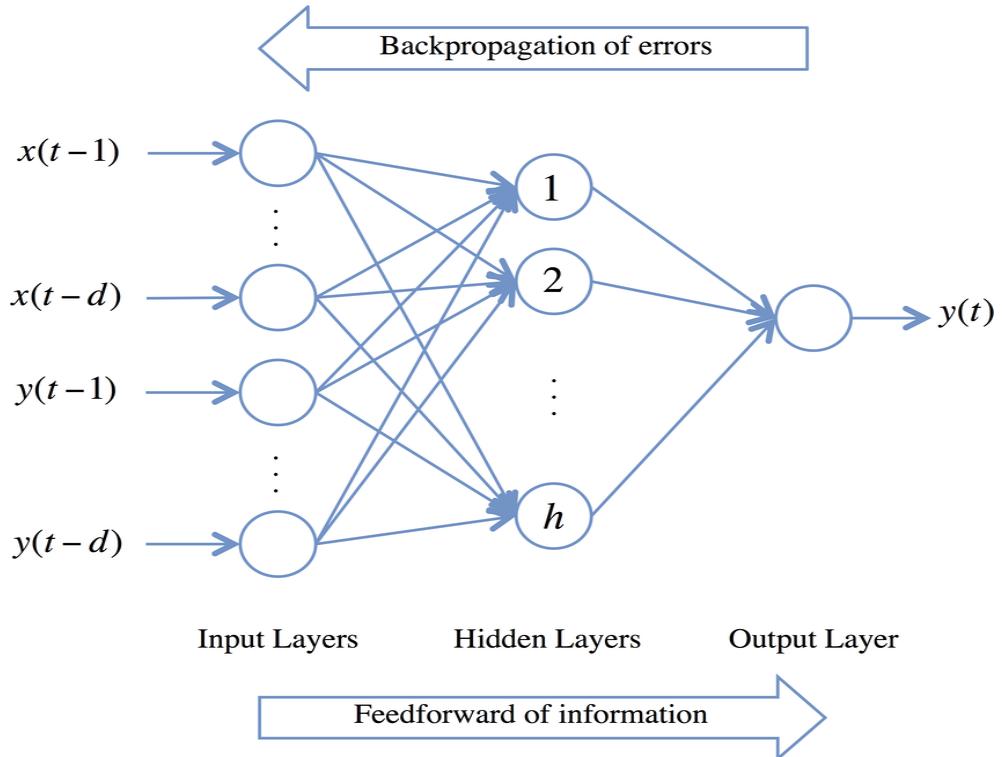


Figura 7:

Schema semplificato di Backpropagation Neural Network. Una Neural Network è un'interconnessione di un gruppo di nodi chiamati neuroni. Ha un livello di neuroni di input, un numero variabile di livelli intermedi (*hidden layers*) e un livello di output. I segnali si propagano dagli input attraverso gli strati nascosti, fino all'output. Nel caso di backpropagation, durante il training, i pesi, e gli altri parametri della rete sono modificati propagando l'errore dall'output verso gli input.

In letteratura è stato dimostrato che Neural Network offre la capacità di prevedere le direzioni del mercato in modo più preciso rispetto ad altre tecniche esistenti. La capacità delle Neural Network di scoprire relazioni non lineari tra le coppie input / output di training li rendono ideali per la modellazione con sistemi dinamici non lineari come i mercati azionari [1]. Uno dei vantaggi è la capacità di poter apprendere le relazioni attraverso i dati stessi, anziché considerare la forma funzionale della relazione. Dato che le Neural Network sono conosciute come approssimatori universali, ogni relazione può essere modellata con un qualsiasi grado di precisione, quando sufficienti dati vengono forniti per la modellazione. Forniscono, inoltre, un livello di

tolleranza al rumore ed alla rappresentazione incompleta dei dati.

Un altro vantaggio è che Neural Network ha proprietà di apprendimento adattive non lineari, non parametriche, ed ha un effetto molto pratico nella modellazione e predizione. La loro natura non lineare mostra un grande potenziale per risolvere molti problemi complessi. A causa delle caratteristiche di cui sopra dei mercati azionari, le Neural Network possono essere applicate alla predizione del mercato azionario. In primo luogo, è difficile modellare i dati di borsa a causa della loro complessità, quindi, il modello non lineare è vantaggioso. È spesso richiesto, inoltre, un ampio set di serie temporali correlate in input per modellare uno specifico asset.

Parlando, invece, degli aspetti negativi, Neural Network ha il problema della scatola nera, cioè che non rivela il senso ed il valore di ogni variabile ed il modo in cui pesa le variabili indipendenti[1]. Dato che il ruolo individuale di ogni variabile non può essere determinato, è impossibile da capire come la rete produce il prezzo futuro delle azioni.

Un altro grosso problema, con le Neural Network, è il sovrallenamento. Quando le reti si adattano troppo bene ai dati, il sistema perde la capacità di generalizzare. Siccome la capacità di generalizzazione è fondamentale per prevedere i prezzi futuri, il sovrallenamento è un problema serio. Questo, di solito, avviene per due motivi principali: quando le Neural Network hanno troppi nodi o periodi di training troppo lunghi (*epoch*¹⁴). Il sovrallenamento, tuttavia, può essere prevenuto eseguendo procedure di test e train o *cross validation*.

Per quanto riguarda il rumore tremendo, e le caratteristiche non stazionarie nei dati di borsa, Lawrence et al. [1] ha sottolineato che, quando l’addestramento delle Neural Network è difficile per dati molto rumorosi, le reti cadono in una soluzione ingenua come prevedere sempre l’output il più comune. Le Neural Network, inoltre, hanno alcune limitazioni nell’apprendimento dei pattern quando i dati di input hanno un’alta dimensionalità. Dash e Liu [28] pongono l’accento sulla selezione delle *feature*¹⁵ e suggeriscono che ridurre il numero di variabili in input, a volte, porta ad un miglioramento di prestazioni del modello per un determinato set di dati. La riduzione e trasformazione delle feature irrilevanti, o ridondanti, possono ridurre il tempo di esecuzione e produrre risultati più generalizzati [28].

¹⁴Un epoch è quando un *intero* dataset di training viene passato in avanti ed indietro attraverso la rete neurale solo una volta per allenarla.

¹⁵Nell’apprendimento automatico e nel riconoscimento di pattern, una ”feature” (caratteristica) è una proprietà misurabile individuale o caratteristica di un fenomeno osservato. La scelta di feature informative, discriminanti ed indipendenti è un passaggio cruciale per algoritmi efficaci nel riconoscimento, nella classificazione e nella regressione dei modelli. Il concetto di feature è correlato a quello della variabile esplicativa utilizzata nelle tecniche statistiche come la regressione lineare.

Ricerche recenti tendono a utilizzare diverse tecniche di AI *ibride*¹⁶. Hiemstra [29] ha proposto *sistemi esperti fuzzy*¹⁷ per prevedere i rendimenti del mercato azionario. Suggerì che, combinando le Neural Network con la logica fuzzy, si possa catturare la complessità del mapping funzionale e non è richiesta la specifica della funzione da approssimare. Tsaih et al. [30] ha integrato la tecnica basata su regole e le Neural Network per prevedere la direzione di cambio dello stock S&P 500¹⁸ *futures*¹⁹ su base giornaliera.

Support Vector Machine

Support Vector Machine (SVM) basato su teoria dell'apprendimento statistico è stato sviluppato da Vapnik [31] ed i suoi colleghi alla fine degli anni '70. È diventato un argomento caldo di studio intensivo grazie al suo successo applicato a compiti di classificazione e regressione, soprattutto nella previsione di serie temporali e applicazioni correlate alla finanza [44]. Support Vector Machine è una tipologia molto specifica di algoritmi di apprendimento caratterizzato dal controllo della capacità della funzione di decisione, l'uso delle funzioni del kernel e la dispersione della soluzione [33]. Fondato sulla teoria del *principio di minimizzazione del rischio strutturale*²⁰, in cui stima una funzione minimizzando l'errore di generalizzazione, Support Vector Machine si dimostra alla fine molto resistente al problema del sovrallenamento (*overtraining*), raggiungendo un ottimo risultato nella generalizzazione.

¹⁶Un sistema intelligente *ibrido* indica un sistema software che impiega, in parallelo, una combinazione di metodi e tecniche provenienti da diversi sottocampi di intelligenza artificiale, come, ad esempio: *neuro-fuzzy systems*, *evolutionary neural network*, *genetic fuzzy systems*, *reinforcement learning* con *fuzzy*, *neural*, o *evolutionary*, etc.

¹⁷La logica fuzzy è una logica in cui si può attribuire a ciascuna proposizione un grado di verità diverso da 0 e 1, e compreso tra di loro. È una logica polivalente, ossia un'estensione della logica booleana. Con grado di verità, o valore di appartenenza, si intende quanto è vera una proprietà: questa può essere, oltre che vera (= a valore 1) o falsa (= a valore 0), come nella logica classica, anche parzialmente vera e parzialmente falsa.

¹⁸L'indice Standard & Poor 500, noto come S&P 500 o semplicemente S&P, è stato realizzato da Standard & Poor's nel 1957 e segue l'andamento di un paniere azionario formato dalle 500 aziende statunitensi a maggiore capitalizzazione.

¹⁹I futures (future contracts) sono contratti a termine standardizzati per poter essere negoziati facilmente in una borsa valori, e sanciscono l'impegno ad un acquisto differito ad un prezzo prefissato.

²⁰La minimizzazione del rischio strutturale (SRM) è un principio induttivo in uso nell'apprendimento automatico. Comunemente, in machine learning, un modello generalizzato deve essere selezionato a partire da un set di dati finito, con il conseguente problema di overfitting: il modello diventa troppo fortemente adattato alle particolarità dei dati di training e generalizza male i nuovi dati. Il principio SRM affronta questo problema bilanciando la complessità del modello con il suo successo nell'adattare i dati di addestramento.

Un'altra proprietà chiave è che addestrare Support Vector Machine equivale a risolvere un problema di programmazione linearmente vincolato. Pertanto, la soluzione di Support Vector Machine è relativamente unica e globalmente ottimale, a differenza del training delle Neural Network, che richiede un'ottimizzazione non lineare con pericolo di rimanere bloccati ai minimi locali.

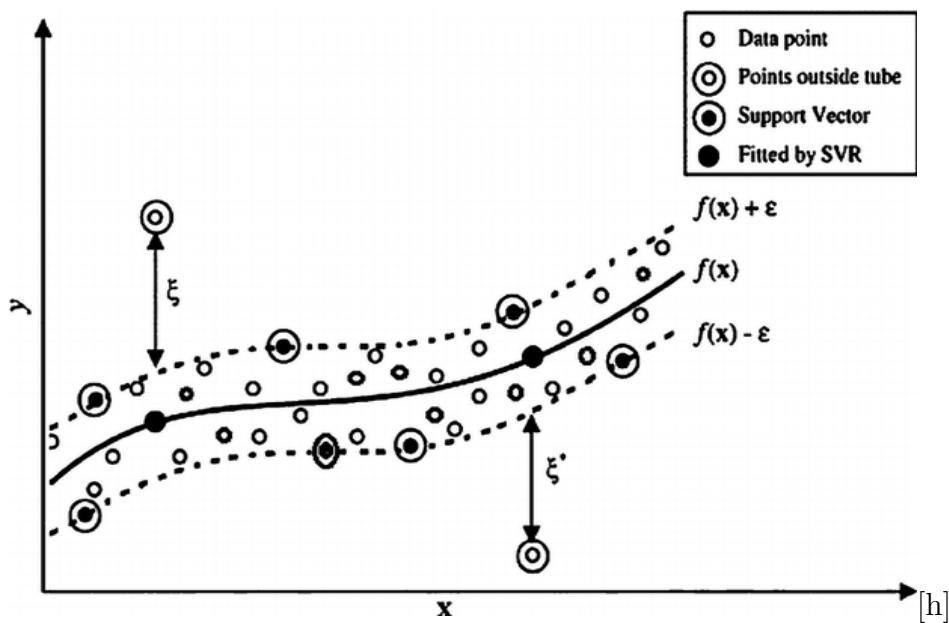


Figura 8:

Support Vector Machine applicato al compito di regressione: nel caso di funzioni non lineari, queste vengono mappate in spazi multidimensionali per mezzo di metodi kernel. Dopodichè viene applicata la tecnica dei vettori di supporto per delineare gli estremi della funzione da approssimare. Si ottiene un'approssimazione che, in ogni punto, discosta al massimo di ε dai valori noti della funzione.

Di recente, diverse applicazioni di Support Vector Machine a problemi di previsioni finanziarie sono state documentate [34][35][36]. Uno degli studi ben noti che utilizzano Support Vector Machine per la predizione del mercato è stato eseguito da Kim [34]. Ha applicato Support Vector Machine alle previsioni finanziarie e confrontato il risultato rispetto a *backpropagation*²¹ Neural Network e Case Based Reasoning

²¹In machine learning, in particolare deep learning, backpropagation è un algoritmo ampiamente utilizzato nel training di reti neurali feedforward per l'apprendimento supervisionato. L'algoritmo

(CBR). I risultati sperimentali hanno mostrato che Support Vector Machine ha avuto performance migliori di Neural Network e Case Based Reasoning. Poiché Support Vector Machine si basa sul principio di minimizzazione del rischio strutturale, la generalizzazione implementata risulta migliore rispetto alle tecniche tradizionali.

Case Based Reasoning

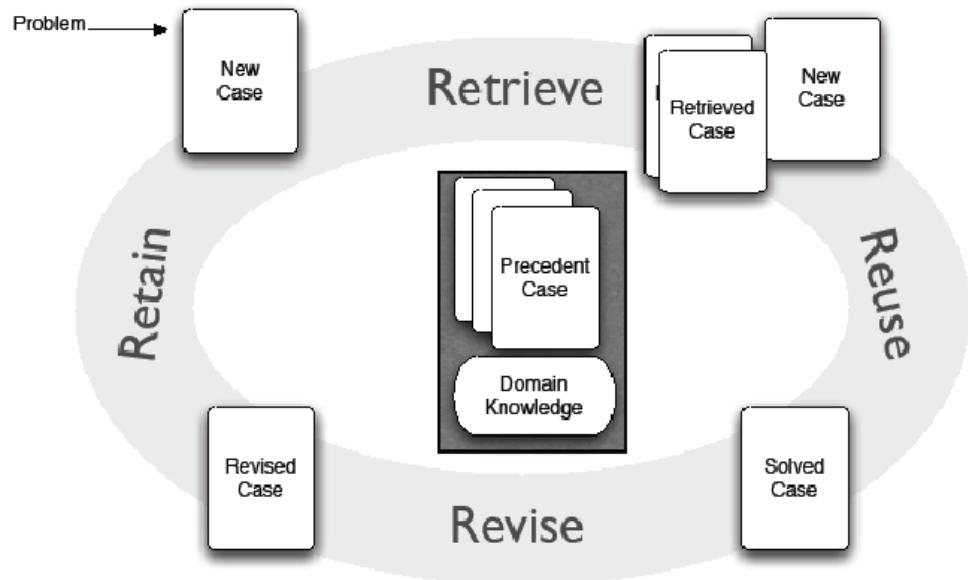


Figura 9:

Il processo di risoluzione di nuovi problemi basandosi sulle soluzioni di problemi anteriori, esempio generale di algoritmo CBR. Dato un problema, recuperare in memoria dei casi rilevanti per risolverlo (*Retrieve*). Adattare la soluzione del caso precedente al problema attuale (*Reuse*). Avendo mappato la soluzione precedente al caso attuale, bisogna provare la nuova soluzione e, se necessario, rivedere la nuova soluzione (*Revise*). Dopo che la soluzione è stata adattata al problema attuale, memorizzare l'esperienza come nuovo caso (*Retain*).

Sebbene le Neural Network offrano capacità di apprendimento relativamente buone rispetto ad altre tecniche, non possono sempre esplicitare il perché arrivano ad una particolare soluzione. Non garantiscono, inoltre, sempre una soluzione completamente certa, non assicurano di arrivare ripetutamente alla stessa soluzione dato lo stesso

di backpropagation funziona calcolando il gradiente della funzione di perdita rispetto a ciascun peso della *chain rule*, calcolando il gradiente uno strato alla volta, ripetendo all'indietro dall'ultimo strato per evitare calcoli ridondanti di termini intermedi.

dataset di training, né possono garantire di trovare la soluzione ottimale[37]. Tuttavia, a differenza delle reti neurali, in genere i *sistemi esperti*²² forniscono spiegazioni per le loro soluzioni. I sistemi esperti acquisiscono, principalmente, la conoscenza da individui esperti nell'ambito. Le organizzazioni hanno conoscenze collettive e competenze che hanno accumulato nel corso degli anni. Queste conoscenze possono essere acquisite, ed archiviate, utilizzando Case Based Reasoning. È una tecnica di ragionamento che riutilizza i casi passati per trovare una soluzione al nuovo problema. Case Based Reasoning non solo acquisisce la conoscenza dell'organizzazione, ma fornisce anche spiegazioni per le soluzioni derivate. Per questa ragione, è comunemente applicato a molti problemi.

Kim [38] ha proposto un nuovo modello ibrido di Genetic Algorithm e Case Based Reasoning per la previsione del mercato azionario. Dal suo studio preliminare, ha scoperto che la scelta del peso delle feature, e del sottoinsieme delle stesse, è molto importante per migliorare le prestazioni delle predizioni del sistema. Ha, quindi, usato Genetic Algorithm come metodo di selezione del sottoinsieme di feature nel sistema Case Base Reasoning. Ha confrontato i risultati di quattro diversi modelli per testare l'efficacia del modello proposto. I modelli a confronto sono: CBR convenzionali (COCBR), pesatura delle feature utilizzando Genetic Algorithm per CBR (FWCBR), selezione delle feature usando Genetic Algorithm per CBR (FSCBR) e ottimizzazione simultanea usando Genetic Algorithm per CBR (SOCBR). I risultati empirici hanno mostrato che SOCBR raggiunge una precisione di previsione superiore rispetto a COCBR, FWCBR e FSCBR. Kim ha concluso che il modello ibrido di GA e CBR offre un valido approccio alternativo alla previsione del mercato azionario.

Informazione sugli eventi

Nei mercati azionari, ci sono molti fattori che possono influenzare il prezzo dei titoli. Questi fattori possono essere derivati dai comunicati stampa sulle piccole aziende o dalle notizie delle superpotenze dell'economia mondiale. Questi episodi sono chiamati *events* [39]. Il motivo principale per cui incorporare conoscenza degli eventi nella previsione del mercato azionario si basa sul presupposto che il prezzo futuro di un'azione dipende in parte da vari aspetti politici ed eventi internazionali, oltre ai vari indicatori economici. Pertanto, molti studi hanno utilizzato informazioni sugli eventi (fattori qualitativi) nonché dati quantitativi nella previsione dei mercati azionari.

²²Un sistema esperto è un programma che cerca di riprodurre le prestazioni di una o più persone esperte in un determinato campo di attività, ed è un'applicazione o una branca dell'intelligenza artificiale. Sono spesso implementati attraverso euristiche o fuzzy logic e si dividono in due categorie principali: basati su regole e basati su alberi.

Uno degli studi popolari che utilizzano la conoscenza degli eventi e la *conoscenza pregressa* è stato realizzato da Kohara et al. [14]. I ricercatori hanno incorporato la conoscenza preliminare nella previsione di mercato, come informazioni sui giornali su eventi nazionali ed esteri. La conoscenza dell'evento è estratta dai titoli dei giornali in conformità con alcune conoscenze preliminari. La conoscenza pregressa è l'informazione che deriva da precedenti esperienze, pertanto, in base alla conoscenza precedente, decisioni differenti possono essere prese se un evento particolare può influenzare positivamente o meno le tendenze del mercato azionario.

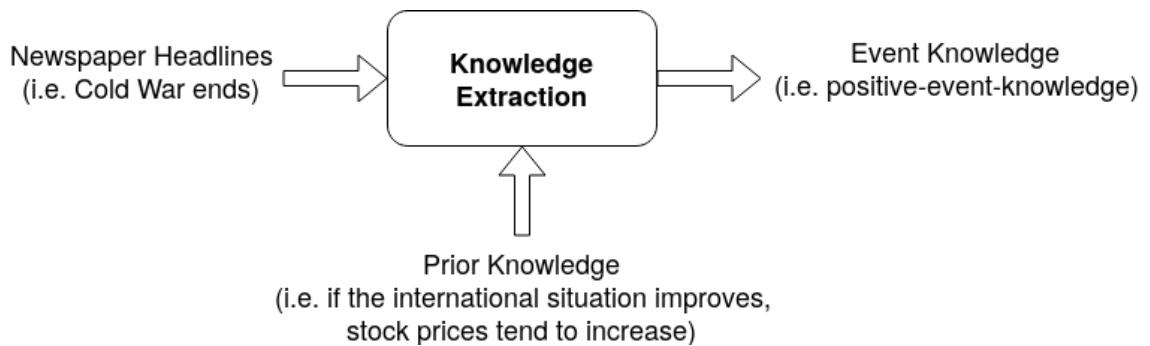


Figura 10:

Estrazione di conoscenza sugli eventi: basandosi sulle notizie di giornale e avendo coscienza dell'impatto che possono avere certi eventi sull'economia, è possibile stimare se un certo evento può influire positivamente o negativamente sul mercato.

Kohara et al. [14] hanno selezionato diversi indicatori economici (tasso di interesse, prezzo del petrolio grezzo e media dei prezzi di chiusura di New York Dow Jones) e forniti insieme alla conoscenza degli eventi come dati di training per Neural Network. I loro risultati sperimentali hanno mostrato che incorporare la conoscenza di eventi ha migliorato la capacità di previsione delle Neural Network, riducendo il tasso di errore del 5% del livello di significatività. Intanto, risulta cruciale *come* incorporare l'impatto delle informazioni sulle notizie nei modelli di serie storiche. Maheu e McCurdy [40] hanno specificato un modello GARCH²³ con salti per serie di ritorno, che può essere direttamente misurato dai dati dei prezzi. Si è supposto che il processo latente delle notizie abbia due componenti separati, eventi di cronaca normali ed insoliti. Queste differenti notizie sono identificate attraverso il loro impatto sulla *volatilità* del

²³In econometria, un modello autoregressivo a eteroschedasticità condizionata generalizzato o modello GARCH (Generalized AutoRegressive Conditional Heteroskedasticity) è un modello utilizzato nell'analisi delle serie storiche. È una funzione dei valori assunti dal processo agli istanti precedenti.

rendimento²⁴.

I modelli precedenti forniscono quadri generali per incorporare gli impatti delle notizie di giornale. Questi metodi non forniscono, però, un approccio per la distinzione delle notizie influenti o significative per un determinato titolo, a fronte di migliaia di articoli di ogni genere. Questi metodi, pertanto, nella pratica, non possono garantire un miglioramento significativo.

In letteratura, un certo numero di ricercatori ha affermato che i prezzi delle azioni sono significativamente correlati con le informazioni relative agli eventi e molti hanno tentato di utilizzare sia informazioni sugli eventi che dati numerici di serie temporali come dati in input. Fawcett e Provost [41] hanno formulato il problema di previsione del mercato come un'attività di monitoraggio della relazione tra le notizie ed i prezzi delle azioni. Fung et al. [13] hanno proposto un sistema che prevede i cambiamenti nell'andamento del mercato analizzando l'influenza di informazioni qualitative (notizie di giornale). In particolare, hanno studiato l'impatto immediato delle notizie sulle serie storiche basate sull'ipotesi di mercato efficiente. Hanno proposto un nuovo algoritmo statistico di *segmented regression*²⁵ per identificare le tendenze nelle serie temporali. In letteratura viene sempre più riconosciuta l'incorporazione della conoscenza pregressa e conoscenza degli eventi nei modelli di previsione come Neural Network.

Informazione sugli eventi nel Web

L'evoluzione di Internet, con l'infrastruttura di informazioni globale, ha provocato un'esplosione della quantità di informazioni disponibili. Enormi moli di dati relativi a eventi che possono avere grande influenza sul mercato azionario sono disponibili sul web. Considerando che i giornali vengono aggiornati una o due volte al giorno, le fonti di notizie in tempo reale sono aggiornate frequentemente quasi istantaneamente. Queste informazioni non contengono solo notizie globali e regionali ma anche citazioni preziose di influenti banchieri, politici e analisti finanziari. Le informazioni che consistono in notizie o citazioni sono viste come principali motori del mercato delle

²⁴In finanza, la volatilità è una misura della variazione percentuale del prezzo di uno strumento finanziario nel corso del tempo. Corrisponde alla *Deviazione standard*. Di solito indica il livello di rischio associato ad un titolo: possedendo un titolo più volatile si è più esposti alle oscillazioni del suo valore.

²⁵La regressione segmentata (segmented regression), nota anche come piecewise regression o broken-stick regression, è un metodo di analisi di regressione in cui la variabile indipendente è suddivisa in intervalli ed un segmento di linea separato è adattato a ciascun intervallo. L'analisi di regressione segmentata può anche essere eseguita su dati multivariati partizionando le varie variabili indipendenti. La regressione segmentata è utile quando le variabili indipendenti, raggruppate in gruppi diversi, mostrano relazioni diverse tra le variabili in queste regioni.

obbligazioni, delle valute e azionario nel mondo. Hong e Han [42] hanno affermato che, con l'aumento della popolarità di Internet, molti giornali espanderanno i loro servizi fornendo notizie sul web, per essere più competitivi e aumentare il profitto. Le notizie includono articoli sulla situazione politica, le condizioni sociali, eventi internazionali, politiche governative, psicologia dei trader e così via; informazioni che vediamo e comprendiamo attraverso Internet. Tali informazioni sono formulate sotto forma di testo, indicati come documenti, quindi tecniche di *text mining*²⁶ sono necessarie.

Hong e Han [42] hanno introdotto un sistema automatizzato (KBNMiner) che acquisisce conoscenza degli eventi da Internet, per la predizione dei tassi di interesse. Questo studio mostra una chiara idea di conoscenza dell'evento usando un sistema automatizzato. KBNMiner è progettato per adottare una base di conoscenza preliminare, che viene vista come "conoscenza esperta", come base su cui raccogliere informazioni sugli eventi da Internet automaticamente e quindi applicare le informazioni a un modello di Neural Network per la previsione dei tassi di interesse. Ciò che si distingue in questo studio è che la tecnica di web mining che hanno applicato per la previsione dei tassi di interesse può anche essere utilizzata per previsione del mercato azionario.

Lavori futuri

In questa sezione si sono esaminati i recenti sviluppi dei modelli di previsione del mercato azionario. Confrontando vari modelli di predizione, si è scoperto che le Neural Network offrono l'abilità di prevedere le direzioni del mercato in modo più accurato rispetto ad altre tecniche esistenti. La capacità di Neural Network di imparare relazioni non lineari dalle coppie di input / output di training consente loro di modellare sistemi dinamici non lineari, come i mercati azionari, più precisamente [24].

Anche altri modelli come Support Vector Machine e Case Based Reasoning sono diventati popolari nella previsione del mercato azionario. Support Vector Machine ha mostrato la sua corretta applicazione nei problemi di classificazione e di regressione, in particolare nella previsione di serie storiche e applicazioni finanziarie correlate [44]. Case Based Reasoning è una tecnica di ragionamento che riutilizza casi passati per trovare una soluzione al nuovo problema. Case Based Reasoning cattura la conoscenza e la competenza di un'organizzazione fornendo spiegazioni per le soluzioni

²⁶Text mining è la scoperta da parte del computer di informazioni nuove, precedentemente sconosciute, estraendo automaticamente informazioni da diverse fonti di testo. È il processo di estrazione di informazioni di alta qualità dal testo. Le informazioni sono in genere derivate dall'elaborazione di modelli e pattern attraverso mezzi come l'apprendimento automatico e statistico di pattern. L'estrazione del testo di solito comporta il processo di strutturazione del testo di input (*parsing*), estrazione di pattern all'interno dei dati strutturati e, infine, la valutazione e l'interpretazione dell'output.

derivate. Per questa ragione, Case Based Reasoning è comunemente applicato a molti problemi.

Studiando diversi punti chiave dei mercati azionari, inoltre, si è scoperto che molti ricercatori hanno riconosciuto che fattori qualitativi come effetti politici ed eventi internazionali possono avere un impatto significativo sui prezzi delle azioni. È stato esaminato che Neural Network basate sia sui fattori quantitativi che qualitativi sono di gran lunga superiori a quelle basate solo su fattori quantitativi. Il web, inoltre, è considerato come la principale fonte di notizie sugli eventi per la predizione del mercato azionario, contenendo informazioni sugli eventi più recenti e latenti. Pertanto, per prevedere il mercato, tecniche di web mining sono necessarie per avere una maggiore accuratezza e per fare predizioni in un tempo più breve.

Per ulteriori ricerche, in primo luogo, un database contenente conoscenza preliminare dovrebbe essere costruito analizzando eventi storici nei mercati azionari. Sulla base delle conoscenze precedenti, verrà sviluppato lo schema di attribuzione di peso degli eventi e ciascun evento dovrebbe essere pesato di conseguenza. Il modello proposto che incorpora i pesi degli eventi nei dati numerici delle serie temporali, infine, dovrebbe essere confrontato empiricamente con altri modelli.

2.2 Sentyment

A differenza degli algoritmi appena citati, Sentyment si concentra sulla creazione di strategie di investimento, invece che sulla previsione dell'andamento nel tempo di un particolare titolo azionario. Gestisce contemporaneamente più asset. Non si può paragonare agli algoritmi nello stato dell'arte sul piano della precisione di previsione, perché la funzione che cerca di massimizzare non è la stessa delle altre AI viste. Questo software commerciale ha come scopo il rispetto delle politiche scelte dall'utente in merito a propensione al rischio e interessi personalizzati e la massimizzazione del guadagno attraverso l'ottimizzazione di alcuni indicatori come, appunto, il guadagno prodotto dalla strategia o il numero di operazioni vincenti.

A differenza degli algoritmi sopra citati, in cui lo scopo è *prevedere quale valore avrà la funzione* in un certo istante, ciò che trova Sentyment è invece *quale operazione fra buy, sell o hold conviene eseguire* al fine di massimizzare il guadagno, in un certo istante.

Ciò premesso, è comunque ovvio il fatto che il sistema abbia una certa capacità di previsione del mercato, di identificazione dei pattern ricorrenti o addirittura un'integrazione di informazioni sugli eventi. Per poter competere con gli altri sistemi presenti sul mercato deve fare sicuramente uso di una o più fra le tecniche descritte in questo capitolo. La AI, tuttavia, verrà trattata come una *black-box* e non saranno, quindi, rivelati dettagli sulla sua implementazione, ma verranno considerati soltanto gli input e gli output.

L'analisi comparativa con altri sistemi risulta, quindi, difficile in quanto gli obiettivi dei software sono differenti. È comunque possibile misurare rispetto ad indicatori assoluti o comparare fra loro le diverse AI che compongono Sentyment. Infatti, il sistema è composto da un certo numero di AI simili ma leggermente differenti fra loro e soltanto una fra queste è scelta come migliore e usata attivamente per il trading. La AI migliore potrebbe cambiare durante il tempo, a intervalli da definire, ed è quindi necessario implementare un metodo per operare una scelta intelligente.

2.2.1 Cenni all'architettura

Sentyment è il prodotto di un'azienda milanese che opera nel settore informatico-finanziario. Nexid Edge è un ramo del gruppo NEXiD votato per perseguire l'innovazione tecnologica e sviluppare soluzioni di digital business rivoluzionarie. SentYment è la nuova Business Unit dedicata all'intelligenza artificiale²⁷.

²⁷Nexid, Srl. URL: <https://nexid.it/>

In questo documento, tuttavia, si fa riferimento a "Sentyment" intendendo l'applicazione sviluppata, e non la divisione AI di Nexid.

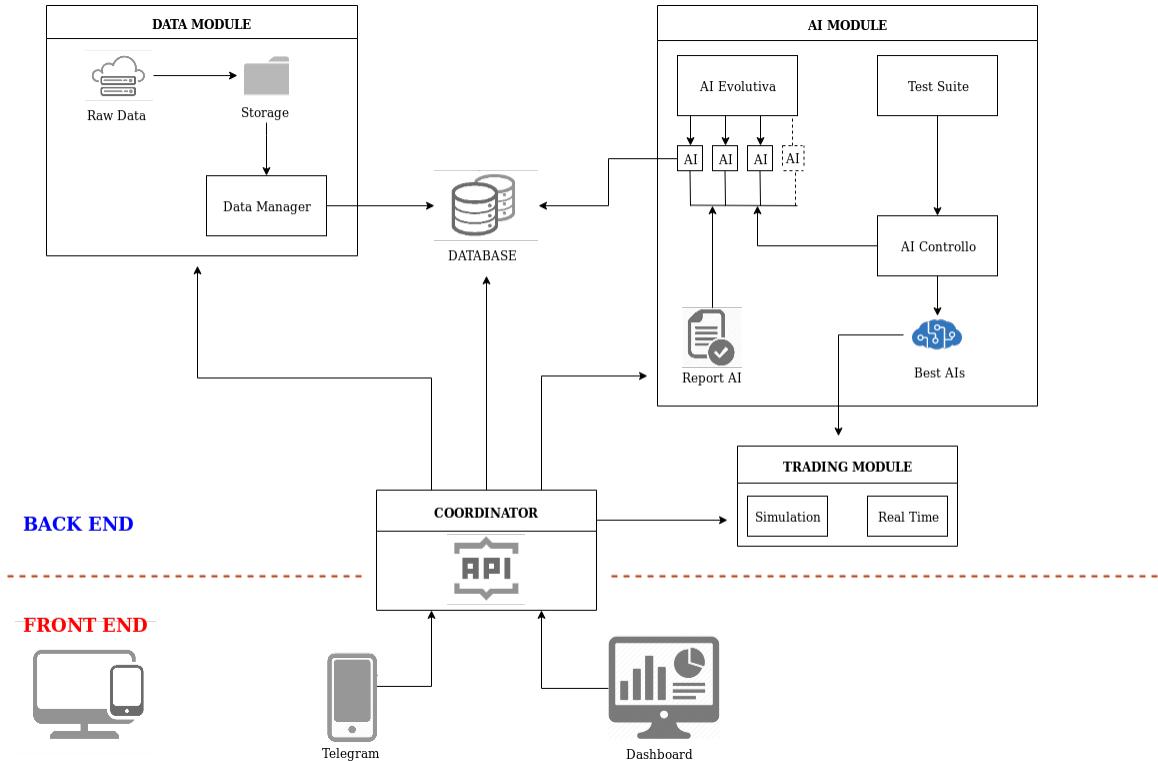


Figura 11:

Architettura di Sentyment. Nei riquadri sono mostrati i quattro principali componenti, suddivisi al loro interno in ulteriori sotto-moduli. Il grafico è diviso in *front-end* e *back-end*, che rappresentano rispettivamente le interfacce esposte all'utente utilizzatore del sistema, e la parte di logica non accessibile dall'esterno

Essendo il fine ultimo della tesi l'inserimento un nuovo modulo operativo in Sentyment, è necessario prendere familiarità con il software ed i dati che tratta. L'applicazione di intelligenza artificiale era già esistente ed in grado di raccogliere dati da Kraken e produrre le strategie, ma mancava di un'architettura modulare, di componenti separati che assolvessero a ruoli precisi e definiti e di un'interfaccia ad API per interrogare il sistema in esecuzione. Era, quindi, necessario sviluppare da zero *data module* e *coordinator* e permettere l'integrazione del resto del sistema già esistente nella nuova architettura a moduli. Era, inoltre, richiesto lo sviluppo della

nuova AI di controllo, supervisore che scegliesse la migliore fra le AI in esecuzione. Tutti i moduli del sistema sono scritti in Python3, comprese le AI stesse. Alcune loro parti sono state riscritte in Cython e Go per migliorare le prestazioni.

Verranno ora brevemente presentati i componenti di Sentyment.

- *data module* raccoglie i record di trade raw tramite le API Kraken, gestisce la creazione e scrittura delle candele OHLCV e permettere l'accesso a tutti i dati ed i database da parte degli altri componenti. I sotto-moduli si spartiscono i compiti: quando i dati di trade sono scaricati e salvati su uno storage temporaneo da *raw data*, allora *data manager* li copia sul database permanente e, in automatico, scatta la creazione delle nuove candele OHLCV, che sono create allo scoccare della nuova ora, a partire dai dati appena scaricati (o minuto/giorno, a seconda delle configurazioni scelte). Le nuove candele create, infine, sono inserite nel database.
- *coordinator* implementa le interfacce esposte per il controllo e l'interrogazione dello stato di Sentyment. È l'unico punto di accesso al sistema in esecuzione. Tramite delle API esposte su server è possibile conoscere quali asset sono attualmente in fase di scaricamento e le informazioni riguardo le configurazioni delle candele che si vogliono creare; è, inoltre, possibile impostare i timeframe delle candele (orarie, giornaliere), attivare o disattivare i download dati e forzare la creazione di specifiche candele.
- *AI module* è la parte preesistente di intelligenza artificiale. Contiene le numerose AI che creano le strategie di investimento ed il supervisore che sceglie la migliore fra queste, a intervalli fissati. È, inoltre, presente un modulo che crea report che descrivono l'andamento delle AI, i dati che stanno processando e le statistiche calcolate per ognuna di esse, per misurarne la performance online.
- *trading module* si occupa di seguire le strategie create dalle AI per effettuare direttamente operazioni di acquisto e vendita tramite le API di Kraken. È collegato ad un portafoglio Kraken che contiene un certo budget ed è libero di fare trading. Oltre ad agire sul mercato vero, le AI operano anche in un ambiente simulato che permette di riprodurre, e monitorare, quale sarebbe stato il loro comportamento nel mercato reale, o in nuovi mercati non ancora raggiunti, al fine di calcolare ulteriori statistiche e tracciare l'evoluzione delle AI.

Come già accennato, sono stati sviluppati *data module* e *coordinator*, ridisegnando la preesistente architettura di Sentyment ed integrando gli altri componenti. Durante la riprogettazione, si sono potute scegliere liberamente la logica di implementazione

e le tecnologie da impiegare, con la restrizione di dover utilizzare Python3, con *sqlalchemy*²⁸ come *ORM*²⁹ e *flask*³⁰ per implementare il server e le API.

Prima di sviluppare la AI supervisore, sono state testate le AI singolarmente attraverso alcune metodologie di test che saranno descritte di seguito. È stato svolto un lavoro di analisi dei dati prodotti dalle AI, confrontandoli con alcune strategie di base e con altre intelligenze artificiali sviluppate al fine di misurarne le prestazioni e, solo dopo, le AI sono state confrontate fra di loro. I metodi ritenuti migliori per il confronto sono stati scelti in seguito ad un'analisi del contesto ed alla natura finanziaria dei dati.

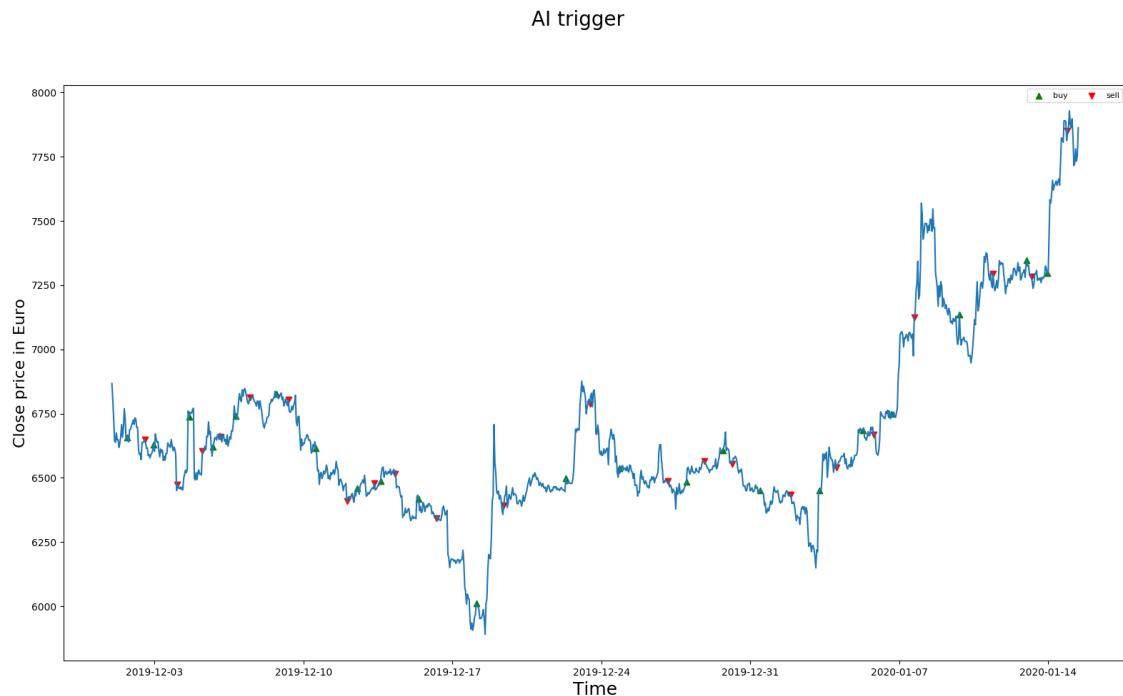


Figura 12: Esempio dei *trigger* generati da una delle AI di Sentyment. Il grafico fa riferimento ai prezzi di chiusura del ticker BITCOIN-EURO relativi ad un certo periodo temporale, da fine 2019 a inizio 2020.

²⁸SQLAlchemy è un toolkit SQL open-source e object-relational mapper (ORM) per Python

²⁹l'Object-Relational Mapping (ORM) è una tecnica di programmazione che favorisce l'integrazione di sistemi software aderenti al paradigma della programmazione orientata agli oggetti con sistemi RDBMS (Relational Database Management System). Un prodotto ORM fornisce, mediante un'interfaccia orientata agli oggetti, tutti i servizi inerenti alla persistenza dei dati, astraendo, nel contempo, le caratteristiche implementative dello specifico RDBMS utilizzato.

³⁰Flask è un micro framework web scritto in Python. Ha un nucleo semplice ma estendibile: ci sono, ad esempio, estensioni per la validazione delle form, la gestione del caricamento dei file, varie tecnologie di autenticazione ed altro

2.2.2 Strumenti e tecnologie utilizzati

Sentyment è quasi totalmente scritta in Python3, tranne per alcune parti in cui si ha bisogno di elevate performance che sono, quindi, state riscritte in Cython³¹. Queste comprendono, ad esempio, gran parte del codice delle AI, ma non solo.

Si fa uso estensivo di librerie di analisi dati quali *numpy*³², *pandas*³³ e *matplotlib*³⁴ per visualizzare i grafici dei dati, soprattutto nella parte di sviluppo e analisi.

Queste librerie sono i classici strumenti utilizzati nell’analisi dati con Python e sono i più utilizzati; le alternative attualmente presenti non offrono le stesse funzionalità con la medesima efficienza e, per questo, erano anche i *tool* già usati nella prima versione di Sentyment.

Pandas, in particolare, permette di importare grandi quantità di dati direttamente da database, o da file, ed è in grado di gestirli in modo efficiente indicizzandoli e garantendo un accesso rapido. Un ampio set di operazioni statistiche sui dati lo rende uno strumento indispensabile in *data science* ed ampiamente utilizzato. Basti pensare ad un AI a cui servono in input moltissimi dati: quando questi sono salvati su database, è facile eseguire piccole letture e query puntuale ed aggregate ma, quando, invece, si ha bisogno di portarne in memoria un gran numero per processarli con funzioni al di là delle potenzialità di *SQL*, serve un ulteriore strumento capace di gestire l’accesso efficiente alla memoria e che offre direttamente queste funzioni di analisi (esempio con funzione *resample* nel Capitolo 3).

Il database scelto è di tipo *relazionale*. Si è scelta l’opzione relazionale, invece di *noSQL*, in quanto i dati sono tutti perfettamente strutturati (schema dati in dettaglio nel Capitolo 3): la sorgente principale da cui partono tutte le elaborazioni è rappresentata dai dati di trading *raw* forniti da Kraken. Questi hanno ciascuno un riferimento al *timestamp*, alla tipologia dell’operazione di trade, la quantità di titoli venduti / acquistati ed il loro volume. Sostanzialmente, non servono altre informazioni per la creazione delle candele OHLCV, che sono anch’esse strutturate ed esplicitano

³¹Cython è un linguaggio di programmazione che mira ad essere un superset del linguaggio di programmazione Python, progettato per fornire prestazioni di tipo C con codice scritto principalmente in Python con sintassi aggiuntiva opzionale ispirata al C

³²NumPy è una libreria open source per il linguaggio di programmazione Python, che aggiunge supporto a grandi matrici ed array multidimensionali, insieme ad una vasta collezione di funzioni matematiche di alto livello, per poter operare efficientemente su queste strutture dati.

³³Pandas è una libreria software scritta per il linguaggio di programmazione Python per la manipolazione e l’analisi dei dati. In particolare, offre strutture dati ed operazioni per manipolare tabelle numeriche e serie temporali. Il nome deriva dal termine ”panel data”, termine econometrico per set di dati che include osservazioni su più periodi di tempo per gli stessi individui.

³⁴Matplotlib è una libreria per la creazione di grafici per il linguaggio di programmazione Python e la libreria matematica NumPy. Fornisce API orientate agli oggetti che permettono di inserire grafici all’interno di applicativi

open, high, low, close, volume, timestamp finale e altri dati secondari. Le AI lavorano tutte sulle candele e necessitano soltanto degli attributi appena citati.

L'unica eccezione, in cui dati non sono perfettamente strutturati, sono le AI. Sono memorizzate come un insieme di parametri che permettono la loro ricreazione in un secondo momento, se necessaria, e sono salvati in formato *JSON*.³⁵ Dato che i parametri sono diversi, non strutturati, e con possibilità di molteplici valori nulli, la memorizzazione in JSON, già adottata nella prima versione di Sentyment, sembrava comunque adatta, permettendo una compressione più efficiente dei dati ed un accesso orientato al documento. Di una specifica AI si vogliono conoscere i suoi parametri, ma il nome / *id* della AI è noto: per esempio, in seguito ad un'analisi durante un nuovo ciclo di test / sviluppo, si vuole continuare dalla AI più promettente fra quelle della precedente iterazione e, quindi, si accede al documento che contiene tutti i parametri della AI scelta.

I JSON dei parametri delle AI non sono scritti su database, ma sono dei file che risiedono direttamente su file system nelle directory di Sentyment, strutturate in modo da poter accedere ai parametri di una certa AI usando il suo identificativo. In sviluppi futuri, potrebbe essere più adatto l'uso di un database non relazionale per la memorizzazione delle AI, oppure sfruttare i tipi di dato JSON presenti in ormai tutti i RDMBS moderni e, quindi, salvare tutto nello stesso database.

Fra i database relazionali è stato scelto *Mysql*³⁶ perché meglio si sposa con l'ORM di *sqlalchemy*.

Sqlalchemy permette di manipolare database SQL direttamente da Python. Mappa lo schema tabellare dei dati in uno a oggetti più facilmente processabile dal linguaggio di programmazione scelto, in questo caso Python. Grazie alla tecnologia degli ORM, il modello dei dati è sempre disponibile come classe del linguaggio e risulta, quindi, più comodo il loro accesso e manipolazione: ora si lavora con collezioni di oggetti, invece che con un semplice elenco di record contenenti alcuni campi.

L'ORM mette anche a disposizione un motore di astrazione per le *query*. Tramite un modello ad API le query eseguite in Python sono tradotte nel dialetto SQL del DBMS sottostante. Al programmatore resta il compito di scrivere le query usando

³⁵JSON, acronimo di JavaScript Object Notation, è un formato adatto all'interscambio di dati fra applicazioni client/server. È basato sul linguaggio JavaScript, ma ne è indipendente.

³⁶MySQL o Oracle MySQL è un Relational database management system (RDBMS) composto da un client a riga di comando e un server. È un software libero rilasciato a doppia licenza ed è sviluppato per essere il più possibile conforme agli standard ANSI SQL e ODBC SQL. I sistemi ed i linguaggi di programmazione che supportano MySQL sono molto numerosi.

l'*expression language*³⁷ di sqlalchemy, il sistema usato per rappresentare strutture dei database relazionali e espressioni usando Python.

Flask, infine, è un microframework per Python ed implementa *coordinator*, il web server che espone le informazioni disponibili sullo stato di esecuzione di Sentyment. È molto minimalista, se mantenute le impostazioni di default, e serve soltanto documenti JSON e non pagine HTML.

In futuri sviluppi, le informazioni fornite dal web server in JSON potrebbero essere processate da un'ulteriore componente che le esporrebbe, in modo più user-friendly, attraverso un software di front-end e permetterebbe l'interrogazione di *coordinator*, attraverso le API, in maniera più intuitiva.

³⁷Expression Language (EL) è un linguaggio per creare una rappresentazione interpretabile di conoscenze specifiche. In questo caso, è un linguaggio creato da sqlalchemy appositamente per rappresentare query SQL nel linguaggio di Python.

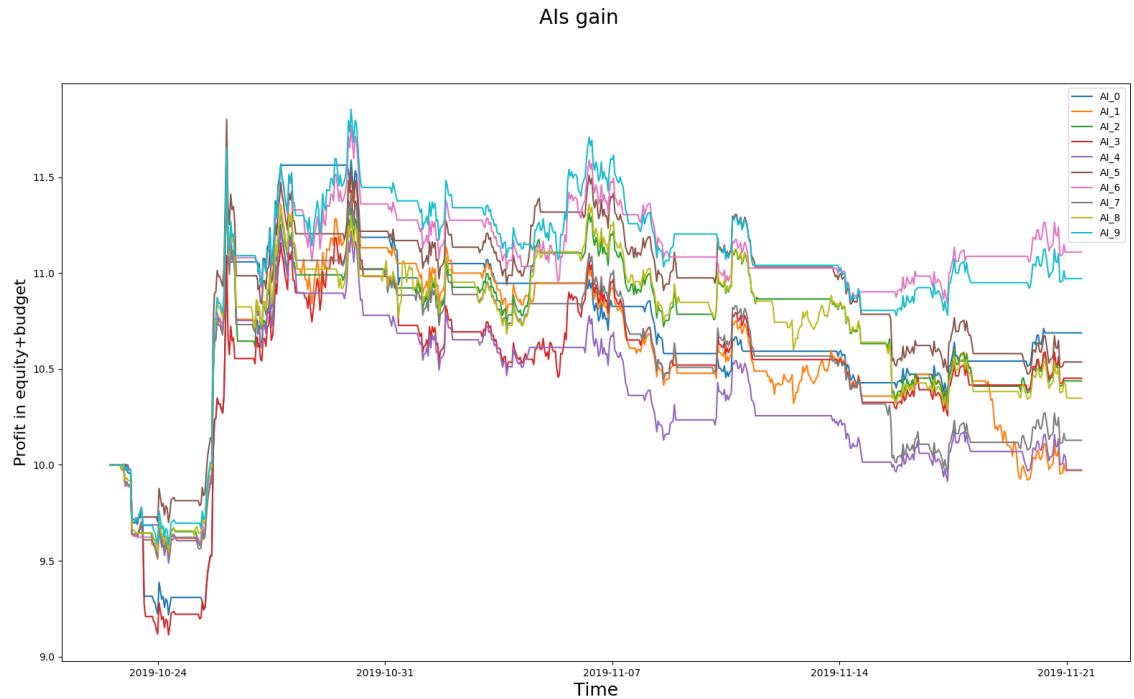


Figura 13: Grafico comparativo dei risultati prodotti da 10 differenti AI di Sentyment. È analizzata la funzione guadagno (*gain*, Formula (3)). Questo frammento di dataset, che fa riferimento ai prezzi di BITCOIN-EURO di fine 2019, è particolarmente rappresentativo. Si nota come le AI abbiano tutte un comportamento diverso, ma ciascuna con pattern simili: nella settimana del 24 ottobre, a inizio grafico, tutte quante prevedono un aumento del prezzo del titolo e investono capitale, ma ognuna lo fa in modo diverso. Seguono altre situazioni simili, in cui le varie linee del guadagno mostrano delle "colline" simili per ciascuna AI.

Capitolo 3

Elaborazione dati di trading

In questo capitolo si andrà principalmente a trattare il lavoro svolto dal software di raccolta e gestione dati, *data module*, ed alcuni dettagli della sua implementazione. Attraverso l'analisi delle operazioni svolte dal programma, si darà anche un ulteriore sguardo ai dati, a come sono organizzati dalla piattaforma di trading ed all'interno del software di Nexid, ed a come si processano per ottenere un formato utile per l'elaborazione da parte delle AI.

Anche se Sentyment comprendeva già una parte di logica di scaricamento dati, in seguito alla sua riorganizzazione, è stato necessario riscrivere questa parte di software convertendola interamente in un modulo separato, sempre attivo e pronto a scaricare, con cui si può interagire attraverso *coordinator*. Svolge diverse attività fra cui scaricare grazie alle API di Kraken, creare le candele OHLCV a partire dai dati scaricati e gestire il database.

I requisiti dell'applicativo sono i seguenti:

- Scaricare dati di trade *raw* per diversi tipi di asset, restando aggiornati in tempo reale con l'arrivo di nuovi
- Fornire la possibilità di scaricare dati passati
- Creare candele OHLCV a partire dai dati scaricati e poter sceglierne i parametri (lunghezza candela, periodo da considerare)
- Salvare su database tutti i dati scaricati

3.1 Download dati Kraken

L'input di tutto il sistema sono i dati di trade. Un modulo intero è dedicato alla logica di scaricamento dati, il cui fine è soltanto fornire i suddetti dati alle intelligenze

artificiali, che rappresentano la parte principale del software. Tutta l'architettura è orientata al dato; le AI stesse ignorano l'esistenza di *data module*, perché, di fatto, accedono direttamente al database per prelevare i dati, ignorando chi lo abbia riempito. Un metodo strutturato per "riempire" il database, e preparare i dati per le AI, è, quindi, un passo iniziale e fondamentale per tutta l'architettura.

Le Web API di Kraken consistono in richieste HTTP GET e POST¹ e si dividono in *pubbliche* e *private*: tutte le attività svolte da *data module* fanno uso soltanto delle interfacce pubbliche, mentre quelle private, le quali necessitano di un API Key², servono per collegare un portafoglio con budget e permettono di eseguire operazioni di acquisto e vendita di titoli. Sono utilizzate soprattutto dal modulo di trading. Le API pubbliche permettono molti tipi di operazioni fra cui, le più usate:

- **Get tradable asset pairs:** elenco di tutti gli asset trattati da Kraken, i *ticker*³, con associato il nome usato ed altre informazioni. Sono 110 in totale le coppie di valute di scambio fra una criptovaluta e la sua controparte in Euro, o Dollaro. Alcuni esempi:

- XXBTZEUR: ticker che rappresenta la valuta Bitcoin-Euro (XBT-EUR)
- XXBTZUSD: Bitcoin-Dollaro (XBT-USD)
- XETHZEUR: Ethereum-Euro (ETH-EUR)
- XETHZUSD: Ethereum-Dollaro (ETH-USD)

Non ci sono soltanto valute di scambio fra Euro/Dollaro e una criptovaluta, ma bensì anche scambi fra criptovaluta e criptovaluta, come, ad esempio:

- ADAETH: valuta di scambio fra le due criptovalute Cardano (ADA)⁴ ed Ethereum (ETH)

¹GET e POST sono metodi di richiesta supportati da HTTP, un protocollo a livello applicativo usato come principale sistema per la trasmissione d'informazioni sul Web.

²Una chiave per API (API Key) è un identificatore univoco utilizzato per autenticare un utente, uno sviluppatore o il programma chiamante un'API. Tuttavia, vengono in genere utilizzati per autenticare un progetto con l'API, anziché un utente umano. Spesso si usano per accedere a quelle API che forniscono servizi a pagamento, o per superare limitazioni di utilizzo.

³Il ticker è un'abbreviazione che identifica le società che vengono quotate su un mercato finanziario. Quando una società decide di 'diventare pubblica', sceglie il tipo di mercato sui cui essere quotata ed il ticker di identificazione.

⁴Cardano è una piattaforma che gestisce la criptovaluta ADA. Il progetto è guidato dall'ex co-fondatore di Ethereum. Lo scopo di Cardano è creare una piattaforma adatta allo sviluppo di smart contracts ed un sistema di pagamento.

- XZECXXBT: Zcash (ZEC)⁵ e Bitcoin (XBT)

Gli asset che scambiano due criptovalute fanno parte dei ticker gestiti da Kraken e, quindi, sono comunque scaricati e memorizzati nel database, ma non verranno usati nel resto del progetto.

L’elenco è scaricato in una fase iniziale di setup del modulo dati ed ha lo scopo di memorizzare, su database, i nomi usati da Kraken per tutti gli asset che fornisce, in modo da poterli utilizzare nelle altre API per la fase successiva, in cui si richiede l’elenco di trade di uno specifico ticker.

Come si può notare, sono usati nomi particolari per rappresentare i ticker ed è, quindi, necessario memorizzare il mapping fra nome reale e nome usato da Kraken, per poi poter utilizzare le API che seguono.

- **Get recent trades:** Ora che si conoscono i nomi dei ticker usati da Kraken, si possono usare per richiedere, per ciascuno di essi, l’elenco delle transazioni di mercato. Questa è, in assoluto, la API più usata dal modulo dati: riceve i dati di trade *raw*, che costituiscono l’input principale da cui partono le successive elaborazioni. I dati di trade sono memorizzati, e saranno usati, per creare le candele OHLCV, l’input per le intelligenze artificiali.

Attraverso questa API, è possibile sia richiedere dei dati storici in un certo periodo di tempo passato, sia ricevere aggiornamenti sui nuovi dati disponibili. In entrambi i casi, va specificato come parametro il ticker di cui si vuole conoscere l’elenco degli scambi. Come secondo parametro, è necessario un riferimento temporale che indichi quali trade sono richiesti di preciso (parametro *since*).

Esempio di documento JSON restituito dalla chiamata a questa interfaccia, usando come parametri il ticker XXBTZEUR ed il riferimento a gennaio 2020:

```
{
  "error": [],
  "result": {
    "XXBTZEUR": [
      ["6410.20000", "0.07588465", 1577833204.9008, "b", "m", ""],
      ...
    ]
  }
}
```

⁵Zcash è una criptovaluta che offre privacy e trasparenza selettiva delle transazioni. I pagamenti Zcash sono pubblicati su una blockchain pubblica, ma il mittente, il ricevente ed il valore della transazione possono rimanere privati. Come Bitcoin Zcash ha una fornitura fissa totale di 21 milioni di unità.

```

        ["6410.00000","0.02911527",1577833220.5154,"s","1","");
        ["6410.00000","0.02910911",1577833261.2323,"s","1","");
        ["6435.50000","0.08000000",1577842069.8177,"s","1"], ...
    ]
    "last": "1577842069817694680"
}
}

```

Oltre ad alcuni metadati utili per riutilizzare le API, sono principalmente elencati una serie molto lunga di record formati da sei campi: *price*, *volume*, *time*, *buy/sell*, *market/limit*, *miscellaneous*.

Prendendo come esempio il primo record:

- *price*: 6410.20000. Siccome la valuta è Bitcoin-Euro, il numero indica il valore di un titolo Bitcoin espresso in Euro.
- *volume*: 0.07588465. La quantità di titoli Bitcoin scambiati in quell’istante.
- *time*: 1577833204.9008. Timestamp associato all’operazione in formato unix timestamp.⁶
- *buy/sell*: ”b”. Indicazione del tipo di operazione: ’b’ (buy) o ’s’ (sell).
- *market/limit*: ”m”. Indica se comprare a prezzo di mercato (m), o se applicare un limit. Un limit order è un ordine inserito nell’order book con un prezzo limite specifico. Quando è inserito un limit order, l’operazione verrà eseguita solo se il prezzo di mercato raggiunge il prezzo limite. È quindi, possibile usare limit order per comprare ad un prezzo più basso o per vendere ad un prezzo più alto rispetto all’attuale prezzo di mercato.
- *miscellaneous*: ””. Spazio per eventuali commenti o casi particolari.

Il campo *last* serve da passare come nuovo parametro *since* nella prossima chiamata, a indicare di scaricare ancora i dati dal punto in cui ci si era fermati. Siccome il volume dei dati di trade è molto elevato,⁷ viene diviso in frammenti; per continuare a scaricare frammento per frammento si sfrutta la combinazione di questi due parametri.

- **Get OHLC data:** Kraken offre anche la possibilità di scaricare direttamente le candele OHLCV create a partire dai dati di trade raw, ma questa funzionalità è già implementata all’interno di Sentyment stesso. Ad ogni modo, sono

⁶Nei sistemi operativi Unix e Unix-like il tempo viene rappresentato come offset in secondi rispetto alla mezzanotte del 1º gennaio 1970. I decimali rappresentano la frazione di secondo.

⁷I dati di trade di Bitcoin, per il solo anno 2019, hanno un’occupazione dell’ordine di grandezza dei 10 Gb.

comunque riportati alcuni esempi di dati scaricati tramite questa API. Esempio di documento JSON restituito dalla chiamata, con parametri il ticker XXBTZEUR, il timestamp di gennaio 2020 e la lunghezza di candela ad un minuto:

```
{
  "error": [] ,
  "result":
  {
    "XXBTZEUR":
    [
      [1582093500,"9384.7","9384.7","9384.6","9384.6","9384.6",
       "0.35398923",9] ,
      [1582093560,"9384.6","9384.7","9384.0","9384.0","9384.5",
       "0.04147968",5] ,
      [1582093620,"9384.1","9384.1","9384.0","9384.0","9384.0",
       "0.40413908",5] ,
      [1582136640,"9433.6","9433.6","9433.5","9433.5","9433.5",
       "0.33265963",4] , ...
    ],
    "last":1582136580
  }
}
```

La struttura del documento è simile a quella restituita nell'esempio precedente, con la differenza che, ora, i record sono composti da attributi diversi. Prendendo come esempio il primo record:

- *time*: 1582093500. Timestamp finale della candela. La candela delle 12:00 comprende i trade dalle 11:00 alle 11:59.
- *open, high, low, close*: rispettivamente 9384.7, 9384.7, 9384.6, 9384.6. Prezzo di apertura e chiusura della candela, minimo e massimo.
- *vwap*: 9384.6. Prezzo medio della candela.⁸
- *volume*: 0.35398923. Somma totale della quantità di titoli scambiati durante l'intervallo di tempo della candela.
- *count*: 9. Numero di operazioni buy / sell realmente eseguite all'interno della durata della candela.

⁸In finanza, Volume Weighted Average Price (VWAP) è il rapporto tra il valore scambiato e il volume totale scambiato in un determinato orizzonte temporale. È una misura del prezzo medio al quale un titolo viene negoziato nell'orizzonte di negoziazione.

Il modulo di scaricamento è molto semplice ed interroga solamente le API di Kraken, memorizzando, di volta in volta, blocchi di dati su database. L'unica difficoltà incontrata risulta essere la comprensione di alcuni parametri delle API, che si dimostrano non perfettamente documentate e, pertanto, è lasciato ai programmatore scoprire il significato di alcune interfacce.

Per esempio, il parametro *last*, che dovrebbe indicare l'ultimo timestamp fra i record ritornati nella chiamata, e che va passato come parametro *since* nella successiva chiamata per ripartire dall'ultimo trade, è controverso. La documentazione, riguardo alla API "Get recent trades" riporta, per il parametro *since*: "*return trade data since given id*", intendendo, quindi, che si tratti di un identificatore. Curiosamente, in moltissimi casi, (soprattutto nei dati OHLC, che hanno tutti timestamp precisi all'ora) ha lo stesso valore del timestamp e, quindi, questo potrebbe significare che i timestamp sono usati come id. Quando, però, il valore ha dei decimali, che nel caso di un unix timestamp indicano frazioni di secondo, questi sono aggiunti in coda all'identificatore con un *padding*⁹ a 0 di nove cifre. Quando si richiedono i dati a partire da un certo timestamp che non ha decimali, quindi, bisogna aggiungere il padding altrimenti l'id richiesto non viene trovato nei database di Kraken. In risposta è invece ritornato il record con id minore di tutti, che è quello che più si avvicina al valore richiesto, estremamente basso rispetto a tutti gli altri id presenti nel database dato che ha nove cifre mancanti.

Tutte queste informazioni si possono inferire analizzando molte chiamate con argomenti particolari, ma non vi è una documentazione completa che espone chiaramente l'uso corretto del parametro e la necessità del padding, e il risultato sono dei comportamenti inaspettati nei programmi che usano le API. Una volta capito il problema, è facile aggirarlo, ma la scelta progettuale di utilizzare i timestamp convertiti a id è discutibile e, inoltre, non chiaramente documentata.

3.2 Creazione candele OHLCV

Una volta memorizzati tutti i dati di trade raw, e resi disponibili per le letture, è il momento di creare il dato aggregato che riassume le operazioni di scambio e che rappresenta la principale fonte di input per le intelligenze artificiali.

⁹Il padding è una delle numerose pratiche che prevedono l'aggiunta di dati all'inizio, a metà, o alla fine, di un messaggio. È usato per rendere uniforme un certo dato, allineandolo alla lunghezza specificata: se la lunghezza del messaggio finale deve assumere precisamente un certo valore, ma il testo contenuto è più breve, si aggiungono tanti valori di padding quanti la differenza fra la lunghezza del messaggio da inviare e quella del testo contenuto.

Le candele sono interamente generate grazie alla libreria di Python *pandas*, chiamando il metodo *resample*¹⁰:

```
def resample(self, rule, ...):
    Resample time-series data.
```

Convenience method for frequency conversion and resampling of time series.
Object must have a datetime-like index (DatetimeIndex, PeriodIndex, or TimedeltaIndex), or pass datetime-like values to the on or level keyword.

È un metodo della classe DataFrame, quindi, viene chiamato a partire da un oggetto contenente il dataset da ricampionare per creare la candela; come parametro è usato *rule*, un *timedelta* che rappresenta la lunghezza della candela.

Supponendo di avere, dunque, tutti i dati raw disponibili nel database, e di voler creare candele con timeframe di un'ora, per tutto il mese di gennaio 2020, il procedimento seguito è il seguente:

```
start_timestamp ← "2020-01-01 00:00"
end_timestamp ← "2020-02-01 00:00"
candle_length ← "01:00"
OHLCVs ← {}

for window ← start_timestamp to end_timestamp, step candle_length do
    end_window ← window + candle_len
    raw_trades ← {db prices with timestamp between window - end_window}
    OHLCVs ← OHLCVs ∪ resample(raw_trades, candle_len)
Commit OHLCVs to database
```

Dal punto di vista concettuale, il metodo citato aggrega i dati creando un nuovo record contenente i quattro attributi *open*, *high*, *low*, *close*, *volume*. Come già accennato:

- *open* è il prezzo di apertura della candela, quindi, il primo fra i prezzi della finestra di ora considerata.
- *high* è il maggiore fra tutti i prezzi nell'intervallo.
- *low* rappresenta il prezzo più basso raggiunto.

¹⁰Da documentazione pandas: <https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.resample.html>

- *close* è l'ultimo fra i prezzi all'interno dell'intervallo di tempo considerato.
- *volume*: numero di titoli scambiati all'interno delle operazioni, durante l'intervallo. Il numero di buy e sell può cambiare da candela a candela, possono esserci situazioni in cui molte operazioni sono effettuate oppure nessuno scambio avviene (e, quindi, la candela assume valori open, high, low, close identici alla candela precedente). Oltre al numero di scambi, cambia anche il numero di titoli comprati e venduti all'interno delle singole operazioni.

Per meglio chiarire come avviene il calcolo dei valori OHLCV, è riportato un esempio numerico. Si considera sempre lo stesso intervallo, inizio gennaio 2020, ma la candela creata è di lunghezza *cinque minuti*, in quanto i dati di una candela oraria sarebbero troppo numerosi da visualizzare, dell'ordine di qualche centinaio di trade.

I trade raw sono raffigurati nella prima tabella (Tabella 1), mentre la candela risultante è contenuta nella seconda (Tabella 2). Sono tralasciati i metadati e tutti gli altri campi superflui.

	time	price	volume
	2020-01-01 00:00:04	6410.2	0.0291153
	2020-01-01 00:00:20	6410	0.0291091
	2020-01-01 00:01:01	6410	0.0619882
	2020-01-01 00:01:03	6410.1	0.00267014
	2020-01-01 00:01:11	6410.6	0.00000692
	2020-01-01 00:01:11	6410.6	0.00000002
	2020-01-01 00:01:11	6410.6	0.001
	2020-01-01 00:01:12	6410.6	0.0389935
	2020-01-01 00:01:42	6410.3	0.015
	2020-01-01 00:02:42	6412.3	0.00443888
	2020-01-01 00:03:00	6412.1	0.0078
	2020-01-01 00:03:06	6411.3	0.01
	2020-01-01 00:03:06	6411.2	0.024148
	2020-01-01 00:03:10	6410	0.220904
	2020-01-01 00:03:12	6410	0.0460962
	2020-01-01 00:03:12	6410	0.00090377
	2020-01-01 00:03:13	6410	0.002
	2020-01-01 00:03:13	6408.5	0.0740962
	2020-01-01 00:03:13	6406	0.00773502
	2020-01-01 00:03:15	6409.9	0.0828255
	2020-01-01 00:04:03	6402.1	0.293174
	2020-01-01 00:04:04	6402.1	0.21
	2020-01-01 00:04:04	6402.2	0.055969
	2020-01-01 00:04:04	6402.9	0.0934388
	2020-01-01 00:04:04	6403.1	0.060497
	2020-01-01 00:04:13	6401.7	0.0515177
	2020-01-01 00:04:41	6401.7	0.0407859
	2020-01-01 00:04:41	6401.7	0.017461
	2020-01-01 00:04:52	6401.6	0.0758846

Tabella 1:

Elenco di prezzi del titolo BITCOIN-EURO per i primi cinque minuti di gennaio 2020. Il primo scambio disponibile è stato effettuato alle 00:00:04 e, in totale, sono presenti 29 scambi.

time	open	high	low	close	volume
2020-01-01 00:05:00	6410.2	6412.3	6401.6	6401.6	1.5576

Tabella 2:

Candela OHLCV risultante dal ricampionamento dei record nella Tabella precedente. Il timestamp della candela è quello dell'ultimo record e rappresenta tutti gli scambi avvenuti fino alle 05:00, partendo dall'ultima candela di cinque minuti.

3.3 Creazione strategie

Le strategie di investimento sono create a partire dalle candele OHLCV appena create e memorizzate nel database. Si possono analizzare diversi tipi di candele ma, per gli scopi di Sentyment, sono usate quelle giornaliere e, soprattutto, orarie.

A prescindere da quale sia la logica secondo cui sviluppare una strategia, discusso nel Capitolo 1, è importante notare che queste producono un elenco di segnali, detti anche *trigger*, che indicano i punti in cui comprare o vendere un titolo. I segnali buy / sell sono rappresentati da un numero fra $[-1, 0, 1]$, dove -1 rappresenta *sell*, 0 *hold* e 1 *buy*. Questi segnali sono associati ad un determinato timestamp, che fa riferimento al momento nel tempo in cui la strategia ha deciso di effettuare la scelta.

Per quanto riguarda Sentyment, i trigger sono disponibili come file csv prodotti da tutte le AI che operano, in uno storico di dati che parte circa dal 2013. Avendo tutta la storia delle AI è possibile analizzarne le scelte per calcolare statistiche come la *performance* o il *guadagno*. Non è noto come Sentyment produca le strategie, se mediante dei metodi di analisi tecnica come MACD, SMA o altri indicatori visti nel Capitolo 1, o se utilizza una Neural Network per tentare di approssimare la funzione dei prezzi. In ogni caso, i dati prodotti sono trattati come il risultato di un'analisi e non ci si pone il problema di evincere la tecnica con la quale sono stati prodotti.

È possibile, però, spiegare come sono prodotti i segnali di acquisto e vendita a partire da dei semplici indicatori. Si prende come esempio una semplice strategia come quella descritta nell'introduzione, RSI (Relative Strength Index), utilizzata nel Capitolo 4 confrontandola con Sentyment. L'indicatore si basa sul fatto che quando il prezzo è forte allora tanto maggiori saranno le chiusure al rialzo rispetto a quelle al ribasso; il contrario accade nel momento in cui prezzo si trova invece in una fase di debolezza, le chiusure al ribasso a quel punto saranno certamente superiori di quelle al rialzo. RSI verifica se pesano di più le chiusure al rialzo rispetto a quelle al ribasso. Delle due direzioni quelle che pesano di più ci indicano la direzione prevalente del mercato

e quindi si possono identificare eventuali cambi di trend, punti in cui è opportuno comprare o vendere.

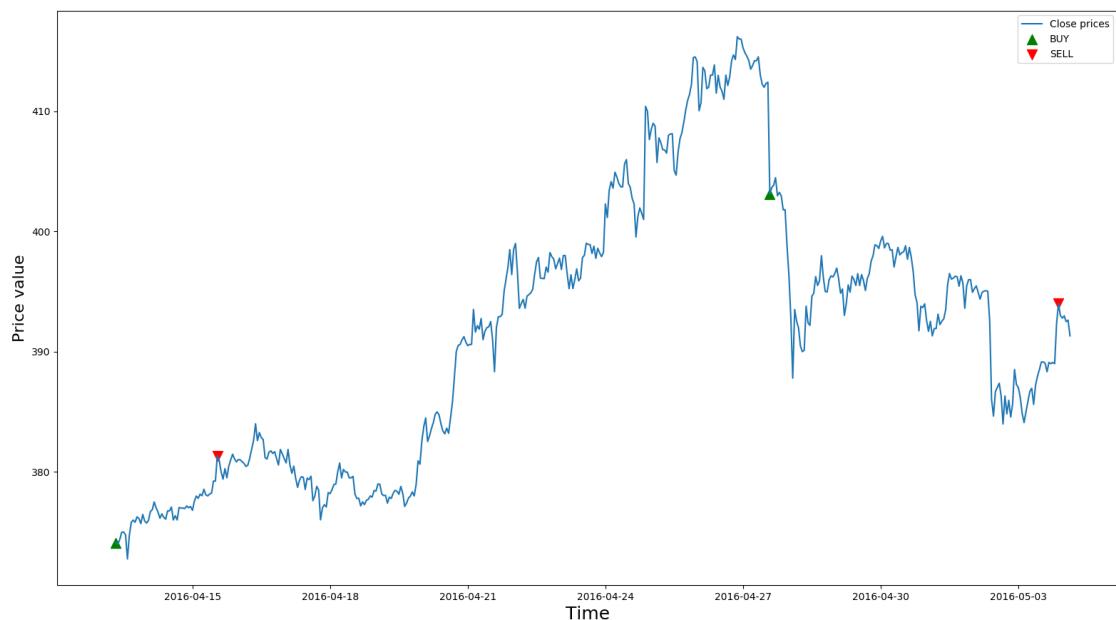
Il valore di RSI in ogni punto del grafico OHLCV è calcolato come segue:

$$RSI = 100 - \left(100 / (1 + U/D) \right) \quad (6)$$

fissata una certa lunghezza di periodo in cui calcolare, U è il guadagno medio ottenuto nel periodo e D la perdita media. Come per SMA e MACD, questo indicatore calcola delle medie mobili di lunghezza pari al periodo considerato e poi le usa per ottenere il risultato finale (RSI), che è un numero che oscilla fra 0 e 100, dove i valori superiori a 70 indicano una sovrastima del mercato e quelli inferiori a 30 una sottostima. Pertanto, quando l'indicatore supera tali valori significa che ci si trova in una zona particolare e sarà opportuno comprare se RSI è minore di 30 o vendere se RSI maggiore di 70. Andrebbe però valutato il momento esatto in cui operare invece di agire non appena l'indicatore tocca gli estremi. Per semplicità è qui considerato un semplice esempio in cui invece i trigger sono generati subito.

Si considera un breve dataset di prezzi OHLCV Bitcoin-Euro e si calcola l'oscillatore RSI in ogni punto. La funzione RSI appare in un altro grafico perché assume solo valori da 0 a 100 e non può essere rappresentata direttamente nel grafico dei prezzi. Nella prima immagine di Figura 14 si vedono i prezzi e i momenti in cui la strategia RSI decide di comprare o vendere. Nella seconda immagine la funzione RSI e i punti in cui tocca gli estremi.

Close prices and triggers



RSI strategy

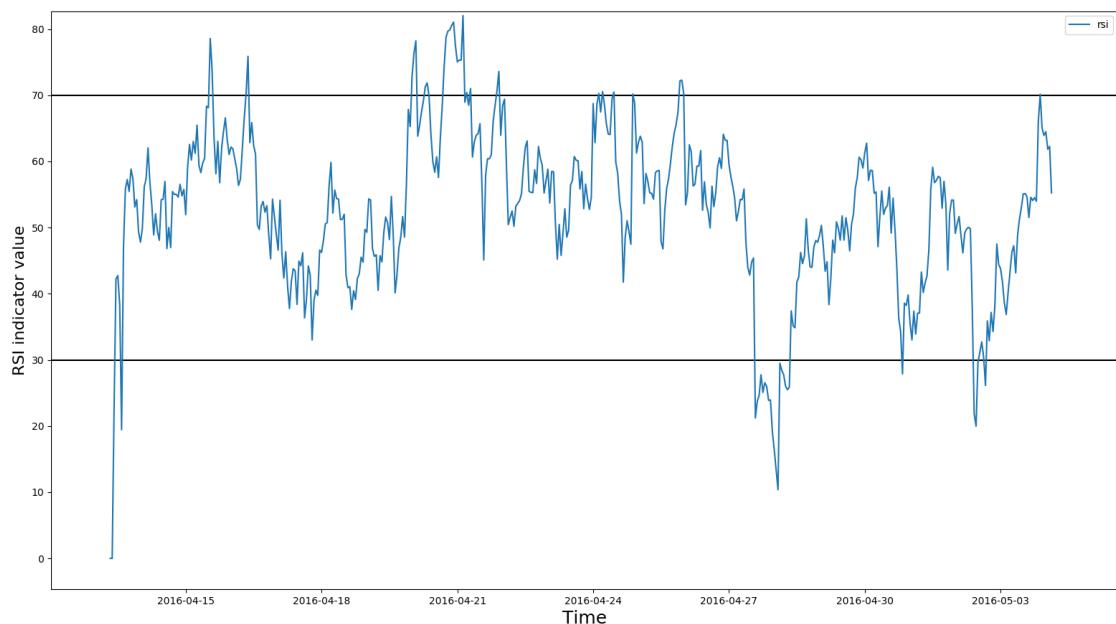


Figura 14: Grafico dei prezzi di chiusura, corredato di segnali buy e sell generati dalla strategia RSI. Nella seconda immagine è mostrato l'andamento del valore dell'indicatore RSI e i punti in cui tocca gli estremi 30 e 70, i quali indicano di generare il segnale compra e vendi.

Una volta calcolata la funzione e i punti di crossover, si ottiene un dataset di prezzi accompagnato dai trigger. Quelli mostrati in figura sono soltanto relativi ai buy e sell; in tutti gli altri punti è segnata un'operazione di hold (0). Per ogni istante si ha quindi l'operazione buy / sell / hold (Tabella 3) e questo è tutto ciò che serve ai passi successivi per le analisi, infatti nel Capitolo 4 si useranno questi record per comparare varie strategie fra di loro, trattando indifferentemente quelle prodotte da indicatori, come RSI, e quelle generate da una AI. Anche le AI di Sentyment, infatti, producono un elenco di record di questo tipo.

time	trigger
...	0
2016-04-13 07:00:00	0
2016-04-13 08:00:00	1
2016-04-13 09:00:00	0
..	0
2016-04-15 12:00:00	0
2016-04-15 13:00:00	-1
2016-04-15 14:00:00	0
..	0
2016-04-27 13:00:00	0
2016-04-27 14:00:00	1
2020-04-27 15:00:00	0
..	0
2016-05-03 20:00:00	0
2016-05-03 21:00:00	-1
2016-05-03 22:00:00	0
...	0

Tabella 3: Elenco di trigger che rappresentano la strategi RSI, ricavati dai crossover dell'indicatore RSI del grafico precedente.

Capitolo 4

Test per intelligenze artificiali

I test sono, in realtà, un elemento chiave per il funzionamento dei progetti di intelligenza artificiale. Non si sviluppa semplicemente un algoritmo AI fornendo dati di training e mettendolo in produzione; è necessario verificare effettivamente che i dati di addestramento svolgano un lavoro sufficientemente buono per classificare accuratamente con una generalizzazione sufficiente senza incorrere in *overfitting*¹ o *underfitting*². Questo viene realizzato usando tecniche di validazione e mettendo da parte un sottoinsieme dei dati di training da usare durante la fase di validazione. In sostanza, si tratta di una sorta di test di qualità in cui si vuole assicurare che l'algoritmo ed i dati, oltre a iperparametri³ e metadati associati, lavorino tutti insieme per fornire i risultati predittivi sperati.

Se si sbaglia nella fase di validazione, bisognerebbe tornare indietro, cambiare i parametri e ricostruire di nuovo il modello, magari con dati di training migliori. Fatto ciò, si torna indietro e si utilizzano nuovi casi di test per verificare che il modello funzioni davvero come dovrebbe. Sebbene sembrino tutti aspetti del test e della validazione, questo accade durante la fase di addestramento della AI, prima che il modello sia messo in funzione.

¹Si parla di overfitting (adattamento eccessivo, "sovradattamento") quando un modello statistico molto complesso si adatta ai dati osservati (il campione) perché ha un numero eccessivo di parametri rispetto al numero di osservazioni. Un modello assurdo e sbagliato può adattarsi perfettamente se è abbastanza complesso rispetto alla quantità di dati disponibili.

²Underfitting ("sottoadattamento") si verifica quando un modello statistico non è in grado di approssimare adeguatamente la struttura sottostante dei dati. Un modello underfitted è un modello in cui mancano alcuni parametri, o termini, che apparirebbero in un modello correttamente specificato. Il sottoadattamento si verificherebbe, ad esempio, quando si adatta un modello lineare a dati non lineari. Tale modello tenderà ad avere scarse prestazioni predittive.

³Iperparametri sono quei parametri i cui valori sono decisi prima che inizi il processo di training. Si differenziano dai parametri che sono, invece, ricavati dal training stesso e vanno a comporre i settaggi dell'algoritmo AI. Gli iperparametri sono, invece, parametri del modello di training

Anche in fase di training, si stanno testando diversi aspetti. Innanzitutto, bisogna assicurarsi che l'algoritmo AI stesso funzioni. Non ha senso modificare gli iperparametri e allenare il modello se l'algoritmo è implementato in modo errato. Tuttavia, in realtà, è difficile avere un algoritmo non corretto perché la maggior parte di questi sono già inseriti nelle varie librerie AI: se si necessita di *K-Means Clustering*, *Support Vector Machine* o diversi tipi di reti neurali, basta semplicemente chiamare quella funzione di libreria in Python *scikit-learn* o qualunque sia lo strumento scelto. Gli sviluppatori AI non dovrebbero scrivere gli algoritmi da zero a meno non si abbia davvero una buona ragione per farlo; ciò significa che se non li si sta codificando da zero, non c'è molto da testare per quanto riguarda la correttezza del codice reale - si suppone che gli algoritmi abbiano già superato i loro test, per quanto possibile. In un progetto AI, il concetto di qualità non sarà mai focalizzato sull'algoritmo AI stesso o sul codice, supponendo che sia stato implementato come dovrebbe.

Restano dunque due cose da testare nella fase di addestramento per il modello AI stesso: i dati di training e le configurazioni degli iperparametri. In quest'ultimo caso, è possibile testare mediante l'uso di metodi di validazione, tra cui *k-fold cross validation* e altri approcci. Ciò contribuirà a determinare se le impostazioni dell'iperparametro sono corrette.

Pertanto, tutto quello che rimane da testare sono i dati stessi per il modello AI. Ciò significa non solo qualità dei dati, ma anche completezza. Il modello di formazione rappresenta adeguatamente la realtà di ciò si sta cercando di generalizzare? Si è inavvertitamente incluso del *bias* informativo, o indotto dall'uomo, nei dati di training? Si sta ignorando parte dei dati che funzionano in allenamento, ma falliranno durante la predizione perché i dati del mondo reale sono più complessi? Il QA per il modello di intelligenza artificiale ha a che fare con la garanzia che i dati di training includano un campione rappresentativo del mondo reale ed eliminino il più possibile *bias* umani.

Un sistema ben validato, e ben generalizzato, che utilizza dati di addestramento rappresentativi ed algoritmi da una fonte già testata e comprovata, dunque, dovrebbe dare i risultati previsti. Ma cosa succede quando non si ottengono quei risultati? La realtà è, ovviamente, più complessa. Nel mondo reale accadono cose che non avvengono nell'ambiente di test e può accadere che, nella fase di "inferenza", quando il modello è reso operativo, non si incontrino i risultati sperati.

I problemi che sorgono con i modelli nella fase di inferenza sono quasi sempre problemi di dati o disallineamenti nel modo in cui il modello è stato addestrato rispetto ai dati del mondo reale. Se si è certi che l'algoritmo funziona, e che i dati del modello di training, e gli iperparametri, sono stati configurati al meglio, significa che quando i modelli falliscono, si hanno problemi di disadattamento dei dati, o del mondo reale. Se dei dati di input sono errati, è necessario trovarli ed analizzarli. Se il modello non sta generalizzando bene, se c'è qualche sfumatura dei dati che deve essere aggiunta

per addestrare ulteriormente il modello, allora, è necessario passare attraverso un ciclo completamente nuovo di sviluppo di un modello di intelligenza artificiale con nuovi dati di addestramento e configurazioni di iperparametri per affrontare il giusto livello di adattamento a tali dati. Indipendentemente dal problema, le organizzazioni che rendono operativi i modelli di intelligenza artificiale hanno bisogno di un approccio solido in base al quale possono tenere sotto controllo le prestazioni dei modelli, e controllare la versione di quelli in funzione.

I progetti di intelligenza artificiale sono davvero unici in quanto ruotano attorno ai dati. I dati sono l'unica cosa che nei test è garantita crescere e cambiare continuamente. E' necessario, pertanto, considerare le AI come anche in continua crescita e cambiamento. [43]

4.1 Test per Sentyment AI

Si è, dunque, introdotto cosa rappresentano i test durante il ciclo di sviluppo di una AI. Quanto descritto fa sicuramente parte del processo di sviluppo di Sentyment e tutti i suoi parametri e iperparametri sono scelti in base a processi iterativi di test. L'ambito di questo capitolo, tuttavia, è descrivere i processi di test per una già esistente AI, al fine di dare una misura puntuale delle sue prestazioni, e non l'integrazione di un processo di testing all'interno del suo sviluppo. Sentyment è una AI da considerarsi *black-box* e, pertanto, non vi è dato modo di aggiungere, o modificare, componenti del modulo di AI, che si considera già completo nella sua implementazione e nelle varie fasi che compongono un processo di sviluppo di una AI. I dettagli della sua implementazione non sono noti e i analizzano soltanto gli input e gli output delle AI. Tuttavia quanto appena descritto è anche parte del framework di test sviluppato per analizzare Sentyment.

La domanda che ci si pone, a questo punto, è *come dare una misura di prestazioni di un algoritmo di AI per previsione e come confrontare questa misura con altri strumenti simili?*

Considerando l'ambito in cui opera l'algoritmo, non esiste un metodo generalmente riconosciuto come migliore. Esistono diversi approcci ed ognuno utilizza una metrica differente. Sentyment è, inoltre, molto diversa dagli altri algoritmi di AI per previsione di mercato e risulta, quindi, difficile trovare dei competitori con cui confrontare, a causa del fatto che, molto spesso, gli strumenti di trading sono proprietari e gli algoritmi accademici effettuano soltanto una previsione di serie storiche, e non una generazione di strategie vere e proprie.

Per i test con altri strumenti, la soluzione pensata rimane all'interno dell'ambito

di Sentyment e prevede di testare, fra di loro, le diverse AI che compongono l’organismo. Come accennato, Sentyment è composta da numerose AI che creano ognuna la sua strategia e soltanto una viene scelta, periodicamente, come migliore e di conseguenza utilizzata in produzione. La scelta della migliore comporta la definizione di una qualche metrica che possa dare modo di confrontare le AI fra di loro. La stessa metrica è implementata nell’algoritmo del *meta-learner* come funzione obiettivo da massimizzare, che comporterà la scelta della migliore AI rispetto a tale misura. Il fine ultimo di questa parte di test è, quindi, definire un metodo di confronto da poter implementare all’interno di *meta-learner*, grazie al quale poter scegliere quantitativamente, ed in automatico, la migliore strategia fra quelle disponibili.

Per quanto riguarda, invece, la misura delle prestazioni assolute dell’algoritmo di Sentyment, vanno fatte alcune considerazioni.

Si vuole dare una misura delle performance di Sentyment tralasciando il fatto che sia composta da diverse AI; si assume che la migliore sia già stata scelta e si prende il software come uno strumento unico da testare. In questo caso, non si vuole dare una percentuale di accuratezza sulle previsioni del valore di un titolo ma, invece, una misura qualitativa di quanto guadagno produce la strategia di investimento, considerando diversi intervalli temporali, anche rispetto ad altre possibili strategie ed algoritmi di investimento.

Il confronto è svolto a posteriori. A differenza dei precedenti test, che monitorano costantemente le AI mentre producono nuovi risultati al fine di aggiornare il sistema scegliendo la migliore, se sta andando meglio della scelta precedente, lo scopo di questa seconda fase di test è, invece, prendere dei risultati e confrontarli con altri algoritmi. Tali algoritmi producono delle semplici strategie di investimento e sono utili al fine di confrontare delle strategie più complesse.

4.2 Metriche

Su quale base si può affermare che una strategia di investimento sia migliore di un’altra? Va definita una metrica secondo cui misurare in modo che, data una certa strategia calcolata in un dataset di prezzi, da un certo istante iniziale ad uno finale, si è in grado di produrre tale metrica come un valore numerico e, quindi, confrontarla con i valori prodotti da un’altra strategia, ordinando in modo decrescente i risultati e prendendo il primo come migliore.

La scelta della metrica è il punto centrale: a seconda di quali *feature* (caratteristiche, indicatori) scegliere si avranno risultati molto diversi. Per esempio, se si sceglie di confrontare due strategie sul piano del guadagno finale apportato, quella che fa guadagnare di più vince. Ma se si considera, per esempio, la propensione al rischio,

la stessa strategia che prima ha vinto potrebbe ora perdere, perché, magari, molto rischiosa e, se si fosse considerato un altro dataset, in certe condizioni, potrebbe condurre ad una perdita disastrosa anche dal punto di vista del guadagno.

Per questi motivi, dopo una attenta analisi delle possibili feature, si è scelto di considerare le seguenti:

- Somma dei guadagni (*gross profit*). Per ogni operazione vincente (apertura e successiva chiusura con ricavo positivo), il guadagno prodotto.
- Somma delle perdite (*gross loss*). Per ogni operazione non vincente, la perdita subita.
- *Profit factor*. Rapporto fra somma dei guadagni e somma delle perdite.

Ognuno di questi aspetti contribuisce alla metrica finale ed esprime una certa caratteristica che si considera importante come feature di una strategia di investimento. I punti chiave sono, in linea con lo scopo di Sentyment, il rispetto per la propensione al rischio (espressa dalla somma delle perdite, una misura di volatilità) e la massimizzazione del guadagno. L'impatto delle operazioni non vincenti è maggiore rispetto al ricavo da operazioni vincenti, o del profitto, perché si vuole tenere sotto controllo la volatilità e le perdite dovute ad operazioni rischiose.

La metrica finale è un *rapporto pesato* dei valori prodotti da queste singole misure, in modo da riassumere le metriche in un singolo risultato numerico finale, confrontabile. Gli indicatori sono pesati per un coefficiente che indica l'importanza del termine all'interno della formula ed i valori dei pesi sono stati scelti empiricamente.

Data la somma dei guadagni delle operazioni vincenti, *gross_profit* e il suo peso, *w_gross_profit*, la somma delle perdite delle operazioni non vincenti e il suo peso, *gross_loss* e *w_gross_loss*, la formula finale della metrica, *performance* (chiamata anche *score* in questo documento), è:

$$\text{performance} = ((\text{gross_profit} * \text{w_gross_profit}) / (\text{gross_loss} * \text{w_gross_loss})) \quad (7)$$

Non sono mai considerati i valori netti di guadagno perché, ai fini del confronto con altre AI, risulta più espressivo considerare il rapporto fra guadagno e perdita, piuttosto che un valore assoluto di guadagno. L'informazione che racchiude un alto guadagno, isolata a sè, non ha lo stesso valore che può avere la stessa vincita in seguito ad una grossa perdita. È inutile puntare a guadagni sempre più alti quando il rischio che si deve correre per ottenerli supera le aspettative di un investitore. Per questo motivo ci si basa soltanto su una misura pesata del profitto (rapporto fra vincite e perdite).

La formula 7 è calcolata su ogni strategia prodotta dalle diverse AI, considerando un periodo di tempo che parte dai primi dati disponibili, fino all'ultimo dato possibile prodotto. Avendo a disposizione l'elenco dei prezzi con associata l'operazione buy / sell / hold prodotta dalla strategia, è possibile simulare le suddette operazioni per ricavare i guadagni delle singole e, quindi, stabilire se si trattava di un'operazione vincente (guadagno positivo) o non vincente ed il guadagno associato. Questa formula è utilizzata soltanto all'interno del *meta-learner* ed è lo *score* usato per allenare l'agente.

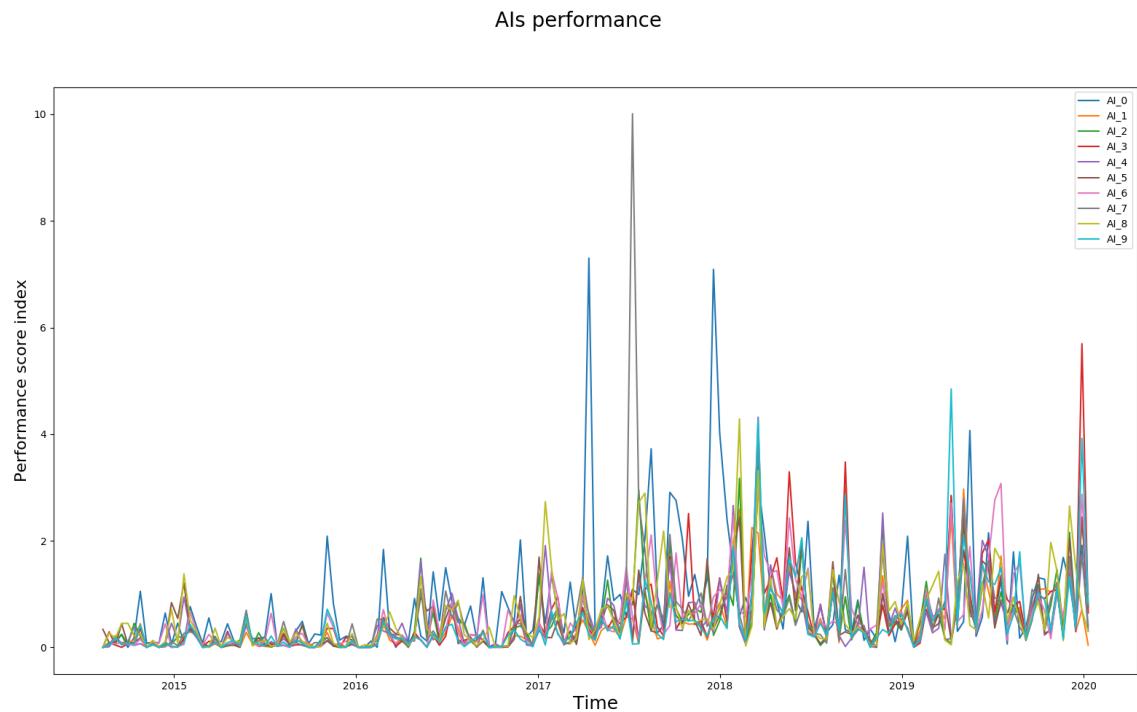


Figura 15: Grafico comparativo dei risultati prodotti da 10 differenti AI di Sentyment. In questo caso è analizzata la funzione *performance* (Formula 7) per ciascuna delle AI prese in considerazione. La funzione è calcolata partendo dall'inizio del dataset (BITCOIN-EURO a partire da inizio 2014, fino ad inizio 2020) in intervalli di 250 candele orarie. Alcune AI si posizionano sempre in alto rispetto ad altre, ed è per questo motivo che è necessario sceglierne una migliore fra tutte.

4.3 Test comparativi e Meta-learner

Verrà ora affrontato il tema centrale della tesi: sviluppare una AI supervisore in grado di scegliere un rappresentante migliore fra una serie di agenti intelligenti.

Sentyment è composta da numerose AI che creano strategie di investimento ed ognuna opera indipendentemente dalle altre, producendo costantemente dati relativi ai nuovi trade aggiornati in tempo reale e differenziandosi leggermente dalle altre. Le AI attive sono solitamente 10, ma il numero potrebbe variare a seconda degli iperparametri scelti durante il ciclo di test. Per ogni AI sono noti i suoi *trigger*, cioè le operazioni di buy / sell relative ad un certo istante di tempo che l'agente ha prodotto.

time	trigger
2020-01-13 18:00:00	1
2020-01-13 19:00:00	0
2020-01-13 20:00:00	0
2020-01-13 21:00:00	0
2020-01-13 22:00:00	0
2020-01-13 23:00:00	-1

Tabella 4: Elenco di record che rappresentano il risultato prodotto da una delle AI di Sentyment. Sono state considerate le candele orarie: per ogni ora si ha il segnale di buy / sell / hold. I trigger sono rappresentati da un valore numerico: 0 per hold, 1 per buy e -1 sell. In questa tabella è stata aperta una posizione di acquisto il 13 gennaio 2020 alle ore 18:00 ed è stato venduto il titolo alle 23:00

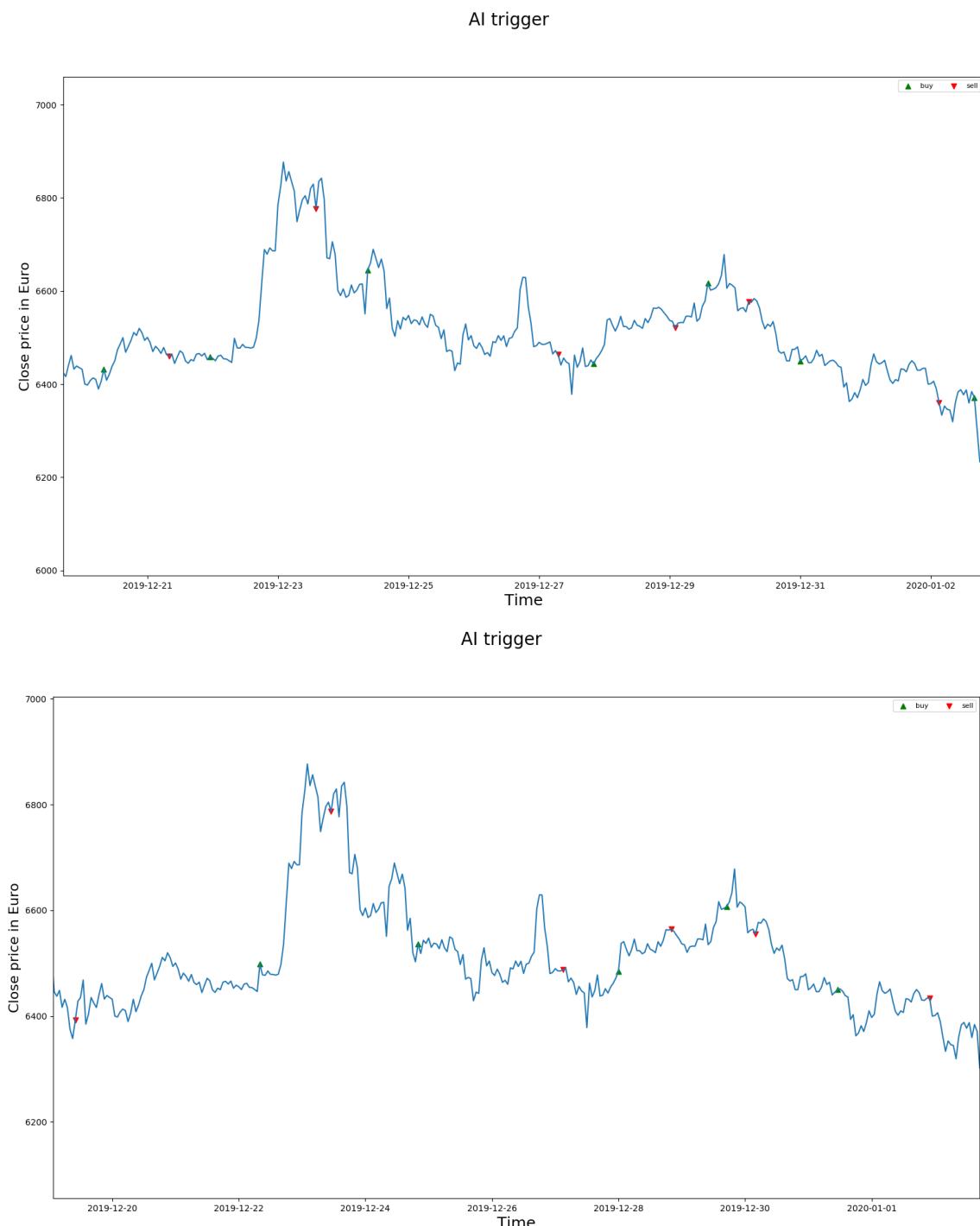


Figura 16: Esempio di grafico dei prezzi di chiusura di BITCOIN-EURO in un certo periodo, ed i *trigger* di due AI di Sentyment. Si nota che i segnali buy / sell non corrispondono esattamente fra una AI e l'altra, confrontando questi due grafici, ma sono leggermente spostati. Ognuna produce strategie differenti e, anche se probabilmente molte AI identificano gli stessi trend nel grafico, alcune potrebbero decidere di comprare, o vendere, leggermente prima, o dopo, rispetto alle altre.

Unendo l'elenco di operazioni di ogni AI per un certo periodo, con il dataset OHLCV dei prezzi (Tabella 5), è possibile calcolare uno *score* basato su numerose statistiche, descritto nella precedente sezione. La AI supervisore, fondandosi su questo score, sceglie qual è la AI che lo massimizza di più rispetto alle altre, tenendo conto anche del fatto che il punteggio varia a seconda del periodo di tempo considerato, ed anche all'interno dello stesso dataset, se si scelgono punti iniziali e finali differenti.

Inoltre la performance (grafico) è fortemente correlata all'andamento dei prezzi, cioè se il mercato è in crescita allora anche la performance ha buone chance di salire, mentre in un mercato in calo è difficile effettuare operazioni vincenti con alto profitto.

time	price	trigger_AI_1	trigger_AI_2	trigger_AI_3
2020-01-13 18:00:00	153.3	1	1	0
2020-01-13 19:00:00	153.3	0	0	0
2020-01-13 20:00:00	152.9	0	0	1
2020-01-13 21:00:00	153.5	0	0	-1
2020-01-13 22:00:00	153.5	0	-1	0
2020-01-13 23:00:00	153.7	-1	0	0

Tabella 5: Elenco di record contenenti le scelte di alcune AI, unito ai prezzi di chiusura delle candele associate. Questi dataset sono tutto ciò che il supervisore necessita in input per essere allenato ed operare la sua scelta.

Un approccio di machine learning che si adatta particolarmente al contesto descritto è il *reinforcement learning*, tecnica di apprendimento automatico che punta a realizzare agenti autonomi in grado di scegliere azioni da compiere per il conseguimento di determinati obiettivi tramite interazione con l'ambiente in cui sono immersi. L'apprendimento per rinforzo [45] è uno dei tre paradigmi principali dell'apprendimento automatico, insieme a *supervised* e *unsupervised learning*⁴. A differenza degli altri due, questo paradigma si occupa di problemi di decisioni sequenziali, in cui l'azione

⁴L'apprendimento supervisionato (*supervised learning*) è una tecnica di apprendimento automatico che mira ad istruire un sistema, in modo da consentirgli di elaborare, automaticamente, previsioni sui valori di uscita di un sistema rispetto ad un input, sulla base di una serie di esempi ideali, costituiti da coppie di input e di output, che gli vengono inizialmente forniti.

da compiere dipende dallo stato attuale del sistema e ne determina quello futuro. La qualità di un'azione è data da un valore numerico di "ricompensa", ispirata al concetto di reinforcement, che ha lo scopo di incoraggiare comportamenti corretti dell'agente. Questo tipo di apprendimento è, solitamente, modellato tramite i processi decisionali di Markov⁵ e può essere effettuato con diverse tipologie di algoritmi.

Questa tecnica si basa sul presupposto che, all'interno di un sistema, si possano: scegliere degli output sulla base degli input ricevuti, valutare l'efficacia degli output rispetto ad un preciso parametro di riferimento e cambiare il meccanismo di scelta degli input per massimizzare la valutazione di efficacia.

Quando si effettua una scelta efficace, allora, la valutazione di efficacia manda in output un premio proporzionale all'efficacia della scelta. Quando, invece, è effettuata una scelta inefficace, l'output della valutazione manda una penalità proporzionale. Osservando le scelte e le ricompense, si cerca di modificare la funzione matematica che regola le scelte, in modo da massimizzare la quantità e la qualità dei "premi".

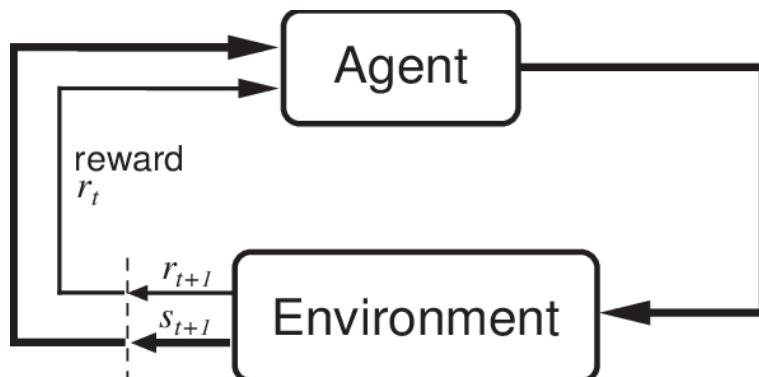


Figura 17: Idea generale di un processo di *reinforcement learning*: un agente interagisce con l'ambiente per imparare dall'esperienza. Per ogni istante di tempo, t l'agente è in un certo stato S_t ed esegue l'azione A_t . Come risultato, passa in un nuovo stato S_{t+1} e riceve una ricompensa R_{t+1}

L'apprendimento non supervisionato (unsupervised) consiste nel fornire, al sistema, una serie di input (esperienza del sistema) che egli riclassificherà, ed organizzerà, sulla base di caratteristiche comuni, per cercare di effettuare ragionamenti, e previsioni, sugli input successivi. Al contrario dell'apprendimento supervisionato, durante l'apprendimento, vengono forniti, all'apprendista, solo esempi non annotati, in quanto le classi non sono note a priori, ma devono essere apprese automaticamente.

⁵Un processo decisionale di Markov (MDP, Markov Decision Process) è un'estensione di Markov Chain e fornisce un framework matematico per la modellazione di processi decisionali. Si basa sull'assunto che la decisione futura dipende soltanto dalle condizioni presenti, e non da quelle passate. È alla base di molti algoritmi di reinforced learning

Nel contesto di Sentyment il supervisore rappresenta l’agente che opera le scelte. L’ambiente (i dati in input) sono i dataset contenenti i trigger e gli score che sono forniti all’algoritmo di reinforcement learning. Si permette all’agente di fare numerosi esperimenti lungo tutto il dataset, fornendo di volta in volta varie finestre progressive di dati. La scelta che il supervisore deve operare ad ogni passo è decidere quale fra le dieci AI sta finora, secondo lui, agendo meglio delle altre; gli score non sono noti all’agente: deve esplorare secondo qualche criterio le varie AI scegliendone una ”casualmente” e ricevendo una ricompensa se quella scelta aveva score maggiore delle altre.

L’output della scelta, ovvero la valutazione di efficacia, è il risultato della funzione di performance calcolata con indicatori statistici finanziari a partire dai trigger della AI scelta (formula 7), il cui risultato viene confrontato con quello prodotto dalle altre AI competitori.

Ritorna ancora una volta il concetto di *score*, o *metrica*. La metrica (Formula 7) guida l’agente supervisore nelle sue scelte: verrà ricompensato positivamente se, ad ogni iterazione, sceglie la AI fra le dieci che produce lo score massimo fra tutte le altre, mentre la scelta si considera negativa se non si verifica questa condizione.

Continuando ad eseguire esplorazioni e scelte, l’agente costruisce un modello di scelta per i vari competitori e sarà in grado, alla fine del dataset, di esprimere quale fra tutte le AI è più conveniente utilizzare.

4.3.1 Multi-armed bandit e Meta-learner

Una soluzione nota che si adatta alla situazione e fa uso di tecniche di reinforcement learning, implementata usando diversi algoritmi. La versione classica di *multi-armed bandit* discosta leggermente da questo ambito, ma può essere revisionata per trarre vantaggio dalla sua semplicità.[45]

Nella teoria delle probabilità, multi-armed bandit è un problema in cui un insieme fisso limitato di risorse deve essere allocato tra scelte concorrenti (alternative) in un modo che massimizza il guadagno atteso, quando le proprietà di ciascuna scelta sono conosciute solo parzialmente al momento dell’allocazione e possono essere meglio comprese col passare del tempo, o allocando risorse per la scelta. Questo è un classico problema di apprendimento per rinforzo che esplicita il dilemma del compromesso tra esplorazione (*exploration*) e sfruttamento (*exploitation*).

Il nome deriva dall’immaginare un giocatore d’azzardo in una fila di slot machine, che deve decidere su quali macchine giocare, quante volte giocare su ogni macchina, in quale ordine giocarle e se continuare con la macchina corrente, o provarne un’altra.

Nel problema, ogni macchina fornisce una ricompensa casuale da una distribuzione di probabilità specifica per quella macchina. L'obiettivo del giocatore è massimizzare la somma dei premi guadagnati attraverso una sequenza di tiri della leva. Il compromesso cruciale che il giocatore deve affrontare ad ogni prova è tra "sfruttamento" della macchina che ha il più alto profitto atteso ed "esplorazione", per ottenere maggiori informazioni sui guadagni previsti delle altre macchine.

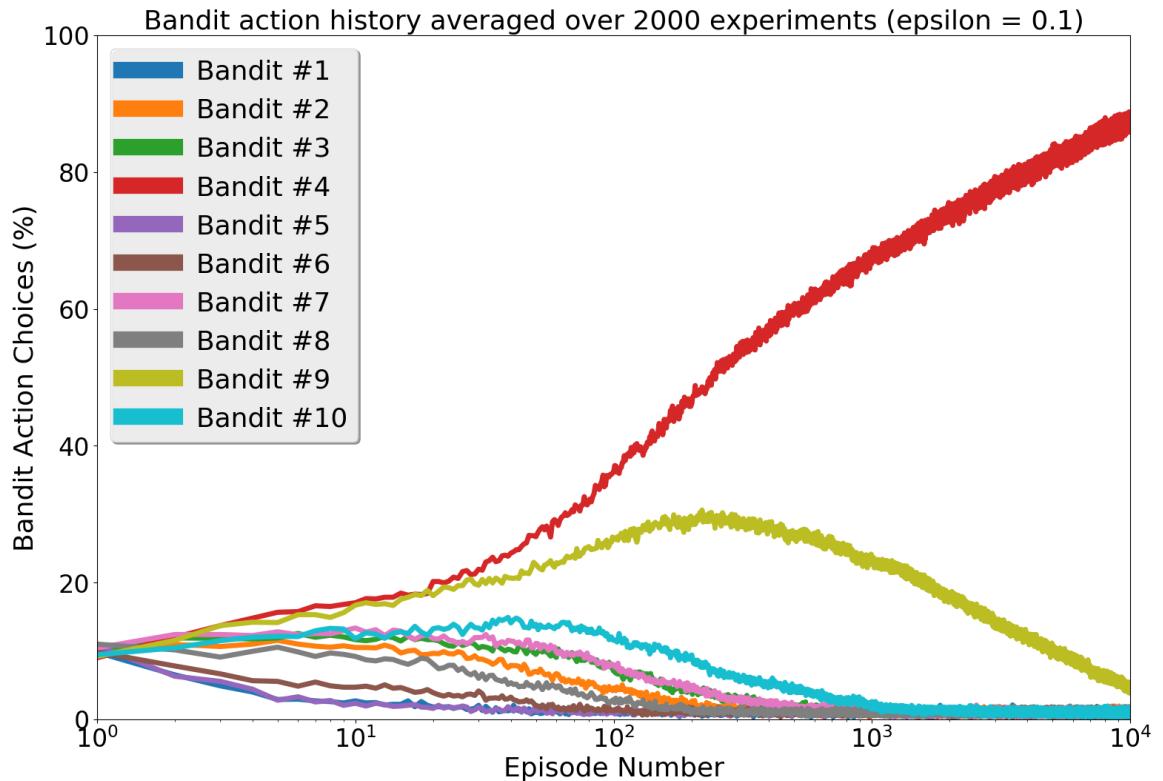


Figura 18:

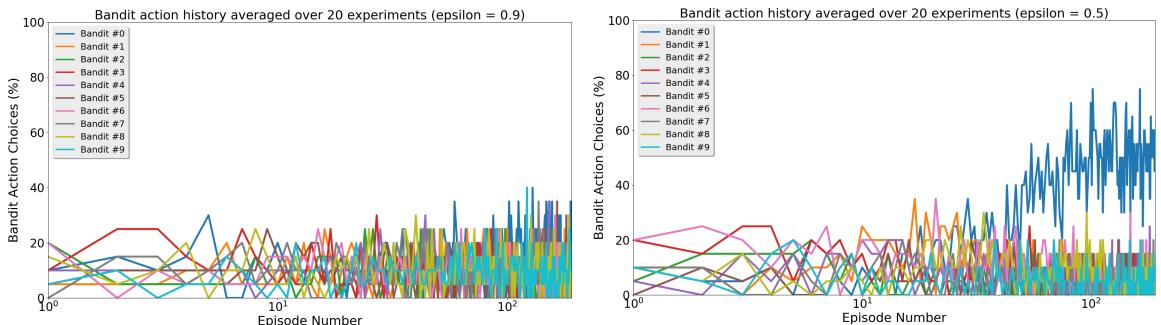
Esempio di scelte effettuate durante il training di un agente multi-armed bandit. Al crescere del numero di scelte effettuate (Episode number), si avrà una concentrazione sempre maggiore della scelta su uno specifico bandit, quello che, secondo l'algoritmo, è il più promettente. In questo esempio, si può notare come il bandit preferito sia il numero 4. Già dopo 10^2 scelte, i competitori si riducono praticamente a due soltanto, il bandit 4 ed il 9. Poco dopo, però, è il numero 4 ad essere scelto sempre più spesso nella fase di training.

La ricompensa a seguito di un'azione è immediata: tirando la leva di una specifica slot machine si può subito osservare se la macchina ha vinto o perso, ricevendo un *reward* di conseguenza. Se l'agente aveva scelto per la slot machine vincente riceve

una ricompensa positiva, viceversa negativa.

La versione adattata prevede di scegliere un dataset di prezzi e trigger delle AI e suddividerlo in un numero di *step*. Per ogni step, è calcolata la *performance* (Formula 7) di ciascuna delle AI in gioco. L'agente effettua la sua scelta valutando fra esplorare nuove soluzioni, con possibilità di trovare una nuova migliore, oppure scegliere di nuovo la AI che si è mostrata migliore nelle precedenti iterazioni. Questa scelta è guidata dal *tasso di apprendimento* (*epsilon*), un parametro che determina il comportamento dell'agente variando da 0 a 1, dove 0 rappresenta il non-apprendimento, la scelta si protrae sempre uguale, mentre 1 farebbe sì che l'agente si interessi solo delle informazioni recenti.

Più precisamente, *epsilon* rappresenta, durante la fase esplorativa, la probabilità della scelta di un nuovo bandit, rispetto a scegliere il migliore finora, in un singolo step. Ad ogni step di training si estrae un numero casuale fra 0 e 1 e, se l'estratto è minore di *epsilon* allora sarà scelto un nuovo bandit mentre, se si estrae un numero maggiore o uguale a *epsilon*, la scelta ricade sul migliore trovato finora.



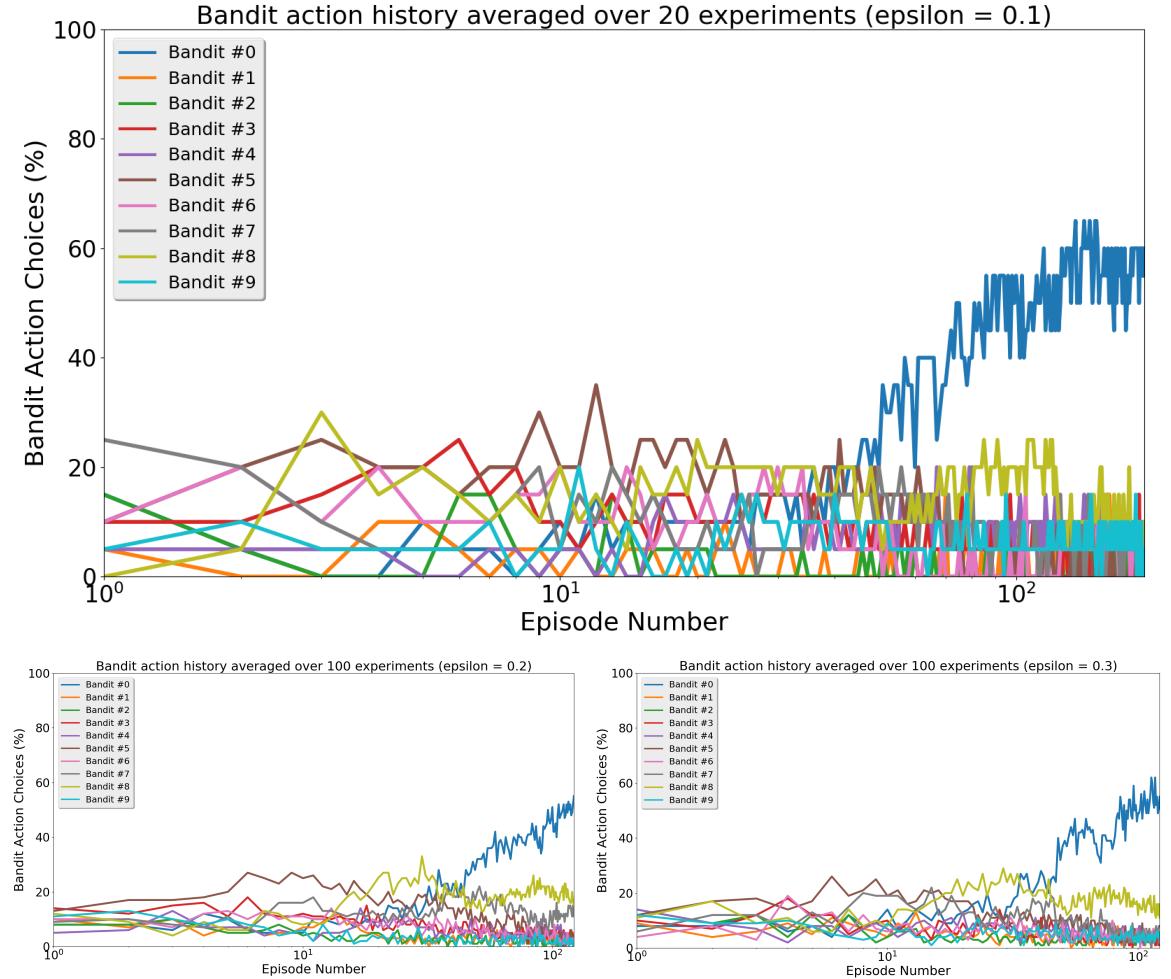


Figura 20:

Scelte effettuate dall'agente di apprendimento al variare del parametro *epsilon*. Con *epsilon* 0.9 si ha una scelta sempre diversa e perciò non riesce ad emergere un migliore, perché l'esplorazione è troppo casuale. Il valore intermedio 0.5 mostra una scelta più concentrata sul bandit 0, ma, anche dopo molti esperimenti, l'esplorazione continua anche su altri bandit, producendo dei picchi alti e bassi verso la fine. Infine, la scelta del parametro 0.1 distanzia ogni altro concorrente e si concentra fermamente sul bandit 0. A seconda delle situazioni, sarbbe adatto utilizzare anche 0.2 / 0.3, i quali lasciano spazio ad un eventuale secondo migliore.

A seconda della scelta effettuata ad ogni step, cioè puntare sul vecchio migliore o esplorare un nuovo concorrente, l'agente riceve una *ricompensa*.

L'implementazione dell'algoritmo multi-armed bandit prevede soltanto una ricompensa positiva (1, il bandit scelto vince questo giro) o negativa (0, il bandit non è vincente). La ricompensa positiva si ha quando l'agente, in questo step, ha scelto l'AI con performance più alta rispetto a tutte le altre. In tutti gli altri casi, la ricompensa sarà negativa.

Continuando nelle scelte attraverso i vari step, fino alla fine del dataset, l'agente aggiorna la sua funzione che regola le scelte e, dopo un sufficiente numero di iterazioni e ripetendo anche più volte lo stesso esperimento, l'algoritmo è in grado di scegliere, con precisione sempre più alta, la AI migliore.

Alla fine di questa fase di training, l'agente fornisce lo storico delle scelte effettuate. La AI migliore è quella scelta il maggior numero di volte rispetto a tutte le altre. Facendo riferimento alla Figura 18, alla fine del training vince il bandit che ha ricevuto più scelte da parte dell'agente quindi, in questo caso, Bandit 4.

4.3.2 Scelta dei parametri e adattamento

L'algoritmo multi-armed bandit originale definisce già una serie di parametri che possono modificare l'esito delle scelte e dovrebbero essere decisi analizzando il contesto e scegliendo i migliori a seconda della situazione. In aggiunta, l'adattamento della soluzione al dominio di Sentyment costringe a prendere alcune decisioni.

Il problema delle slot machine prevede, come step, la scelta di un bandit e la successiva ricompensa immediata: tirando la leva si vince o si perde. Poi si ripete l'esperimento un gran numero di volte.

Nel caso di Sentyment, è evidente che questa situazione non si può modellare così facilmente, non esiste vincita o perdita immediata, ma soltanto un bandit che, all'interno di uno step, si sta comportando meglio degli altri. È stato pensato un adattamento in cui si divide il dataset di training in step da 250 candele, si calcola lo score che ogni AI produce considerando quei 250 record e si sceglie la migliore all'interno di questo frame. Il numero 250 è scelto dopo numerosi esperimenti empirici e rappresenta il *trade-off* fra un calcolo attendibile della performance e il poter disporre di abbastanza punti per il training.

Dividendo tutto il dataset disponibile (di circa 50000 record) in frame da 250 si ottengono 200 punti per poter allenare l'agente. Sono molto pochi e, per raggiungere un risultato accettabile, si deve compensare facendo ripetere molte più volte il training sugli stessi dati.

Il problema risulterebbe mitigato se si scegliesse una dimensione di step più piccola di 250 candele, avendo così a disposizione più punti. Così facendo, però, il calcolo della performance all'interno di un frame da poche candele risulta poco preciso e produce

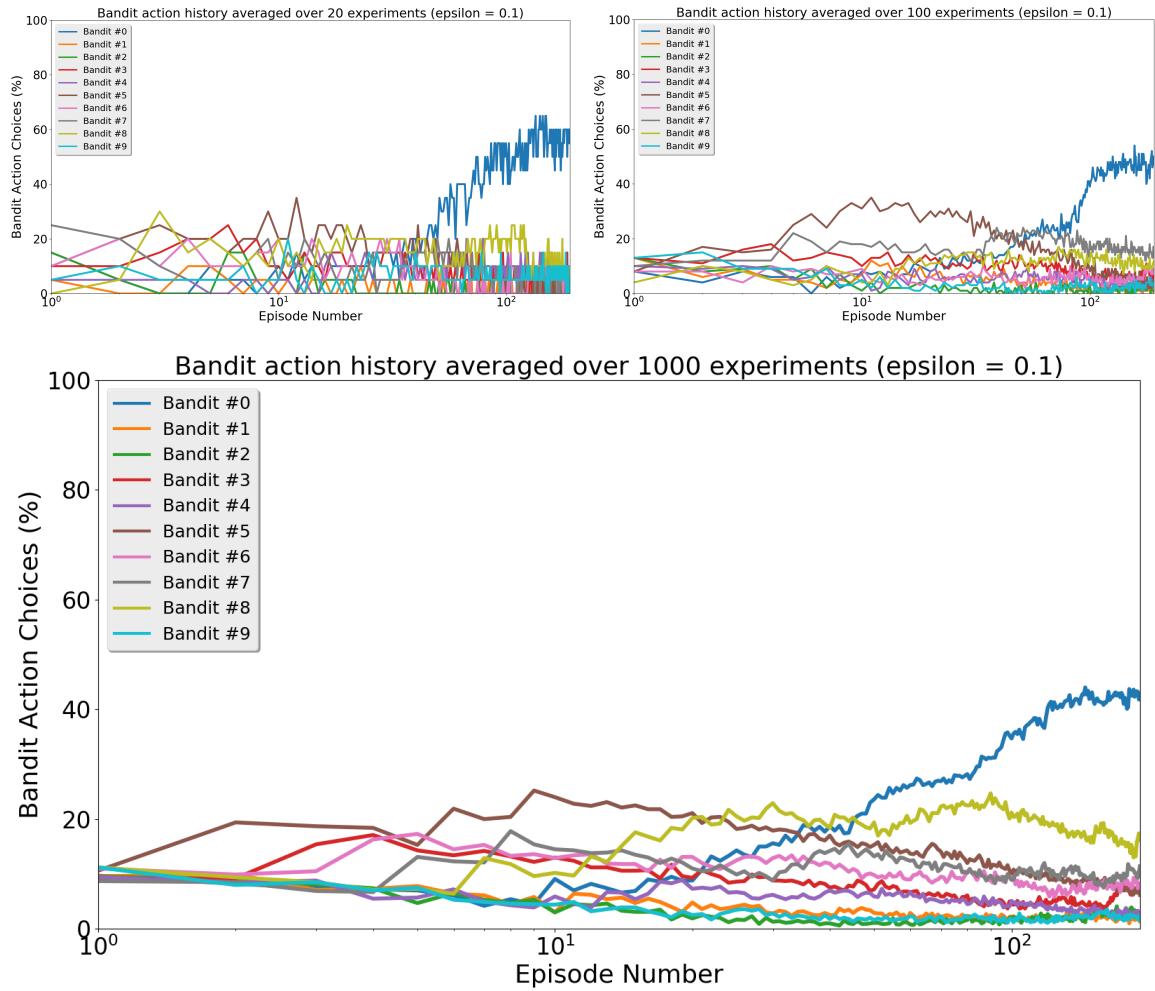
un risultato sempre molto simile rispetto a tutte le AI. Considerando il basso numero di operazioni buy / sell prodotti dalle AI in relazione ai molti più numerosi hold e vista la distanza media fra un buy e il successivo sell di circa 25-30 candele, si è scelto di lasciar operare le AI per almeno 10 giorni prima di calcolare la performance. Aspettando almeno una decina di operazioni di acquisto e vendita fra uno step e l'altro si lasciano le AI libere di effettuare qualche azione e di differenziarsi fra di loro, senza costantemente monitorare ogni singola operazione e quindi si effettua il controllo circa su base settimanale.

Proseguendo con l'esecuzione dell'algoritmo, l'agente effettua la sua scelta ed ottiene la ricompensa di conseguenza. L'agente ovviamente non conosce a priori le varie performance. Si procede considerando i prossimi 250 record e si ripete fino alla fine del dataset.

A causa della gran quantità di dati necessari per un singolo step, scelta dovuta in seguito alla proprietà intrinseche delle AI e del contesto finanziario, sono, di conseguenza, necessarie moltissime candele per allenare l'algoritmo di reinforcement learning. L'agente compie, come minimo, 200 step per il training, con esempi di anche 2000 step [44], arrivando ad usare candele fino al 2014 (50000 candele nel passato) per un training efficace per il 2020.

Tutta questa fase intermedia di "training" sull'intero dataset, detto *experiment*, infine, va ripetuta per molte volte. Il numero deve essere abbastanza alto per compensare il basso numero di punti presi sul dataset ed è quindi stato scelto, in accordo con altri esempi, almeno 100 come numero di experiment. Si tratta quindi di ripetere il training su tutto il dataset almeno 100 volte. Per ottenere questo numero si sono analizzati i grafici ottenuti dalle scelte dell'agente con parametro *numero di esperimenti* che varia da uno più basso, 20, fino a 100 e infine 1000. Un numero di esperimenti basso offre senza dubbio una velocità di esecuzione molto alta rispetto ai concorrenti (circa 30 minuti per 20 esperimenti, diverse ore per 100 e qualche giorno per 1000), che risultano troppo lenti e quindi costringono a poter eseguire l'intero algoritmo soltanto di rado. Scegliendo troppi pochi esperimenti, però, non si hanno poi abbastanza dati per un training efficace e si rischia di semplificare eccessivamente il modello.

Guardando le figure si può osservare come il numero di esperimenti cambia il grafico delle scelte, mettendo a disposizione sempre più punti per poter meglio effettuare il calcolo della migliore.

**Figura 22:**

Stesso grafico dei precedenti, scelte dell'algoritmo multi-armed bandit, questa volta al variare del numero di esperimenti. Si può notare una linea sempre meno spezzata al crescere dei punti, mentre con basso numero di esperimenti i punti sono molto pochi.

Infine è d'obbligo parlare dei dati scelti per il training, cioè del dataset OHLCV vero e proprio. Si è usato soltanto BITCOIN-EURO, perché questo era l'ambito dell'applicazione. Ulteriori dataset potrebbero essere disponibili in un futuro e l'algoritmo di *meta-learner* potrebbe essere applicato a nuove monete, adattando i parametri ai dati. Una volta testata la sua efficacia applicato ad uno strumento finanziario, potrebbe risultare utile anche a qualsiasi altro tipo di valuta o criptovaluta.

Parlando di qualità dei dati, va considerato che i primi trade di Bitcoin dal 2013 al

2014 rappresentano una situazione diversa da quella attuale: il titolo era appena comparso e, per qualche ragione, in forte decrescita, a differenza dei dati del 2019-2020, e questo comportava un calcolo delle performance molto diverso da quelle attuali. Il mercato era molto diverso e le performance ai tempi erano decisamente più basse; ora in qualche punto del dataset (ad esempio fra il 2017 e il 2018) si ottengono performance molto alte (Figura 22), in seguito ad una crescita del valore di Bitcoin, mentre nei periodi successivi si ritorna ad una situazione altalenante. Per questi motivi i primi dati di Bitcoin sono stati esclusi per evitare distorsioni nei calcoli.

Inoltre va considerato il fatto che si dovrebbero sempre usare i dati più recenti possibili per allenare algoritmi di trading, perché meglio rappresentativi del mercato attuale, rispetto a dati più vecchi. Tuttavia, a causa della scarsità di dati per il training, sono comunque stati inclusi tutti i dati possibili, tranne quelli iniziali. In future iterazioni dell'algoritmo di *meta-learner*, quando abbastanza nuovi dati saranno disponibili, potrebbe essere più opportuno escludere i dati più vecchi in favore di quelli più aggiornati, per meglio rappresentare il mercato attuale.

4.4 Test framework workflow e risultati

Ora si analizzeranno i risultati ottenuti applicando l'algoritmo di multi-armed bandit, nella sua implementazione *meta-learner*, al contesto descritto. È considerato un dataset di prezzi che parte da fine 2014 fino ai primi 2020, contenente circa 50000 record, cioè 50000 candele orarie. Le AI i cui trigger sono stati resi disponibili sono organizzate in un pool di 10 competitori ed è calcolata la performance secondo la Formula 7. Il dataset è diviso in step da 250 candele, in modo da dare all'agente la possibilità di fare diverse prove durante ogni esperimento.

Prima di eseguire l'algoritmo finale su tutte e 10 le AI, qui riportato, sono stati considerati molti casi di test prendendo dei sottoinsiemi casuali e di varie dimensioni di AI ed analizzando la scelta prodotta dall'algoritmo. Calcolando prima il grafico delle performance si può chiaramente vedere in quasi tutti i casi, usando anche strumenti come la media, quale è la AI che risulta migliore nei vari sottoinsiemi e quindi si è costruita una base di verità etichettando i casi di test manualmente. Controllando poi, per ogni test, se l'algoritmo sceglieva come migliore la AI etichettata in precedenza, si sono sempre ottenuti buoni risultati. Il metodo di etichettatura "manuale", però, non è precisamente affidabile e una eventuale scelta differente da parte di *meta-learner* potrebbe comunque essere giustificata da altri fattori, motivo per cui è appunto necessario un algoritmo intelligente per eseguire la scelta e per cui una misura di *precision* e *recall*⁶ non è molto attendibile in questo caso. Volendo fare un paragone, si tratta

⁶Precision e recall sono due comuni classificazioni statistiche, utilizzate in diversi ambiti. Precision può essere vista come una misura di esattezza o fedeltà, mentre recall è una misura di completezza.

più di un problema di apprendimento non supervisionato in cui la *ground truth* non è fornita in precedenza, ma va scoperta.

Tutti i test sono inoltre stati eseguiti considerando soltanto l'istante finale del dataset, ovvero si è decisa la migliore AI come se fosse stato necessario sceglierne una da attivare nel presente e mantenere tale scelta per il futuro. Non è stato quindi possibile effettuare ulteriori test in altri punti passati del dataset, come, ad esempio, verso metà 2017 (Figura 23), quando AI_0 è stata superata per un breve momento da un picco di AI_7. In quel periodo sarebbe stato interessante scoprire se la scelta fosse ricaduta, appunto, su AI_7 o se l'agente avrebbe comunque mantenuto AI_0. Il motivo per cui altri test in momenti precedenti non sono possibili è il troppo basso numero di dati. Un training con meno di 200 punti non può produrre un risultato accettabile e l'unico momento in cui si può disporre di tali numeri è nell'istante finale del dataset.

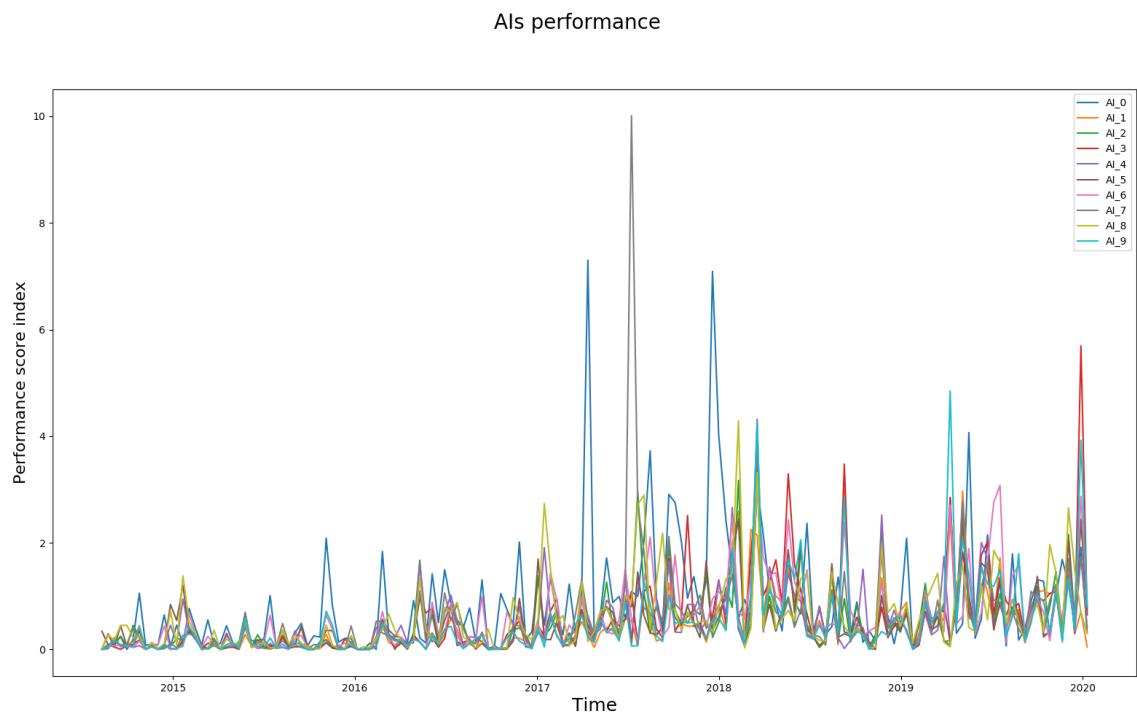


Figura 23:

Grafico riassuntivo delle performance delle 10 AI considerate per il training di *meta-learner*. Le performance sono calcolate a step di 250 candele (circa 10 giorni) ed utilizzando la Formula 7. Le performance e i rendimenti sono fortemente correlate all'andamento dei prezzi. Questi sono i dati calcolati dal supervisore per il suo training: ha una ricompensa positiva se, all'interno di ogni step, sceglie la AI che ha un valore più alto di performance. Come si può notare, all'inizio, ci sono pochi picchi di performance e soprattutto da parte di AI_0 (blu). Continuando più avanti nel tempo, i picchi compaiono sempre più spesso, sono più accentuati e sono prodotti da sempre più AI, risultato di un training di Sentyment che si adatta sempre di più ai dati e produce risultati sempre migliori.

Per prima cosa è calcolata la performance in tutto il dataset, diviso nei vari step, e creato il grafico in Figura 23 per analizzarne l'andamento. Si notano già da subito dei picchi di performance da parte di AI_0, mentre le altre AI emergono solo dalla seconda metà del dataset. Questi valori sono quelli usati per il training: all'interno di ogni step, l'agente deve riuscire a scegliere la AI che sta avendo il picco più alto di tutte le altre. Siccome AI_0 ha molti picchi e molto più alti delle altre, soprattutto nei primi periodi, allora un training con parametro *epsilon* 0.1 può già dare dei risultati positivi, identificando il bandit che si rivela sempre migliore a partire dall'inizio, senza dover scommettere su nuovi competitori.

La scelta di *meta-learner* infatti, come già accennato nei grafici di scelta, premia AI_0.

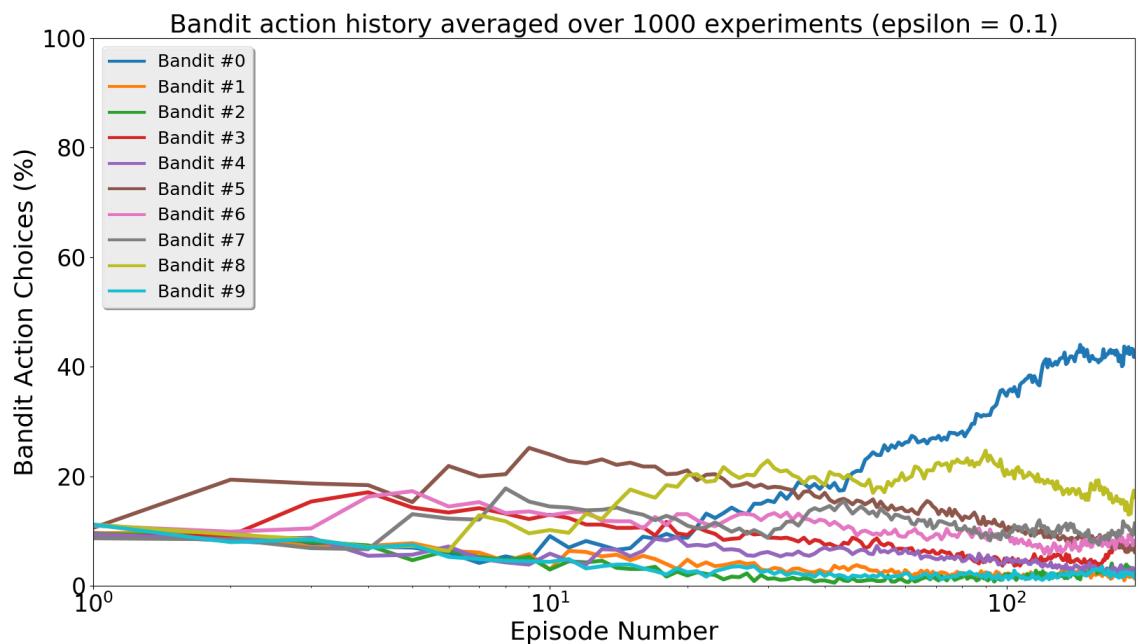
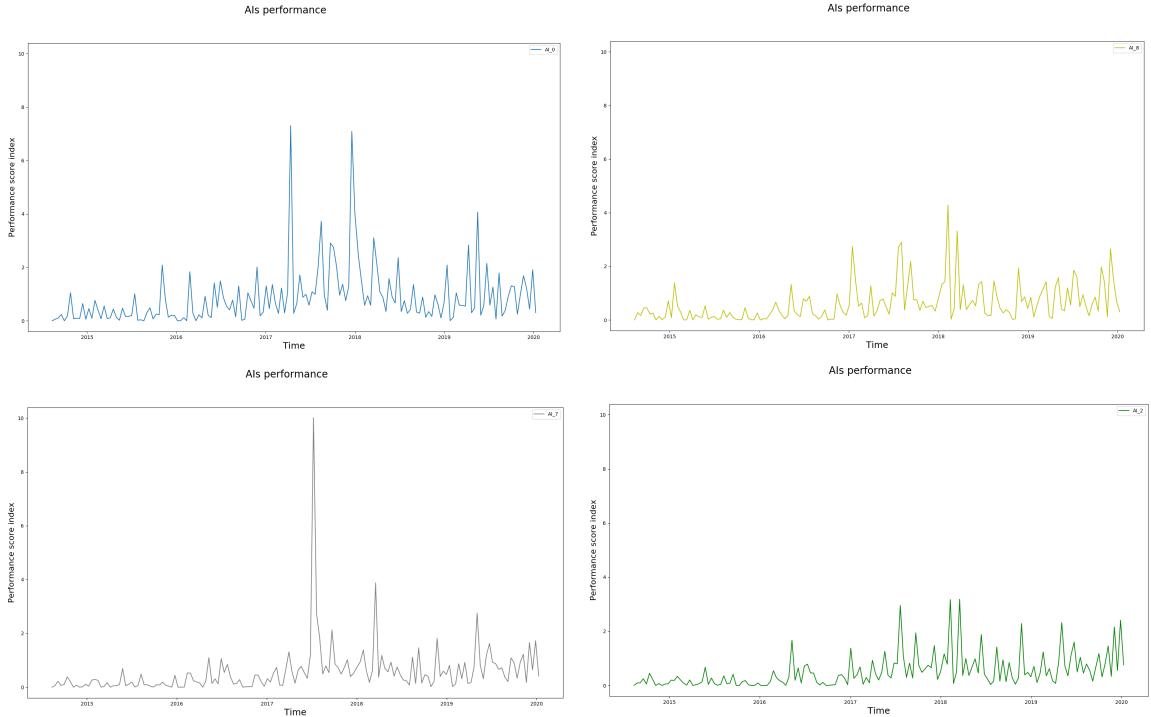


Figura 24:

Risultato del training dell'agente di *meta-learner*. In figura sono mostrate le scelte dell'agente all'aumentare del numero di esperimenti (step): quando si è allenato a sufficienza, sceglie la AI che più rispetta i criteri imposti. Il risultato finale che interessa è quale AI ha un valore più alto di scelta percentuale rispetto alle altre, a fine training. In questo caso l'agente è stato allenato con un *tasso di apprendimento* di 0.1, quindi, tende ad effettuare una scelta che tiene molto conto dei risultati passati. Se si considerano, infatti, le performance che hanno avuto tutte le AI nel corso del tempo, indubbiamente la AI_0 risulta la migliore.

Nelle successive immagini è analizzata la funzione di performance di AI_0, isolandola dalle altre in Figura 23. Si può notare che la performance di AI_0 è quasi sempre superiore di tutte le altre, considerando picco per picco.



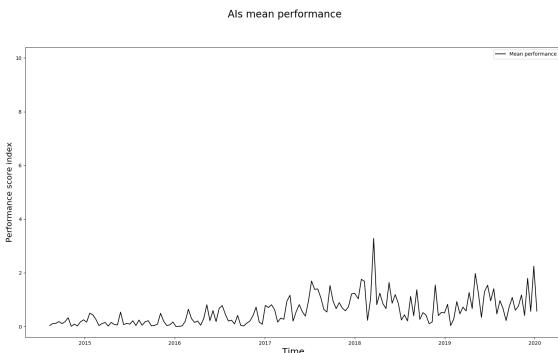


Figura 26:

Performance di diverse AI a confronto, isolate dal grafico : la migliore (AI_0), la seconda e terza posizione (AI_8 e AI_7), l’ultima (AI_2) e la media di tutte.

La performance media di AI_0, infatti, risulta essere di 0.86, rispetto a una media di 0.61 per AI_7, al secondo posto. La media globale, calcolata come media in ogni punto di tutti i valori delle 10 AI e, successivamente, estratta la media, risulta di 0.58: AI_0 supera quindi la media di performance di ben 0.28, cioè quasi il 32.5%. Per quanto riguarda altre metriche, si analizza un guadagno totale delle operazioni vincenti di 1672 per AI_0 w una perdita totale delle operazioni non vincenti di 11167, per un profitto totale medio di 0.15, rispetto alla media di guadagno di 3392, una perdita pari a 13568 e un profitto medio di 0.25 per la seconda posizione, mentre in media le AI ottengono 1368.5 di guadagno, 11368.5 come somma delle perdite e un profitto di 0.117.

Curiosamente si osserva che la AI in seconda posizione ha un profitto ed un guadagno molto maggiori di AI_0, ma non è stata scelta da *meta-learner*. Questo perché la maggior parte del suo alto guadagno deriva da un singolo picco altissimo di performance, che alza notevolmente tutti i valori (anche la media di tutte le altre AI), ma non è costante e non viene quasi mai più registrato. Di conseguenza, è difficile che l’agente cambi scelta in così poco tempo, seppur esplorando un nuovo bandit. Anche se l’algoritmo avesse scelto correttamente AI_7 durante ogni step all’interno del suo picco, *prima* e *dopo* quel periodo questa AI non mostrava comunque di essere particolarmente meglio delle altre. Il breve picco di AI_7 non è quindi stato sufficiente per permettere all’algoritmo di abbandonare la sua precedente scelta e premiare il nuovo bandit fino alla fine, modificando la sua funzione di scelta in favore di AI_7.

È molto difficile essere in grado di prevedere un picco del genere, soprattutto quando il competitore non mostra quasi mai un comportamento simile in tutto il resto dei dati. La difficoltà di riuscire a prevedere tali valori è paragonabile allo sforzo di

previsione dell'andamento dei mercati azionari, cosa che uno strumento semplice come questo supervisore non è in grado di fare.

Si può quindi affermare che l'introduzione di un algoritmo di questo tipo all'interno di un sistema come Sentyment sia in grado di migliorare le performance del 33% effettuando una scelta intelligente rispetto ad eseguire una scelta casuale, che si può dire produrre un valore atteso di performance pari a 0.58 rispetto a 0.86. Rispetto alla scelta peggiore, invece, le prestazioni migliorando di molto, cioè 47% considerando la performance media della peggiore delle AI, AI_2, che risulta di 0.45.

4.4.1 Considerazioni finali su test comparativi

Innanzitutto un commento sui numeri. I risultati ottenuti da tutte le AI potrebbero sembrare strani: i guadagni sono bassi e le perdite molto alte. Questo è dovuto al fatto che i trigger delle AI forniti rappresentano soltanto una parte del sistema di Sentyment. La reale totalità dei dati prodotti dalle AI non comprende soltanto questi trigger, analizzati in questo documento, ma anche altre informazioni aggiuntive che cambiano l'esito delle scelte.

Per esempio, non si è tenuto conto di altri segnali di livello più alto, che interrompono le normali scelte della AI quando sta andando in *leva*, cioè quando il profitto sta scendendo troppo sotto una certa soglia scelta, oppure dei suggerimenti che permettono, quando il profitto è particolarmente alto, di guadagnare molto più velocemente investendo più capitale.

Una strategia di investimento, infatti, contempla anche il fatto di vendere completamente tutti i titoli quando il mercato va male, e introdurre del budget aggiuntivo per guadagnare ancora di più, quando la situazione è favorevole.

Qua sono soltanto analizzati i "profitti", cioè il rapporto unitario fra vincita e perdita, considerando di investire soltanto un certo budget iniziale e senza mai introdurne di nuovo. I guadagni e le perdite totali qua riportati sono anch'essi ottenuti in questo modo, non sono una misura *reale* del guadagno che può produrre una strategia di Sentyment.

Per semplicità si è considerata una situazione in cui gli investimenti sono soltanto legati ad un budget iniziale e sono analizzati i rendimenti, escludendo fattori aggiuntivi da parte delle AI e trattandole tutte come un elenco di operazioni di acquisto e vendita. In questo modo è possibile testarle tutte sulla stessa base, invece di introdurre ulteriori fattori che avrebbero cambiato le misurazioni, ma per tutte quante le AI, quindi, di fatto, non alterando i test.

Inoltre Sentyment è uno strumento di investimento a breve termine, analizza i brevi trend dei prezzi e va quindi sfruttato tempestivamente sul breve periodo, non

ha senso considerare risultati applicati al lungo termine, su tutto il dataset, ma soltanto su brevi intervalli.

Si è visto come l'algoritmo di reinforcement learning sceglie la migliore AI a seconda dei vari picchi di performance. La scelta potrebbe risultare poco chiara, ma l'obiettivo di questo strumento è sempre stato quello di minimizzare i rischi, massimizzando il guadagno. Sono premiate di più le strategie che offrono dei picchi di performance costanti seppur non eccessivamente accentuati, rispetto a pochi picchi di altissimo valore. È questo il motivo per cui AI_7 non compare nelle prime posizioni della scelta, mentre AI_0 risulta la migliore, seguita da AI_8, che pure mostra dei picchi costanti, ma di altezza minore del vincitore.

Rimane la questione *ogni quanto far girare l'algoritmo*. Purtroppo la scarsità di dati non permette molte esecuzioni lungo diversi istanti di tempo. Tutti i test sono stati eseguiti sull'istante più recente possibile quindi è difficile misurare qual è il tempo medio che trascorre da una scelta fino al sorpasso da parte di una nuova AI. È solo possibile fare qualche ipotesi osservando il grafico in Figura 23.

Si può affermare che AI_8 potrebbe sorpassare la migliore, AI_0, in un qualche momento nel tempo, mostrando sempre più picchi come quelli a inizio dataset prodotti dalla 0. Avendo altri dati a disposizione e ipotizzando, appunto, un'ascesa di AI_8, probabilmente dopo qualche mese *meta-learner* potrebbe modificare la sua scelta optando per AI_8, che era già al secondo posto in molti test. Sarebbe sicuramente inutile ripetere l'esecuzione dell'algoritmo ogni settimana o ogni mese, in questo dataset e considerando queste specifiche AI, perché non si osserva un così repentino cambio di tendenza della performance da parte dei competitori.

Osservando le performance delle AI relative ad altre monete, probabilmente la situazione potrebbe essere diversa e potrebbe essere necessario un cambio di AI più frequente, ma dipende da caso a caso e ognuno va analizzato con attenzione, con le dovute considerazioni e scegliendo opportuni parametri.

Infine, durante lo sviluppo di *meta-learner*, è emerso come si potrebbe ulteriormente migliorare lo strumento decisionale.

Se invece di considerare soltanto una AI migliore e mantenere quella per un lungo periodo (mesi), si scegliesse una differente AI per ogni picco di performance, la performance finale potrebbe ancora aumentare. I picchi sono circa di lunghezza sempre uguale e separati da periodi regolari (circa 25-30 giorni di picco, 2 settimane in salita e 2 in discesa), quindi è facile capire se ci si trova all'interno della salita o della discesa. Durante la salita, se fosse possibile scegliere una nuova AI ogni volta, discriminando secondo quella che ha una maggior *pendenza* rispetto alle altre, si potrebbe essere in grado di scegliere la migliore abbastanza velocemente, prima che

raggiunga la vetta del picco, e in questo modo guadagnare molto di più rispetto a scegliere sempre quella che produce picchi moderati ma costanti.

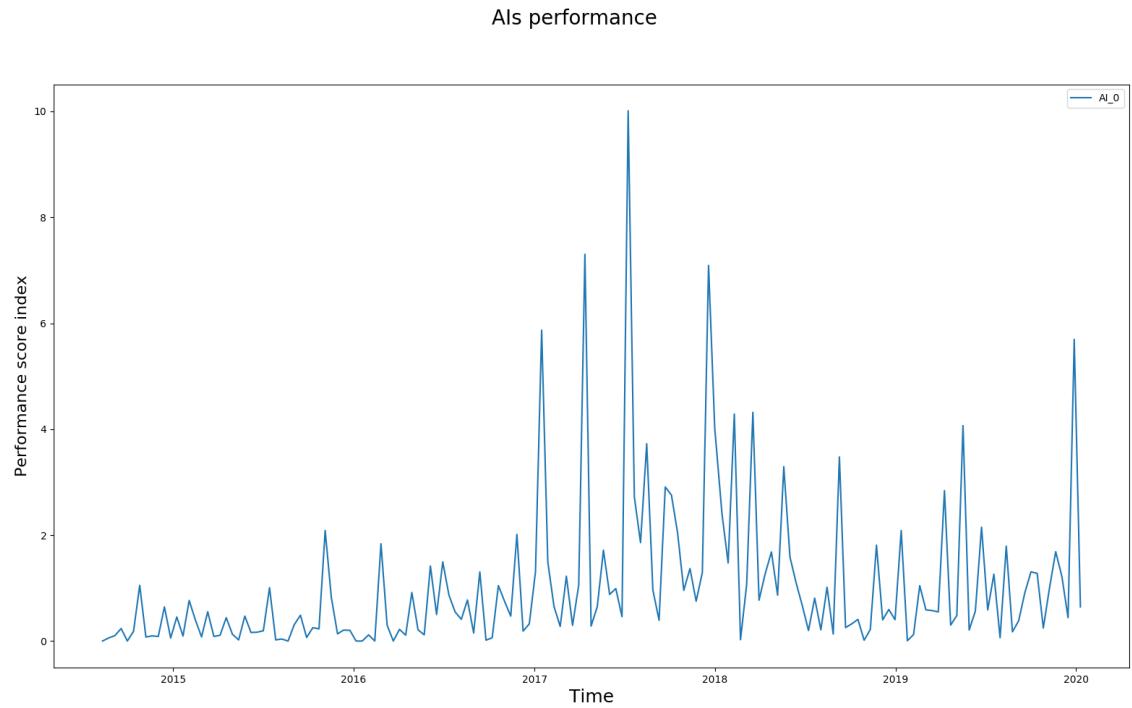


Figura 27:

Grafico ideale di una scelta perfetta. Se in ogni punto del grafico delle performance (Figura 23) si fosse sempre scelta la AI con performance migliore, alla fine, si avrebbe una performance finale che è sempre il massimo fra le performance delle varie AI nel tempo. Sono stati scelti alcuni punti nel grafico precedente in cui scegliere la migliore AI e scambiare con la precedente migliore. Il risultato approssima la funzione di massimo, ma resta la questione di come scegliere una AI quando sta avendo un picco migliore delle altre.

Per identificare la AI migliore all'interno di un picco, si potrebbe ricorrere a tecniche che studiano la derivata o il gradiente, oppure utilizzando a un ulteriore livello di apprendimento. Una Neural Network il cui scopo è prevedere i picchi da parte delle AI.

Risulta comunque un altro problema complesso, di entità simile alla previsione del mercato azionario, visto che si tratta di prevedere quando una AI avrà una alta performance, ed al di fuori dell'ambito di questo documento.

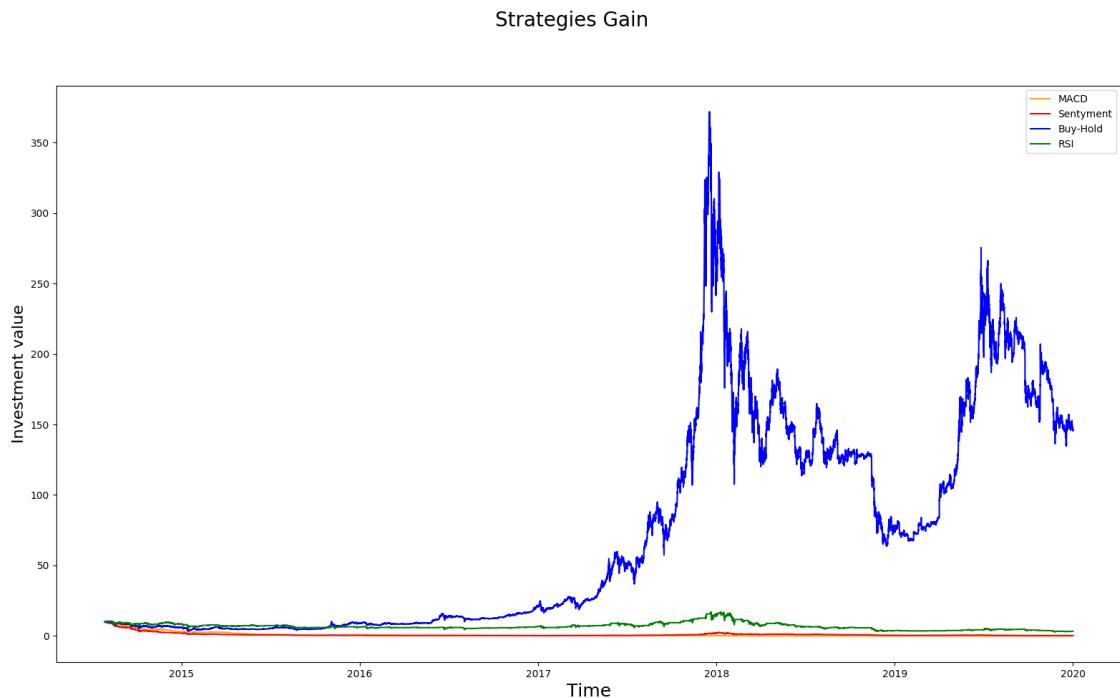
4.5 Confronto con altre strategie

Sentyment è ora trattata come un'unica AI, non più distinta in molti componenti differenti. Si suppone composta soltanto dalla AI che si è scelta come migliore nei passi precedenti e confrontata con altri strumenti simili attraverso l'analisi di dati e grafici.

Confrontare diverse strategie è un compito non banale, vanno calcolati per ognuna molti indicatori e statistiche e si confrontano quelli che si ritengono più importanti a seconda della situazione e del tipo di dati. Difficilmente una strategia risulta in assoluto migliore di tutte le altre, da tutti i punti di vista; si tratta più di imporre dei vincoli e operare delle scelte, sacrificando alcuni aspetti in favore di altri. In questo ambito si affronta un'analisi qualitativa e saranno esposti alcuni punti chiave che sono emersi studiando i grafici di profitto di varie strategie, ma non si intende in alcun modo comparare numericamente i risultati, in quanto strumenti più sofisticati e adatti allo scopo esistono già e un'analisi finanziaria dettagliata è al di fuori dell'ambito di questo documento.

Il confronto risulta quindi difficile. I guadagni di ogni strategia sono fortemente correlati ai prezzi e quindi, in generale, se il mercato sta andando male difficilmente le strategie riescono ad avere un alto profitto. Inoltre, alcune strategie sono più adatte ad un certo tipo di tendenza dei prezzi e si rivelano inefficaci in qualsiasi altra situazione.

Dato che si parla di Bitcoin, si prende come esempio una strategia buy-hold calcolata per l'intero periodo disponibile nel dataset (2014-2020), quindi analizzando sul lungo termine. Il titolo Bitcoin ha visto il suo prezzo aumentare sempre più, in una salita crescente fino a circa il 2018. In questa situazione, comprare un titolo a inizio 2014 e mantenerlo fino al 2018, senza effettuare nessun'altra operazione, produce un rendimento altissimo e non c'è strategia che riesca a batte questo profitto. Un asset comprato in precedenza per una certa quantità ora vale almeno 100 volte tanto, senza eseguire alcuna operazione. Per questi motivi, sul lungo periodo e in una situazione di crescita costante, nonostante eventuali massimi e minimi locali, una strategia di tipo buy-hold risulta molto redditizia e adatta alla situazione.

**Figura 28:**

Rendimento di varie strategie applicate circa all'intero dataset Bitcoin-Euro. Si nota che buy-hold sovrasta tutte le altre, perché è una strategia adatta a questa situazione, in cui il valore di Bitcoin ha delle crescite di prezzo altissime. Le altre strategie elencate sono per le analisi a breve termine.

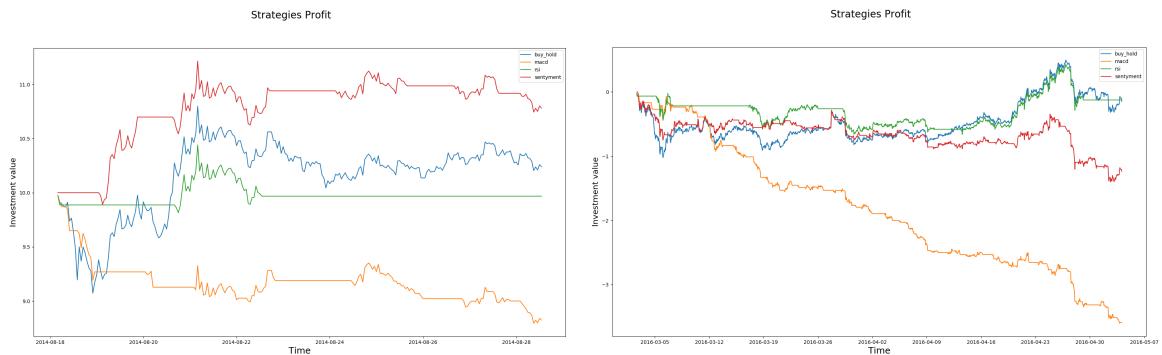
Il problema che sorge poi, però, è che una volta che i prezzi smettono di salire e iniziano a scendere, buy-hold segue il mercato e crolla di conseguenza. Qua questa strategia si rivela inefficace, non bisogna seguire il mercato se è in discesa, ma cambiare strategia e sfruttare comunque i massimi e minimi locali che si presentano lungo la discesa. Si avranno probabilmente dei profitti molto bassi rispetto a quando il mercato era in crescita, ma si evita di perdere tutto il capitale accumulato durante la salita. Se invece di vendere nel 2018 e guadagnare il massimo dalla salita, si aspetta che i prezzi calino, i titoli che si possiedono perdono di valore e il profitto scende.

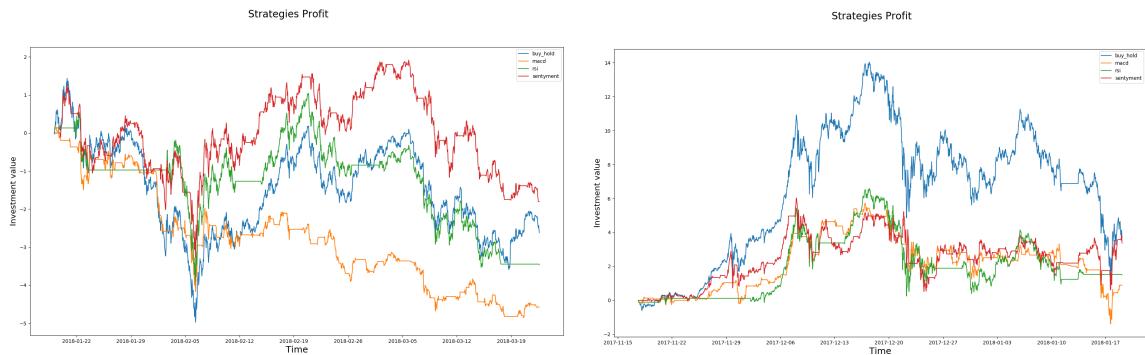
Quindi, una strategia buy-hold è utile per effettuare un confronto rispetto a una situazione di crescita dei prezzi, dove un "compra e detieni" risulta vincente. Per le altre situazioni, dove i prezzi sono oscillanti e non si è in grado di anticipare se ci si trova in un periodo di netta crescita o decrescita, le strategie migliori sono quelle che analizzano il breve periodo. Nel breve periodo è possibile sfruttare le oscillazioni dei prezzi per guadagnare. Ora il grafico in Figura 28 risulta più chiaro: buy-hold

è assolutamente imbattibile per quanto riguarda Bitcoin nel lungo termine. Un trader che ha avuto la possibilità di acquistare dei titoli Bitcoin nel 2014 ha visto il loro valore aumentare e ha guadagnato il massimo. Sentyment e le altre strategie sono incentrate sul breve termine e non si sono preoccupate quindi di prevedere questa crescita. Se si osserva con attenzione il grafico, però, si vede che a inizio 2019 buy-hold rischia di scendere di prezzo e ritornare ai suoi valori prima del picco: quindi se non si presta attenzione, anche buy-hold può fallire, se non si cambia strategia nei periodi peggiori.

Per questi motivi è necessario quindi analizzare periodi più corti. Sentyment dà il massimo in brevi periodi da 100-200 candele (su base settimanale). In questa dimensione è possibile fare trade attivo e continuare a effettuare scambi che traggono vantaggio dai brevi picchi, oltre a essere in grado di evitare le ripide discese vendendo in tempo e non dover aspettare le lunghe tendenze di crescita per detenere per lunghi periodi. In periodi così corti allora si nota che buy-hold non è più una scelta così efficace. Nei brevi picchi di prezzo, buy-hold detiene anche in tendenze di discesa, mentre Sentyment vende prima delle discese e mantiene il capitale.

Le altre strategie usate per il confronto sono MACD (Moving Average Convergence Divergence, trattata nel Capitolo 1) e RSI (Relative Strength index). Anche RSI, che nel lunghissimo periodo sembrava ben superiore a Sentyment, in realtà in certe situazioni si vede in netta difficoltà.



**Figura 30:**

Analisi di un breve periodo (dieci giorni) e un alcuni medi periodi di qualche mese di lunghezza. Si notano scenari differenti, in cui nel primo Sentyment è comparabile ad altre strategie. Alcune strategie falliscono in certi tipi di situazioni, come MACD che vede il suo rendimento crollare nella seconda immagine. Spesso quando si parte con un'impennata nei prezzi, buy-hold risulta la migliore. Quando invece si parte con un calo o si verifica una decrescita nel periodo, Sentyment reagisce meglio di altre.

Dai grafici si osservano situazioni in cui Sentyment risulta la migliore strategia rispetto alle altre considerate, mentre in altre situazioni ha circa gli stessi risultati. Altre volte Sentyment non riesce a guadagnare tanto come i concorrenti, soprattutto quando ci sono le forti crescite di prezzo, dove buy-hold domina. In compenso, però, quando successivamente il prezzo decresce con la stessa velocità con cui è salito, altre strategie perdono mentre Sentyment è quasi sempre in grado di anticipare e vendere in tempo. Ai fini di un'analisi completa vanno analizzati sia i picchi che le valli, sia situazioni favorevoli che sfavorevoli a tutte le strategie o soltanto ad alcune, per mostrare i punti di forza e le debolezze di ciascuna.

Ripetendo il concetto, in ogni caso se si considera un lungo periodo in cui c'è soltanto una crescita dei prezzi, anche con una serie di massimi e minimi locali e non una retta perfetta ma comunque con un trend sempre in crescita, allora difficilmente una strategia può battere buy-hold. È difficile però capire se si sta attraversando un periodo di questo tipo o se il crollo è dietro all'angolo (cosa che, per buy hold, comporterebbe una grave perdita) e quindi è per questo che buy-hold è usata più come una base di confronto, perché non è una strategia in cui si può operare attivamente e, dato che funziona soltanto in certe situazioni, come anche altre strategie, adatte solo a certi pattern, è difficile prevedere se la situazione adatta si sta presentando o no.

Sentyment invece attua un mix di strategie adatte a varie situazioni: si notano periodi in cui compra e detiene per molto tempo, quando riesce ad identificare correttamente un lungo periodo di crescita e ricalca un buy hold, mentre ha periodi in cui riesce a vendere prima di un calo senza ricomprare per un tempo considerevole.

Righe perfettamente orizzontali significano che c'è stato un *sell* e quindi l'unico profitto è quello del budget ricavato dalla vendita che ora è "congelato" al suo valore. Naturalmente non è sempre facile identificare il trend e decidere la migliore strategia da adottare in ogni istante e bisogna anche rischiare. Per questo motivo ogni tanto non è possibile vendere tutto prima di un crollo, perché magari ci si aspetta una rapida ricrescita, viste situazioni simili in precedenza, o viceversa si esita a mantenere un investimento in crescita perché non ci si aspetta che il titolo continui a salire incessantemente per tanto tempo.

Si registra che in media Sentyment è più proficua nei periodi in cui i prezzi sono calanti, altalenanti o non presentano crescite repentine. Negli altri casi è forse meglio adottare altre strategie.

Nel periodo di fine 2017 in cui Bitcoin ha avuto una grande salita, Sentyment ha ovviamente fatto fatica a competere con buy-hold ma, dall'altra parte, appena i prezzi sono nuovamente scesi, il profitto di Sentyment ha continuato a rimanere quasi costante senza crollare, come invece hanno fatto molte altre strategie.

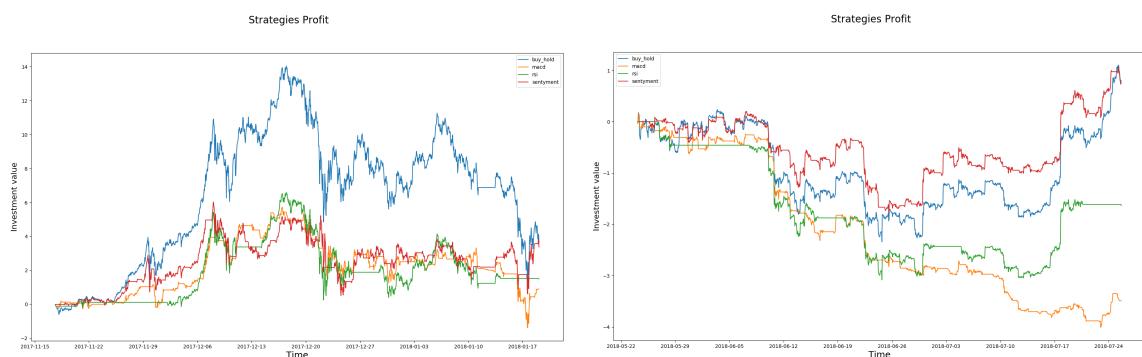


Figura 32:

Grafici di profitto di due particolari periodi per Bitcoin. Il primo, relativo al picco di prezzi di fine 2017 / inizio 2018, vede buy-hold in prima posizione, grazie al crescente valore del titolo. Sentyment vende spesso al posto di mantenere e perde parte del profitto (probabilmente non sta cercando di identificare il trend globale). Nella seconda immagine si vede che, però, una volta finito il picco ascendente, quando i prezzi sono calati e stanno tornando piatti o oscillanti, Sentyment ritorna in prima posizione.

Il punto chiave per Sentyment rimane cercare di ottenere guadagni continui e costanti senza troppe perdite, soprattutto senza perdite improvvise e considerevoli. Si rischia meno e, di conseguenza, si perde poco e si sfruttano con cautela le situazioni potenzialmente favorevoli.

Altre strategie, invece, potrebbero indicare quando il mercato è favorevole e spingere ad investire molto e mantenere per più tempo, soltanto per poi finire per perdere tutto quello guadagnato. Infatti sia buy-hold che RSI spingono in questa direzione

e, se non si presta attenzione, sono strategie che da sole risultano abbastanza estreme e indicano soltanto situazioni molto particolari in cui bisogna, secondo le loro statistiche, cimentarsi in vendita o in acquisto soltanto. Per esempio, RSI identifica i cambi di trend e indica quando un titolo sta venendo sottovalutato o sopravvalutato, il che potrebbe condurre ad una prossima inversione dell'andamento che, se sfruttata in anticipo, produrrebbe guadagno. L'indicatore però, preso da solo, mostra solamente la possibilità di un cambio di trend ma non ne indica l'esatto momento: il valore potrebbe rimanere per molto tempo su "vendi" ma bisogna aspettare l'effettivo verificarsi della situazione e non vendere subito.

Per questi motivi spesso le strategie si usano in combinazione fra di loro, per ottenere il massimo da più punti di vista e unire gli aspetti positivi. Alcuni esempi sono l'unione di RSI con MACD o Bollinger Band.

Per quanto riguarda un'analisi di tipo più quantitativo, è possibile calcolare, per ognuno dei periodi in cui è stato diviso il dataset, diverse statistiche per poterle confrontare con quelle prodotte dalle altre strategie, sempre all'interno dei periodi. È possibile analizzare sotto diversi aspetti e gli indicatori sono molti e ognuno esprime una qualità differente.

Al fine di questa analisi, che non ha la pretesa di essere un'analisi finanziaria completa sulle strategie di Sentyment ma soltanto una discussione sulle sue performance, sono trattate le metriche già esposte, quindi rendimento e profitto. Il rendimento, qua chiamato anche "guadagno", esprime l'andamento della strategia rispetto ai prezzi mentre il profitto pesa anche il fattore perdite ed esprime quindi anche un fattore di rischio. Non è esattamente la Formula 7 usata nel *meta-learner*, perchè pesava le perdite più delle vincite; qua invece sono usati entrambi i pesi a 1. Purtroppo, però, non è possibile calcolare la performance per buy-hold o per periodi troppo brevi (problema già affrontato in questo capitolo), in quanto basa le sue statistiche sull'analisi delle singole operazioni, vincenti e non, mentre buy hold non esegue mai alcuna operazione, ad eccezione di quella iniziale. Buy hold sarà quindi esclusa da questo paragone.

A seconda della lunghezza del periodo considerato si hanno risultati differenti: in periodi più corti Sentyment lavora meglio e buy hold perde di potenza, se il periodo considerato non ha una forte crescita senza perdite. In assoluto i risultati migliori si hanno per periodi molto brevi, per esempio di 100 candele (circa mezza settimana), dove Sentyment raggiunge risultati comparabili a altre strategie come RSI, che sembra spesso andare bene. Sono stati considerati anche periodi di 250 e 720 candele, ma l'analisi e lo scopo del software Sentyment mostrano che è opportuno considerare soltanto 100 o, al massimo, 250 candele.

Sono tracciati i grafici di rendimento e performance per i periodi di 100 e 250 candele

e sono analizzate le medie calcolate per ogni periodo.

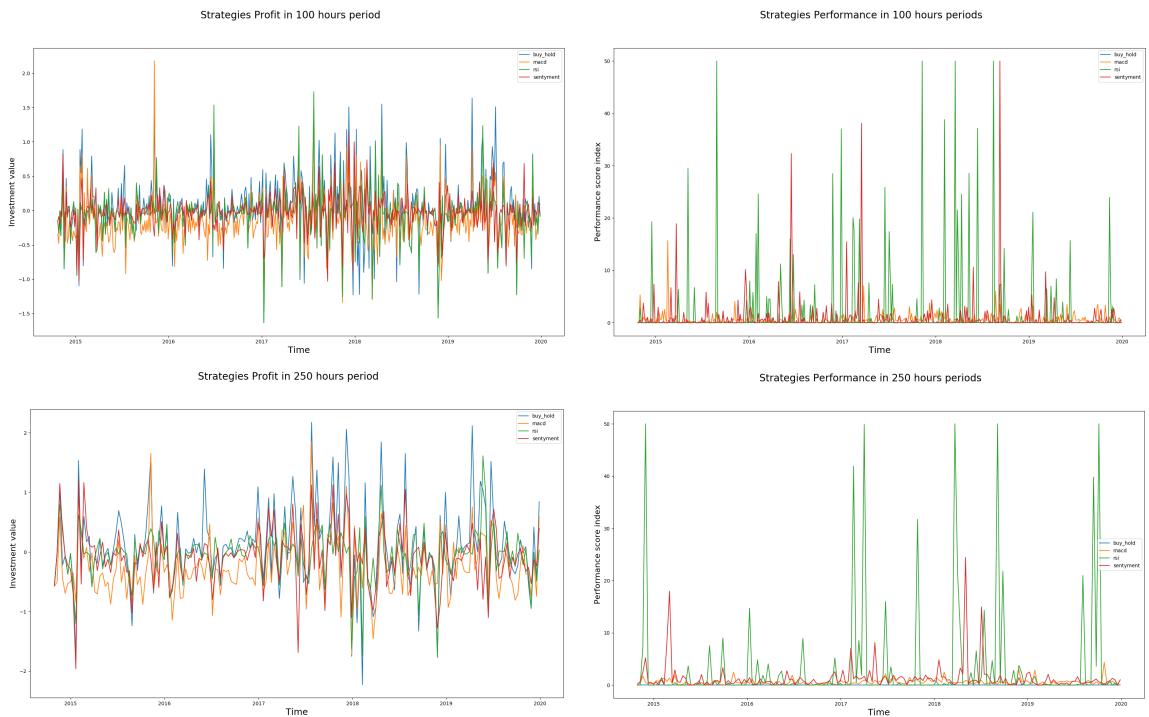


Figura 34:

Analisi di rendimento e performance in brevi periodi da 100 e 250 candele. I grafici sono simili a quelli usati da *meta-learner* in cui si analizzano le performance, ma qui le perdite sono pesate allo stesso modo delle vincite, risultando in performance medie più alte. I profitti di Sentyment sono inversamente proporzionali al periodo di analisi. Sentyment compete con RSI in 100 / 250 candele. La performance di buy-hold non può essere calcolata in quanto necessiterebbero molte operazioni di buy / sell.

Dalle immagini si può evincere quanto già affermato. Sentyment in generale è abbastanza costante ed è spesso nelle prime posizioni, subisce moderate perdite anche quando le altre strategie ne risentono, ma non approfitta estensivamente dei picchi in salita di prezzo come, invece, fanno le altre (a parte macd). Queste altre, però, mostrano anche consistenti perdite per ogni vincita, soprattutto buy-hold. RSI ha poche perdite, seppur di notevole quantità, accompagnate da vincite più frequenti e di grande intensità: per questo è premiata nel grafico delle performance (profitto), dove però anche Sentyment risulta essere decisamente buona.

La scelta della lunghezza di periodo incide sulle performance: più si aumenta il periodo preso in considerazione e più le performance di Sentyment calano. Potrebbe essere a causa del fatto che Sentyment è appunto uno strumento di analisi di brevi

periodi e non interpreta i pattern dei prezzi a lungo termine, lo stesso motivo per cui fa fatica nel confronto con buy-hold a lungo periodo (in una situazione di crescita dei prezzi).

100h period			250h period		
strategy	profit	performance	strategy	profit	performance
buy-hold	0.0297	/	buy-hold	0.091	/
macd	-0.139	0.619	macd	-0.244	0.585
rsi	-0.034	2.013	rsi	-0.049	3.302
sentyment	-0.021	0.969	sentyment	-0.077	1.295

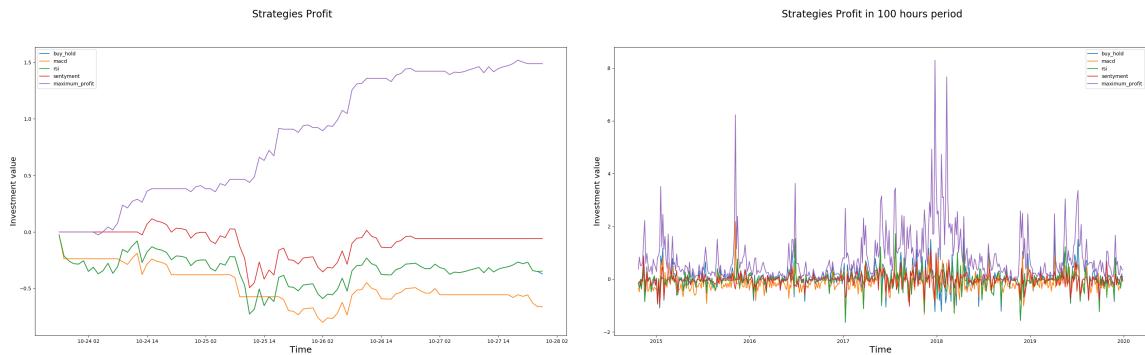
Tabella 6:

Rendimenti e performance medie delle strategie considerate per il confronto, relative a periodi di 100 e 250 candele.

4.5.1 Maximum profit

A scopo comparativo è possibile costruire a posteriori un modello ottimo di scelte, prendendo in considerazione un breve periodo di tempo e avendo a disposizione tutto il dataset di prezzi. Si calcola quali sarebbero state le scelte da compiere per produrre il *massimo guadagno assoluto*: esiste una precisa combinazione di segnali buy / sell per ogni intervallo di prezzi che, se attuata con un certo budget iniziale e delle fee, produce un rendimento massimo rispetto a qualsiasi altra possibile combinazione di segnali a partire dagli stessi dati.

Questa tecnica non anticipa i prezzi né si può dire una vera e propria strategia, perchè non esprime delle regole secondo cui comprare o vendere a seconda dell'andamento dei prezzi, ma è soltanto ricavabile da prezzi passati e rappresenta un'ulteriore base di confronto per altre strategie. Attraverso un modello così costruito è possibile comparare ulteriormente Sentyment o qualsiasi strategia per capire quale sarebbe il modello di riferimento ottimale di scelte da attuare a seconda delle varie situazioni, oppure capire quale sarebbe la lunghezza o il periodo migliore da scegliere per ottenere più guadagno, o indagare sul valore di altri possibili parametri o statistiche, o, ancora, fornire una stima delle performance di una AI o di altre strategie in funzione di quella ottenuta dal modello ottimo.

**Figura 37:**

Immagini che mostrano l'efficacia dell'algoritmo maximum profit. È mostrata prima la sezione di un breve periodo di qualche giorno e poi l'intero dataset. Per ciascuno dei due grafici è mostrato l'andamento del rendimento (nel secondo è considerata la media dei periodi da 100 candele) ed entrambi sono analizzati con le stesse strategie utilizzate per i confronti precedenti. Si può notare come la "strategia" prodotta da maximum profit ha rendimenti nettamente superiori rispetto a qualsiasi altra, ma è un algoritmo teorico e non realizzabile nella pratica perché necessita di conoscere in anticipo i prezzi.

Implementazione

L'algoritmo non è ottimo, esegue un'esplorazione ordinata di tutte le possibili combinazioni di operazioni e mantiene la sequenza che produce il guadagno maggiore, prendendo sempre le differenze massime fra vendita alta e acquisto basso. Ha infatti una complessità teorica di $O(kn^2)$ nel numero degli elementi (n , la lista di prezzi) e nel numero massimo di operazioni di compravendita (k , che sarà al massimo $n/2$ dato che le operazioni sono divise in compra + vendi). Esiste però una versione ottimizzata che fa uso di tecniche di programmazione lineare, riutilizzando alcuni casi già calcolati, che aiuta a ridurre la complessità a $O(kn)$. Per semplicità è soltanto riportato un esempio della versione non ottimizzata e senza fee.

Nel trading azionario, un acquirente acquista azioni e vende in una data futura. Dato il prezzo delle azioni di n giorni, al trader è consentito effettuare al massimo k transazioni, dove una nuova transazione può iniziare solo dopo il completamento della transazione precedente. Si vuole scoprire il massimo profitto che un trader avrebbe potuto realizzare.

È facile comprendere la soluzione al problema così formalizzato attraverso un semplice esempio. Si supponga di avere il seguente elenco di prezzi:

$$[10, 22, 5, 75, 65, 80]$$

Allora, il guadagno massimo è ottenuto comprando a 10 e vendendo a 22, comprando a 5 e vendendo a 80, per un totale di 87. La strategia prodotta è quindi:

$$[1, -1, 1, 0, 0, -1]$$

87 rappresenta il guadagno finale, mentre i valori intermedi ($12 = 22-10$ e $75 = 80-5$) sono i guadagni delle singole operazioni, che aiutano a tracciare la funzione del rendimento.

La soluzione non è unica: in alcune situazioni è possibile che due diverse combinazioni di segnali producano lo stesso guadagno massimo. Per esempio se ci sono due prezzi ripetuti, ([10, 22, 5, 70, 70, 80]) comprare e vendere a ogni posizione, quindi con $k=3$, avrebbe prodotto lo stesso risultato.

Inoltre, l'esecuzione del massimo numero di buy e sell possibile ($n/2$) non garantisce sempre la massimizzazione del risultato, infatti ci sono casi in cui si arriva alla soluzione eseguendo un numero inferiore di operazioni, come nell'esempio in cui sono eseguite soltanto 2 operazioni buy e 2 sell, rispetto ad un massimo possibile di 3. È necessario quindi calcolare quale sarebbero i vari esiti anche al variare di k . Bisogna decidere se, una volta comprato il titolo ad un basso prezzo, conviene vendere ad un alto prezzo oppure aspettare che il prezzo si alzi ulteriormente, quindi eseguendo una operazione in meno ma con la prospettiva di guadagnare di più rispetto ad aver venduto e ricomprato.

Applicando l'algoritmo che produce la strategia e tracciando il grafico del rendimento in ogni punto (Formula 3) come si è già fatto più volte all'interno di questo documento, si ottiene una funzione monotona non decrescente. Non è assicurato che in ogni punto si abbia sempre il massimo rendimento possibile, poichè non si tratta di una tecnica *greedy*, infatti spesso si aspetta a vendere in vista di un'ulteriore crescita dei prezzi. È garantito che si ottenga il massimo assoluto soltanto *nell'istante finale* dell'elenco dei prezzi. La funzione, però, non decresce: le operazioni sono eseguite solo se apportano al rendimento totale un valore maggiore di quello precedente, cioè uno scambio con guadagno positivo.

Questa semplice versione dell'algoritmo non è applicabile al mercato azionario perchè non contempla le fee. In presenza di tasse la situazione cambia: un maggior numero di operazioni comporta un pagamento di più soldi in tasse e quindi, quando ci si trova nella situazione precedentemente descritta, in cui si valuta se vendere a un alto prezzo o aspettare che salga ulteriormente, spesso la risposta è aspettare, perchè vendere e ricomprare è meno redditizio che aspettare che il prezzo salga ancora.

Inoltre, come si nota dalla prima immagine di Figura 37, in presenza di fee la funzione non è più monotona non decrescente, ma presenta delle perdite. Questo è dovuto al pagamento della tassa quando si esegue un'operazione. Quando si acquista o vende, il valore dei titoli posseduti (unito al budget) decresce per un momento di una minima percentuale dovuta alla tassa.

L'algoritmo di maximum profit va quindi adattato alla fee, cambia la formula in cui si calcola il rendimento delle singole operazioni, che non sono più differenze di prezzi ma acquisti di equity e vendite con ricavo di un budget, che poi va reinvestito (secondo quanto descritto nel primo capitolo). Dal punto di vista concettuale, però, continua a funzionare: esprime comunque la miglior combinazione possibile di operazioni da eseguire per avere massimo rendimento.

Approssimazioni

Siccome la complessità dell'algoritmo è alta o non lineare, è difficile applicarlo a periodi di tempo non piccoli. Anche per brevi intervalli (100) risulta comunque scomodo, se il calcolo deve essere ripetuto per analizzare tutto il dataset. Per questo è introdotta una funzione che approssima il risultato che, anche se non arriva esattamente agli stessi numeri dell'algoritmo maximum profit, è comunque ben oltre qualsiasi strategia e rappresenta un modello a cui rifarsi. È inoltre molto più performante rispetto all'algoritmo originale.

Un altro motivo per cui è irrilevante avere un guadagno perfetto è legato al fatto che combinazioni completamente diverse di segnali potrebbero comunque produrre un alto guadagno, molto superiore a quello prodotto da qualsiasi altra strategia e magari anche comparabile con il massimo assoluto. Risulterebbe quindi inutile concentrarsi sulle singole scelte prese dall'algoritmo, perché raggiungere la precisione delle scelte è irrilevante per strumenti intelligenti che approssimano soltanto le soluzioni. Sarebbe più produttivo, invece, porre l'attenzione sul *valore* o sulla funzione di guadagno prodotta dalla sequenza di scelte ottime. E a questo scopo, allora, è più utile avere uno strumento performante che produca risultati per diversi test velocemente, seppur con buone approssimazioni di maximum profit, piuttosto che disporre dell'algoritmo perfetto, che risulta purtroppo molto lento. Il secondo grafico in figura 37 è infatti stato prodotto con l'algoritmo approssimato e non con quello originale, perché avrebbe richiesto troppo tempo e ad ogni modo si nota che non in un singolo punto altre strategie sorpassano in performance la funzione di massimo.

Anche questa funzione fa uso di algoritmi intelligenti. Si è sviluppata una semplicissima AI genetica che tenta di massimizzare la funzione di guadagno

(Formula 3).

Da questo punto di vista, la formalizzazione del problema del massimo guadagno cambia leggermente. Ora si considera non più una sequenza di trigger buy / sell, la cui relativa differenza di prezzo produce un certo guadagno, ma una funzione a n variabili dove n è la lunghezza del dataset. Ogni variabile, che era un punto nel dataset in cui fissare l'operazione compra o vendi, può assumere un valore discreto compreso fra [-1, 0, 1]. La combinazione dei diversi valori dei coefficienti delle variabili produce un risultato di guadagno differente dalle altre e bisogna scoprire attraverso un'esplorazione "casuale" dei valori delle incognite qual è la combinazione che massimizza la funzione di guadagno.

Si parte da una popolazione iniziale formata da un certo numero di soluzioni casuali al problema, cioè alcune sequenze di [-1, 0, 1] (vendi / mantieni / compra, una per ogni candela nel dataset). A ogni generazione è calcolata la funzione di *fitness* per ogni soluzione, ovvero la funzione di guadagno (Formula 3), e sono mantenute solo le prime soluzioni che producono un valore maggiore delle altre. Poi, attraverso *mutazioni* e *crossover* degli individui della popolazione, si cerca di cambiare la soluzione introducendo una coppia di compra / vendi in una posizione casuale o spostandone alcune già esistenti. Nelle varie generazioni si continua ad introdurre il fattore della mutazione casuale, incrociando gli individui più promettenti e mantenendo sempre i migliori.

Ovviamente bisogna assicurarsi che le soluzioni create rispettino alcuni criteri, per esempio un *sell* deve occorrere solo dopo un *buy* senza altri *sell* in mezzo.

Dopo molte iterazioni il guadagno inizia a salire, in seguito all'esplorazione casuale e, in poco tempo, ci si ritrova in una situazione di *convergenza* in cui gli individui non mutano più e le popolazioni rimangono identiche nel corso delle generazioni. A questo punto si è probabilmente raggiunto un massimo (non vi è modo di capire se locale o globale per questo semplice strumento) e si accetta la soluzione migliore come sequenza di trigger che approssimano la strategia di massimo profitto.

Sarebbe possibile esplorare ulteriormente le soluzioni introducendo altre sequenze casuali una volta raggiunta la convergenza, per smuoversi dal massimo locale, ma i test rivelano che, per brevi periodi, questa tecnica approssima già abbastanza bene *maximum profit*.

Inoltre il suo punto di forza è rappresentato dalla velocità: fissato un certo numero di *generazioni*, che non dipende dalla lunghezza del periodo considerato (cioè il numero di variabili), l'algoritmo è certo di trovare una soluzione. Il numero di generazioni può essere abbassato per brevi periodi, ma non si va mai oltre le 3000-5000 anche per il calcolo su tutto il dataset (50000 candele circa). La soluzione non è garantita essere ottima, ma alcuni test ne confermano la buona approssimazione.

La complessità di questo algoritmo risulta quindi minore di quella di *maximum*

profit secondo la sua precedente definizione, perchè dipende soltanto linearmente dal numero di candele. All'interno di ogni generazione, le quali sono sempre in numero costante, si scorre soltanto una volta ogni elemento della soluzione per calcolare la funzione di fitness e si ripete per ogni soluzione della popolazione, anch'essa un numero costante. Si può affermare che la complessità sia quindi sicuramente più bassa di quella precedentemente trattata. Considerando i generazioni in cui si ripete j volte il calcolo della funzione di guadagno (j numero di individui per popolazione), la quale deve scorrere tutte le n variabili, l'unico costo che rimane non costante è appunto quello dato dal calcolo della funzione di fitness. Le mutazioni e i crossover sono eseguiti in tempo costante tramite accesso diretto agli elementi. Rimane soltanto mantenere, per ogni generazione, le prime x soluzioni migliori a seconda del loro risultato della la funzione di fitness ma, con strutture dati apposite, si può ridurre ad una complessità in funzione di x , dato che si itera già sulle soluzioni allo step precedente e quindi è sufficiente mantenere un riferimento ai primi x migliori.

Risultati

Tutto ciò rappresenta un valore per Sentyment. Sarebbe possibile allenarsi sulle scelte migliori da fare, cercando di approssimare la funzione del massimo guadagno. Ora risulta più chiaro come ci si dovrebbe comportare in certe situazioni, anche se le scelte di maximum profit sono tutte prese conoscendo il futuro quindi non sempre possibili da seguire. In effetti, questo algoritmo opera delle scelte probabilmente irrazionali perchè eseguite soltanto in vista di determinati prezzi futuri, che ancora non sono disponibili e non è detto che si ripresentino. Un'analisi più approfondita potrebbe rivelare che alcune scelte possono invece risultare abbastanza razionali e operabili anche da un umano o da un algoritmo intelligente, se allenati correttamente su questo modello.

Per il momento, è utile usare l'algoritmo per stabilire un *upper bound* rispetto alle performance delle varie strategie e quindi un valore a cui fare riferimento ogni volta che sono confrontate. Per esempio, prodotti i valori di performance o di qualsiasi altra statistica per alcune strategie in un dataset di prezzi, potrebbe essere utile calcolare la stessa metrica anche per la strategia di maximum profit, per poi esprimere i risultati delle altre in funzione di quelli ottenuti del modello massimo.

In questo modo si può capire se la differenza in prestazioni di due strategie è minima o rappresenta un risultato più sostanziale. Se una strategia A ottiene 10 per la statistica X, una strategia B ottiene 15 e maximum profit ottiene 100000, B è migliore di A solo di poco, rispetto a maximum profit. Se, invece, il risultato massimo fosse stato 20, allora il vantaggio di B è molto più considerevole.

Per brevi periodi qualche tecnica potrebbe avvicinarsi alle prestazioni di maximum

profit, cosa che sicuramente non succede per periodi più lunghi. Inoltre potrebbe rappresentare anche una base di test per eventuali prossimi algoritmi da implementare, sapendo che nessuna strategia potrà mai superare i risultati ottenuti da maximum profit.

100h period		2014-2020 period	
strategy	profit	strategy	profit
buy-hold	0.0297	buy-hold	97.856
macd	-0.139	macd	-8.895
rsi	-0.034	rsi	-2.716
sentyment	-0.021	sentyment	-6.052
maximum profit	0.842	maximum profit	4395810751585198.5

250h period	
strategy	profit
buy-hold	0.091
macd	-0.244
rsi	-0.049
sentyment	-0.077
maximum profit	2.45

Tabella 7:

Rendimenti medi di alcune strategie e di maximum profit, relative a periodi di 100 e 250 candele e su tutto il periodo 2014-2020.

Come si evince dalle tabelle, a conferma di quanto già sostenuto in precedenza, Sentyment non compete per il lungo periodo. Anche buy-hold, che sembrava molto superiore a qualsiasi altra strategia sul lungo termine, ha un rendimento che non è assolutamente paragonabile a quello massimo (da notare che il valore è relativo all'algoritmo genetico quindi non è garantito essere esattamente il massimo). Questo perchè buy-hold parte bene e guadagna in salita, ma poi perde moltissimo in discesa, cosa che maximum profit non fa, continua soltanto a crescere. Buy-hold e maximum profit potrebbero competere quando la prima tocca il suo massimo (inizio 2018): ha comunque perso parte del guadagno nelle brevi discese prima di arrivare al picco più alto, ma ha un valore abbastanza alto, tuttavia di molti ordini di grandezza inferiore a quello di maximum profit, perchè per molti anni prima della salita Bitcoin ha avuto un valore altalenante, dove buy hold sale e scende mentre maximum profit continua a salire e in poco tempo

distanzia qualsiasi altra strategia. Per questo va usata soprattutto per intervalli corti.

Ora, se si analizza il breve periodo, dove Sentyment è senz'altro efficace, si scopre che in media le sue prestazioni non sono così lontane dal massimo quanto lo sono quelle calcolate in periodi sempre più lunghi. Questo fatto conferma le potenzialità dello strumento in brevi periodi: in generale su breve periodo si guadagna meno ma è più facile essere il più efficienti possibile, a differenza dei lunghi periodi in cui è molto difficile indovinare i trend e, anche se le potenzialità sono di gran lunga più ampie, è impensabile sviluppare uno strumento che si avvicini anche solo minimamente ai massimi risultati possibili.

Si è scoperto che il lungo periodo produce guadagni assurdi per quanto riguarda il profitto massimo. Ma anche questo è un fatto da analizzare con attenzione. Come già ripetuto, i guadagni sono fortemente correlati ai prezzi e quindi all'andamento del mercato. Il risultato ottenuto su tutto il periodo da parte di maximum profit è tale soltanto grazie al tipo di mercato a cui è applicato. Bitcoin, come già detto, ha avuto una forte crescita da quando è nato fino al 2018. Questi numeri non si sarebbero potuti ottenere se non fosse stato per questa impennata e quindi, applicando lo stesso algoritmo di massimo ad un altro tipo di mercato, i risultati sarebbero stati diversi. Sicuramente sempre molto più alti di qualsiasi altra strategia, ma diversi. Quindi anche comparare i risultati di altre strategie a maximum profit va fatto con attenzione e bisogna pesare il fattore mercato: è inutile affermare che una strategia è di pochissimo migliore di un'altra in relazione a maximum profit, quando il risultato ottenuto da quest'ultimo è relativo ad una situazione ideale mai raggiungibile nella pratica e dovuta in gran parte alla forte crescita dei prezzi.

Capitolo 5

Conclusioni e lavori futuri

Attraverso l'implementazione di nuovi strumenti si è analizzato il software Sentyment in diversi suoi aspetti, aggiungendo nuovi componenti e sviluppando dei metodi di test. Si è visto ad alto livello come Sentyment opera e come agiscono le sue AI, utilizzando i test creati per confrontarle fra di loro e paragonando il software a strumenti simili.

Lo sviluppo di *meta-learner* ha costretto ad alcune scelte ma ha prodotto dei risultati tangibili e ha aperto la strada per eventuali futuri miglioramenti per lo strumento. Il confronto con altre strategie ha evidenziato i punti di forza e le possibili mancanze, producendo anche modelli di riferimento sempre validi e per possibili test futuri.

Ogni aspetto trattato è accompagnato dallo sviluppo di strumenti intelligenti, piuttosto che da un approccio classico statistico.

Per quanto riguarda *meta-learner*, un approccio intelligente per un problema di scelta e massimizzazione offre diversi vantaggi. Non sono utilizzati degli indicatori come la media perché non interessa scegliere la funzione che ha un singolo picco di performance maggiore di tutti gli altri, ma una che sia costante e affidabile, quindi picchi costanti anche se non esagerati, e spesso maggiori degli altri.

Un approccio intelligente permette eventuali modifiche cambiando i parametri: formula della performance, pesi dei termini che la compongono; lunghezza step (quando ci saranno più dati si potrà allungare) e epsilon.

L'algoritmo di reinforcement learning è in grado di adattarsi ai nuovi dati e permette ad un nuovo competitore di affermarsi, se costantemente scelto come nuovo migliore. È adattabile anche a diverse situazioni: trattando diverse criptovalute, ognuna con il suo sottoinsieme di differenti AI che operano, Sentyment ha bisogno di un supervisore per ogni asset. I titoli sono differenti l'uno dall'altro e ciascuno ha bisogno di un'attenta analisi e test per la scelta dei migliori parametri che meglio si adattano alla situazione. L'agente è poi in grado di operare la migliore scelta

osservando l'ambiente e facendo esperimenti.

Per migliorare ulteriormente le performance del sistema, si potrebbe aggiungere un altro livello di learning sopra meta-learner, che apprende dal supervisore qual è l'intervallo di tempo migliore per eseguire di nuovo la scelta di una AI, o che riesce a riconoscere quando una (o più) AI stanno salendo un picco di performance e, quindi, scegliere temporaneamente quella che sta andando meglio all'interno di un piccolo step.

La situazione ideale sarebbe quella in cui ci si ricorda di chi è il migliore, ma ogni tanto lo si lascia se c'è un competitore che da poco sta andando meglio, scalando un picco in modo più veloce degli altri. Lo sviluppo di uno strumento in grado di prevedere i picchi di performance, però, è un compito non banale e richiederebbe maggiori sforzi.

Infine non resta che applicare lo strumento a molte più AI: lo stesso esperimento andrebbe ripetuto per ogni asset trattato da Sentyment, che comprende molte criptovalute. Ogni titolo ha il suo pool di AI che sono differenti da quelle di altre monete e si sono allenate sui quegli specifici dati, quindi, dovrebbe esistere un supervisore per moneta.

Il supervisore può quindi essere ampliato in molti modi, si possono aggiungere tanti livelli di *meta-learning* al di sopra delle AI di Sentyment, sopra *meta-learner* o affiancandoli a questo, in maniera simile a quanto visto in letteratura recentemente.

Purtroppo la carenza di strumenti simili a Sentyment e open-source non ha permesso lo sviluppo di test comparativi soddisfacenti rispetto al resto delle AI per trading presenti attualmente. Quasi tutti i software intelligenti di trading sono proprietari o molto restrittivi e non permettono una personalizzazione a livello fine, necessaria per poterli confrontare con Sentyment, che opera in modo particolare.

In futuro un passo fondamentale sarà sicuramente il confronto con altre AI per trading. Allo stato attuale una soluzione sarebbe stata sviluppare da zero una AI partendo dalle note librerie disponibili (per esempio, una Neural Network in Python) ed allenarla a predire il mercato o a implementare dei mix di strategie di investimento. Anche questo però sarebbe risultato un compito lungo e sicuramente non avrebbe prodotto dei risultati comparabili con uno strumento consolidato negli anni.

Invece, il confronto con altre strategie è comunque riuscito a far emergere dei punti chiave per la AI soggetto dei test, che si è dimostrata comparabile rispetto agli altri strumenti comunemente usati in ambito finanziario, fornendo eventuali spunti di miglioramento.

Infine, il modello ottimo rappresentato dall’algoritmo di *maximum profit* potrebbe essere utilizzato per testare anche le eventuali altre AI presenti sul mercato, per confrontarle con Sentyment. Potrebbe tornare utile anche all’interno dello sviluppo di una nuova AI o di successive iterazioni di Sentyment.

Bibliografia

- [1] Lawrence, R. (1997). Using neural networks to forecast stock market prices. University of Manitoba, 333, 2006-2013.
- [2] White, H. (1988). Economic prediction using neural networks: The case of IBM daily stock returns.
- [3] Phua, P. K. H., Ming, D., & Lin, W. (2000, July). Neural network with genetic algorithms for stocks prediction. In Fifth Conference of the Association of Asian-Pacific Operations Research Societies, 5th-7th July, Singapore. sn.
- [4] Kim, K. J., & Han, I. (2000). Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert systems with Applications*, 19(2), 125-132.
- [5] Garliauskas, A. (1999, October). Neural network chaos and computational algorithms of forecast in finance. In IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028) (Vol. 2, pp. 638-643). IEEE.
- [6] Schumann, M., & Lohrbach, T. (1993, January). Comparing artificial neural networks with statistical methods within the field of stock market prediction. In [1993] Proceedings of the Twenty-sixth Hawaii International Conference on System Sciences (Vol. 4, pp. 597-606). IEEE.
- [7] Yoon, Y., Swales Jr, G., & Margavio, T. M. (1993). A comparison of discriminant analysis versus artificial neural networks. *Journal of the Operational Research Society*, 44(1), 51-60.
- [8] Kim, J. W., Weistroffer, H. R., & Redmond, R. T. (1993). Expert systems for bond rating: a comparative analysis of statistical, rule-based and neural network systems. *Expert systems*, 10(3), 167-172.
- [9] Patuwo, E., Hu, M. Y., & Hung, M. S. (1993). Two-group classification using neural networks. *Decision Sciences*, 24(4), 825-845.

- [10] Subramanian, V., Hung, M. S., & Hu, M. Y. (1993). An experimental evaluation of neural networks for classification. *Computers & operations research*, 20(7), 769-782.
- [11] Chalup, S., & Maire, F. (1999, July). A study on hill climbing algorithms for neural network training. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406) (Vol. 3, pp. 2014-2021). IEEE.
- [12] Hong, T., & Han, I. (2002). Knowledge-based data mining of news information on the Internet using cognitive maps and neural networks. *Expert systems with applications*, 23(1), 1-8.
- [13] Fung, G. P. C., Yu, J. X., & Lam, W. (2002, May). News sensitive stock trend prediction. In Pacific-Asia Conference on Knowledge Discovery and Data Mining (pp. 481-493). Springer, Berlin, Heidelberg.
- [14] Kohara, K., Ishikawa, T., Fukuhara, Y., & Nakamura, Y. (1997). Stock price prediction using prior knowledge and neural networks. *Intelligent Systems in Accounting, Finance & Management*, 6(1), 11-22.
- [15] Fung, G. P. C., Yu, J. X., & Lam, W. (2003, March). Stock prediction: Integrating text mining approach using real-time news. In 2003 IEEE International Conference on Computational Intelligence for Financial Engineering, 2003. Proceedings. (pp. 395-402). IEEE.
- [16] Kuo, R. J., Chen, C. H., & Hwang, Y. C. (2001). An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. *Fuzzy sets and systems*, 118(1), 21-45.
- [17] Kuo, R. J., Lee, L. C., & Lee, C. F. (1996, October). Integration of artificial neural networks and fuzzy delphi for stock market forecasting. In 1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems (Cat. No. 96CH35929) (Vol. 2, pp. 1073-1078). IEEE.
- [18] Lowe, D., & Webb, A. R. (1991, February). Time series prediction by adaptive networks: A dynamical systems perspective. In IEE Proceedings F (Radar and Signal Processing) (Vol. 138, No. 1, pp. 17-24). IET Digital Library.
- [19] Maddala, G. S., & Lahiri, K. (1992). *Introduction to econometrics* (Vol. 2). New York: Macmillan.
- [20] Hellstrom, T., & Holmstrom, K. (1998). Predicting the stock market. Unpublished Thesis, Malardalen University, Department of Mathematics and Physics, Vasteras, Sweden.

- [21] Malkiel, B. G., & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2), 383-417.
- [22] Tsibouris, G., & Zeidenberg, M. (1995). Testing the efficient markets hypothesis with gradient descent algorithms. In *Neural networks in the capital markets* (Vol. 8, pp. 127-136). Chichester, UK: Wiley.
- [23] Van Eyden, R. J. (1996). *The Application of Neural Networks in the Forecasting of Share Prices* (Finance and Technology Publishing, Haymarket, VA).
- [24] Lawrence, R. (1997). Using neural networks to forecast stock market prices. University of Manitoba, 333, 2006-2013.
- [25] Van Eyden, R. J. (1996). *The Application of Neural Networks in the Forecasting of Share Prices* (Finance and Technology Publishing, Haymarket, VA).
- [26] Refenes, A. P., Zapranis, A. D., & Francis, G. (1995). Modeling stock returns in the framework of APT: a comparative study with regression models. In *Neural networks in the capital markets* (Vol. 7, pp. 101-126). John Wiley & Sons Chichester.
- [27] Steiner, M. (1995). Neural networks as an alternative stock market model.
- [28] Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(3), 131-156.
- [29] Hiemstra, Y. (1995). Modeling structured nonlinear knowledge to predict stock market returns. *Chaos & Nonlinear Dynamics in the Financial Markets: Theory, Evidence and Applications*, Irwin, Chicago, IL, 163-175.
- [30] Tsaih, R., Hsu, Y., & Lai, C. C. (1998). Forecasting S&P 500 stock index futures with a hybrid AI system. *Decision Support Systems*, 23(2), 161-174.
- [31] Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5), 988-999.
- [32] Yang, H., Chan, L., & King, I. (2002, August). Support vector machine regression for volatile stock market prediction. In *International Conference on Intelligent Data Engineering and Automated Learning* (pp. 391-396). Springer, Berlin, Heidelberg.
- [33] Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5), 988-999.

- [34] Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2), 307-319.
- [35] Tay, F. E., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *omega*, 29(4), 309-317.
- [36] Tay, F. E. H., & Cao, L. J. (2001). A comparative study of saliency analysis and genetic algorithm for feature selection in support vector machines. *Intelligent Data Analysis*, 5(3), 191-209.
- [37] Trippi, R. R., & Turban, E. (1992). Neural networks in finance and investing: Using artificial intelligence to improve real world performance. McGraw-Hill, Inc..
- [38] Kim, K. J. (2004). Toward global optimization of case-based reasoning systems for financial forecasting. *Applied intelligence*, 21(3), 239-249.
- [39] Ng, A., & Fu, A. W. C. (2003, April). Mining frequent episodes for relating financial events and stock trends. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (pp. 27-39). Springer, Berlin, Heidelberg.
- [40] Maheu, J. M., & McCurdy, T. H. (2004). News arrival, jump dynamics, and volatility components for individual stock returns. *The Journal of Finance*, 59(2), 755-793.
- [41] Fawcett, T., & Provost, F. J. (1996, August). Combining Data Mining and Machine Learning for Effective User Profiling. In *KDD* (pp. 8-13).
- [42] Hong, T., & Han, I. (2004). Integrated approach of cognitive maps and neural networks using qualitative information on the World Wide Web: the KBNMiner. *Expert Systems*, 21(5), 243-252.
- [43] Schmelzer, R. (2020, January 3). How Do You Test AI Systems?. *Forbes*. Last accessed 21th Feb 2020: <https://www.forbes.com/sites/cognitiveworld/2020/01/03/how-do-you-test-ai-systems>.
- [44] Wong, A. (2017, September 25). Solving the Multi-Armed Bandit Problem. Last accessed 21th Feb 2020: <https://towardsdatascience.com/solving-the-multi-armed-bandit-problem-b72de40db97c>
- [45] Sutton, R. S., & Barto, A. G. (1998). *Introduction to reinforcement learning* (Vol. 135). Cambridge: MIT press.

- [46] Guijarro-Berdiñas, B., & Alonso-Betanzos, A. (2002). Empirical evaluation of a hybrid intelligent monitoring system using different measures of effectiveness. *Artificial Intelligence in Medicine*, 24(1), 71-96.