POLITECNICO DI TORINO

Corso di Laurea Magistrale in
Ingegneria Informatica

Tesi di Laurea Magistrale

Tecniche di data mining per il trading intraday di criptovalute

Relatori
prof. Paolo Garza
prof. Luca Cagliero

Candidato
Nicolò Montesano

A.A. 2017-2018

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Negli ultimi anni si è assistito ad una vera e propria rivoluzione nel mondo digitale, le *criptovalute*.

La storia delle criptovalute è relativamente breve, in quanto le origini derivano soltanto dalla seconda metà degli anni 90. Già nel 1998 fu pubblicato da Wi Dai quello che si chiamava "B-Money": un sistema di denaro elettronico anonimo e distribuito. Successivamente, Nick Szabo creò "Bit Gold". Proprio come il Bitcoin ed altre Criptovalute che avrebbero seguito a ruota la tecnologia, il Bit Gold era un sistema di criptovaluta elettronica che richiedeva che gli utenti completassero lo schema proof of work. Si era tuttavia ancora agli albori delle criptovalute.

Nel 2008 fu creata la prima criptovaluta decentralizzata: il Bitcoin (ForexItalia24, 2017).
Il Bitcoin è un protocollo di comunicazione online che facilita l'uso di una valuta virtuale, compresi i pagamenti elettronici. Fin dalla sua istituzione nel 2009 da un gruppo di sviluppatori anonimi conosciuti come Satoshi Nakamoto, il Bitcoin ha eseguito oltre 300 milioni di transazioni e ha percepito una crescita eclatante. Tale sviluppo ha interessato e successivamente richiamato l'attenzione di numerose persone divenendo opportunità di ingenti guadagni.
Queste sono le condizioni che hanno reso celebre il Bitcoin e che hanno permesso l'evoluzione dell'intero sistema. Le criptovalute hanno ulteriormente acquisito una componente virale introducendo una sorta di "*caccia all'oro*" grazie alla loro diffusione nei social netowork e nei media di maggiore rilievo.

Sono molte le aziende americane che sfruttando tale viralità diedero origine ad

un fenomeno singolare. Molte di queste cambiando il proprio nome, apponendo in esso un termine legato al mondo delle criptovalute, percepirono una enorme crescita della loro quotazione in borsa.

Celebre è il caso della compagnia americana produttrice di the la quale trasformando il solo nome della compagnia da "The Long Island Iced Tea" a "Long Blockchain Corp" percepì una crescita istantanea delle proprie azioni pari al 200%.

Il lavoro svolto ha riguardato l'analisi dei dati storici di diverse criptovalute basate su tecniche di data mining di tipo supervisionato (classificazione e regressione) con l'obiettivo di generare in maniera automatica segnali di trading intraday.

Molti studi sono stati portati avanti in questo ambito sui mercati azionari e forex, mentre l'applicabilià di tali tecniche in un contesto differente come le criptovalute è ancora oggetto di studio.

A differenza del mercato azionario tradizionale tale mercato non è ancora regolamentato e non vi sono enti centrali o complicati sistemi finanziari, e il valore di ogni moneta è strettamente legata alla domanda e alla offerta. Tale sistema ha manifestato un elevato grado di volatilità e non stazionarità dei prezzi.

In particolare i modelli sviluppati saranno addestrati per la generazione di un segnale di compra-vendita con granularità oraria. I modelli computati si pongono l'obiettivo di predire la variazione del prezzo di chiusura ed in base alle delle soglie stabilite e priori generare tre tipologie di segnali:

- *Buy*: si apre una posizione long con la aspettativa che la moneta in questione percepisca una crescita
- *Hold*: la posizione attuale viene mantenuta poichè non si prevede una variazione importante del prezzo
- *Sell*: si apre una posizione short con la aspettativa che la moneta in questione percepisca un calo

In un secondo luogo, in base ai segnali generati, si sono studiate diverse strategie di trading con l'intento di massimizzare il guadagno. Le strategie prese in considerazione sono:

- *Keep Trend*: evita di aprire e chiudere la posizione allo scadere di ogni ora. La chiusura della posizione, quindi, avverrà solamente al verificarsi delle seguenti condizioni:

  - Predizione errata: il modello genera dei segnali non concordi al trend
  - Predizione di un cambio trend: il modello segnala un cambiamento di trend

- *Stop Loss*: propone una strategia finalizzata al contenimento delle perdite nel caso in cui l'andamento si discosti fortemente dalle aspettative. Questo implica la creazione di ordine ad un prezzo prefissato, il quale verrà eseguito in maniera automatica nel caso in cui il trend raggiunga il valore fissato.

- *Trailing Stop*: assume l'obiettivo di limitare al massimo le perdite cercando in caso di sbagliata predizione di assicurarsi comunque una situazione di non perdita includendo nel calcolo dello stop loss anche i costi di commissione

Infine si sono paragonati i guadagni ottenuti con le prestazioni di altri tre sistemi di riferimento:

- *Random*: vi è la generazione di segnali di compravendita del tutto casuali, priva di qualsiasi strumento per il supporto decisionale.

- *Lasthour*: : ripropone lo stesso segnale percepito l'ora precedente presupponendo che il trend in atto continui senza variazioni. Propone un numero di compravendite adeguate e quindi costi di commissioni di riferimento.

- *Buy&Hold*: simulazione volta a calcolare il guadagno percepito nel caso in cui si compri il primo giorno e si venda tutto l'ultimo giorno dell'intervallo temporale considerato. Questa è una delle sperimentazioni di riferimento perchè offre il minimo costo di spese di transizione ed il più basso tasso di rischio.

Le sperimentazioni effettuate hanno confermato come l'implementazione di strumenti di supporto, volti all'analisi tecnica dei dati, hanno generato un notevole miglioramento delle prestazioni di trading.

I risultati ottenuti dal lavoro di tesi mostrano un guadagno, rispetto all'investimento iniziale, pari al 1040,7% sul Bitcoin, al 1122,9% sull'Ethereum ed al 1677,4% sul Ripple.

Inoltre i risultati ottenuti hanno notevolmente superato le sperimentazioni di riferimento, confermando ulteriormente come il lavoro svolto in questa tesi possa supportare le decisioni intraprese da un trader.

Capitolo 2

Data mining

Negli ultimi decenni grazie all'evoluzione tecnologica è oggi possibile accedere facilmente ad una quantità enorme di dati. Questo fenomeno è stato supportato e reso possibile dalla presenza e dall'evoluzione di altri fattori fra cui si può identificare l'evoluzione delle capacità computazionali dei calcolatori e dai progressi ottenuti nei sistemi hardware, che permettono oggi di elaborare e salvare una quantità notevole di dati.
Da qui nasce l'esigenza di elaborare questi dati mediante il data mining, termine ispirato all'attività svolta dai minatori. Cosi come i minatori scavano la materia grezza in cerca dell'oro, il data mining "mina" i dati per estrapolare la conoscenza. In altre parole diviene ora indispensabile riuscire ad elaborare ed estrarre da questa grossa mole di dati, apparentemente senza valore, delle informazioni utili.

Il data mining è una tecnica oramai diffusa ed utilizzata nei campi più disparati e svolge un ruolo sempre più importante nel supportare le decisioni. Tale tecnica non si limita ad una analisi statistica dei dati ma piuttosto ricerca:

- L'informazione implicita, nascosta
- La presenza di pattern rilevanti

È facile immaginare come un fenomeno di questa portata abbia reso completamente inadeguati gli strumenti tradizionali andando a sconvolgere l'approccio di una considerevole mole di problematiche, concedendo ai dati e quindi all'informazione

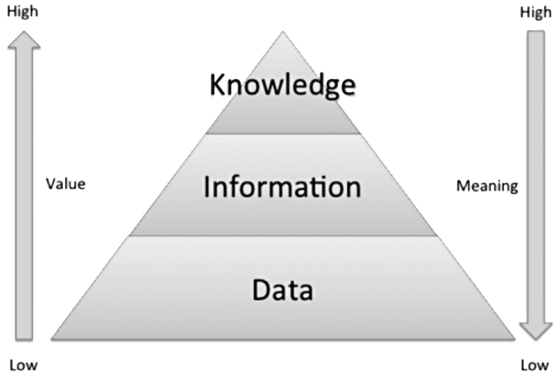

Figura 2.1: Data mining come supporto decisionale[1]

un importante ruolo per il supporto decisionale.

In letteratura il processo di analisi dei dati viene denominato *Knowledge Discovery Process* oppure per semplicità con l'acronimo KDD. Il KDD costituisce l'intero processo di estrazione della conoscenza ed è composto da cinque fasi distinte:

- **Selezione**: consiste nel selezionare un sottoinsieme di dati da quello di partenza, mantenendo solo i dati ritenuti importanti per l'analisi da effettuare. Questo è un processo delicato in quanto la selezione dei dati implica una conoscenza a priori del dominio applicativo, ed influenzerà l'intero processo di estrazione della conoscenza
- **Data clean e preprocessing**: fase dedicata alle rielaborazioni dei dati pervenuti dalla selezione, la quale si occupa di eliminare tutti quei dati ritenuti non conformi per l'analisi (rumori o outlier)
- **Trasformazione**: predispone i dati per gli algoritmi di data mining applicando diverse tipologie di trasformazioni (discretizzazione, normalizzazione ecc.)
- **Data mining**: applicazione di diversi algoritmi alla ricerca di pattern specifici conformi all'obiettivo prefissato
- **Interpretazione**: analisi dei pattern identificati

[1]Immagine tratta da: `https://www.giarts.org/article/knowledge-centric-arts-organization`

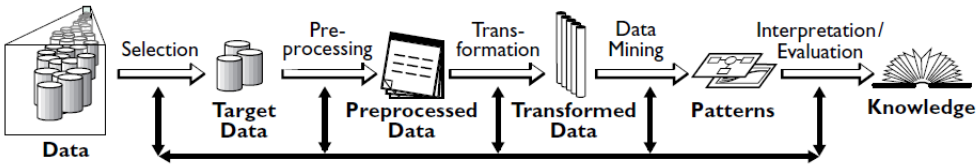

Figura 2.2: Knowledge Discovery Process[2]

2.1 Tipologie di apprendimento

Si possono indentificare due macro-tipologie di apprendimento:

- Apprendimento supervisionato
- Apprendimento non supervisionato

La scelta tra le due tipologie è dettata esclusivamente dalla conoscenza a priori disponibile sui dati e dalla finalità dello studio.
In particolare l'apprendimento supervisionato è una tecnica automatica che necessita di una conoscenza a priori sul dominio applicativo e un chiaro obiettivo sullo studio e gli obbiettivi da raggiungere. Questo rende possibile un approccio completamente automatizzato nel quale l' algoritmo viene istruito attraverso dati già etichettati (conoscenza a priori) e sulla base di questi costruisce un modello generalizzato che permetta la classificazione (obiettivo target) di dati futuri non ancora etichettati.
Differente è invece l'apprendimento non supervisionato il quale è caratterizzato da una completa mancanza di conoscenza a priori e viene quindi definita per tale motivo una ricerca di tipo esplorativa. L'obiettivo di questa tipologia è quella di riuscire ad accumunare dati apparentemente simili in gruppi denominati cluster basandosi esclusivamente sulle proprietà geometriche dei dati di input.

[2]Immagine tratta da: `http://www.ceine.cl/the-kdd-process-for-extracting-useful-knowledge-from-volumes-of-data/`

2.2 Alberi decisionali

Gli alberi decisionali sono adatti sia per problemi di classificazione che di regressione. In generale l'algoritmo si pone come obiettivo la creazione di un albero decisionale, in cui ogni nodo rappresenta una variabile all'interno del dataset, ogni arco rappresenta un valore o un range di valori possibile per la variabile interessata e ogni foglia le possibili predizioni del modello. Questo algoritmo si contraddistingue quindi per la sua semplicità e per la creazione di modello interpretabile in quanto espone la presenza di un albero decisionale.

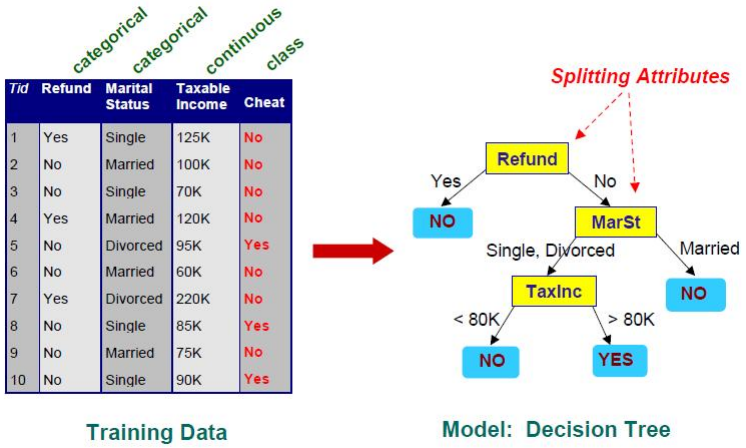

Figura 2.3: Esempio di decision Tree, Training set[3]

Generalmente per la creazione del modello si suddividono i dati in due porzioni distinte, la prima il train set destinata alla creazione del modello, mentra la seconda, il test set, finalizzata invece al test. Si può osservare come in figura 2.3 ci sia la generazione dell'albero decisionale attraverso l'impiego del train set mentra nella figura 2.4 ci sia invece la generazione della predizione attraverso l'uso dell'albero precedentemente sviluppato. Questo algoritmo presenta però alcune criticità :

- La scelta dell'ordine degli attributi di split

[3]Immagine tratta da: `http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/lguo/decisionTree.html`

[4]Immagine tratta da: `http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/lguo/decisionTree.html`

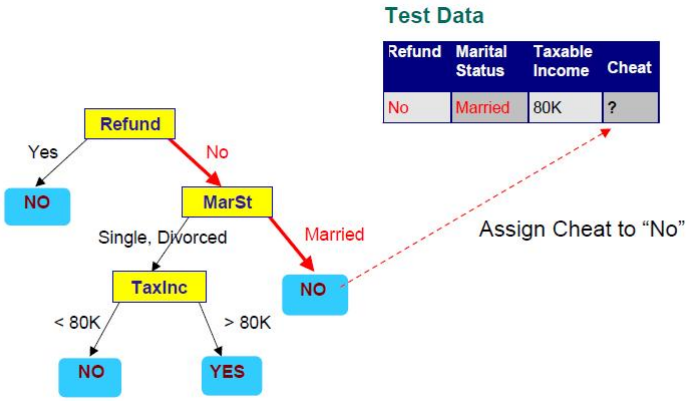

Figura 2.4: Esempio di decision Tree, Test set[4]

- Capire quando terminare la computazione del modello in caso di creazione di soluzioni non pure

Selezione dell'attributo di split

L'ordine di selezione degli attributi di split ha un impatto notevole sulla creazione dell'albero e per tale motivo vi sono vari algoritmi dedicati. Uno degli algoritmi più utilizzati è l'*algortmo di Hunt*, il quale implica ad ogni livello di split la selezione dell'attributo localmente più selettivo, con la preferenza a generare sottoinsiemi con un alto grado di purità. Per la generazione del grado di purità di un attributo vi sono vari indici tra cui:

- Gini index

$$GINI = 1 - \sum_{j} [p(j|t)]^2 \tag{2.1}$$

Dove $p(j|t)$ è la frequenza relativa della classe j al nodo t.

Il Gini Index assume quindi valori compresi tra 0 (massimo valori di purità) e 0.5 (minimo valore di purità).

- Entropia

$$Entropy = -\sum_{j} p(j|t) log_2 * p(j|t) \tag{2.2}$$

L'Entropia genera invece valori compresi tra 0 (massimo valore di purità) e 1 (minimo valori di purità)

Entrambi gli indici assumono valori pari a 0, con un massimo grado di purità, quando lo split interessato genera dei sottoinsiemi totalmente sbilanciati, mentre invece assumo valori maggiori, con quindi un minore grado di purità, quando invece l'attributo tende a dividere omogeneamente i dati.

Terminazione dell'algoritmo

La terminazione dell'algoritmo avviene al verificarsi di tre condizioni:

- La creazione di sottoinsiemi completamente puri
- L'esaurirsi degli attributi di split
- L'aggiunta di ulteriori nodi non contribuisce più alla creazioni di sottoinsiemi utili

Tuttavia quando si computa un modello vi sono altri fenomeni che costringono ad imporre ulteriori limiti alla creazione del modello. In particolare si pone maggior attenzione ai fenomeni di *underfitting* e di *overfitting*. Questi fenomeni si presentano nelle seguenti condizioni:

- *Underfitting*: il modello è stato computato utilizzando pochi attributi, generando un albero semplice, non in grado di gestire la vera complessità del problema. Questo porterà nella fase di test ad un numero elevato di errori.
- *Overfitting*: il modello è stato generato con l'implicazione di troppi nodi, computando così un albero estremamente specializzato sul train set. Vi è così la perdita di una maggiore visione generale del problema che porterà pure in questo caso ad un maggiore errore sul test set.

Per ovviare le criticità esposte sopra vi sono due strategie:

- *Pre-prunig* : si ferma lo sviluppo dell'albero nel caso in cui si determini che i successivi split futuri non siano più affidabili
- *Post-pruning* : in una prima fasi si genera l'intera albero, mentre nella fase successiva si eliminano i nodi che non portano veri contributi al modello

2.3 Classificatore bayesiano

Il classificatore bayesiano è una tecnica che affonda le proprie radice nel calcolo probabilistico ed in particolare nel Terema di Bayes. Il Teorema di Bayes permette in generale di calcolare la probabilità condizionata, ovvero la probabilità del verificarsi di una ipotesi h data l'evidenza di un evento D come

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)} \tag{2.3}$$

Dove $P(D|h)$ è la probabilità di osservare D dato il verificarsi dell'ipotesi h mentre $P(h)$ e $P(D)$ sono le probabilità a priori del verificarsi rispettivamente dell'ipotesi h e dell'evento D. Questo tipo di approccio ci permette quindi di esprime la probabilità a posteriori $P(h|D)$ sotto forma di $P(D|h), P(h)$ e $P(D)$.
Supponendo quindi di avere in input un dato $X = (x_i, x_2, .., x_n)$, l'obiettivo di questo tipo di approccio è quello di riuscire a determinare l'ipotesi h che riesca a massimizzare la probabilità condizionata $P(h|X)$, cioè di determinare l'etichettatura di classe più probabile per il dato presentato, che in termini matematici diviene

$$argMaxP(C|x_1, x_2, .., x_n) = ArgMax\frac{(P(x_1, x_2, ..., x_n|C)P(C))}{P(x_1, x_2, ..., x_n)} \tag{2.4}$$

Per poter risolvere l'equazione (1.2) bisognerebbe calcolare quindi tutte le probabilità condizionate per ogni possibile combinazione di x_i. Questo tipo di approccio risulta essere estremamente oneroso in termini di sforzo computazionale e in molti casi impraticabile, perchè necessita la conoscenza di tutte le probabilità di classe per ogni combinazione degli attributi x_i.
Per poter superare questo tipo di problematica si è deciso quindi di applicare una

sostanziale semplificazione assumendo l'indipendenza statistica di tutti gli attributi x_i (*ipotesy Naive*). Questa assunzione implica

$$P(x_1, x_2, .., x_n|C) = P(x_1|C)P(x_2|C)...P(x_n|C) \tag{2.5}$$

Da cui per poter calcolare l'assegnazione di classe bisogna risolvere

$$argMaxP(C = c_i) \prod_{k=1}^{N} P(A_k = a_k|C = c_i) \tag{2.6}$$

Questa piccola ma sostanziale semplificazione permette quindi di:

- Agevolare notevolmente il carico computazionale.
- Rendere applicabile questo classificatore anche su piccole porzioni di dataset (non è più necessario avere all'interno della base dati tutte le possibili combinazioni degli x_i).
- Rendere il modello più interpretabile dall'uomo in quanto si possono osservare nella realizzazione delle etichette di classe quale attributo pesa maggiormente sulla decisione di appartenenza.

Un ulteriore vantaggio del classificatore bayesiano è che questo è incrementale, quindi ogni singolo dato all'interno del dataset di addestramento contribuisce in maniere incrementale al calcolo della classe di appartenenza. Diviene quindi molto semplice, una volta reperiti nuovi dati di addestramento, calcolare il contributo di queste nel modello già definito.

La forza di questo algoritmo quindi sembra risiede nella validità dell'ipotesi naive, la quale però raramente può essere soddisfatta. Nonostante questa proprietà sia di rado valevole si è riscontrato però che in situazioni di non soddisfacimento si hanno comunque buone prestazione.

Il classificatore bayesiano per la sua semplicità è stato quindi nel tempo adottato in molti campi tra cui si possono considerare il Junk Mail Filtering e la classificazione di documenti testuali.

2.4 Reti Bayesiane

Le reti bayesiane possono essere utilizzate in ogni situazione in cui sia necessario modellare una realtà dominata da incertezza, situazioni in cui siano coinvolte quindi delle probabilità. La rete Bayesiana è costituita da un modello grafico probabilistico che aiuta la comprensione delle condizioni di dipendenza fra le variabili analizzate. In particolare il modello grafico è costituito da un grafo aciclico diretto (DAG) dove ogni nodo rappresenta una variabile e gli archi le dipendenze fra questi. Si può quindi dedurre che l'assenza di una arco tra due variabili ne consegue la loro indipendenza.

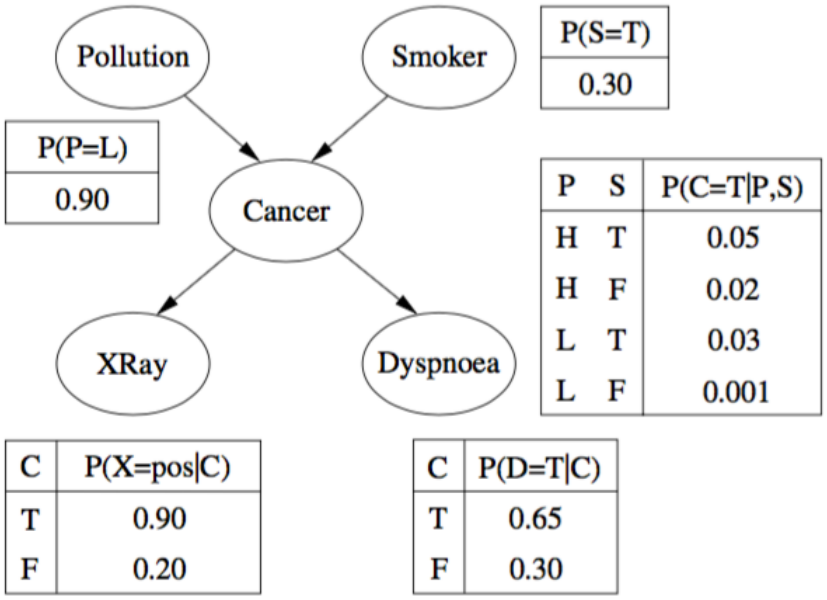

Figura 2.5: Esempio Rete Bayesiana[5]

[5]Immagine tratta da:`https://stats.stackexchange.com/questions/254631/calculating-priors-of-leaf-nodes-in-bayesian-networks`

Si può osservare in figura 2.5 un esempio di rete Bayesiana. E' evidente in figura il rapporto di dipendenza espressa dalla direzionalità degli archi che evidenziano una relazione di parentela padre-figlio tra i vari nodi. Si può quindi denotare che in questo grafo aciclico il cancro dipenda direttamente da due variabili, l'inquinamento ed il fumo, e che all'interno di ogni nodo vi siano espresse le probabilità condizionate del verificarsi di una ipotesi al variare dei valori assunti dai padri. E' quindi di facile deduzione, una volta creata la rete Bayesiana, calcolare la probabilità che una variabile assuma un determinato valore, dato il verificarsi di alcune ipotesi. Per esempio nel caso si volesse calcolare la probabilità di P(X) dato il verificarsi di P(P=L) si otterrà applicando il teorema di Bayes:

$$P(X|P) = \frac{P(P,X)}{P(X)} \tag{2.7}$$

dove

$$\begin{aligned} P(P|X) = P(P,X) = P(P,X,S,C,D) + P(P,X,\neg S,C,D) + P(P,X,\neg S,\neg C,D) + \\ + P(P,X,\neg S,C,\neg D) + P(P,X,\neg S,\neg C,\neg D) + P(P,X,S,\neg C,D) + P(P,X,S,C,\neg D) + \\ + P(P,X,S,\neg C,\neg D) = \\ = P(X|C)P(D|C)P(C|L\wedge S)P(L)P(S) + P(X|C)P(D|C)P(C|L\wedge\neg S)P(L)P(\neg S) + \\ + P(X|\neg C)P(D|\neg C)P(\neg C|L\wedge\neg S)P(L)P(\neg S) + P(X|\neg C)P(\neg D|C)P(C|L\wedge\neg S)P(L)P(\neg S) + \\ + P(X|\neg C)P(\neg D|\neg C)P(\neg C|L\wedge\neg S)P(L)P(\neg S) + P(X|\neg C)P(\neg D|C)P(C|L\wedge S)P(L)P(S) + \\ + P(X|C)P(\neg D|C)P(C|L\wedge S)P(L)P(S) + P(X|\neg C)P(\neg D|(\neg C)P((\neg C|L\wedge S)P(L)P(S) \end{aligned} \tag{2.8}$$

A questo punto basterà semplicemente sostituire i valori nella espressione 2.8 con i corrispettivi valori presenti all'interno delle tabella dei nodi della rete per ottenere la probabilità desiderata.

2.5 Support vector machine

La *support vector machine* (SVM) è un approccio sviluppato durante gli anni 90' e nel passare del tempo ha riscosso sempre più popolarità tra le comunity divenendo uno degli algoritmo "*out of box*" prediletti.

La SVM è un tipico esempio di apprendimento supervisionato e si predispone come obiettivo principe quello di riuscire a disgiungere i dati presenti nel dataset di allenamento, attraverso un iperpiano. In particolare se i nostri dati giacciono in uno spazio *p-dimensionale*, l'iperpiano è un sottospazio di dimensione *p-1* che si pone l'obiettivo di dividere i dati presenti nel training set rispettando l'etichettaura di classe. Si può quindi esprimere l'iperpiano come

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p = 0 \tag{2.9}$$

Una volta definito il dato di input $X = x_1 + x_2 + ... + x_p$ in base al risultato della equazione si può definire la porzione di spazio in cui ricade e quindi conseguentemente l'etichettatura assegnata dall'algoritmo

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p > 0 \tag{2.10}$$

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_p x_p < 0 \tag{2.11}$$

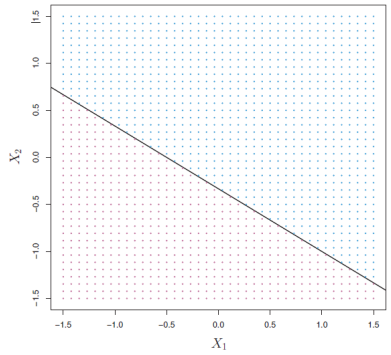

Figura 2.6: Esempio di iperpiano[6]

[6]Immagine tratta da: G. James ,D. Witten, T. Hastie e R. Tibshirani *An Introduction to Statistical Learning*

Dalla figura 2.6 si può quindi desumere che in base al risultato conseguito si può ricadere nella sottospazio superiore oppure in quello inferiore

La SVM inoltre non si limita soltanto a calcolare l'iperpiano sopra definito, ma si pone l'obiettivo di ottenere l'iperpiano migliore. Per poter conseguire questo obiettivo si deve definire un ulteriore parametro, il margine.
Il margine viene definito come la minima distanza tra l'iperpiano selezionato ed i punti appartenenti ai diversi sottospazi. Definito il margine si può allora concludere che ricavare l'iperpiano migliore coincida con massimizzare il margine. Si è quindi davanti ad un problema di ottimizzazione.

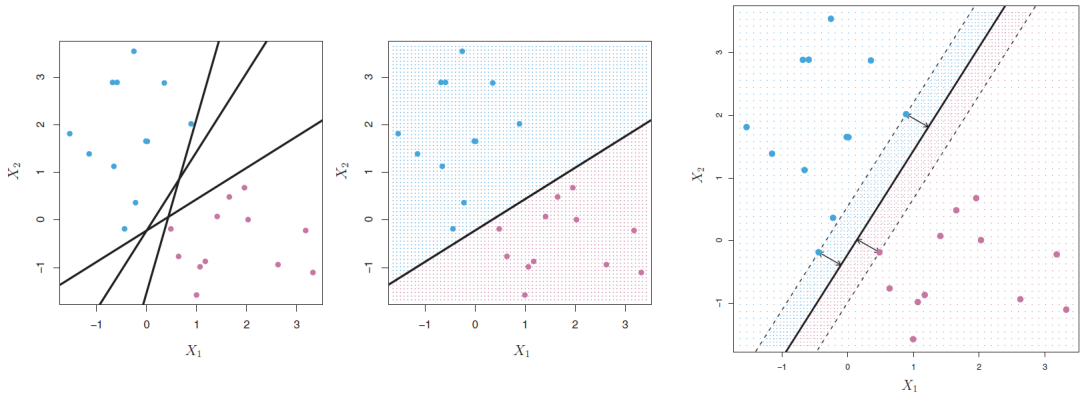

Figura 2.7: Esempio di iperpiano con margine[7]

Tuttavia purtroppo considerando casi reali ci si trova frequentemente davanti a problemi non strettamente lineari e quindi i dati possono non essere separabili da un semplice iperpiano. In questa situazione la SVM ricorre a varie trasformazioni spaziali definite con il nome di *kernel*. In generale dati due vettori x_1 e x_2 il kernel è una generica funzione K applicata ai vettori iniziali dove nel caso di kernel lineare diviene

$$K(x_1, x_2) = \sum_{i=1}^{n} (x_1)_i (x_2)_i \tag{2.12}$$

[7]Immagine tratta da: G. James ,D. Witten, T. Hastie e R. Tibshirani *An Introduction to Statistical Learning*

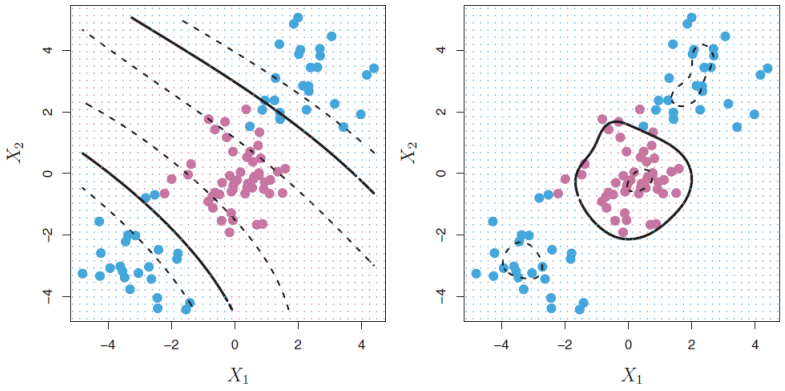

Figura 2.8: Esempio di kernel polinomiale e radiale[8]

Modificando quindi la trasformazione K si possono ricavare diverse tipologie di kernel tra cui si può definire quella polinomiale come

$$K(x_1, x_2) = (1 + \sum_{i=1}^{n} (x_1)_i (x_2)_i)^d \tag{2.13}$$

Si può notare come in figura 2.8 vi siano due diverse tipologie di approccio per risolvere il medesimo problema non lineare. Nel primo caso si ha un kernel di tipo polinomiale mentre nel secondo un kernel di tipo radiale.

[8]Immagine tratta da: G. James ,D. Witten, T. Hastie e R. Tibshirani *An Introduction to Statistical Learning*

2.6 Reti neurali

Le rete neurale è una tecnica nata negli anni 80' ispirata alla sofisticata funzionalità dei cervelli umani dove centinaia di miliardi di neuroni interconnessi elaborano le informazioni in parallelo. Alcuni ricercatori hanno quindi provato a riproporre lo stesso modello sul silicio (Wang, 2003). L'elemento chiave di questo paradigma è la nuova struttura del sistema di elaborazione delle informazioni. È composto da a numero elevato di elementi di elaborazione altamente interconnessi (neuroni) che lavorano all'unisono per risolvere problemi specifici (Nahar, 2012).

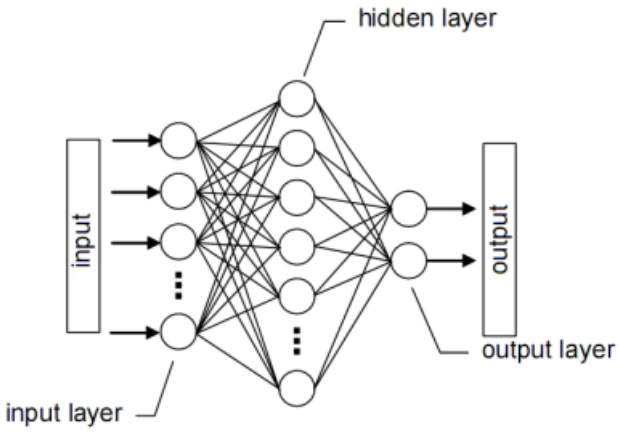

Figura 2.9: Struttura della rete neurale[9]

Analizzando la struttura delle rete neurale (figura 2.9) si possono osservare diversi livelli, ognuno di questi composto da nodi. Ogni nodo è collegato a tutti i nodi del livello successivo ed in base al livello di appartenenza acquisisce un nome ed una funzionalità differente. I nodi appartenenti al primo livello vengono denominato *nodi di ingresso*, quelli appartenenti all'ultimo livello *nodi di uscita* mentre quelli appartenenti ai livelli intermedi *nodi nascosti*.
Ogni nodo, o unità elaborativa, riceve quindi un insieme di valori di input ed al proprio interno elabora un valore di output, il quale verrà propagato a tutti i nodi appartenente al livello successivo.

[9]Immagine tratta da: *Reti Neurali* URL: `http://www.electroyou.it/c1b8/wiki/rete_neurale`

Il segnale elaborato per poter essere propagato da una unità elaborativa all'altra transita su delle *connessioni*. Ogni connessione è definita da un peso w il quale può specificare una connessione eccitatoria se presenta $w > 0$ oppure una connessione inibitoria se presenta $w < 0$.
È interessante osservare come le connessioni siano apprese in maniera completamente autonoma e costituiscono la vera conoscenza appresa dal sistema durante il periodo di addestramento. In altre parole la memoria risiede nelle connessioni.

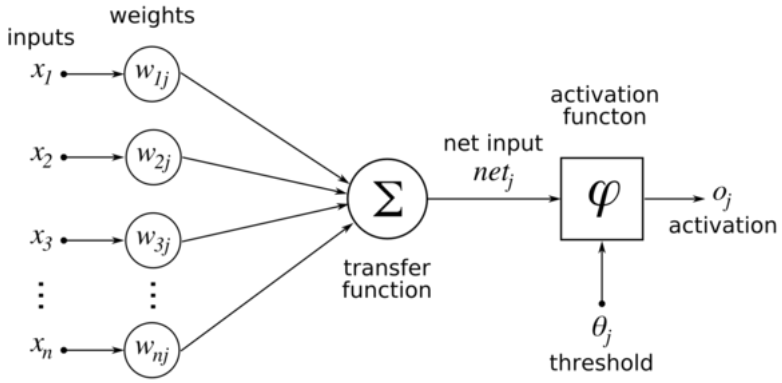

Figura 2.10: Struttura interna di una unità elaborativa[10]

Come si può osservare dalla figura 2.10 inizialmente vi è una semplice sommatoria di tutti gli input pesati rispettivamente per il peso attribuito alla singola connessione su cui avviene la propagazione, in formula

$$net_j = \sum_{j=1}^{n} x_j w_j \tag{2.14}$$

L'output del sommatore viene infine trasmesso all'ultimo modulo il quale applica a net_j la funzione di attivazione φ.

[10]Immagine tratta da: *Artificial Neural Network* URL: `http://www.saedsayad.com/artificial_neural_network.htm`

$$o_j = \varphi(\sum_{j=1}^{n} x_j w_j) \tag{2.15}$$

La funzione di attivazione φ svolge un ruolo fondamentale in quanto in base alla funzione prescelta avremmo per ogni neurone delle uscite differenti. Di seguito sono illustrate alcune delle funzioni di attivazioni più usate

- Funzione a soglia θ

$$o_j = \theta(\sum_{j=1}^{n} x_j w_j - t) \tag{2.16}$$

dove θ è la funzione di *Heaviside*

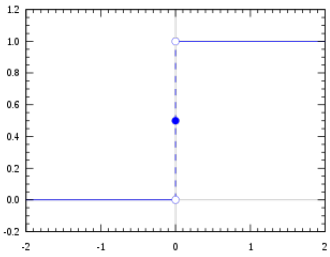

Figura 2.11: Funzione di Heaviside[11]

Se net_j supera una soglia prefissata t allora l'unità di elaborazione si accende con $o_j = 1$, in caso contrario l'output risultante è 0

- Funzioni *sigmoid* e *hyperbolic tangent*

$$\varphi(x) = \frac{1}{1 + e^{-x}} \qquad \varphi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.17}$$

[11]Immagine tratta da: *Funzione gradino di Heaviside* URL: `https://it.wikipedia.org/wiki/Funzione_gradino_di_Heaviside`

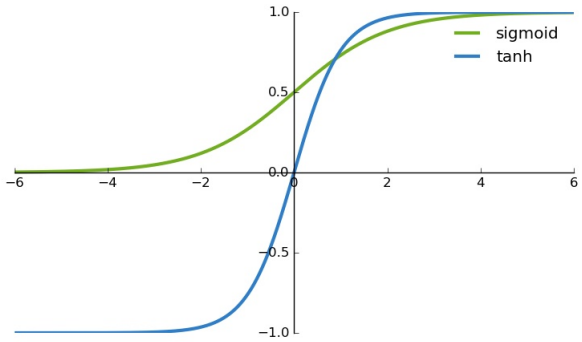

Figura 2.12: Funzioni sigmoid e hyperbolic tangent [12]

Sicuramente più interessanti in quanto consentono di avere un uscita continua, permettendone una interpretazione probabilistica

Quindi una volta definita l'architettura della rete neurale si procede verso l'addestramento del modello. La fase di addestramento si protrae in due distinte fasi

- Fase di *feed-forward*
- Fase di *back-propagation*

L'intero processo apprende in maniera completamente autonoma e iterativa rieseguendo le due fasi elencate sopra. Nella prima fase si generano gli output della rete, mentre nella seconda si protrae all'indietro l'errore commesso correggendo di volta in volta i pesi di ogni singola connessione. L'addestramento termina nel momento in cui l'errore di uscita della rete neurale si colloca sotto una soglia prestabilita.

2.7 K-Nearest Neighbors

Questo algoritmo usa un approccio completamente diverso dai modelli visti precedentemente. Solitamente si cerca attraverso l'utilizzo del train set di estrapolare delle regole su cui poi successivamente costruire il modello. Il *K-Nearest Neighbors*

[12]Immagine tratta da URL: `http://ronny.rest/blog/post_2017_08_16_tanh/`

invece si discosta da questo approccio e fa del train set il modello stesso. Per assegnare successivamente l'etichetta di classe ai dati del test set, questo algoritmo si limita a calcolare la distanza del nuovo dato con i k più vicini presenti nel train set. Ovviamente in questo contesto il successo di questo algoritmo è legato profondamente alla scelta del valore di k, il quale giocherà un ruolo fondamentale per l'etichettatura di classe.

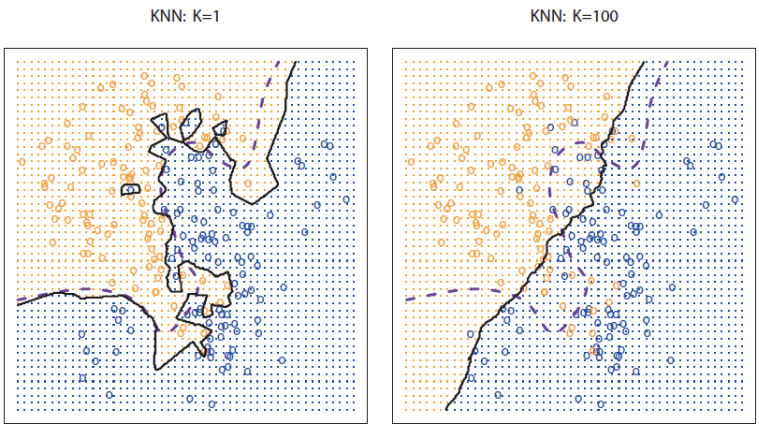

Figura 2.13: Esempio di KNN[13]

Si può osservare in figura 2.13 come la scelta di due valori distinti di k, 1 e 100, portino a due esiti completamente differenti. Nel primo caso, la scelta di un k molto piccolo genera un limite tra le due classi molto flessibile, mentre nel secondo caso, aumentando il valore di k fino a 100 vi è la creazione di un limite molto più rigido. Con la crescita di K, il metodo diventa meno flessibile e produce un limite di decisione che è prossimo al lineare. Guardando i grafici riportati in figura 2.13 sembrerebbe quindi più promettente utilizzare un valore di k piccolo piuttosto che usarne uno grande. In generale questo non è vero, in quanto il tasso di errore commesso nel train set non è assolutamente legato in alcun modo con il tasso di errore nel test set.

[13]Immagine tratta da: G. James ,D. Witten, T. Hastie e R. Tibshirani *An Introduction to Statistical Learning*

[14]Immagine tratta da: G. James ,D. Witten, T. Hastie e R. Tibshirani *An Introduction to Statistical Learning*

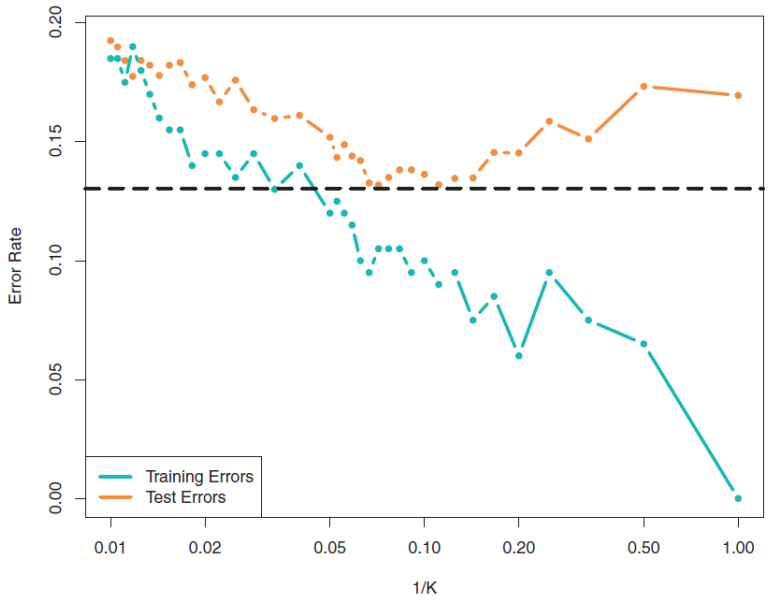

Figura 2.14: Esempio di tasso di errore del KNN[14]

Osservando infatti il grafico presente in figura 2.14 si può dedurre come l'aumento del valore di k porti nel train set ad un miglioramento generale del modello arrivando con k = 1 ad un minimo di error rate prossimo allo 0. Tuttavia osservando il comportamento del test set si percepisce un comportamento differente. In particolare si osserva un comportamento analogo fino a k = 10, dove assume un valore minimo di error rate, oltre il quale, aumentando il valore di k, vi si presenta anche un aumento del tasso di errore. Questo comportamento è dovuto al fenomeno dell'overfitting, in quanto scegliendo un valore troppo piccolo di k si ha una perdita di flessibilità del modello, specializzandosi sui dati presenti nel train set.

Capitolo 3

Criptovalute

Negli ultimi anni si è assistito ad una vera e propria rivoluzione nel mondo digitale, le *criptovalute*.

La storia delle criptovalute è relativamente breve, in quanto le origini derivano soltanto dalla seconda metà degli anni 90. Già nel 1998 fu pubblicato da Wi Dai quello che si chiamava "B-Money": un sistema di denaro elettronico anonimo e distribuito. Successivamente, Nick Szabo creò "Bit Gold". Proprio come il Bitcoin ed altre Criptovalute che avrebbero seguito a ruota la tecnologia, il Bit Gold era un sistema di criptovaluta elettronica che richiedeva che gli utenti completassero lo schema proof of work. Si era tuttavia ancora agli albori delle criptovalute.

Nel 2008 fu creata la prima criptovaluta decentralizzata: il Bitcoin (ForexItalia24, 2017).

3.1 Bitcoin

Il bitcoin con il passare degli anni è divenuto il simbolo di questo fenomeno ed è quindi doveroso introdurre il discorso sulle criptovalute con un breve riassunto della sua storia.
Il Bitcoin nasce nel 2008 con la pubblicazione da parte di Nakamoto di un documento dal il titolo "Bitcoin: A Peer-to-Peer Electronic Cash System"(Nakamoto 2008).

Dopo la diffusione del documento, la piattaforma reale per le transazioni di bitcoin è nato grazie al rilascio del primo client Bitcoin open source e della concomitante

emissione dei primi Bitcoin. Dopo l'emissione di una una quantità di circa 1 milione di bitcoin Nakamoto decise di sparire e recidere il coinvolgimento al movimento. Gavin Andresen è diventato quindi lo sviluppatore principale al Bitcoin Foundation, e successivamente divenne il "volto pubblico" dei Bitcoin (Chohan, 2017).

Da li a pochi anni il bitcoin percepirà una crescita esponenziale, se pur con oscillazioni molto ampie, raggiungendo nel dicembre 2017 un valore massimo di circa 20000 dollari per bitcoin (figura 3.1) e nel 2018 un totale di transazioni superiore ai 300 milioni.

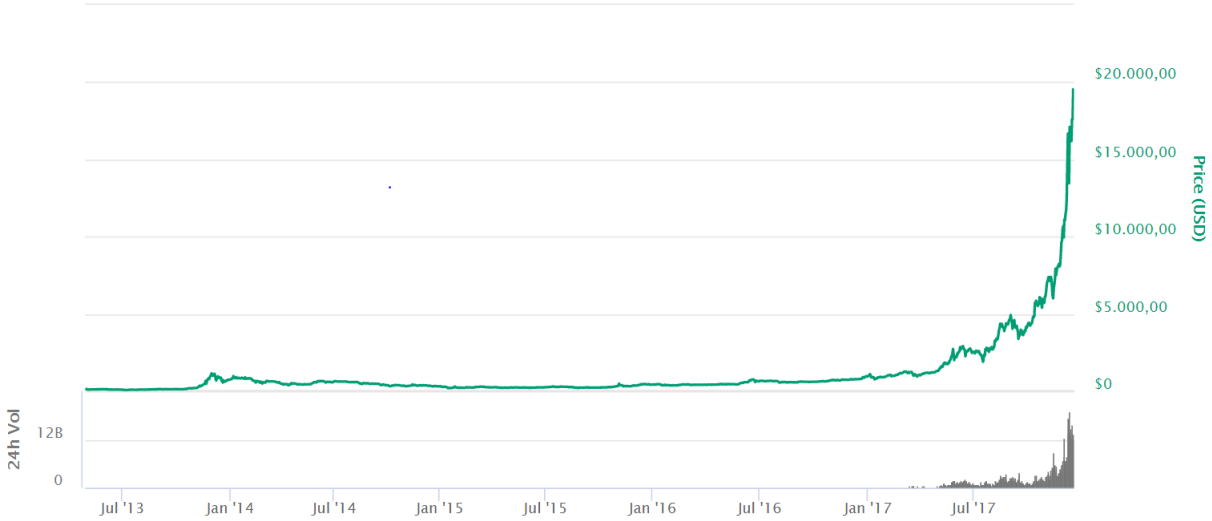

Figura 3.1: storico del prezzo del BTC[1]

Grazie a questa incredibile crescita molte sono le aziende che hanno creduto nel sistema. La prima grossa azienda fu Microsoft nel dicembre 2014 a cui si sono aggiunte Expedia, Overstock, Newegg, TigerDirect, Paypal, Square e Dell.

In seguito il codice open source del Bitcoin ha reso possibile la creazione di nuove criptomonete e più in generale ha contribuito al fenomeno degli *Altcoin*, ovvero le menete alterntive. Le monete alternative dovendo competere con il bitcoin hanno cercato di emularne il comportamento tentando però di incrementarne l'efficienza.

[1]Immagine tratta da: *coinmarketcap* URL: `https://coinmarketcap.com/currencies/bitcoin`

Per esempio tra il 2011 e il 2012 sono usciti i *Litecoin* ed i *PeerCoin* entrambi con l'intento di migliorare la velocità di approvazione delle transizioni. Altre monete come *Dash* invece hanno cercato di creare un concetto di anonimato tentando di offuscare gli indirizzi pubblici a cui erano legate le transazioni.

3.2 Valute digitali vs tradizionali

Di seguito verranno illustrate in maniera superficiale le differenze tra le valute tradizionali e quelle digitali. In particolare ci si è focalizzati sull'intero processo di gestione delle valute e come queste possano essere trasferite da un conto ad un altro.

3.2.1 Sistema tradizionale

Nel sistema tradizionale ogni individuo possessore di un conto presso la banca possiede il numero del conto e l'autenticazione dell'intestatario avviene tramite la possessione delle credenziali del conto e della chiave segreta (pin). La persona può quindi identificarsi nella propria banca come titolare dell'account e può richiedere che il denaro associato al proprio numero di conto venga trasferito sul conto di qualcun altro presso un'altra banca. La banca modificherà nel database interno la liquidità associata al conto e comunicherà alla banca destinataria la transazione in atto.
In definitiva il normale sistema di pagamento bancario attraversa una serie limitata di intermediari privati che controllano e quindi informano i titolari dell'account che le transazioni sono avvenute(Scott, 2016).

3.2.2 Valute digitali

Per il mondo delle criptomonete vi è un discorso analogo ma con due sostaniziali differenze:

- Non esistono entità private
- I database migrano da un'architettura centralizzata e privata ad un'architettura distribuita e pubblica

La peculiarità di questo sistema è che la registrazione presso il sito `https://bitcoin.org/en/choose-your-wallet` è completamente gratuita senza alcun tipo di controllo centralizzato. Questo approccio rende quindi il sistema notevolmente più flessibile e soggetto a meno verifiche (si può anche non specificare l'identità proprietaria dell'account). Dopo la registrazione l'utente dovrà procedere prima con la crazione di un *wallet* dove conserverà le proprie valute e poi di un nodo Bitcoin peer-to-peer per poter accedere alla rete e quindi alla blockchain (Böhme, Christin, Edelman, & Moore, 2015).

Il possessore di un portafoglio possiede un indirizzo pubblico e l'autenticazione dell'intestatario avviene tramite una chiave privata. In questo caso la persona può richiedere il trasferimento delle criptovalute verso un ulteriore persona specificando l'indirizzo pubblico del destinatario. La modifica del database distribuito come già specificato non avviene più tramite intermediari privati ma avviene invece tramite una rete di persone decentralizzate chiamate *minatori*(Scott, 2016). La presenza della transazione all'interno del database distribuito è pubblica ed entrambe le parti possono quindi accertare la presenza di questa.

A differenza delle monete tradizionali quindi non vi sono enti centrali o complicati sistemi finanziari ed il valore di ogni moneta è strettamente legata alla domanda e alla offerta. La decentralizzazione è in definitiva il suo vero punto di forza, permettendo in generale una maggiore privacy e ostacolando concentrazioni di potere in singole persone o organizzazioni.

3.3 Blockchain

La *blockchain* può in generale essere considerata come un libro mastro pubblico dove tutte le transazioni sono registrate in strutture dati a blocchi(Zheng, Xie, Dai, Chen, & Wang, 2017).

In figura 3.2 si può osservare la composizione di ogni singolo blocco all'interna della catena. Ogni blocco ha un header composto dai seguenti campi:

[2]Immagine tratta da: *Data Models for Blockchain* URL: `http://www.databaseanswers.org/data_models/blockchain/print_version.htm`

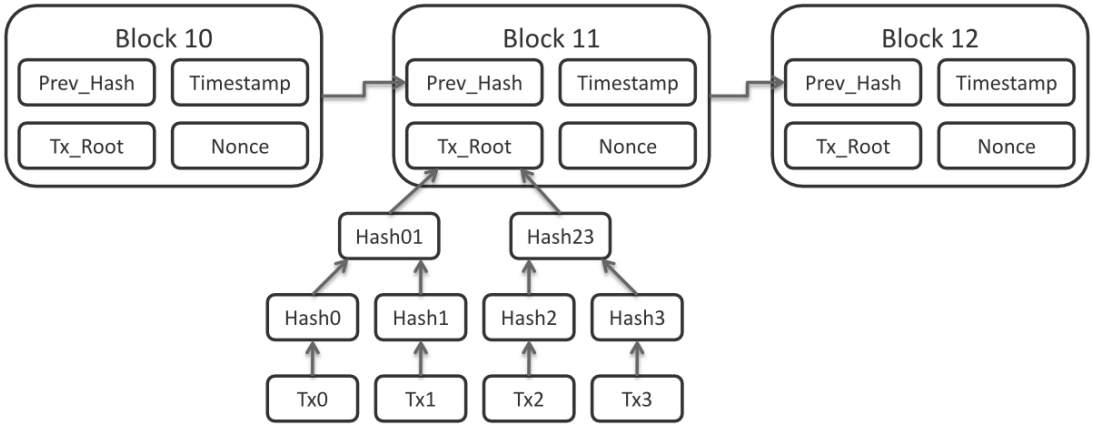

Figura 3.2: esempio di sequenza di blocchi nella blockchain[2]

- *PrevHash*, l'ash del blocco antecedente
- *Timestamp*, timestamp relativo alla transazione più recente
- *TxRoot*, hash comlessivo di tutte le transazioni presenti nel blocco
- *Nonce*, valore di 8 byte casuale

Si può notare come il *PrevHash* leghi ogni blocco con il suo antecedente ed il *TxRoot* garantisca l'integrità delle transazioni registrate nel nodo.

La blockchain inoltre garantisce autenticità ed integrità su ogni singola transazione attraverso l'utilizzo della *firma digitale*. Ogni utente che desidera quindi eseguire una transazione deve necessariamente possedere una coppia di chiave pubblica-privata.

L'architettura proposta e le strategie adottate rendono la blockchain *immutabile*, non permettendo alcun tipo di modifica dei dati una volta che questi entrano a far parte della catena.

3.4 Sistema *Proof of work*

Ogni nodo all'interno della rete possiede due liste distinte:

- Lista delle transazione validate
- Lista delle transazioni in attesa di convalidata

Quando un utente genera una transizione invia un pacchetto broadcast all'interno della rete, ed i nodi percepiranno questa transazione inserendola nella lista delle transizioni da convalidare. Ogni nodo quindi dopo un intervallo regolare stabilito, per esempio di 10 minuti, propone il proprio pool di transazioni da convalidare alla rete. Questo tipo di protocollo è quindi soggetto all'introduzione di transizioni non valide ed è quindi di fondamentale importanza introdurre un protocollo di validazione. Il protocollo di validazione processerà quindi le richieste dei vari nodi ed aggiungerà alla catena solamente un pool di richieste validate.
In generale questo tipo di architettura si presta a diversi tipi attacchi come il *Sybil attack* o il *double spend attemp*, entrambi basati sull'idea di introdurre all'interno della rete delle transazioni malevole e con l'intento quindi di minare l'efficienza dell'intero processo di validazione

Il sistema di prevenzione, per incentivare un comportamento corretto, prevede una ricompensa per i minatori che contribuiscono alla creazione del blocco mentre tende a scoraggiare qualsiasi tipo attacco perchè computazionalmente troppo oneroso. Questa strategia, denominata *proof of work*, dovrebbe quindi promuovere un comportamento leale in quanto tutte le parti coinvolte nell'approvazione della transazione percepiscono un guadagno. In particolare i minatori si assicurano due fonti di guadagno.

La prima è derivante della creazioni dei blocchi. Inizialmente la ricompensa per ogni blocco assumeva un guadagno pari a 50 bitcoin, ma il sistema prevede un dimezzamento del valore percepito periodicamente nel tempo (ogni 210 blocchi), in base alla totalità dei bitcoin creati. Quando la totalità dei bitcoin coniati assumerà il suo massimo valore pari a 21 milioni allora nessun minatore potrà più percepire nessun guadagno su questa attività.

La seconda fonte di reddito deriva invece dalla verifica del puzzle della transizione, ovvero il minatore percepisce una *fee* sulla avvenuta transazione. La ricompensa è in realtà opsionale a discapito delle parti coinvolte nell'operazione ma nel 2014 ben il 97% delle transazioni prevedeva una ricompensa per i minatori (Narayanan, Bonneau, Felten, Miller, & Goldfeder, 2016).

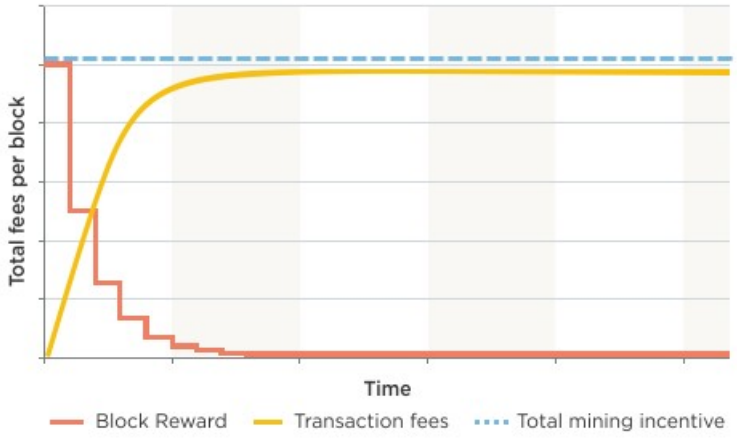

Figura 3.3: Block Reward[3]

$$minerReward = transactionfee(opsional) + blockreward \tag{3.1}$$

Il costo energetico per la risoluzione di un blocco è molto elevata in quanto il problema da risolvere è oneroso dal punto di vista computazionale. È comprensibile quindi che da un certo periodo in poi il profitto ricavato dalla creazione di blocco non basti più per sostenere le spese energetiche. Per questo motivo come si può vedere dalla figura 3.3 sarà quindi deducibile un incrementale crescita delle tasse di transazione per mantenere alto l'incentivo del minatore.

[3]Immagine tratta da: `https://bitsonblocks.net/2015/09/21/a-gentle-introduction-to-bitcoin-mining/`

Capitolo 4

Trading

Con il termine trading, in generale, si intende la compravendita di titoli azionari in borsa. L'obiettivo principale di questa attività è quello di trarre del profitto da operazioni di tipo speculativo volte a sfruttare le oscillazioni dei mercati. Attualmente il trading viene sempre associato alla parola online. Prima dell'avvento di internet, il trading era una attività accessibile solamente da pochi eletti, mentre oggi attraverso l'uso del web chiunque ha la possibilità accedervi. Per tale motivo è importante la comprensione del sistema e dei rischi che questo comporta. I vantaggi di operare in questo settore sono molteplici tra cui:

- Bassi costi di commissioni
- La possibilità di accedere ad una grossa di mole di dati, attraverso la quale il trader può informarsi riguardo ad un particolare titolo

Da questo scenario nasce il bisogno di costruire strumenti adeguati che possano sostenere le scelta del trader. È facile comprendere, come il data mining ad altri strumenti di analisi abbiano da subito occupato un posto di rilevanza in questo settore.

4.1 Trading intraday

Con trading intraday si intende un approccio al mercato con l'obiettivo di profitto nel brevissimo termine. Il trading intraday, infatti, implica l'apertura e la chiusura dell'operazione entro la giornata mediante l'utilizzo di time frame molto bassi

(Magalotti, 2016).

Gli operatori intraday negoziano con alcuni dei più alti livelli di rischio nell'intero mercato, ma con la possibilità che vi siano guadagni maggiori sull'investimento.I trader giornalieri operano a tempo pieno affinchè possano rimanere sempre aggiornati con le ultime attività di trading. Questi monitorano continuamente un elevata quantità di dati nel tentativo di rintracciare le migliori condizioni di mercato e i tempi ideali per la negoziazione. Sebbene i livelli di rischio siano elevati per i trader infragiornalieri, la vendita delle loro posizioni entro la fine della giornata implica un rischio limitato (Harvey, 2018).

In conclusione i vantaggi e gli svantaggi nell'affrontare il trading intraday sono:

Vantaggi

- Richiesta di un capitale inferiore: il trading intraday, a parità di investimento, permette guadagni molto più elevati
- Si limita il rischio overnight: attraverso l'impiego del trading intraday ogni posizione deve essere chiusa al termine della giornata limitando quindi rischi di oscillazioni

Svantaggi

- Richiede una costante attenzione del trader
- Spesso le operazioni a lungo termine danno profitti maggiori in quanto non impongono la chiusura della posizione a fine giornata
- Notevole aumento dei costi di transazione

4.2 Indicatori

Gli indicatori sono impiegati nel supporto decisionale del trader. Questi sono composti da costruzioni matematiche volte ad analizzare l'andamento dei prezzi e dei volumi scambiati con l'obiettivo di riuscire a prevedere l'andamento futuro delle quotazioni interessate. Di seguito sono riportati gli indicatori maggiormente usati dai trader.

4.2.1 Medie mobili

Le medie mobili si ottengono sommando i valori della serie e dividendo il risultato ottenuto per il numero delle osservazioni. Una delle proprietà fondamentali è il periodo su cui si calcola la media mobile. Per determinare il giusto valore è importante capire l'intervallo di tempo entro il quale si vuole operare e, di conseguenza, determinare quanto la media debba essere sensibile alle variazioni del mercato.
Minore è l'intervallo temporale di una serie, maggiore sarà la sua reattività è la probabilità di generare falsi segnali.

Figura 4.1: Esempio di Media Mobile[1]

Nella figura 4.1 è illustrato un esempio riguardante tre medie mobili con differente periodo. La media mobile identificata con il colore verde è calcolata lungo un breve periodo, infatti risulta molto reattiva e segue bene il trend. Le altre medie riportate nel grafico, caratterizzate da un periodo più lungo, hanno minor reattività e un andamento più rilassato.

SMA

La SMA (Simple Moving Average), detta anche media aritmetica, è una delle più usate per la sua semplicità.

[1]Immagine tratta da: *Media Mobile* URL: `https://opzionibinariechiare.wordpress.com/strategie/media-mobile/`

$$SMA = \frac{Sum\ of\ prices\ of\ the\ last\ n\ periods}{n} \tag{4.1}$$

Lo svantaggio della SMA è quello di assegnare uguale importanza a tutte le osservazioni. Sarebbe più utile, invece, assegnare maggior peso alle osservazioni recenti.

EMA

La EMA (Exponential Moving Average), media mobile, rimedia le mancanze enunciate della SMA. Il calcolo, infatti, viene generato attraverso un sistema più complesso, il quale da maggior rilievo ai prezzi recenti e minor peso a quelli passati. Per tale motivo si tengono in considerazione nel calcolo molte più osservazioni rispetto alla media aritmetica.

$$EMA = (ClosingPrice * K) + (EMA_{yesterday} * (1 - K)) \tag{4.2}$$

dove:

- K È uguale a $\frac{2}{N+1}$
- N È il periodo su cui è calcolata la media

VWAP

Il VWAP (volume-weighted average price) è il rapporto tra la somma del valore scambiato per ogni transazione ed il totale delle azioni scambiate lungo un arco temporale prestabilito.

$$P_{vwap} = \frac{\sum_j P_j * Q_j}{\sum_j Q_j} \tag{4.3}$$

dove

- P_{vwap} È la media mobile ponderata per il volume
- P_j È il prezzo di acquisto del titolo durante la compravendita j

- Q_j È la quantintà di titoli comprati durante compravendita j

Uso di una media mobile

Le medie mobili sono considerate come trendline e quindi possono creare delle zone di supporto e di resistenza. In particolare, quando il prezzo interessato salirà al di sopra o al di sotto della media mobile, si otterrà la generazione di segnali di acquisto o di vendita .

Figura 4.2: Segnali generati da una Media Mobile[2]

In questo caso è importante la scelta del periodo su cui operare. Una media con un intervallo temporale ristretto genera un valore che segue molto bene il trend, ma è soggetta alla generazione di falsi segnali. Di contro la scelta di un periodo più ampio genera una media meno pronunciata e più smussata che ha maggiori difficoltà a seguire il trend.

Uso di due medie mobili

L'uso di due medie mobili è una tra le tecniche più utilizzate dai trader. La possibilità di utilizzare due medie mobili (ovviamente di due periodi differenti) permette di generare segnali attraverso la loro intersezione (*doppio crossover*).

[2]Immagine tratta da: *La media mobile nel trading binario: ecco come sfruttarla!* URL: `https://www.on-line-trading.it/media-mobile.html`

Figura 4.3: Segnali generati da due Medie Mobili[3]

I segnali sono generati nel modo seguente:

- Segnale di acquisto: generato nel momento in cui la media più veloce (con il periodo minore) incrocia al rialzo quella più lenta
- Segnale di vendita: generato nel momento in cui la media più veloce (con il periodo minore) incrocia al ribasso quella più lenta

Anche in questo caso è necessario scegliere con attenzione i periodi delle medie per la creazioni di segnali adeguati.

Canali di medie

In questo caso vi è la creazione di un canale capace di contenere il trend analizzato e di riconoscere, al superamento delle due soglie prestabilite, un discostamento dal trend. Le soglie sono costruite calcolando una media mobile e sommando e sottraendo a questa una determinata quantità. L'obiettivo di questo strumento è percepire le grandi variazioni del titolo e come queste si discostano dalla sua media mobile.

Questi concetti sono utilizzati per la creazione di un secondo strumento finanziario più accurato e performante, le *Bande di Bollinger*.

---

[3]Immagine tratta da: *Segnali di acquisto e vendita con le medie mobili* URL: `http://espertoforex.com/media-mobile`

[4]Immagine tratta da: *Analisi tecnica canale medie envelopes.png* URL: `https://it.wikipedia.org/wiki/File:Analisi_tecnica_canale_medie_envelopes.png`

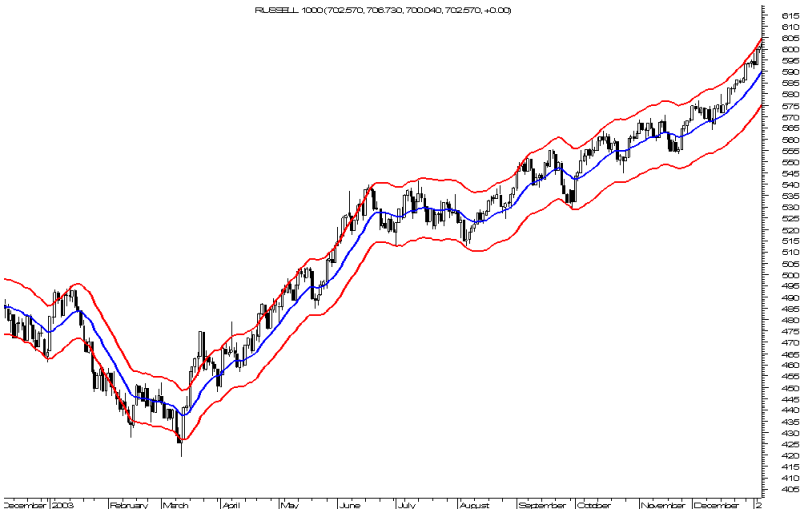

Figura 4.4: Esempio di un canale di medie[4]

4.2.2 BBands

Le BBands (Bande di Bollinger) si ottengono calcolando prima la media mobile dei prezzi a n giorni (spesso mantenuto uguale a 20) a cui successivamente viene addizionato e sottratto il valore di deviazione standard moltiplicato per un determinato fattore f (spesso uguale a 2). Si ottengono quindi le Bande di Bollinger attraverso la definizione di 3 soglie:

- La soglia centrale ottenuta dalla media mobile
- La soglia superiore ottenuta sommando alla media mobile f volte la deviazione standard
- La soglia inferiore ottenuta sottraendo alla media mobile f volte la deviazione standard

Delle bande ampie suggeriscono una alta voltalità del prezzo, al contrario, delle bande ravvicinate indicano una bassa volatilità.

Le BBands generano segnali di acquisto e vendita quando si verificano le seguenti condizioni:

[5]Immagine tratta da: *Strategie con le Bande di Bollinger* URL: `https://www.confrontobroker.it/strategia-bande-bollinger/`

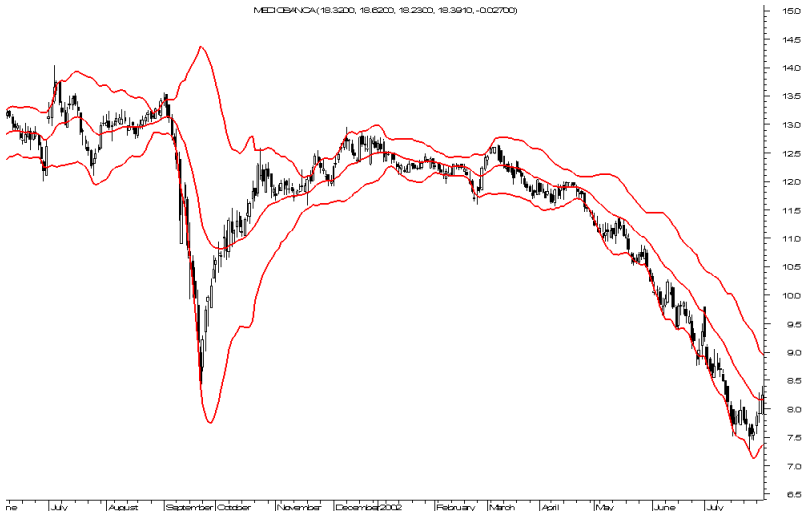

Figura 4.5: Esempio di BBands[5]

- Segnale di vendita: il prezzo del titolo supera la banda superiore per poi rientrare. Questo corrisponde ad un aumento repentino del prezzo seguito da un rallentamento o aggiustamento.
- Segnale di acquisto: il prezzo supera la banda inferiore per poi rientrare. Questo corrisponde ad un improvviso calo del prezzo seguito probabilmente da un cambiamento di trend.

4.2.3 MACD

Il MACD, per la generazione dei segnali, fa uso di due medie mobili, una di breve e una una di lungo periodo. Si osserverà quindi:

- Un incrocio rialzista: la media di corto raggio incrocia, dal basso verso l'alto, quella di lungo raggio suggerendo un trend rialzista
- Un incrocio ribassista: la media di corto raggio incrocia, dall'alto verso il basso, quella di lungo raggio suggerendo quindi un trend ribassista

Il MACD, introdotto da *Gereld appel*, nasce stabilendo la lunghezza dei periodi delle medie, una di 12 e una di 26. IL punto di forza di questo strumento risiede nella sua diffusione, in quanto, la sua attendibilità aumenta insieme al numero di operatori che lo utilizzano.

IL MACD viene infatti definito come

$$MACD = EMA12 - EMA26 \tag{4.4}$$

Dalla formula si osserva che la generazione del segnale avverà quando MACD assumerà un valore pari a zero. Si osserva un trand rialzista quando il MACD è maggiore di zero, viceversa, un trand ribassista quando il valore è minore di zero.

Frequentemente si usa un ulteriore media mobile a periodo 9 che definirà la *signal line*. La signal line, essendo più breve di quella a periodo 12, assume uno scopo predittivo poichè ,essendo più reattiva, prevedere in anticipo i segali generati dal MACD.

4.2.4 RSI

L'*RSI* (Relative Strenght Index) è un oscillatore largamente utilizzato per l'analisi tecnica del Forex e si pone l'obiettivo di identificare due fasi: ipervenduto e ipercomprato. L'RSI, basandosi sull'analisi dello storico dei prezzi, si pone l'obiettivo di quantificare la forza o debolezza di una quotazione.

$$RSI = \frac{U}{U + D} * 100 \tag{4.5}$$

dove

- U è la media delle differenze di chiusura al rialzo degli ultimi n giorni
- D è la media del valore assoluto delle differenze di chiusura al ribasso degli ultimi n giorni

I valori assunti dall'oscillatore sono compresi tra 0 e 100. Valori maggiori di 70 identificano un segnale di ipercomprato, mentre valori inferiori a 30 un segnale di ipervenduto. Il superamento di una delle due soglie ci identifica una possibile inversione del trend.

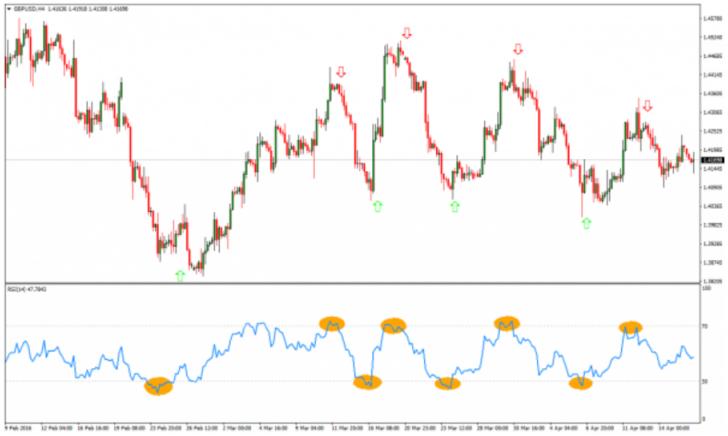

Figura 4.6: Esempio di RSI[6]

Si può osservare dalla immagine 4.6 come siano stati evidenziati in arancione i momenti in cui l'RSI si è protratto in zone di ipercomparto e di ipervenduto. In corrispondenza di tali zone si sono verificate le inversioni di trend preannunciate dall'oscillatore.

4.2.5 ROC e MOMENTUM

Il Roc (*Rate of Change*) è un oscillatore di momentum che determina la variazione percentuale del prezzo nel periodo in analisi. In definitiva viene definito confrontando l'attuale prezzo con i precedenti n. La formula del ROC è definita secondo l'equazione 4.6

$$ROC = \frac{P - Pn}{Pn} * 100 \tag{4.6}$$

dove

- P: ultima chiusura

[6]Immagine tratta da: *Indicatore RSI* URL: `https://www.money.it/indicatore-RSI-calcolo-significato-esempio`

- Pn: chiusura dell'ennesimo periodo a ritroso

L'oscillatore assume valori prossimi allo zero e definisce i seguenti segnali:

- Segnale di acquisto: superamento dal basso verso l'alto della linea dello 0
- Segnale di vendita: superamento dall'alto verso il basso della linea dello 0

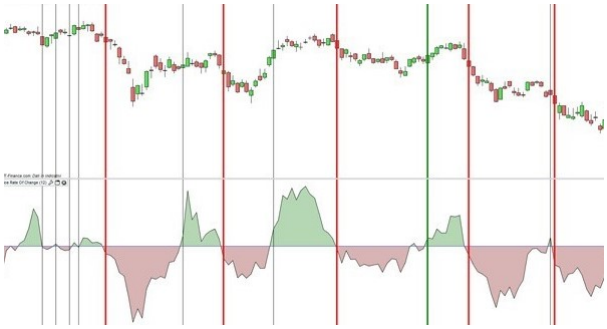

Figura 4.7: Esempio di ROC[7]

Si può osservare come in figura 4.7 siano stati evidenziati i segnali generati. In verde e in rosso sono evidenziati i segnali di acquisto e di vendita andata a buon fine, ovvero quelli che avrebbero fatto percepire un guadagno.

Si può anche osservare come all'inizio del grafico vi sia la generazione di falsi segnali dovuti a piccole oscillazione del prezzo rispetto al n-esimo prezzo a cui si fa riferimento.

Per quanto riguarda l'oscillatore *Momentum* vale il medesimo discorso ma con una piccola variazione, il Momento non studia la variazione percentuale del prezzo ma la sua differenza.

4.2.6 CCI

Il CCI (*Commodity Channel Index*) è uno degli strumenti maggiormente utilizzati e oramai presente nella maggior parte delle piattaforme. Viene utilizzato per definire

[7]Immagine tratta da: *Oscillatore ROC price rate of change* URL: `https://www.ig.com/it/oscillatore-roc-price-rate-of-change`

delle zone di ipercomprato e ipervenduto e per la generazione di segnali di acquisto e vendita. Il CCI si ottiene come:

$$CCI = \frac{TP - SMATP}{0{,}015 * DevMedia} \tag{4.7}$$

dove:

- TP è il Typical Price, il quale si ottiene attraverso la seguente media $\frac{high+low+close}{3}$
- SMATP è la simple moving average di TP
- DevMedia è la deviazione media ottenuta dapprima sommando le differenze in valori assoluto tra SMATP e TP per ciascuno degli n periodi passati per poi successivamente dividere il risultato per n.

Il CCI assume valori positivi quando il valore assunto è maggiore della sua media e valori negativi quando assume un valore al di sotto della sua media. I range dei valori assunti si suddividono quindi in due zone:

- Zona 1: compresa tra -100 e 100 in cui vi ricade circa il 75% dei valori
- Zona 2: in cui ricade il restante 25% dei valori

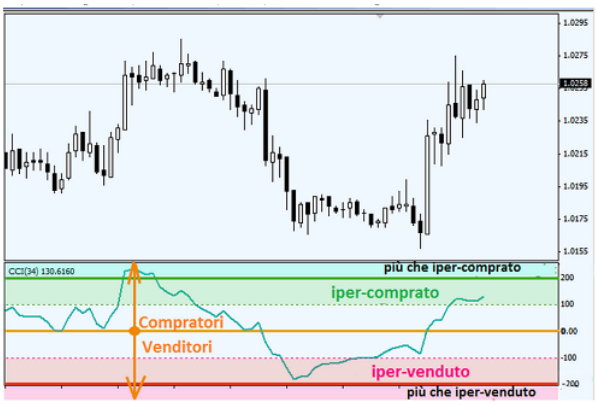

Figura 4.8: Esempio di CCI[8]

Generazione dei segnali:

[8]Immagine tratta da: *Commodity channel index (CCI)* URL: `https://www.forexitalia24.com/strategie/cci.php`

- Segnale di acquisto: derivante dal superamento del valore +100, con conseguente entrata nella zona di ipermercato. Conferma una trand rialzista con la conseguente possibilità di aprire delle posizioni long.
- Segnale di vendita: generato durante il superamento del valore -100 e quindi alla presenza di una situazione di ipervenduto.

Se il trend oltrepassa valori maggiori (in valore assoluto) di 200 allora si ricade in zone di *più che ipercomprato* e *più che ipervenduto* le quali indicano un possibile cambiamento di trend.

4.2.7 OBV

L'OBV consiste in una somma cumulativa di volumi. Si può definire l'OBV corrente nel seguente modo:

- Se il prezzo di chiusura corrente è maggiore del precedente

$$OBVcorrente = PrecedenteOBV + VolumeCorrente \tag{4.8}$$

- Se il prezzo di chiusura corrente è minore del precedente

$$OBVcorrente = PrecedenteOBV - VolumeCorrente \tag{4.9}$$

- Se il prezzo di chiusura corrente è uguale al precedente

$$OBVcorrente = OBVprecedente \tag{4.10}$$

Questo strumento fu ideato da *Grossville* in seguito ad una semplice osservazione, i volumi anticipano i prezzi. In particolare l'indicatore percepisce una risalita quando i volumi nelle sedute rialziste superano i volumi delle sedute ribassiste, e viceversa, tende a scendere quando vi sono maggiori volumi nelle giornate ribassiste. L'utilizzo dell'OBV si basa sull'analisi delle divergenze tra l'andamento dell'indicatore e quello del prezzo. In particolare si possono enunciare due tipi di divergenze:

[9]Immagine tratta da: *On Balance Volume (OBV), come sfruttare questo indicatore per fare trading online* URL: `http://www.universoforex.it/news/534/on_balance_volume_obv_come_sfruttare_questo_indicatore_per_fare_trading_online.html`

Figura 4.9: Esempio di OBV[9]

- Divergenza rialzista: si verifica nel momento in cui l'OBV forma minimi crescenti in disaccordo con il prezzo, che forma invece minimi decrescenti

Figura 4.10: Esempio di divergenza rialzista[10]

- Divergenza ribassista: si verifica nel momento in cui l'OBV forma minimi decrescenti in disaccordo con il prezzo, che forma invece minimi crescenti

[10]Immagine tratta da: *OBV - Indicatore on balance volume* URL: `https://www.ig.com/it/indicatore-on-balance-volume-`

[11]Immagine tratta da: *OBV - Indicatore on balance volume* URL: `https://www.ig.com/it/indicatore-on-balance-volume-`

Figura 4.11: Esempio di divergenza ribassista[11]

Su può vedere, come in figura 4.11, che l'indicatore OBV non riesce a confermare l'andamento del trend, in quanto genera un doppio massimo segnalando una possibile ricaduta del prezzo.

4.2.8 Oscillatore stocastico

L'oscillatore stocastico, sviluppato da George C. Lane alla fine degli anni '50, non segue nè il prezzo nè il volume, ma la velocità o il ritmo del prezzo (stockcharts Staff, 2018).

L'oscillatore stocastico è un indicatore di momentum che confronta il prezzo di chiusura di un titolo con l'intervallo dei suoi prezzi in un determinato periodo di tempo. La sensibilità dell'oscillatore ai movimenti del mercato è riducibile regolando quel periodo di tempo o prendendo una media mobile del risultato (Investopedia-Staff, 2018b).

L'oscillatore viene calcolato utilizzando le seguenti formule:

$$\%K = \frac{C - L_{14}}{H_{14} - L_{14}} * 100 \tag{4.11}$$

$$\%D = 3day\ SMA\ of\ \%K \tag{4.12}$$

dove

- C è il prezzo di chiusura più recente
- L_{14} è il prezzo minimo delle 14 precedenti sessioni di trading
- H_{14} è il prezzo massimo delle 14 precedenti sessioni di trading

L'oscillatore stocastico è composto quindi da due indici $\%K$ e $\%D$, che assumono valori compresi tra 0 e 100. La linea K è la più veloce mentre la D è la più lenta . L'investitore deve osservare come la linea D e il prezzo di emissione si spostino nelle posizioni di overbought (oltre il valore 80) o di oversold (sotto il valore 20). Si deve prendere in considerazione la possibilità di vendere il titolo quando l'indicatore supera il livello 80. Al contrario, si prende in considerazione l'acquisto di un titolo se si osserva un valore inferiore a 20, e sta iniziando a salire, con un aumento parallelo del volume (Investopedia-Staff, 2018c), figura 4.12.

Figura 4.12: Esempio di oscillatore stocastico[12]

Esistono tre versioni dell'oscillatore stocastico:

- L'oscillatore stocastico veloce: indicatore creato e pansato da George C. Lane
- L'oscillatore stocastico lento: definito per generare segnali maggiormente interpretabili, in quanto lo stocastico veloce ha la caratteristica di essere molto reattivo e alcune volte di difficile applicazione. In questa versione dello stocastico la linea %K è costruita come media mobile (a 3 periodi) della linea %D della versione dello stocastico veloce, mentre la linea %D rappresenta a sua volta la media mobile (a 3 periodi) della nuova linea %K.
- L'oscillatore stocastico completo: variante personalizzabile dello stocastico lento in cui si possono impostare i parametri necessari, tra cui il numero di periodi per il %K e per il %D. Questa versione è la più utilizzata dalla maggior parte dei trader (Ig-Staff, 2018).

4.2.9 CMO

L'oscillatore di momento, inventato da Tushar Chande, è un oscillatore tecnico. Il CMO è generato calcolando la differenza tra la somma di tutti i guadagni recenti e la somma di tutte le perdite recenti, dividendo il risultato per la somma di tutti i movimenti di prezzo nel periodo (Investopedia-Staff, 2018a).

$$CMO = \frac{S_u - S_d}{S_u + S_d} * 100 \tag{4.13}$$

dove:

- S_u è la somma di tutti i guadagni recenti
- S_d è la somma di tutte le perdite recenti

In generale il CMO è considerato una variazione del RSI con la presenta però di alcune differenze:

[12]Immagine tratta da: *Strategia forex EMA, stocastico lento e RSI* URL: `https://www.manualeforex.it/strategie-forex/ema-stocastico-lento-rsi/`

- L'utilizzo anche a numeratore dei valori al ribasso (l'RSI impiega solo i valori al rialzo)
- Il range di valori varia da -100 a 100 (L'RSI da 0 a 100)

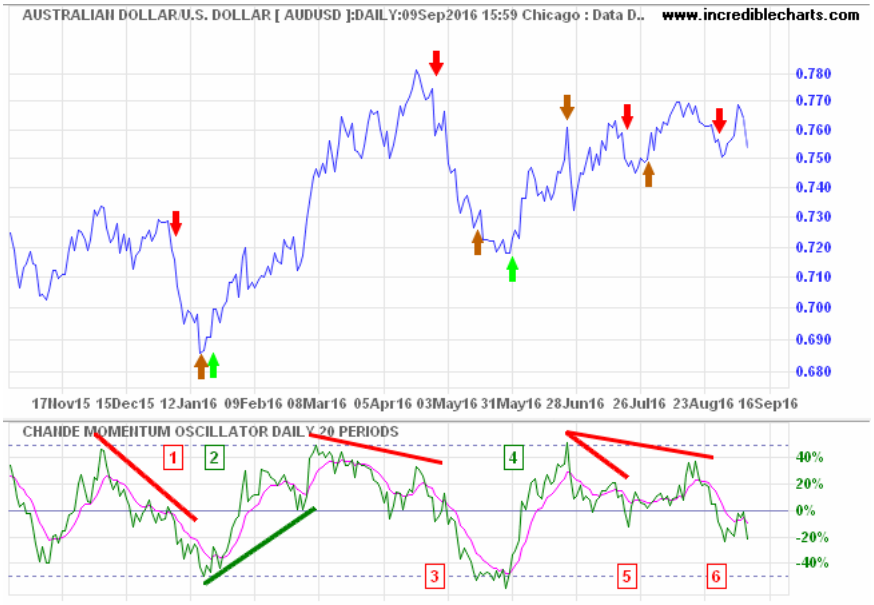

Figura 4.13: Esempio di CMO[13]

4.2.10 DPO

L'indicatore DPO non essendo un indicatore di momentum non genera segnali di acquisto o di vendita in seguito a fasi di ipervenduto o di ipercomprato, come avviene ad esempio con l'oscillatore stocastico o con il MACD, ma si muove in ritardo rispetto ai prezzi proprio per la sua natura atta a "detrendizzare" il grafico. Si propone pertanto, non come sostituto di altri indicatori, ma come strumento destinato ad arricchire l'analisi tecnica. Questo indicatore mette in evidenza gli alti e i bassi livelli di prezzo, con il fine di stimare la durata dei cicli dei prezzi (Lucchetti, 2018). Il calcolo dell'indicatore è dato dalla formula:

[13]Immagine tratta da: *Strategia forex EMA, stocastico lento e RSI* URL: `https://www.manualeforex.it/strategie-forex/ema-stocastico-lento-rsi/`

$$DPO = chiusura - media\_mobile(\frac{x}{2} + 1) \tag{4.14}$$

Il DPO è qundi caratterizzato da una media mobile solitamente di 20 o 30 giorni, traslata di $X/2 + 1$ giorni.

4.2.11 UO

L'ultimate oscillator si computa calcolando il BP (Buying Pressure), ovvero la pressione di acquisto, per determinare la direzione del trend del titolo. Successivamente si procede a calcolare il TR (True Range), il quale aiuta a capire il vero range di un possibile guadagno o di una possibile perdita.

$$BP = close - low\ (minimum\ price) \tag{4.15}$$

$$TR = high - low \tag{4.16}$$

Si procede con il creare una media mobile di tre periodi (7,14 e 28) su cui poi si eseguirà infine una media ponderata delle tre. Si definisce:

$$MA\_7 = \frac{BP\_Sum_7}{TR\_Sum_7} \tag{4.17}$$

$$MA\_14 = \frac{BP\_Sum_{14}}{TR\_Sum_{14}} \tag{4.18}$$

$$MA\_28 = \frac{BP\_Sum_{28}}{TR\_Sum_{28}} \tag{4.19}$$

dove BP_Sum_i e TR_Sum_i sono la somma dei BP e dei TR degli ultimi i giorni. Infine si esegue la media ponderati di questi come:

$$UO = \frac{(4 * MA\_7) + (2 * MA\_14) + MA\_28}{4 + 2 + 1} * 100 \tag{4.20}$$

L'indice assume valori compresi tra 0 e 100, dove si identifica la regione di ipercomprato quella maggiore di 70, e la regione di ipervenduto quella che assume valori minori di 30.

Capitolo 5

Trading system proposto

Prima della computazione dei modelli è stato essenziale stipulare ed organizzare la modellizzazione delle analisi. Come riportato in figura 5.1 il data flow è suddiviso in sei fasi specifiche:

- Acquisizione dei dati
- Preprocessing
- Creazione dei data set di train e test
- Computazione dei modelli
- Generazione delle predizioni
- Attuazione delle strategie di compravendita

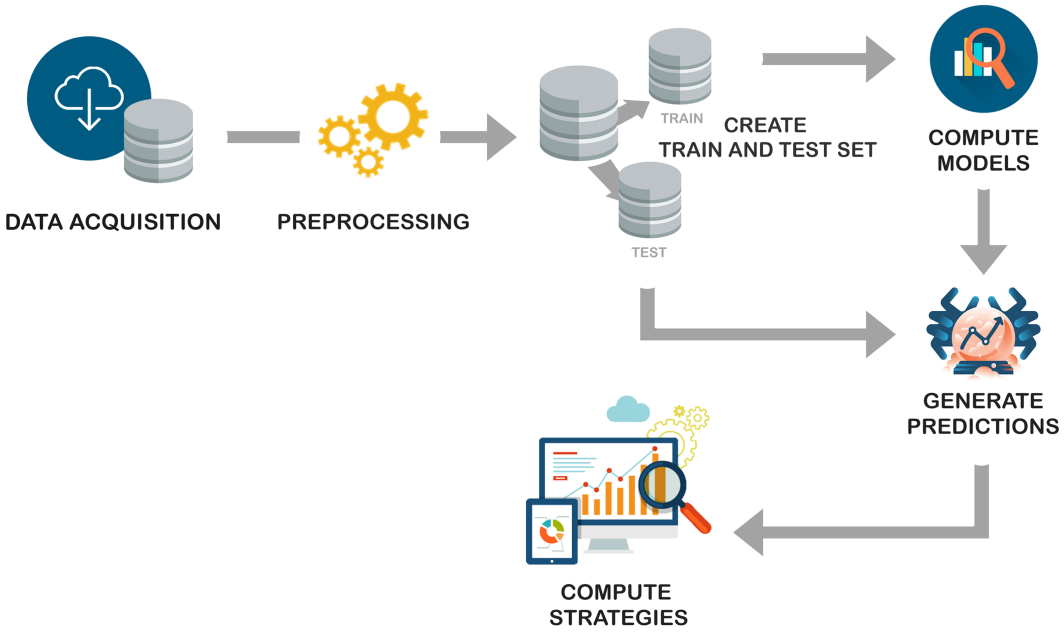

Figura 5.1: Modellizzazione[1]

5.1 Acquisizione dei dati

L'acquisizione dei dati, utilizzati per le sperimentazioni, sono stati ottenuti tramite l'utilizzo di un *crawler*. Un crawler (detto anche web crawler, spider o robot), è un software che analizza i contenuti di una rete (o di un database) in un modo metodico e automatizzato, in genere per conto di un motore di ricerca. Un crawler è un tipo di bot (programma o script) che automatizza delle operazioni, solitamente acquisisce una copia testuale di tutti i documenti visitati e le inserisce in un indice (Wikipedia, 2018a) per facilitare possibili analisi future.
Il crawler ha memorizzato e restituito due tabelle(5.1 e 5.2),una rappresenta le criptovalute, l'altra lo storico dei prezzi per tutte le criptovalute presenti nella prima tabella.

La tabella 5.1 è composta da tre attributi che rappresentano rispettivamente: un id univoco collegato ad ogni criptovaluta, un codice e un nome. La tabella 5.2 è costituita invece da diversi attributi tra cui si può notare: il currency_id

Tabella 5.1: Tabella criptovalute

id	code	name
1	BTC	Bitcoin
2	BCY	BitCrystals
3	BLK	Blackcoin
4	BTCD	BitcoinDark
5	BTS	Bitshares
6	CLAM	Clams
...	...	...

Tabella 5.2: Tabella storico prezzi

id	currency_id	date	price	volume	year	month	day	hour	minute
1	1	27-01-2017 01:40	922.044	126551000	2017	1	27	1	40
2	12	27-01-2017 01:41	10.534.924	8852870	2017	1	27	1	41
3	40	27-01-2017 01:40	0.006334	1363090	2017	1	27	1	40
4	17	27-01-2017 01:41	3.857.187	5275800	2017	1	27	1	41
5	39	27-01-2017 01:41	12.054.794	1979840	2017	1	27	1	41
6	13	27-01-2017 01:41	1.339.868	895580	2017	1	27	1	41
...	...	...	...	...	...	...	...	...	...

(corrispondente all'identificare id nella tabella 5.1), la data (con successivi attributi ridondanti collegati ad essa), il prezzo e il volume.

5.2 Preprocessing

La prima operazione è stata quella di eseguire un merge delle tabelle 5.1 e 5.2 in funzione dell'attributo comune *currency_ id* e *id*. Successivamente si sono eliminati gli attributi poco rilevanti ottenendo come risultato la tabella 5.3.

Per transitare verso un strategia intraday i dati sono stati adattati alla granularità temporale necessaria, portando ad un ora l'unità di tempo che intercorre tra un dato ed il suo successivo(tabella 5.4). Si sono inoltre introdotte le variabili close, open, high e low; ricavate dalla manipolazione dell'attributo price. Infine, dal data base, si sono selezionate le 3 monete di interesse principale: BTC, ETC e XRP.

Durante la fase di preprocessing è possibile procedere attraverso la scelta di un'analisi univariata o multivariata.

Tabella 5.3: Tabella risultante del merge

Code	date	price	volume	year	month	day	hour	minute
BTC	27-01-2017 01:40	922.044	126551000	2017	1	27	1	40
ETH	27-01-2017 01:41	10.534.924	8852870	2017	1	27	1	41
XRP	27-01-2017 01:40	0.006334	1363090	2017	1	27	1	40
LTC	27-01-2017 01:41	3.857.187	5275800	2017	1	27	1	41
XMR	27-01-2017 01:41	12.054.794	1979840	2017	1	27	1	41
ETC	27-01-2017 01:41	1.339.868	895580	2017	1	27	1	41
...	...	...	...	...	...	...	...	...

Tabella 5.4: Tabella intraday

code	date	hour	close	open	low	high
BTC	2017-01-27	1	0.428037745307	0.082329089396	-0.0581187665723	0.470298218881
BTC	2017-01-27	2	0.532346635433	-0.716096160909	-0.965314768575	0.372631858802
BTC	2017-01-27	3	0.237862960213	-0.299246983012	-0.464793746507	0.375976958148
BTC	2017-01-27	4	0.038480144218	0.401292846661	-0.391093861207	0.928580880226
BTC	2017-01-27	5	-0.144545304879	-0.425930246054	-0.891342211538	0.462325204415

5.2.1 Analisi univariata

L'analisi univariata si occupa dello studio di una sola variabile, che in questo contesto riguarda la variabile target delle analisi: il prezzo di chiusura. È stata eseguita una normalizzazione del prezzo di chiusura, trasformando il valore assoluto in un valore percentuale relativo alla seduta precedente. Insieme alla target feature sono stati aggiunti altri attributi, nei quali sono rappresentati i prezzi di chiusura delle ore precedenti. Tali colonne sono denominate *close%(t-i)*, dove i rappresenta l'i-esima ora a ritroso, ottenendo la configurazione mostrata in tabella 5.5.

5.2.2 Analisi multivariata

L'analisi multivariata si occupa dello studio della variazione simultanea di due o più variabili. Durante queste simulazioni si sono introdotti alcuni indici finanziari per aiutare i nostri modelli nel comprendere e anticipare i cambiamenti di trend. Attraverso l'uso della libreria TTR si sono aggiunti i seguenti indici finanziari:

- MACD (Moving Average Convergence/Divergence, paragrafo 4.2.3)
- EMA (Exponential Moving Average, paragrafo 4.2.1)

Tabella 5.5: Tabella risultante: modello intraday

close(t)	close(t-1)	close(t-2)	close(t-3)
0.428037745307	0.082329089396	-0.0581187665723	0.470298218881
0.532346635433	0.428037745307	0.082329089396	-0.0581187665723
0.237862960213	0.532346635433	0.428037745307	0.082329089396
0.038480144218	0.237862960213	0.532346635433	0.428037745307
-0.144545304879	0.038480144218	0.237862960213	0.532346635433

- SMA (Simple Moving Average, paragrafo 4.2.1)
- VWAP (Volume Weighted Average Price, paragrafo 4.2.1)
- RSI (Relative Strength Index, paragrafo 4.2.4)
- MOM (Momentum, paragrafo 4.2.5)
- BBANDS (Bollinger Band, paragrafo 4.2.2)
- STOCH (Stochastic Oscillator, paragrafo 4.2.8)
- CMO (Chande Momentum Oscillator, paragrafo 4.2.9)
- DVI (DV Intermediate Oscillator)
- DPO (De-Trended Price Oscillator, paragrafo 4.2.10)
- UO (Ultimate Oscillator paragrafo 4.2.11)

Nella tabella 5.6 è visibile la modifica descritta. Come si può osservare per ognuna delle medie mobili prese in considerazione (EMA, SMA e VWAP) si è deciso di utilizzare tre diverse configurazioni con periodi differenti pari a 20,50 e 200. Per i rimanenti indici utilizzati si sono mantenute le configurazioni di default.

Come avvenuto nelle simulazioni viste nel paragrafo precedente 5.2.1 si sono poi aggiunti ulteriori attributi rappresentanti i prezzi ed i relativi indici finanziari pervenuti nelle tre ore precedenti. La tabella 5.7 mostra la composizione dei dati dopo la fase di preprocessing.

Tabella 5.6: Tabella intraday con indici finanziari

		financialIndexClose(t)								
close	volume	macd	macdSignal	ema20	ema50	ema200	sma20	sma50	sma200	...
0.32599	0.26625	-0.492	-7.8784	0.18930	0.00804	-0.00687	0.1516	-0.1190	0.00124	...
-0.7156	-194.83	0.3674	6.4012	0.20255	0.02053	-0.00358	0.1334	-0.0995	0.00561	...
-0.0803	-67.366	0.9729	-5.0475	0.11342	-0.0084	-0.01064	0.0958	-0.1232	0.00071	...
-0.9798	-338.36	0.6583	-3.8434	0.09477	-0.0112	-0.01133	0.0790	-0.1407	-0.0029	...
...	...	...	...	...	...	...	...	...	...	...
financialIndexClose(t)										
vwap20	vwap50	vwap200	rsi	mom	bbandsPctB	stochFastK	stochFastD	stochSlowD	cmo	...
0.10077	-0.06237	-0.0033	56.9947	5	1.1413	0.8325	0.9182	0.8841	27.5364	...
0.07358	-0.05253	-0.0003	58.2987	3.26	1.0452	0.8752	0.8869	0.9143	53.0049	...
0.05217	-0.06518	-0.0041	54.3871	-7.18	0.9225	0.7151	0.8076	0.8709	35.7678	...
0.03188	-0.07335	-0.0053	53.9527	-0.8	0.8205	0.6988	0.7630	0.8192	30.0959	...
...	...	...	...	...	...	...	...	...	...	...
financialIndexClose(t)										
DVI	dpoPrice	ultOscil								
0.8619	6.7577	64.3522								
0.9507	8.5569	66.3758								
0.9690	-0.7821	66.410								
0.9404	-2.2951	69.939								
...	...	...								

Tabella 5.7: Tabella risultante: modello intraday con indici finanziari

close(t)	volume(t)	finIndexClose(t)	close(t-1)	volume(t-1)	finIndexClose(t-1)	close(t-2)	...
0.32599	0.26625	...	0.13984	34.971	...	0.31927	...
-0.7156	-194.83	...	0.32599	0.26625	...	0.13984	...
-0.0803	-67.366	...	-0.7156	-194.83	...	0.32599	...
-0.9798	-338.36	...	-0.0803	-67.366	...	-0.7156	...

Feature selection

Durante la fase di preprocessing nell'analisi multivariata è inoltre possibile applicare un processo di feature selection. La *Feature selection* è il processo di riduzione degli attributi dedicato alla individuazione delle caratteristiche maggiormente significative all'interno del data set. La selezione delle caratteristiche diviene un processo indispensabile per ridurre la dimensionalità dei dati ed eliminare tutti gli attributi ridondanti. Questo semplifica le analisi a venire, riducendo la complessità del problema, migliorando quindi l'efficienza dei modelli.

La Feature selection è stata applicata mediante l'uso del pacchetto Boruta disponibile nel repository di CRAN. L'algoritmo Boruta è un wrapper costruito attorno

all'algoritmo di classificazione *Random Forest* implementato nel pacchetto R randomForest (Liaw, Wiener, et al., 2002). Il Random Forest è un algoritmo di classificazione relativamente veloce, il quale può essere eseguito senza ottimizzazione dei parametri, fornendo una stima numerica dell'importanza delle features (Kursa, Rudnicki, et al., 2010). L'algoritmo è stato eseguito separatamente per le tre criptovalute e successivamente sono stati creati dei grafici per poterne analizzare i risultati.

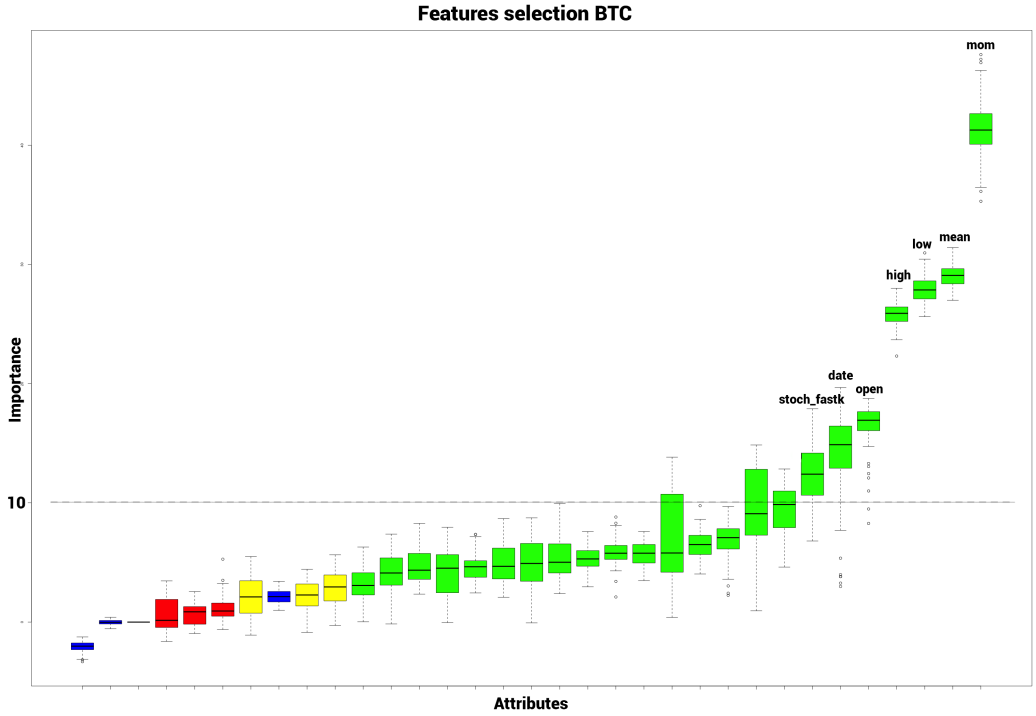

Figura 5.2: Feature selection BTC[2]

Le figure 5.2, 5.3 e 5.4 illustrano il ranking degli attributi generato dalla feature selection. I grafici mostrano come per le tre criptovalute vi siano risultati simili, in particolare mostrano una maggior affinità per i valori di Importance maggiore di 10. Per tale motivo si è deciso di imporre come soglia di selezione tale valore e di includere solo gli attributi con maggior sovrapposizione lungo i tre risultati. Gli attributi che si è deciso di selezionare sono:

- Mom
- Mean
- Low

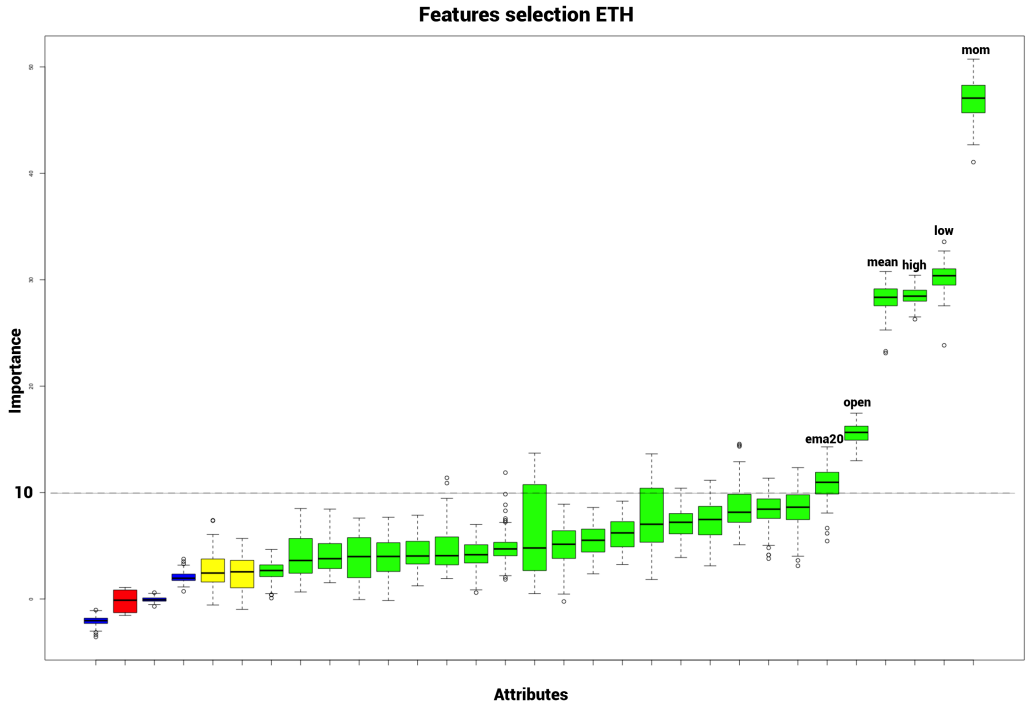

Figura 5.3: Feature selection ETH[3]

- High
- Open
- Stoch_fastk
- Ema20

Tali risultati mostrano come gli indicatori ema20 e stock_fastk, indicatori molto veloci e reattivi, abbiano preso il sopravvento sulle loro versioni più lente. Questi risultati testimoniano la grossa reattività di questo mercato, dove indicatori con periodi più brevi riescono a seguire maggiormente le grosse oscillazioni percepite. Si sono quindi selezionate le features interessate, tabella 5.8.

Come per le precedenti simulazioni si sono aggiunti gli attributi relativi alle tre ore precedenti, tabella 5.9

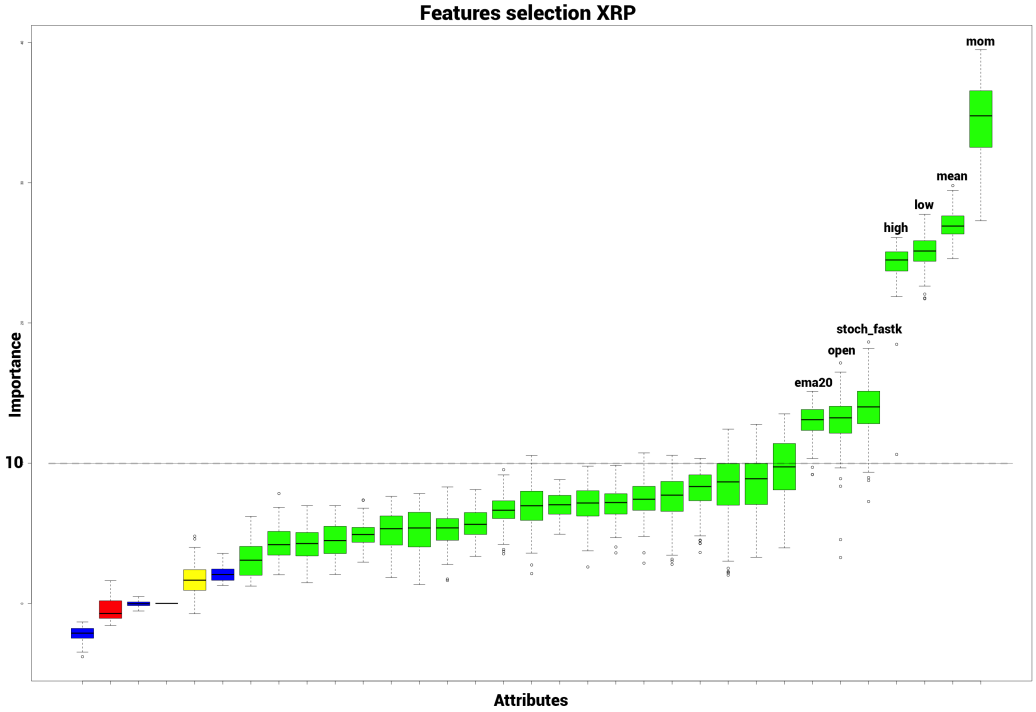

Figura 5.4: Feature selection XRP[4]

Tabella 5.8: Tabella intraday, features selection

prices(t)					financialIndexClose(t)		
close	open	high	low	mean	mom	ema20	stoch_fatsk
-5.4521	0,10234	3.0647	-6.4778	-3.4219	-125.4	-1.03911	0.06141
3.69816	-2.46437	4.9807	-2.5103	1.72303	80.42	-0.63294	0.32555
-0.66518	-0.84877	0.5002	-2.6101	-1.3437	-15	-0.63583	0.28938
3.52632	0.312052	4.9481	0.31205	2.61753	78.99	-0.26353	0.50582
-4.06985	-0.00689	2.1134	-4.0698	-1.2312	-94.38	-0.61694	0.26713

5.3 Creazione dei dataset di train e test

All'interno del processo la realizzazione dei data set di train e test avvengono seguendo due strategie differenti.

La prima strategia consiste nella suddivisione dei dati in due porzioni, in cui rispettivamente vengono collocati il 65% dei dati nel train set ed il rimanente 35% nel test set, ovviamente mantenendo l'ordine temporale dei dati. In particolare il train è costituito da 3005 osservazioni rappresentanti lo storico dei prezzi compresi

Tabella 5.9: Tabella risultante: modello intraday con indici finanziari, features selection

prices(t)	finIndexClose(t)	prices(t-1)	finIndexClose(t-1)	prices(t-2)	finIndexClose(t-2)	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...

tra le date 2017/06/05 e 2017/11/18 mentre il test set e composto da 1619 dati che compongono lo storico dei prezzi dal 2017/11/18 al 2018/1/24. Successivamente i dati di train verranno sfruttati per la realizzazione del modello mentre quelli di test per la generazione delle predizioni, figura 5.8.

Per la realizzazione dei modelli questa strategia risulta semplice, ma si possono intuire già dal principio alcuni limiti che questa strategia impone.

Analizzando i grafici temporali dell'andamento del prezzo delle criptomonete (figure 5.5, 5.7 e 5.6) si nota come tali andamenti attraversino delle chiare fasi distinte con andamenti differenti. Alcune di queste risultano stabili, altre con un andamento rialzista, mentre altre ancora con un andamento ribassista. Questo fenomeno potrebbe portare ad un addestramento non sufficientemente adeguato dei modelli, in quanto la fase di test potrebbe comprendere dati non coerenti con quelli utilizzati nella fase di train.

Date tali premesse si è presa in considerazione una seconda strategia per mitigare il comportamento sopra citato, passando ad una configurazione più dinamica: dalla creazione di una sola porzione di train e test, e quindi della computazione di un unico modello, alla creazione di una finestra scorrevole rappresentata in figura 5.8.

Come descritto in figura vi è la creazione di una finestra scorrevole per ciascuna osservazione presente nella base dati. In questo modo tutti i dati presenti parteciperanno sia alla fase di train che a quella di test per la creazione di un modello che studi specificatamente il comportamento locale del prezzo.

Si sono eseguite varie simulazioni testando diverse dimensioni per la finestra in considerazione. Si sono valutate finestre pari a 1 giorno, 3 giorni, 7 giorni e 14 giorni.

Figura 5.5: BTC candlestick chart[5]

Figura 5.6: ETH candlestick chart[6]

Per poter confrontare le simulazioni ottenute, con le due tipologie di creazioni del test e del train, si è deciso di utilizzare le seguenti configurazioni:

- Partizionamento statico

Figura 5.7: XRP candlestick chart[7]

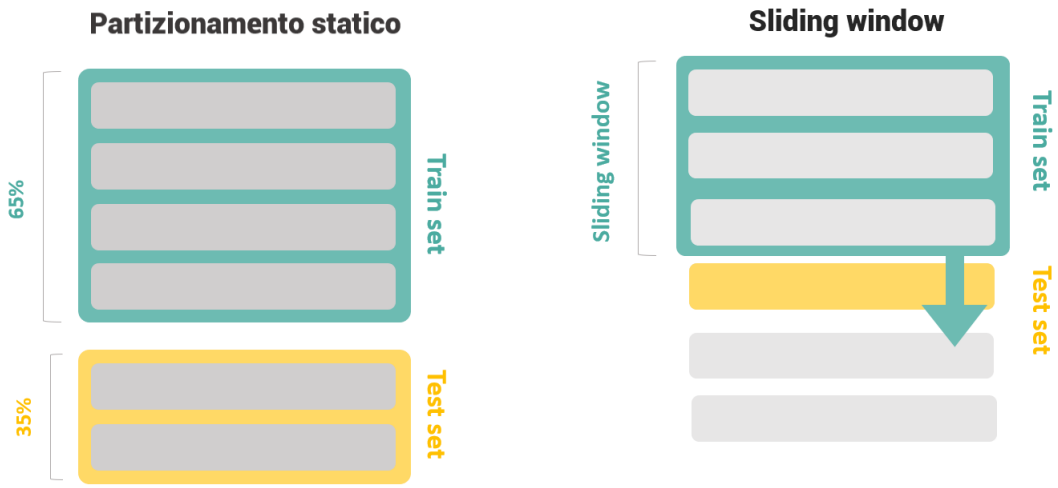

Figura 5.8: Creazione dei dataset di train e test tramite partizionamento statico e finestra scorrevole[8]

- Train: 3005 osservazioni comprese dal 2017/06/05 e 2017/11/18
- Test: 1283 (1619 - 336) osservazioni per il test, dove 336 sono i dati inclusi nella sliding window più grande a 14 giorni, comprese dal 2017/12/02 e 1018/1/24

- Sliding window
  - Train/Test: 1619 osservazioni, le medesime utilizzate nel test per il partizionamento statico, comprese dal 2017/11/18 al 2018/1/24

Attraverso l'utilizzo di questa strategia viene accertato che tutte le simulazioni testino le medesime osservazioni. In conclusione quindi le porzioni di train e test saranno creati con diverse configurazioni. Tali configurazioni si otterranno attraverso l'utilizzo di tutte le possibili combinazioni dei seguenti parametri (figura 5.9):

- Modalità di creazione del train set (figura 5.8):
  - Train 65% e test 35% della base dati
  - Uso di una finestra scorrevole di dimensione 1 giorno
  - Uso di una finestra scorrevole di dimensione 3 giorni
  - Uso di una finestra scorrevole di dimensione 7 giorni
  - Uso di una finestra scorrevole di dimensione 14 giorni
- Numero di prezzi di chiusura, *close(t-i)*, presenti in una sola osservazione (tabella 5.5):
  - Nclose 1: per ogni osservazione vi è il prezzo odierno e quello di ieri
  - Nclose 2: per ogni osservazione vi è il prezzo odierno e quello dei due giorni passati
  - Nclose 3: per ogni osservazione vi è il prezzo odierno e quello dei tre antecedenti

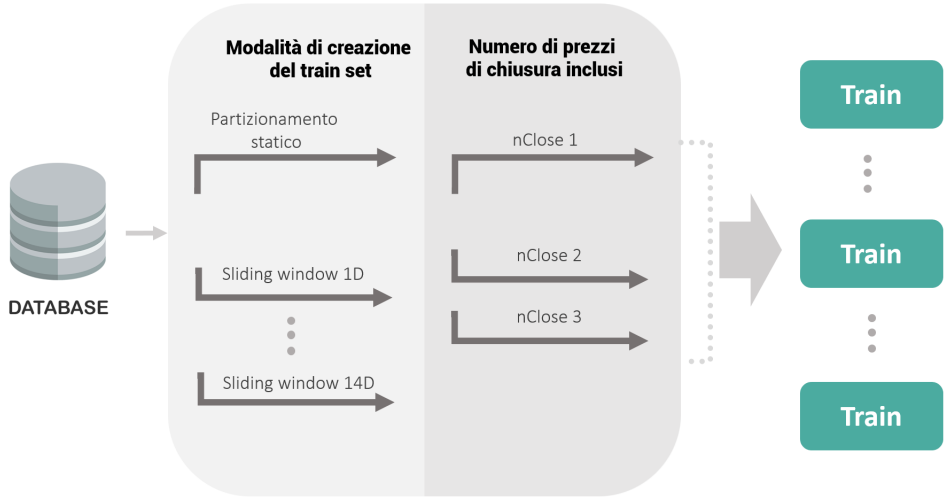

Figura 5.9: Generazione dei modelli in seguito alla variazione dei parametri[9]

5.4 Computazione dei modelli

Una volta stabilita la porzione di train all'interno della base dati questa verrà inviata al modulo successivo il quale provvederà alla computazione dei modelli. Per assicurarsi una buona visione generale, si è posta attenzione a testare adeguatamente sia i modelli di regressione che quelli di classificazione. In particolare i modelli di regressione usati sono stati:

- Support Vector Machine, nelle due configurazioni lineare e polinomiale
- Reti neurali
- Alberi di decisione
- Regressione lineare

mentre i modelli di classificazione utilizzati sono stati:

- Support Vector Machine, nelle due configurazioni lineare e polinomiale
- Naive Bayes
- Alberi di decisione

- Reti Bayesiane
- K-nearest neighbors

Come visibile in figura 5.10 per i modelli di regressione basterà inviare la porzione di train interessata al modulo successivo per generare il corrispettivo modello di regressione. Per i modelli di classificazione viè una generazione dei modelli discordante, derivante dalla differente natura dei modelli. Essendo dei modelli di classificazione necessitano un dataset già etichettato. Per tale motivo, in base ai prezzi di chiusura, è stato necessario generare anticipatamente i segnali di compravendita BUY, SELL e HOLD. E' stato quindi necessario aggiungere un ulteriore passaggio in cui sono state create ulteriori sette porzioni di train. Ogni porzione di train è stata ottenuta applicando un soglia differente per la generazione dei segnali. La soglia (thr) è un valore compreso tra 0 e 0.6 per il quale:

- Se chiusura $>$ thr si avrà la generazione di un segnale di BUY
- Se chiusura $<$ - thr si avrà la generazione di un segnale di SELL
- Se chiusura $<$ |thr| si avrà la generazione di un segnale di HOLD

Una volta etichettato il train set, questo verrà poi utilizzato per la creazione del modello di classificazione.

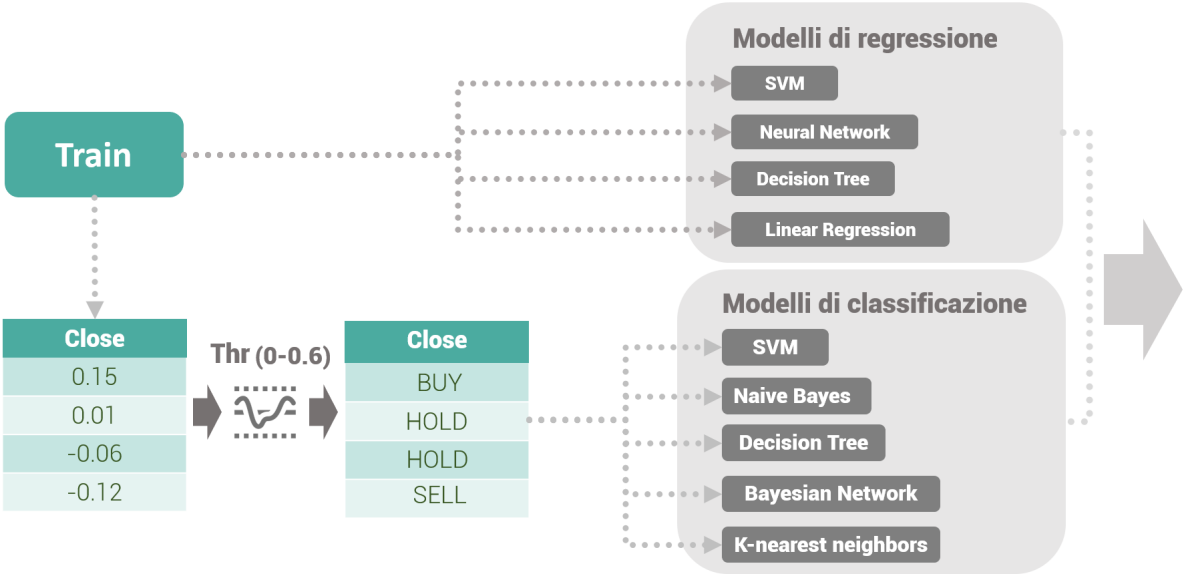

Figura 5.10: Computazione dei modelli[10]

5.5 Generazione delle predizioni

Una volta terminata la fase di computazione dei modelli, vi è quella della generazione delle predizioni. In particolare per ogni modelli generato da una porzione di train, viene assegnato la corrispettiva porzione di test, con l'obiettivo di concepire le predizioni su quest'ultima. Come si può osservare dalla figura 5.11 vi è una sostanziale differenza nella produzione dei segnali, derivante dalla modalità di creazione dei modelli vista nella fase precedente. Si può osservare come per i modelli di classificazione vi siano in uscita già i segnali desiderati, mentre nei modelli di regressione vi sia invece la predizione sulla variazione del prezzo. Per tale motivo è stato necessario applicare una soglia (thr) alla predizione, in base alla quale è stata calcolata la generazione dei segnali.

Le tabelle risultanti si possono osservare in tabella 5.10 e 5.11.

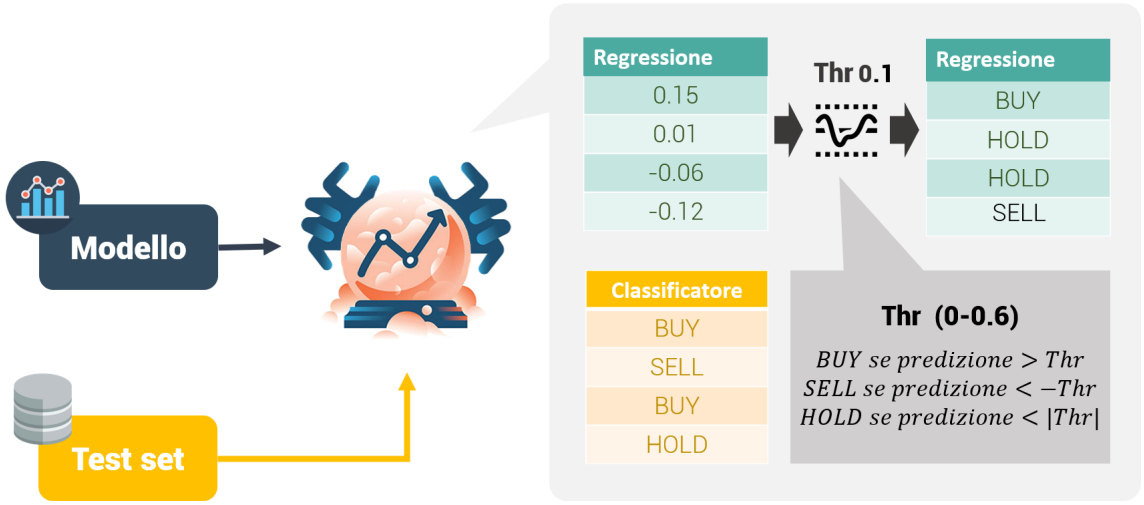

Figura 5.11: Generazione delle predizioni[11]

Tabella 5.10: Tabella risultante: modelli di classificazione

date	hour	signThr0	signThr0.1	signThr0.2	signThr0.3	signThr0.4	signThr0.5	signThr0.6
03-12-2017	1	1	1	1	0	0	0	0
03-12-2017	2	1	1	1	0	0	0	0
03-12-2017	3	1	1	1	0	0	0	0
03-12-2017	4	1	1	1	0	0	0	0
03-12-2017	5	1	1	1	0	0	0	0

5.6 Attuazione delle strategia di compravendita

Per ognuno dei modelli concepiti vengono generati 7 flussi di segnali, derivanti dalle diverse soglie applicate alle predizioni. Per ognuno dei flussi generati, in quest'ultima fase, occorre interpretare la sequenza dei segnali al fine di generare una strategia di compravendita. In particolare si sono sviluppate tre differenti strategie:

- Keep Trend
- Stop Loss
- Trailing stop

L'idea alla base di ciascuna strategia impiegata è quella di limitare le perdite derivate da più fattori. Le perdite su cui si è posta maggiore attenzione sono:

Tabella 5.11: Tabella risultante: modelli di regressione

date	hour	prediction	signThr0	signThr0.1	signThr0.2	signThr0.3	signThr0.4	signThr0.5	signThr0.6
03/12/2017	1	0.008416416	1	0	0	0	0	0	0
03/12/2017	2	0.126670864	1	1	0	0	0	0	0
03/12/2017	3	-0.013150547	-1	0	0	0	0	0	0
03/12/2017	4	0.023160	1	0	0	0	0	0	0
03/12/2017	5	0.005467847	1	0	0	0	0	0	0

- Limitazioni delle perdite in caso vi si presenti una brusca variazione al ribasso della moneta interessata
- Perdite derivanti dai costi di transazione

In particolare non si è sfruttata nessuna tecnica di Take profit (ovvero di ordini di chiusura al rialzo) per non essere limitati nella percezione di grosse variazioni al rialzo osservate nei grafici temporali 5.5, 5.6 e 5.7.

5.6.1 Keep trend

Questa è sicuramente la strategia più semplice, che prefigge come unico obiettivo quello di limitare i costi di transazione. La strategia in questione evita di aprire e chiudere la posizione allo scadere di ogni ora. La chiusura della posizione avverrà solamente al verificarsi delle seguenti condizioni:

- Predizione errata: per esempio si apre una posizione nella speranza che il prezzo salga e invece si percepisce una variazione al ribasso
- Predizione di un cambio trend: per esempio si è all'interno di un trend rialzista e la predizione segnala una variazione al ribasso

5.6.2 Stop Loss

Lo Stop Loss invece, al contrario del Keep Trend, non si pone solo l'obiettivo di ridurre i costi di transazione, ma anche di proporre una strategia finalizzata al contenimento delle perdite, nel caso in cui l'andamento si discosti fortemente dalle aspettative. Questo comporta l'apertura di un ordine ad un prezzo prefissato, che nel caso venga raggiunto dal trend, implica la chiusura della posizione e quindi una limitazione delle perdite.

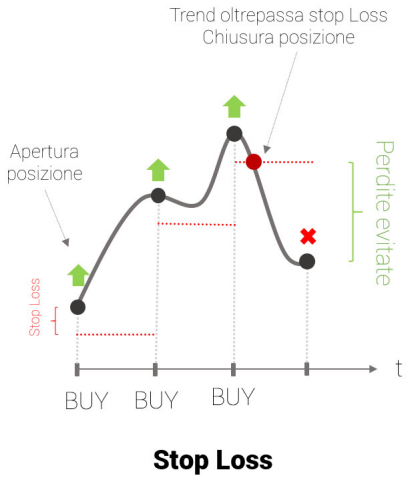

Figura 5.12: Stop Loss[12]

Si può osservarne un esempio in figura 5.12, nella quale lo Stop Loss ci permette di chiudere in tempo la posizione ed evitare perdite più onerose.

5.6.3 Trailing stop

Il trailing stop invece fissa l'obiettivo di limitare al massimo le perdite cercando, in caso di sbagliata predizione, di assicurarsi comunque una situazione di non perdita.

Figura 5.13: Trailing stop[13]

[12]Immagine tratta da: `https://tradingqna.com/t/is-there-any-chance-to-modify-stop-loss-for-completed-orders/2085`

Come si può osservare in figura 5.13 lo Stop Loss viene calcolato in base alla combinazione di due fattori, lo Stop Loss statico ed i costi di commissione. Lo Stop Loss viene quindi calcolato nel seguente modo:

$$Trailingstop = max[StaticStopLoss, (previusPrice + transactionfee)] \quad (5.1)$$

Capitolo 6

Risultati

6.1 Software

Tutte le sperimentazioni effettuate ed i grafici prodotti sono stati creati utilizzando come software R. R è un linguaggio di programmazione e un ambiente di sviluppo specifico per l'analisi statistica dei dati. Venne scritto inizialmente (circa nel 1993) dal matematico e statistico canadese Robert Gentleman, e dallo statistico neozelandese Ross Ihaka. R è un software libero in quanto viene distribuito con la licenza GNU GPL, ed è disponibile per diversi sistemi operativi (ad esempio Unix, GNU/Linux, macOS, Microsoft Windows). Il suo linguaggio orientato agli oggetti deriva direttamente dal pacchetto S distribuito con una licenza non open source e sviluppato da John Chambers e altri presso i Bell Laboratories. (Wikipedia, 2018b). R è una suite integrata di funzioni software per la manipolazione dei dati, il calcolo e la visualizzazione grafica. Include

- Un'efficace sistema di gestione e stoccaggio dei dati,
- Una suite di operatori per calcoli su array, in particolare matrici,
- Una raccolta ampia, coerente e integrata di strumenti intermedi per l'analisi dei dati,
- Strutture grafiche per l'analisi dei dati e visualizzazione su schermo o su supporto cartaceo e

- Un linguaggio di programmazione ben sviluppato, semplice ed efficace che include condizionali, cicli, funzioni ricorsive definite dall'utente e strutture di input e output.

R, come S, è progettato basatosi su un vero linguaggio informatico e consente agli utenti di aggiungere funzionalità definendone di nuove. Gran parte del sistema è essa stessa scritta in linguaggio R, che rende ulteriormente semplice agli utenti seguire le scelte algoritmiche fatte (R, 2018).

In definitiva la popolarità di R è dovuta quindi a due fattori principali:

- In primo luogo alla possibilità di accedere ad esso in maniera completamente gratuita
- In secondo luogo alla possibilità di estendere le sue funzionalità attraverso un insieme di moduli, sempre usufruibili con licenza GLP, disponibili attraverso il repository CRAN.

Per agevolare ulteriormente l'interazione con le funzionalità di R, si è deciso di non interagire direttamente con R, ma di utilizzare invece RStudio, un ambiente di sviluppo integrato open-source, sviluppato appositamente per R.

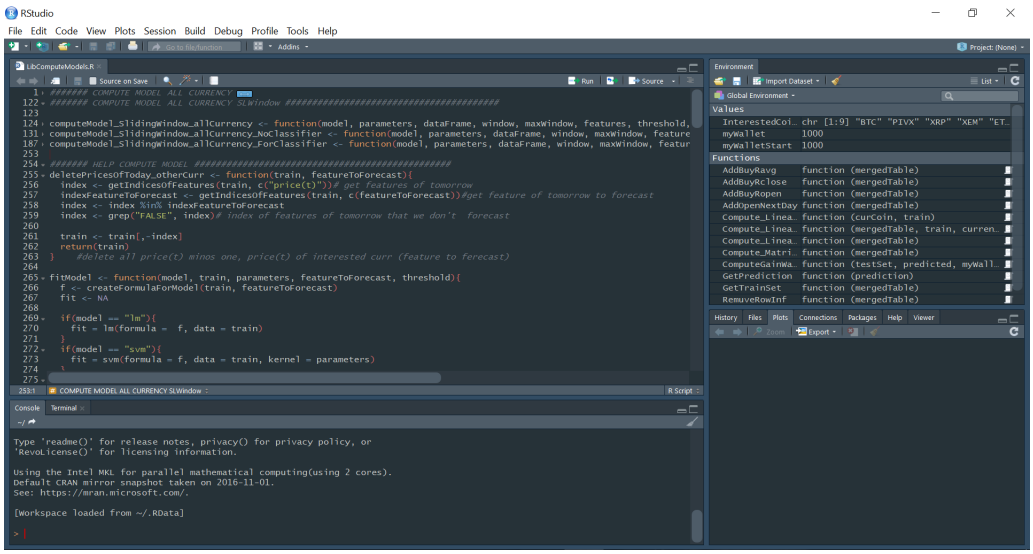

Figura 6.1: RStudio[1]

6.2 Hardware

Per l'esecuzione delle sperimentazioni sono state utilizzate prevalentemente due configurazioni di macchine:

- Il computer principale: un IMac con processore Intel Core I5 a 3,5 GHz con 8 Gb di memoria RAM DDR3 a 1600 MHz, su cui sono state eseguite quasi il totale delle sperimentazioni.
- Il computer secondario: un Dell XPS 13 con processore Intel Core I5 a 2,2 GHz con 8 Gb di memoria RAM DDR3RS, attraverso il quale è stato sviluppato il codice ed eseguiti task secondari per la parallelizzazione del lavoro.

Fin dalle prime sperimentazioni è stato chiaro quale fosse il vero limite delle macchine impiegate: la RAM. Infatti è stato necessario successivamente un upgrade dell'hardware del IMac da 8 a 16 Gb di Ram DDR3 per poter eseguire più rapidamente i task.

In questo capitolo verranno discussi i risultati ottenuti tramite le sperimentazione in funzione dei guadagni ottenuti dalle strategie implementate. In particolare i guadagni saranno rappresentati come un valore percentuale relativo all'investimento iniziale. Si considereranno costi di commissione con una valore pari al 0,01% e saranno inoltre evidenziati gli effetti derivanti dalla variazione dei seguenti parametri:

- Dalla procedura di realizzazione del train e del test set, il quale implica la variazione di
  - *NClose*, descritto nel paragrafo 5.3
  - *Modalità di creazione del train set*, descritto nel paragrafo 5.3
- Dalla strategia di compravendita utilizzata, la quale dipende da:
  - *Thr*, descritto nel paragrafo 5.4 e 5.5
  - *StopLoss*, descritto nel paragrafo 5.6
  - *Strategy*, descritto nel paragrafo 5.6
- Dal modello impiegato per la generazione delle predizioni, descritto nel paragrafo 5.4

6.3 Simulazioni di riferimento

I risultati ottenuti saranno inoltre comparati con delle simulazioni di riferimento, volte ad offrire un metodo di giudizio sui guadagni ottenuti dagli esperimenti. Le simulazioni di riferimento subiranno il medesimo processo a cui sono state sottoposte le altre, ottenendo quindi diversi risultati al variare dei parametri descritti sopra. Le simulazioni di confronto sono le seguenti:

- *Buy & Hold*: simulazione volta a calcolare il guadagno percepito nel caso in cui si compri il primo giorno e si venda tutto l'ultimo giorno dell'intervallo considerato. Questa è una delle sperimentazioni di riferimento, perchè offre il minimo costo di spese di transizione ed il più basso tasso di rischio.
- *Last hour*: ripropone lo stesso segnale percepito l'ora precedente presupponendo che il trend in atto continui senza variazioni. Propone un numero di compravendite adeguate e quindi costi di commissioni di riferimento.

- *Random*: vi è la generazione di segnali di compravendita del tutto casuali. Si propone questo caso, in quanto propone una strategia totalmente randomica, priva di qualsiasi strumento per il supporto decisionale.

6.4 Risultati delle simulazioni di trading intraday

Nel grafico 6.2 si possono osservare i risultati ottenuti dalle sperimentazioni riguardanti il Bitcoin. In tale grafico si può osservare la totalità dei guadagni pervenuti al variare dei parametri descritti precedentemente nel capitolo 5.
Nelle tabelle 6.1, 6.2 e 6.3 sono illustrate le configurazioni ottenute rispettivamente per ogni moneta. Si possono osservare diverse similitudini tra le configurazioni in analisi. Di seguito verranno illustrati per ogni parametro preso in considerazione l'influenza che questo ha avuto su i corrispettivi risultati:

- Model: si può analizzare come per tutte le criptomonete i modelli che raggiungo migliori risultati siano il Naive Bayes, la Bayesian network, la Support Vector Machine ed il Decision tree, con una dominanza quindi dei modelli di classificazione.

- Strategy: in tutte le simulazione la strategia vincente risulta essere il trailing stop

- NPrices: l'inclusione del solo prezzo antecedente (nPrices = 1) ha portato in tutte le monete ad ottenere la configurazione migliore, ma non vi si può riconoscere un pattern lungo tutte le simulazioni. In generale nPrices varia assumendo in maniera del tutto casuale il range dei valori possibili.

- NWinodws: il parametro nWindows varia assumendo valori compresi tra 0 e 14, dove 0 corrisponde all'assenza della sliding window e dell'utilizzo del partizionamento statico dei dati, mentre valori maggiori di 0, implicano l'utilizzo di una finestra scorrevole, di dimensione pari a quella specificata. In questo caso è chiaro che l'utilizzo della sliding window ha portato alla rilevazione di maggiori introiti, specialmente con l'impiego di sliding window con dimensioni pari a 7 e 14. Il fenomeno descritto si verifica maggiormente nelle criptovalute ETH e XRP dove nei grafici 6.3 e 6.4 vi è quasi la completa assenza di nWindows pari a 0. Il grafico 6.2 mostra come l'utilizzo del partizionamento statico ha invece portato a maggiori risultati sul BTC.

- Thr: la soglia thr è il parametro dedicato alla generazioni dei segnali e varia assumendo valori tra 0 e 0.6. Dai grafici sottostanti vi è la predominanza di valori compresi tra 0 e 0.3. Questo fenomeno rimarca il comportamento visto nei precedenti capitoli dove si è evidenziata la presenza di un mercato estremamente volatile, in cui è necessaria una strategia che riesca a seguire improvvise variazioni di trend. Si è quindi confermato come valori di thr bassi aiutino i modelli nel prevedere questi fenomeni.

- Stop Loss: lo stop loss influisce invece sul contenimento delle perdite nel caso in cui il modello generi una previsione scorretta. Il parametro in questione assume valori compresi tra 1 e 4. Questo è l'unico valore che lungo tutte le sperimentazione ha riscontrato il medesimo comportamento in tutte e tre le criptovalute. In particolare nei grafici 6.2, 6.3 e 6.4 vi è la sola presenza dello stop loss con valori assunti pari a 1. Probabilmente questo comportamento è dovuto alla grossa instabilità del mercato dove quindi stoploss ridotti aiutano maggiormente a prevedere e contenere grosse perdite.

Tabella 6.1: Risultati sperimentazioni BTC trading intraday

Classification model						
Gain	Model	Strategy	nPrices	nWindows	Thr	StopLoss
141.10	NaiveBayes	TrailingStop	1	14	0,2	1
140.34	BayesianNetwork	TrailingStop	2	14	0,3	1
138.83	SvmPolynomial	TrailingStop	3	0	0,2	1
120.75	DecisionTree	TrailingStop	2	0	0.2	1
99.03	Knn	TrailingStop	3	0	0.2	1
Regression model						
Gain	Model	Strategy	nPrices	nWindows	Thr	StopLoss
120.75	NeuralNetwork	TrailingStop	2	0	0	1
120.75	DecisionTree	TrailingStop	1	0	0	1
81.07	SvmPolynomial	TrailingStop	1	0	0	1
60.48	SvmLinear	TrailingStop	2	0	0	1
51.83	LinearRegression	TrailingStop	3	14	0	1

Nella tabella 6.4 sono illustrati i risultati ottenuti comparati con gli esiti raggiunti tramite le simulazione di confronto. Si può rilevare come per tutte le monete prese in considerazione, ci sia un notevole miglioramento dei guadagni rispetto alle

Tabella 6.2: Risultati sperimentazioni ETH trading intraday

Classification model						
Gain	Model	Strategy	nPrices	nWindows	Thr	StopLoss
263.03	BayesianNetwork	TrailingStop	1	7	0,2	1
260.82	NaiveBayes	TrailingStop	1	14	0,2	1
254.68	SvmPolynomial	TrailingStop	3	14	0	1
244.22	SvmLinear	TrailingStop	1	14	0.3	1
146.37	Knn	TrailingStop	3	7	0.3	1
140.63	DecisionTree	TrailingStop	3	14	0.3	1
Regression model						
246.14	SvmPolynomial	TrailingStop	1	14	0	1
169.34	SvmLinear	TrailingStop	1	14	0	1
163.73	NeuralNetwork	TrailingStop	1	3	0	1
157.56	LinearRegression	TrailingStop	1	1	0	1
133.87	DecisionTree	TrailingStop	2	1	0.3	1

Tabella 6.3: Risultati sperimentazioni XRP trading intraday

Classification model						
Gain	Model	Strategy	nPrices	nWindows	Thr	StopLoss
560.91	BayesianNetwork	TrailingStop	1	14	0,2	1
552.34	NaiveBayes	TrailingStop	1	14	0,2	1
550.16	SvmLinear	TrailingStop	1	14	0,2	1
535.82	SvmPolynomial	TrailingStop	1	14	0.2	1
464.85	DecisionTree	TrailingStop	3	14	0.1	1
421.25	knn	TrailingStop	3	1	0	1
Regression model						
547.16	SvmPolynomial	TrailingStop	2	7	0	1
532.10	LinearRegression	TrailingStop	1	7	0	1
488.16	DecisionTree	TrailingStop	1	14	0	1
478.95	NeuralNetwork	TrailingStop	2	7	0	1
477.22	SvmLinear	TrailingStop	3	14	0	1

strategie di riferimento. In particolare si percepiscono notevoli aumenti di guadagni anche rispetto alla strategia Buy&Hold, rimarcando la possibilità di un considerevole aumento dei guadagni nonostante un consistente aumento dei rischi .

Si può inoltre evincere come la strategia random e lasthour restituiscano dei risultati pessimi confermando l'importanza della necessità della presenza di un strumento per il supporto decisionale. In particolare si sono eseguite all'incirca 36000 simulazioni sulla strategia Random, e solo l'1% ha raggiunto introiti positivi, e di questi solo alcuni hanno raggiunto buoni risultati. Il modelli creati hanno comunque abbondantemente superato i migliori risultati ottenuti dalla strategia randomica.

Tabella 6.4: Risultati sperimentazioni trading intraday

Currency	Gain Model	Gain Buy&Hold	Gain LastHour	Gain Random
BTC	141.10	2.36	-121.40	99.86/-358.48
ETH	263.03	31.31	-137.16	154.27/-373.26
XRP	560.91	423.01	-35.10	426.42/-479.77

6.5 Risultati delle simulazioni di trading intraday con indici finanziari

L'introduzione degli indici finanziari ha notevolmente aumentato gli attributi inclusi nella simulazione passando da una dimensione massima del train pari a 4 attributi (5.5) ad un massimo di 92 attributi (5.7), determinato dal numero di nPrices inclusi nella simulazione. Per tale motivo le simulazioni in questione hanno impegnato un tempo notevolmente maggiore passando da un tempo medio di 48 ore per le simulazioni intraday, a un tempo di 480 ore invece per la nuova configurazione.

Come per le precedenti simulazioni i grafici 6.5, 6.6 e 6.7 illustrano i risultati ottenuti dalle sperimentazioni. Gli esiti ottenuti risultano essere differenti, facendo emergere configurazioni discordanti rispetto a quelle comparse nella capitolo predente 6.4. Di seguito si analizzeranno gli i risultati presenti nelle tabelle 6.5, 6.6 e 6.7 in funzione dei parametri utilizzati:

- Model: Le tabelle mostrano come i modelli più performanti sono la regressione lineare e la support vector machine lineare nelle due configurazioni di regressione e di classificazione. La nuova conformazione del train set ha quindi supportato modelli lineari ed ha invece ostacolato modelli più complessi come la rete neurale o i modelli di classificazione emersi nelle precedenti analisi come la Bayesian Network ed il Naive Bayes.

- Strategy: contrariamente a quanto avvenuto nelle sperimentazioni intraday del capitolo precedente, la strategia vincente non risulta più essere il trailing stop ma il keep trend. Probabilmente in queste simulazioni lo stop loss evidenzia dei limiti sui guadagni, imponendo forzatamente la chiusura delle propria posizione quando si percepisce un grossa oscillazione. Il keep trend invece non imponendo

nessun ordine di chiusura della posizione fino alla successiva sessione, permette al trend di riguadagnare punti implementando quindi una maggiore flessibilità.

- NPrices: contrariamente a quanto avvenuto nella precedente simulazione le migliori configurazioni sono state ottenute con nPrices uguale a 3. Tale valore rimane costante per tutte le configurazioni presenti nei grafici 6.5, 6.6 e 6.7.

- NWindows: per tutte e tre le criptovalute emerge essere più promettente la configurazione con un partizionamento statico dei dati, inversamente a quanto accaduto precedentemente.

- Thr: la soglia è l'unico parametro confrontabile a quanto avvenuto nelle sperimentazioni precedenti, confermando la presenza di soglie relativamente basse.

- Stop Loss: impiegando la strategia keep trend lo stop loss non assume nessuna rilevanza in quanto non è utile nelle strategie di compravendita.

Infine nella tabella 6.8 si sono messi a confronto i risultati pervenuti dalle due simulazioni fatte in funzione di ogni moneta di interesse. Si può osservare in generale come con l'introduzione degli indici finanziari ci sia un notevole aumento dei profitti.

Tabella 6.5: Risultati sperimentazioni BTC trading intraday con indici finanziari

Classification model						
Gain	Model	Strategy	nPrices	nWindows	Thr	StopLoss
1006.20	SvmLinear	TrailingStop	3	0	0,1	1
830.71	SvmPolynomial	StopLoss	3	0	0	4
775.96	DecisionTree	KeepTrend	1	0	0.5	1
703.33	knn	KeepTrend	1	0	0	1
348.55	NaiveBayes	StopLoss	1	14	0	1
121.88	BayesianNetwork	TrailingStop	1	0	0	1
Regression model						
1040.75	LinearRegression	KeepTrend	3	0	0,1	3
1014.36	SvmLinear	KeepTrend	3	0	0,1	1
889.63	NeuralNetwork	KeepTrend	2	0	0.4	1
770.78	DecisionTree	KeepTrend	1	0	0.4	1
767.90	SvmPolynomial	KeepTrend	1	0	0.1	1

Tabella 6.6: Risultati sperimentazioni ETH trading intraday con indici finanziari

Classification model						
Gain	Model	Strategy	nPrices	nWindows	Thr	StopLoss
1051.22	SvmLinear	KeepTrend	3	0	0	1
1025.8	SvmPolynomial	KeepTrend	3	0	0	1
923.41	DecisionTree	KeepTrend	1	0	0.4	1
758.69	Knn	TrailingStop	2	0	0	1
358.50	NaiveBayes	TrailingStop	2	0	0.4	1
79.41	BayesianNetwork	TrailingStop	1	0	0	1
Regression model						
1122.97	SvmLinear	KeepTrend	3	0	0,1	1
1110.17	LinearRegression	KeepTrend	3	0	0,2	1
1049.27	SvmPolynomial	KeepTrend	3	0	0	1
935.69	DecisionTree	KeepTrend	3	0	0.3	1
848.26	NeuralNetwork	KeepTrend	3	0	0.2	1

Tabella 6.7: Risultati sperimentazioni XRP trading intraday con indici finanziari

Classification model						
Gain	Model	Strategy	nPrices	nWindows	Thr	StopLoss
1610.37	SvmLinear	KeepTrend	3	0	0.1	1
1605.54	DecisionTree	KeepTrend	3	0	0.5	1
1536.14	svmPolynomial	KeepTrend	1	0	0.5	1
787.19	NaiveBayes	StopLoss	1	14	0	1
484.41	Knn	StopLoss	1	0	0	1
334.76	BayesianNetwork	TrailingStop	1	0	0	1
Regression model						
1677.41	LinearRegression	KeepTrend	3	0	0	1
1625.76	SvmPolynomial	KeepTrend	3	0	0,2	1
1621.89	SvmLinear	KeepTrend	3	0	0,5	1
1597.17	DecisionTree	KeepTrend	1	0	0	1
517.78	NeuralNetwork	TrailingStop	2	0	0	1

6.5.1 Feature selection

La Feature selection ha permesso una notevole riduzione della dimensionalità passando da un numero di 92 attributi presenti in tabella 5.7 ad un numero di 32 attributi presenti in tabella 5.9. Si è quindi ridotta la dimensionalità dei dati di circa un terzo rispetto alla simulazione precedente, ma tale riduzione non ha però riscontrato il medesimo impatto sul tempo di computazione del modello. La creazione dei modelli ha impiegato circa 384 ore risparmiando solamente un centinaio di ore sulla precedente sperimentazione. Le figure 6.8, 6.9 e 6.10 mostrano i risultati

Tabella 6.8: Risultati sperimentazioni, trading intrady vs trading intraday con indici finanziari

	Trading intraday		Trading intraday with financial indices	
Code	Gain	model	Gain	model
BTC	141.10	NayveBayes	1040.75	LinearRegression
ETH	263.03	BayesianNetwork	1122.97	SvmLinearRegression
XRP	560.91	BayesianNetwork	1677.41	LinearRegression

ottenuti dalle sperimentazioni. Come fatto per le precedenti simulazioni le tabelle 6.9, 6.10 e 6.9 mostrano, per ogni moneta, i guadagni in funzione dei modelli usati. Di seguito si analizzeranno i risultati in funzione dei seguenti parametri:

- Model: i modelli che ottengono il miglior rendimento risultato essere la regressione lineare e la support vector machine lineare confermando anche in questo caso che l'utilizzo di modelli lineari realizzano superiori introiti. Diversamente da quanto avvenuto nel paragrafo 6.5 ci sono buone perfomance anche per modelli più complessi come la rete neurale. Questo fenomeno può essere ricondotto alla riduzione della dimensionalità dei dati derivanti dall'uso della features selection.

- Strategy: la strategia trailing stop sembra essere la strategia più promettente anche se in alcuni casi si può osservare come la strategia vincente sia il keep trend

- NPrices: i grafici 6.8, 6.9 e 6.10 mostrano come le configurazioni migliori si ottengano con nPrices uguale a 1. In particolare tale comportamento è rilevabile in tutte e tre le criptovalute.

- NWindow: in generale le performance migliori si ottengono con un valore di nWindow maggiore di zero, cioè con la presenza della Sliding Window. Questo non è vero per alcuni casi specifici rilevati nel BTC e nell'Ethereum, dove si nota che utilizzando il partizionamento statico si riescono ad ottenere risultati migliori, rispetto all'utilizzo della finestra mobile.

- Thr: non si riesce in questo caso a rilevare dei pattern significativi utili all'analisi dei risultati.

- StopLoss: come per il Thr sono presenti delle importanti oscillazioni nei grafici illustrati, i quali non permettono di rilevare dei pattern significativi.

Tabella 6.9: Risultati sperimentazioni BTC trading intraday con indici finanziari e features selection

Classification model						
Gain	Model	Strategy	nPrices	nWindows	Thr	StopLoss
551.91	SvmLinear	KeepTrend	3	14	0	0,5
473.44	DecisionTree	TrailingStop	1	0	0.1	0.5
390.71	SvmPolynomial	TrailingStop	1	0	0.2	0.5
246.14	BayesianNetwork	TrailingStop	1	0	0	0.5
219.50	Knn	TrailingStop	1	0	0.2	0.5
140.67	NaiveBayes	TrailingStop	1	0	0	0.5
Regression model						
582.56	SvmLinear	TrailingStop	1	0	0	0,5
542.41	LinearRegression	KeepTrend	1	14	0,2	0,5
486.41	NeuralNetwork	TrailingStop	1	0	0	0.5
458.11	SvmPolynomial	TrailingStop	1	0	0	0.5
403.31	DecisionTree	TrailingStop	1	0	0	0.5

Tabella 6.10: Risultati sperimentazioni ETH trading intraday con indici finanziari e features selection

Classification model						
Gain	Model	Strategy	nPrices	nWindows	Thr	StopLoss
621.37	SvmLinear	KeepTrend	2	14	0.2	0.5
434.91	DecisionTree	TrailingStop	1	0	0.3	0.5
373.64	Knn	TrailingStop	1	0	0.1	0.5
289.71	SvmPolynomial	TrailingStop	2	0	0	0.5
207.37	BayesianNetwork	TrailingStop	1	0	0	0.5
186.36	NaiveBayes	StopLoss	1	14	0.5	1
Regression model						
733.85	SvmLinear	TrailingStop	1	0	0	0.5
699.40	LinearRegression	StopLoss	1	14	0	1
670.86	NeuralNetwork	StopLoss	1	14	0	1
466.72	SvmPolynomial	KeepTrend	2	14	0	0.5
336.21	DecisionTree	TrailingStop	2	0	0	0.5

La tabella 6.12 mette a confronto le simulazioni svolte. Si può in generale osservare come l'introduzione degli indici finanziari abbia notevolmente migliorato i guadagni. La features selection non ha invece portato ai risultati sperati, in quanto

Tabella 6.11: Risultati sperimentazioni XRP trading intraday con indici finanziari e features selection

Classification model						
Gain	Model	Strategy	nPrices	nWindows	Thr	StopLoss
1109.82	SvmLinear	StopLoss	2	14	0.1	1
939.69	SvmPolynomial	TrailingStop	1	0	0	0.5
912.15	Knn	TrailingStop	1	0	0	0.5
689.86	DecisionTree	StopLoss	1	14	0.1	1
488.14	BayesianNetwork	TrailingStop	1	0	0	0.5
467.76	NaiveBayes	StopLoss	2	0	0.6	0.5
Regression model						
1276.66	SvmLinear	StopLoss	1	14	0,1	1
1149.22	LinearRegression	StopLoss	3	14	0,2	1
1122.82	NeuralNetwork	StopLoss	1	14	0,1	1
821.51	SvmPolynomial	TrailingStop	1	0	0	0.5
515.35	DecisionTree	TrailingStop	1	0	0	0.5

attraverso la riduzione della dimensionalità dei dati si percepisce anche un peggioramento delle performance. In particolare si possono rilevare guadagni dimezzati per le criptovalute BTC e ETH.

Tabella 6.12: Risultati, sperimentazioni a confronto

Trading intraday			Trading intraday with financial indices			
			All index		Features Selection	
Code	Gain	model	Gain	model	Gain	model
BTC	141.10	NaiveBayes	1040.75	LinearRegression	582.56	SvmLinearRegression
ETH	263.03	BayesianNetwork	1122.97	SvmLinearRegression	733.85	SvmLinearRegression
XRP	560.91	BayesianNetwork	1677.41	LinearRegression	1276.66	SvmLinearRegression

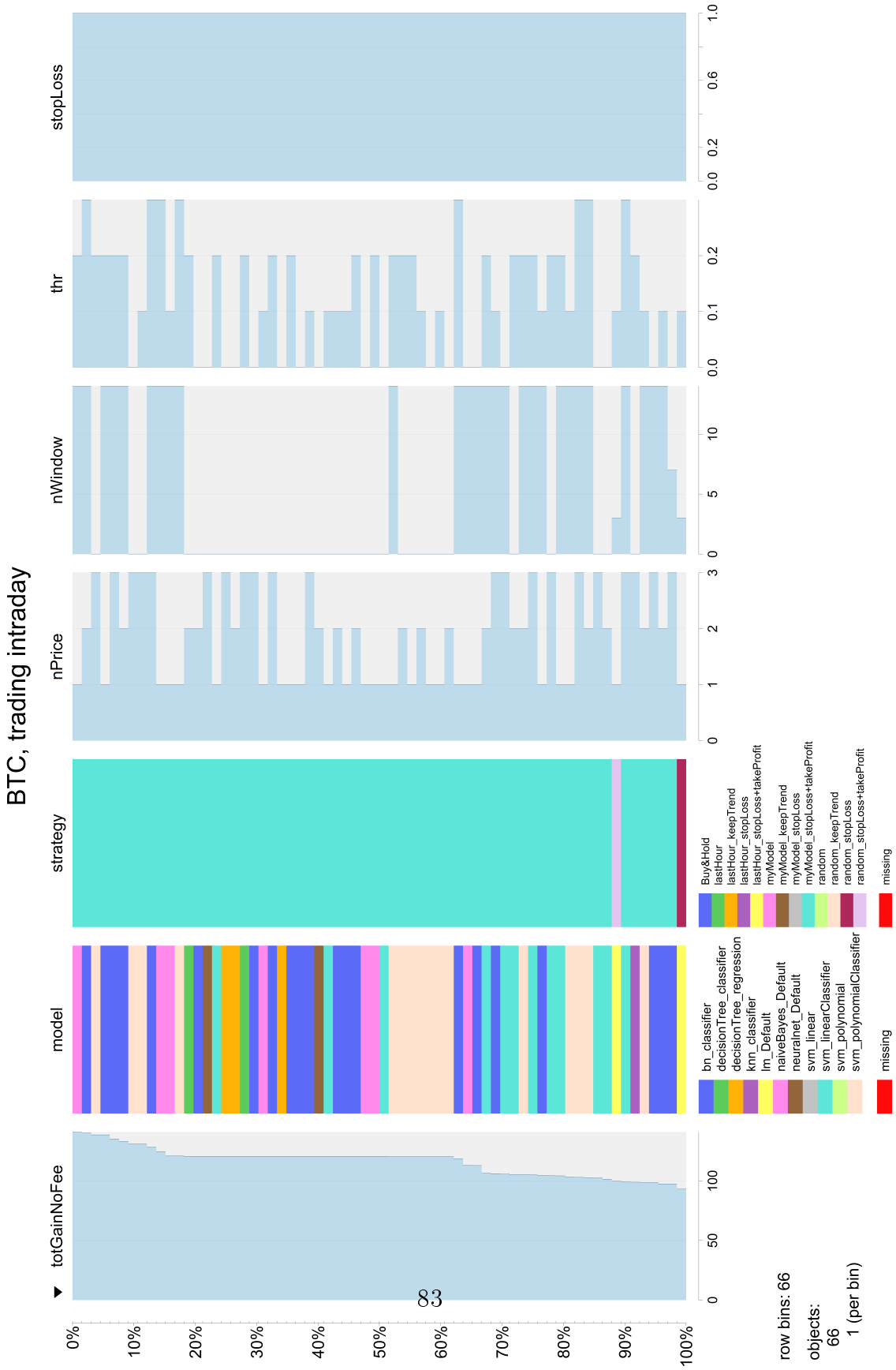

Figura 6.2: Risultati delle sperimentazioni intraday sul BTC[2]

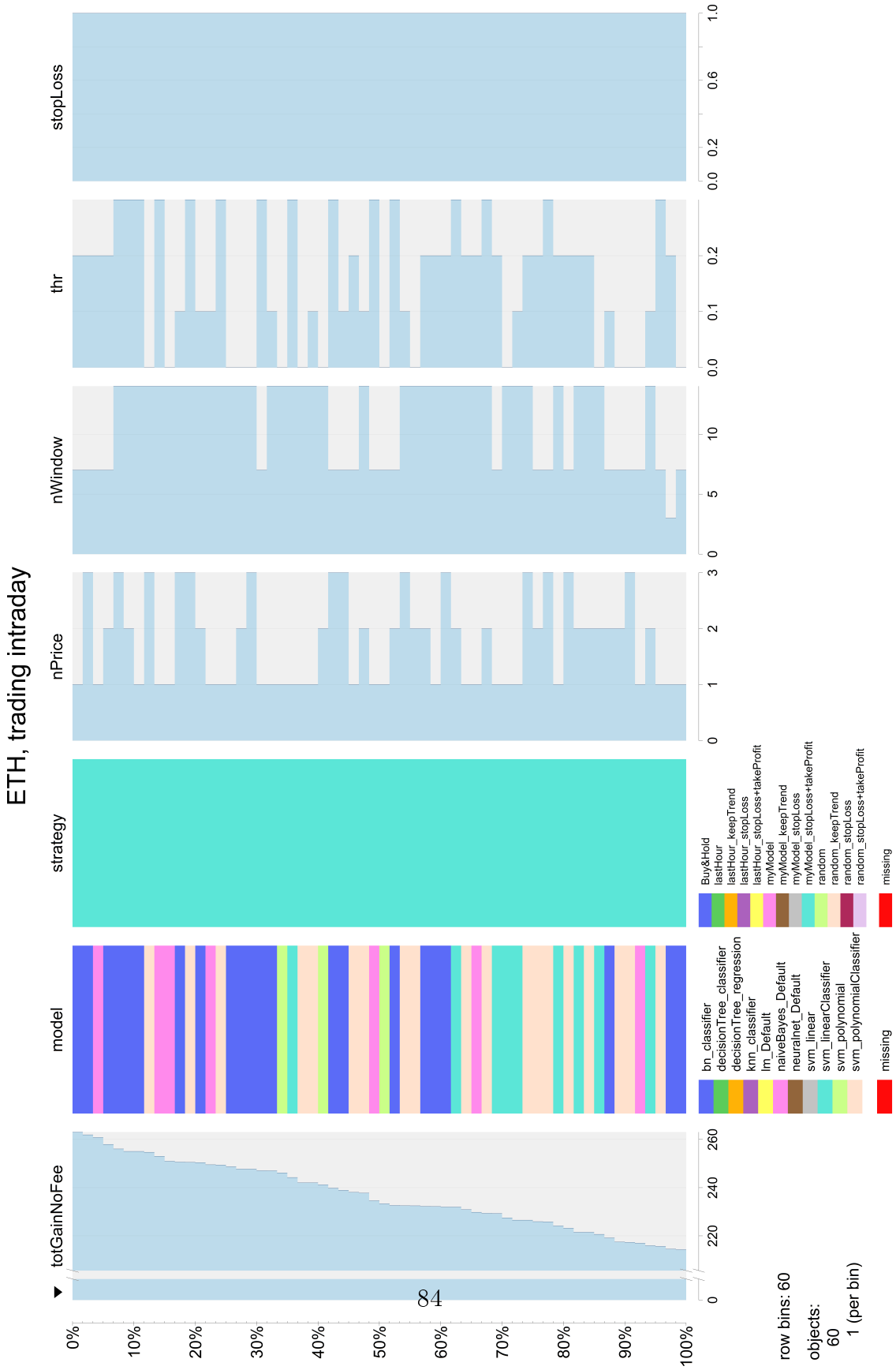

Figura 6.3: Risultati delle sperimentazioni intraday sul ETH[3]

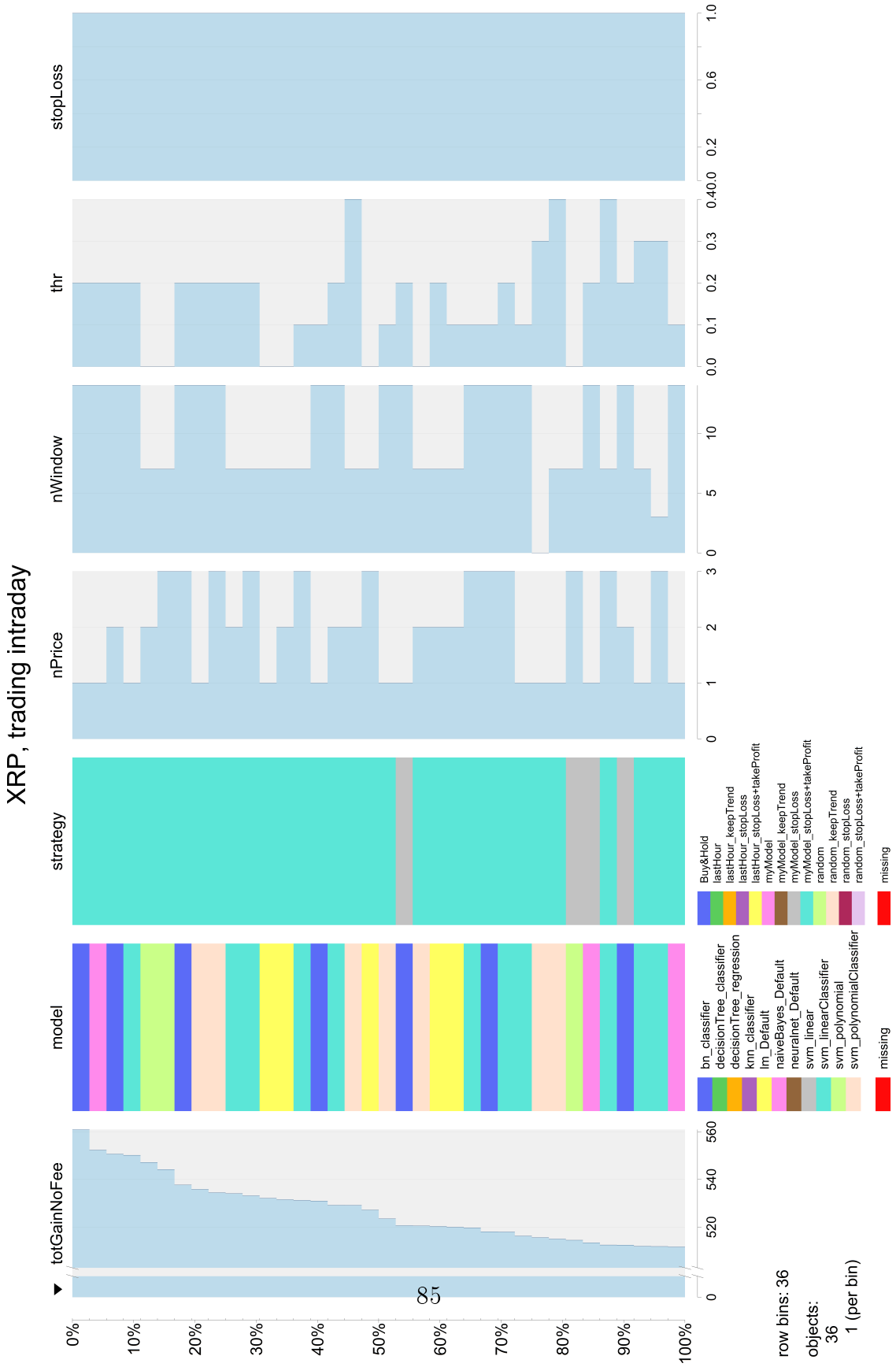

Figura 6.4: Risultati delle sperimentazioni intraday sul XRP[4]

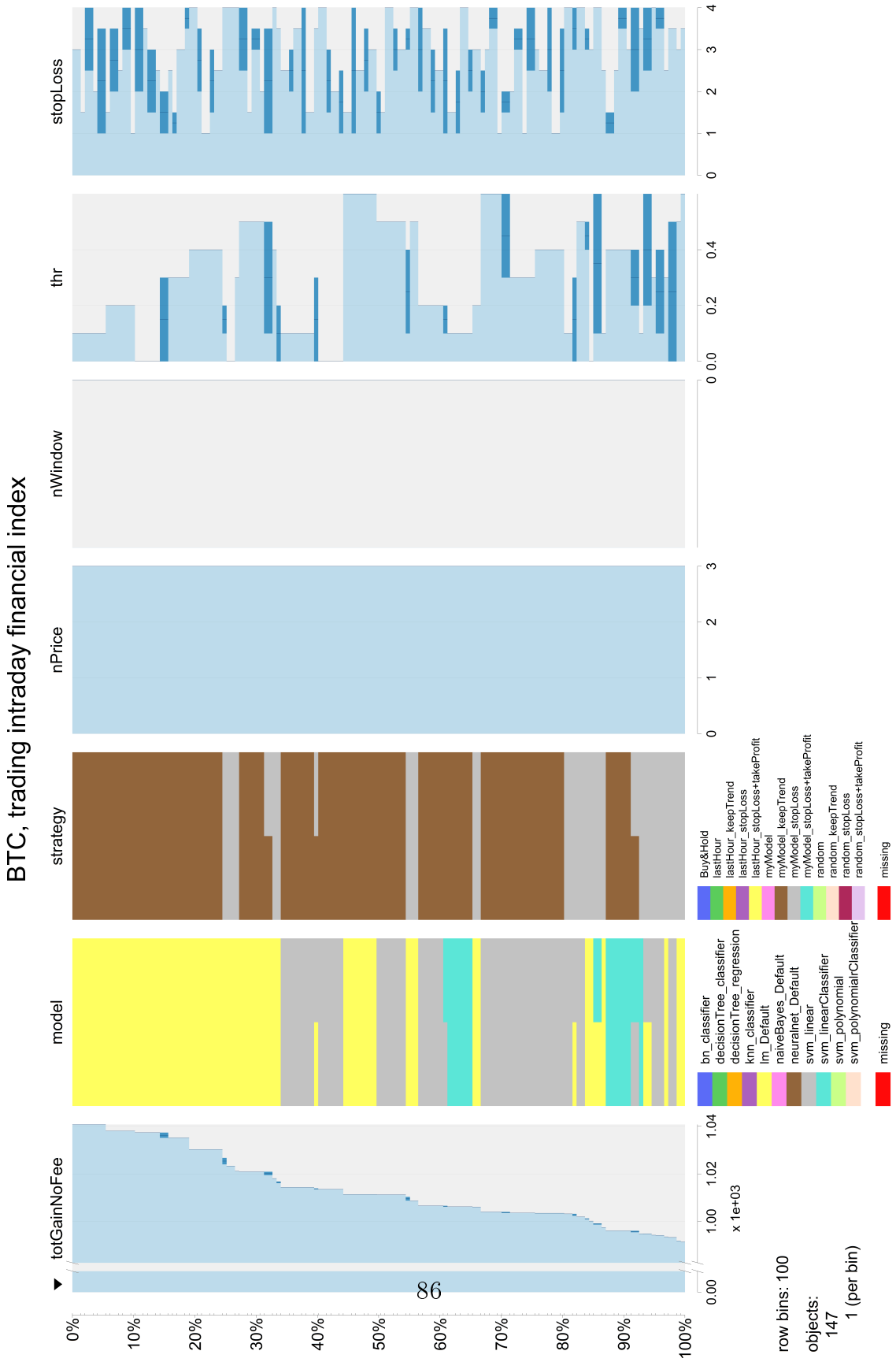

Figura 6.5: Risultati delle sperimentazioni intraday con indici finanziari su BTC[5]

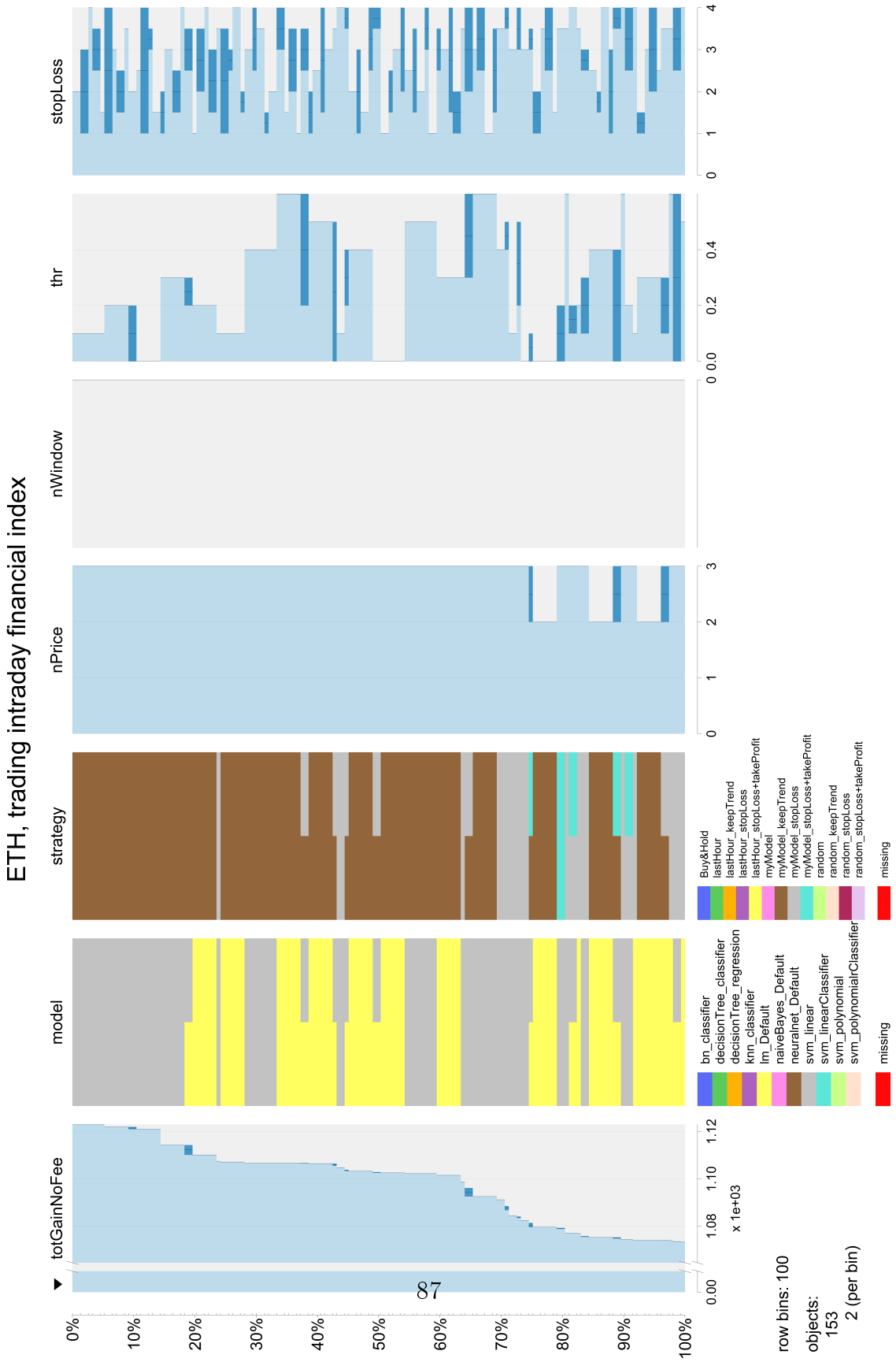

Figura 6.6: Risultati delle sperimentazioni intraday con indici finanziari su ETH[6]

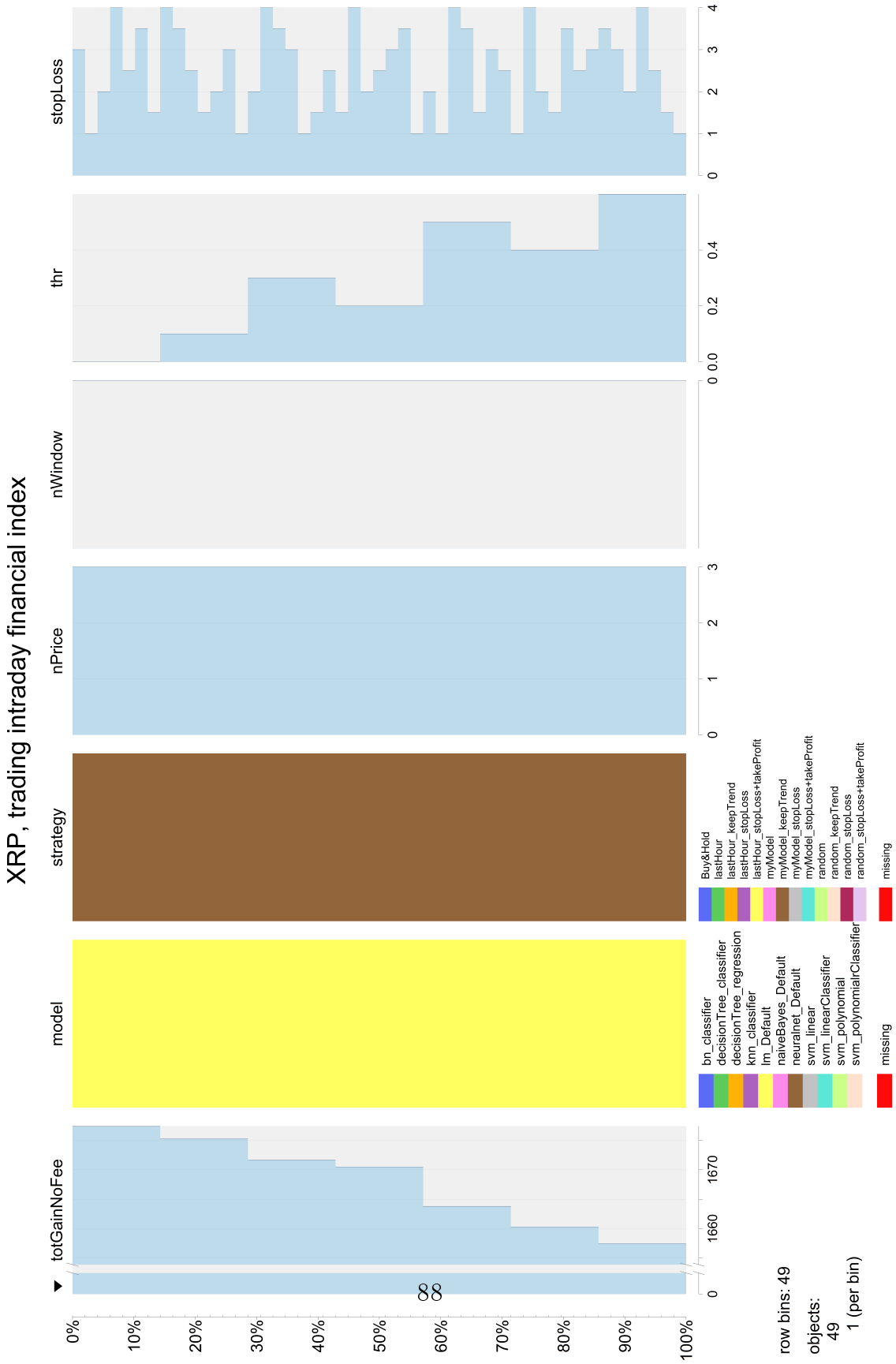

Figura 6.7: Risultati delle sperimentazioni intraday con indici finanziari su XRP[7]

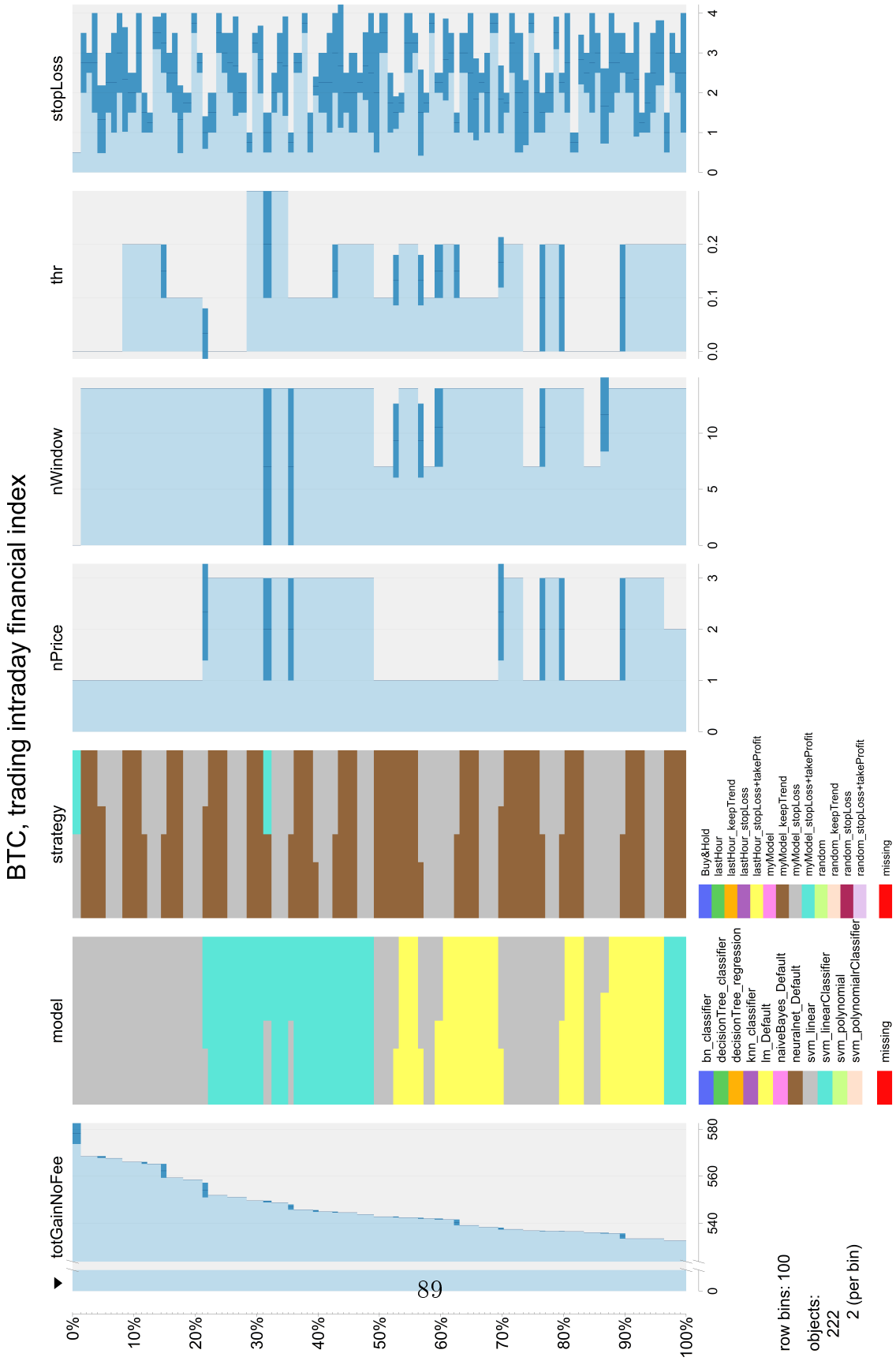

Figura 6.8: Risultati delle sperimentazioni intraday sul BTC con indici finanziari e Features selection[8]

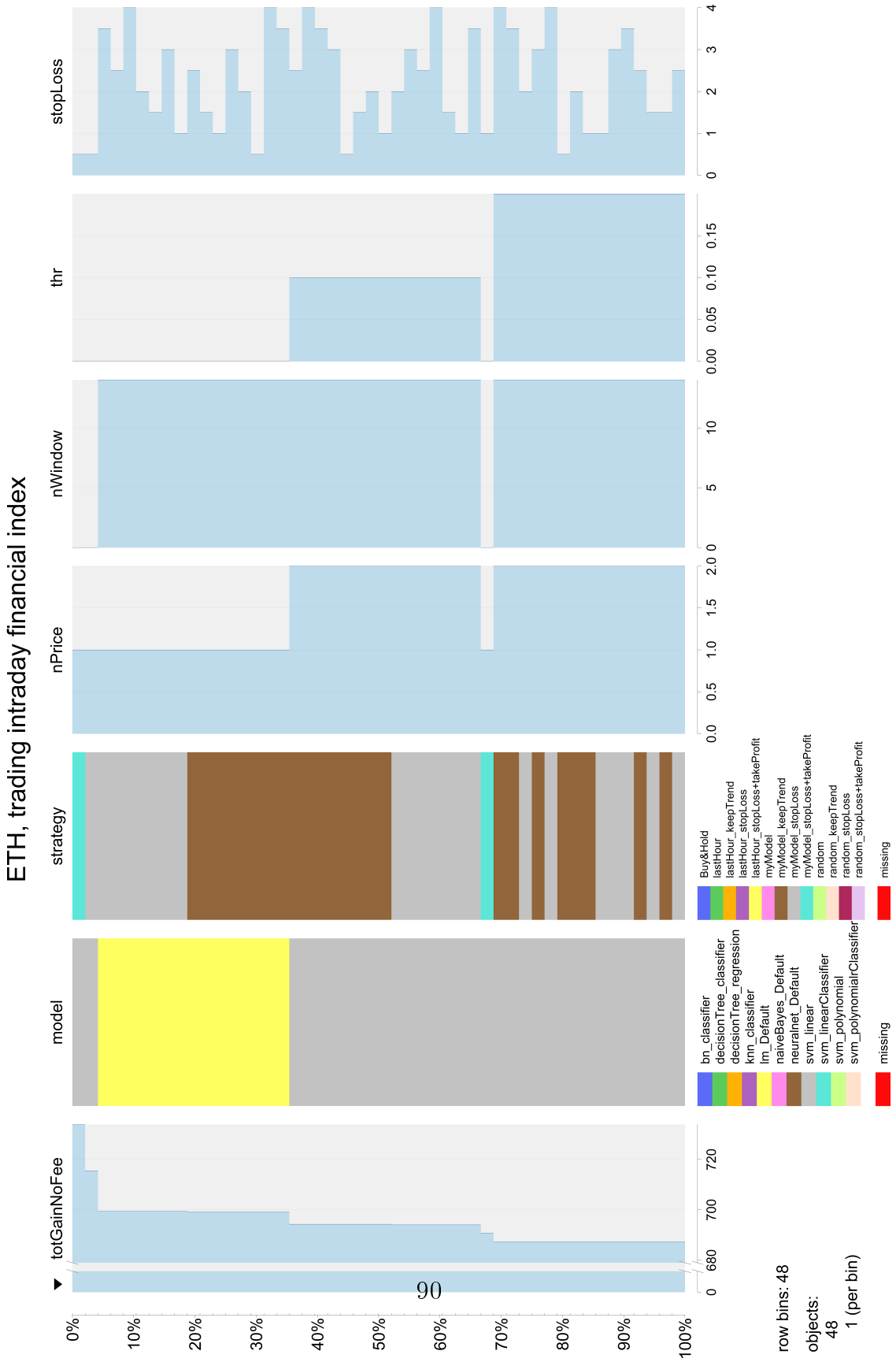

Figura 6.9: Risultati delle sperimentazioni intraday sul ETH con indici finanziari e Features selection[9]

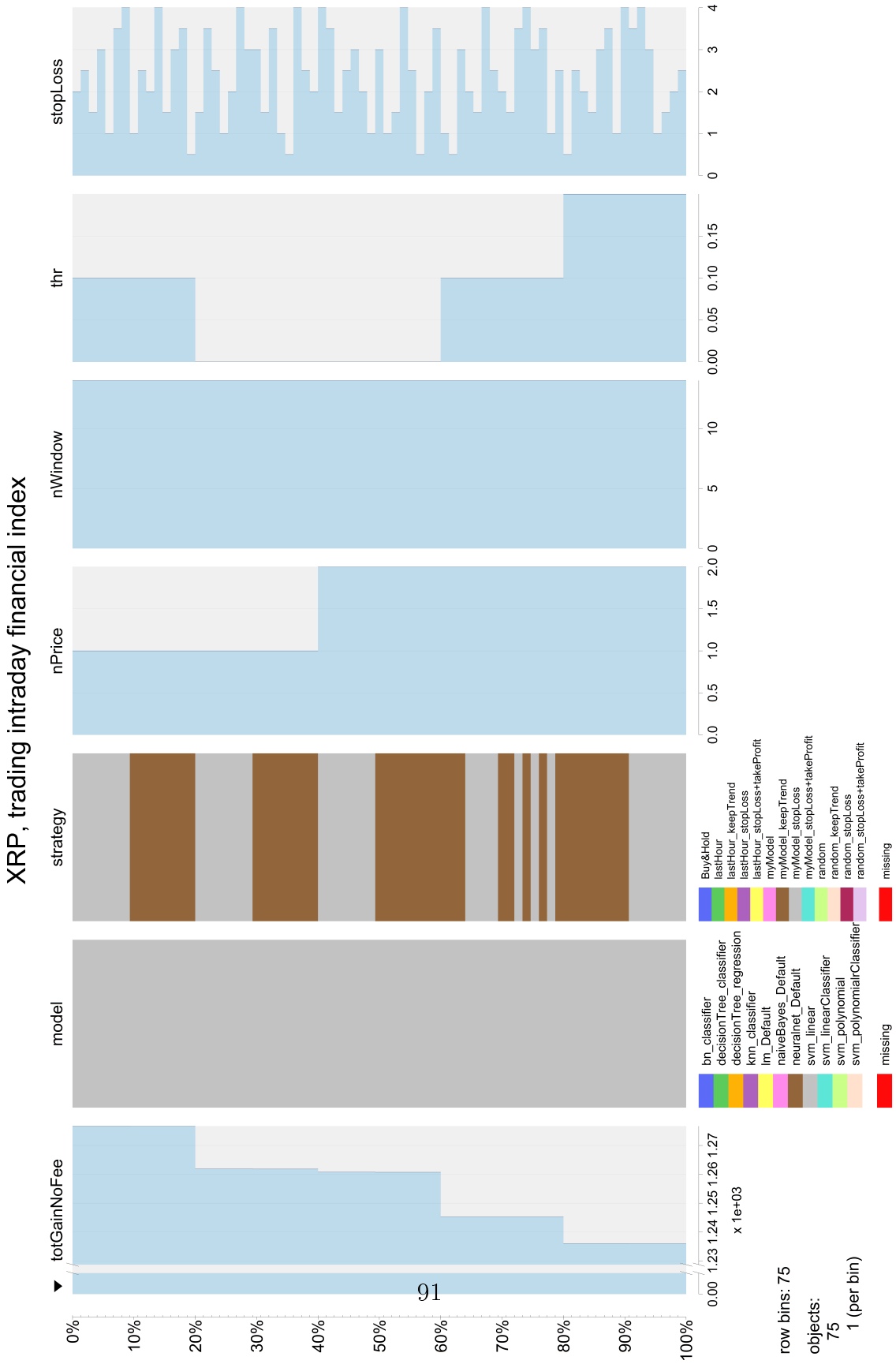

Figura 6.10: Risultati delle sperimentazioni intraday sul XRP con indici finanziari e Features selection[10]

Capitolo 7

Conclusioni

L'obiettivo della ricerca è stato quello di riuscire a prevedere l'andamento del mercato delle criptovalute e creare in un secondo luogo delle strategie di compravendita per riuscire a massimizzare i guadagni. In particolare si è voluto dimostrare se attraverso l'implementazione di uno strumento per il supporto decisionale si riesca ad aumentare in maniera considerevole i ricavi.

L'analisi dell'andamento dei prezzi relativi alle criptovalute ha rilevato da subito un mercato estremamente volatile con delle importanti variazioni di prezzo. Per tale motivo si è deciso di optare per una strategia di tipo intraday per cercare di riuscire a prevedere e sfruttare queste grosse oscillazioni. In particolare si è deciso di prediligere una strategia a livello orario, la quale ha inoltre annesso alle sperimentazioni un notevole aumento dei rischi. Per tale motivo per confermare l'efficacia dei nostri modelli, tutte le sperimentazioni sono state confrontate con delle simulazioni di riferimento volte a testare l'efficienza dei nostri modelli. Le simulazioni di riferimento sono le seguenti:

- *Buy & Hold*: simulazione volta a calcolare il guadagno percepito nel caso in cui si compri il primo giorno e si venda tutto l'ultimo giorno dell'intervallo considerato. Questa è una delle sperimentazioni di riferimento perchè offre il minimo costo di spese di transizione ed il più basso tasso di rischio.

- *Last hour*: ripropone lo stesso segnale percepito l'ora precedente, presupponendo che il trend in atto continui senza variazioni. Propone un numero di compravendite adeguate e quindi costi di commissioni di riferimento.

- *Random*: vi è la generazione di segnali di compravendita del tutto casuali. Si propone questo caso in quanto propone una strategia totalmente randomica, priva di qualsiasi strumento per il supporto decisionale.

In generale le sperimentazioni effettuate hanno confermato come l'implementazione di strumenti di supporto, volti all'analisi tecnica dei dati, ha generato un notevole miglioramento delle prestazioni di trading. In particolare si è percepito, nelle configurazioni migliori, un guadagno rispetto all'investimento iniziale pari al 1040,7% sul Bitcoin, al 1122,9% sull'Ethereum ed al 1677,4% sul Ripple.

Partendo dal lavoro svolto si potrebbero approfondire diversi aspetti delle simulazioni per poter cercare di aumentare l'efficacia del trading system proposto. Un particolare aspetto ,che meriterebbe un maggior approfondimento, sarebbe la possibilità, durante la fase di preprocessing, di aumentare l'efficienza della features selection. In particolare si potrebbe provare ad includere nella computazione dei modelli un maggior numero di attributi, testando anche altre tipologia di algoritmi per la selezione dei dati.

Durante la fetures selection è inoltre emerso come gli indici finanziari più promettenti siano il mom, lo stoch_fatsk e l'ema20, rimarcando come indici più reattivi alle piccole oscillazioni siano di maggior rilevanza. Sarebbe quindi interessante provare ad utilizzare degli indici finanziari più adeguati ad un mercato così volatile.

Un ulteriore approfondimento potrebbe inoltre aggiungere un ulteriore fase al trading system proposto, attraverso l'implementazione della tecnica denominata come *Ensemble model* o *Stacking model* .

La figura 7.1 mostra l'implementazione di tale tecnica. Spesso lo stacking model, con l'introduzione di un secondo livello, sovraperforma i singoli modelli compensando le scorrette predizioni da questi generati.
Infine, sarebbe di interesse rieseguire le sperimentazioni viste aggiornando i dati relativi alla criptovalute, in quanto lungo la prima metà del 2018 si è osservato un calo generale dei prezzi ed un mitigamento della volatilità del mercato.

[1]Immagine tratta da: `https://blogs.sas.com/content/tag/ensemble-models/`

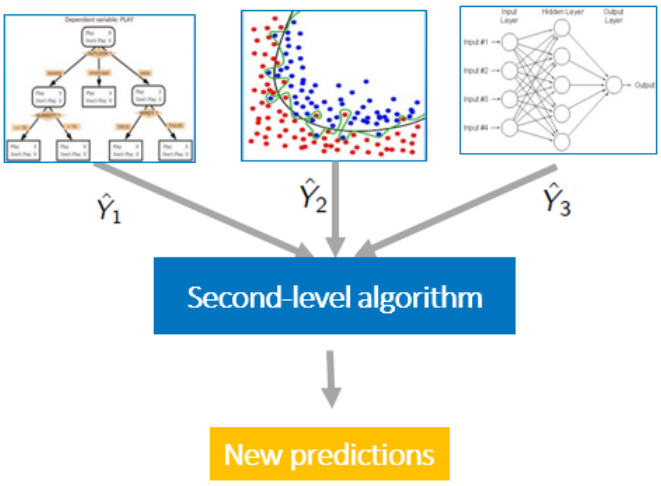

Figura 7.1: Model stacking[1]

Bibliografia

Böhme, R., Christin, N., Edelman, B., & Moore, T. (2015). Bitcoin: Economics, technology, and governance. *Journal of Economic Perspectives*, *29*(2), 213–38.

Chohan, U. W. (2017). A history of bitcoin.

ForexItalia24. (2017). *Cosa sono le criptovalute.* Retrieved from `https://www.forexitalia24.com/trading/criptovalute.php` ([Online; in data 12-luglio-2017])

Harvey, I. (2018). *Intraday.* Retrieved from `https://www.investopedia.com/university/introduction-stock-trader-types/intraday-traders.asp`

Ig-Staff. (2018). *Oscillatore stocastico - prima parte.* Retrieved from `https://www.ig.com/it/oscillatore-stocastico-prima-parte`

Investopedia-Staff. (2018a). *Chande momentum oscillator.* Retrieved from `https://www.investopedia.com/terms/c/chandemomentumoscillator.asp`

Investopedia-Staff. (2018b). *Stochastic oscillator.* Retrieved from `https://www.investopedia.com/terms/s/stochasticoscillator.asp`

Investopedia-Staff. (2018c). *Stochastics: An accurate buy and sell indicator.* Retrieved from `https://www.investopedia.com/articles/technical/073001.asp`

Kursa, M. B., Rudnicki, W. R., et al. (2010). Feature selection with the boruta package. *J Stat Softw*, *36*(11), 1–13.

Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, *2*(3), 18–22.

Lucchetti, F. (2018). *Indicatore dpo, detrended price oscillator.* Retrieved from `https://www.investopedia.com/terms/c/chandemomentumoscillator.asp`

Magalotti, R. (2016). *Trading intraday: significato e consigli pratici. la guida.* Retrieved from `https://www.money.it/Trading-intraday`

-significato-consigli-guida

Nahar, K. (2012). Artificial neural network. *COMPUSOFT, An international journal of advanced computer technology*, *1* (2), 25–27.

Narayanan, A., Bonneau, J., Felten, E., Miller, A., & Goldfeder, S. (2016). *Bitcoin and cryptocurrency technologies.* s Princeton University Press.

R. (2018). *What is r?* Retrieved from https://www.r-project.org/about.html ([Online; controllata il 26-aprile-2018])

Scott, B. (2016). *How can cryptocurrency and blockchain technology play a role in building social and solidarity finance?* (Tech. Rep.). UNRISD Working Paper.

stockcharts Staff. (2018). *Stochastic oscillator.* Retrieved from http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:stochastic_oscillator_fast_slow_and_full

Wang, S.-C. (2003). Artificial neural network. In *Interdisciplinary computing in java programming* (pp. 81–100). Springer.

Wikipedia. (2018a). *Crawler.* Retrieved from https://it.wikipedia.org/wiki/Crawler (2018)

Wikipedia. (2018b). *R (software).* Retrieved from https://it.wikipedia.org/wiki/R_(software) ([Online; controllata il 26-aprile-2018])

Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. In *Big data (bigdata congress), 2017 ieee international congress on* (pp. 557–564).