POLITECNICO DI TORINO

Corso di Laurea Magistrale in
Ingegneria Informatica (Computer Engineering)

Tesi di Laurea Magistrale

Trading azionario basato su tecniche di classificazione e sul riconoscimento di pattern di analisi tecnica

Relatori
prof. Paolo Garza
prof. Luca Cagliero

Candidato
Roberto Brescia

A.A. 2016/2017

Ringraziamenti

Dopo un lungo e faticoso cammino, questo traguardo è stato finalmente raggiunto.

Un ringraziamento particolare va ai miei relatori, i professori P. Garza e L. Cagliero, per avermi supportato durante tutta la stesura di questo lavoro. Sin dall'inizio hanno dimostrato una grande disponibilità nonostante il poco tempo a mia disposizione e sono riusciti a guidarmi nei migliori dei modi fino al compimento della mia tesi.

Desidero ringraziare tutti voi, amici miei, per essere stati la mia valvola di sfogo e aver sopportato il mio caratteraccio.

Grazie anche a voi, Miche, Gianna e Davide, che in questi anni siete diventati la mia seconda famiglia. Con voi ho condiviso momenti splendidi e devo dirvi grazie per aver partecipato alla mia crescita.

Vorrei ringraziare i miei genitori per aver sempre creduto in me, anche quando ero io stesso a non crederci. Grazie per avermi sostenuto e sopportato durante questo percorso. Grazie per esservi lasciati scivolare tutto addosso anche quando avreste dovuto reagire e dirmene quattro. Grazie per essermi sempre accanto. Semplicemente grazie di essere mio padre e mia madre.

Grazie a mia sorella che, nonostante negli ultimi anni sia più lontana, è sempre nei miei pensieri.

Infine, voglio ringraziare te, Roberta, per essere, da sempre, il mio più grande sostegno nonostante i miei modi un po' così. In questi anni hai imparato a sopportare le mie paranoie ed ansie e sappiamo entrambi che senza di te non sarei mai arrivato dove sono ora. Sappi che te ne sarò sempre grato.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Le istituzioni finanziarie hanno, da sempre, prodotto grandi quantità di dati da analizzare e studiare affinché i loro profitti fossero i più alti possibile; con il tempo questi dataset hanno iniziato ad incrementare sempre più la loro mole rendendo perciò l'analisi manuale un opzione non più percorribile. Grazie all'avvento di numerosi tools di *data mining*, comincia quindi a prendere piede l'analisi automatica o semi-automatica; con il tempo, l'incremento della potenza computazionale ha reso possibile, inoltre, l'esplorazione e la sperimentazione di diverse tecniche, anche tra le più complesse. L'avvento di internet e il continuo progresso tecnologico hanno reso il trading molto più *user-friendly*: tutt'ora, qualsiasi persona può iniziare ad operare in borsa senza alcuna particolare difficoltà e conoscenza del settore. In questo scenario, emergono sempre più figure professionali, i cosiddetti *trader*, che cercano di rendere di questa attività il proprio mestiere; è emersa quindi, da parte di tutto il settore tecnologico e finanziario, la necessità di creare strumenti che possano facilitare e migliorare il lavoro. Per queste motivazioni, l'interesse verso la compravendita di strumenti finanziari è, di anno in anno, in continua crescita. Fatte queste premesse, si può capire come il data mining abbia preso sempre più piede, anche in maniera preponderante. L'utilizzo di queste tecniche rende possibile l'estrazione di informazioni, pattern e modelli a partire da grandi quantità di dati, anche di natura diversa tra loro. Uno degli aspetti più importanti è che questa metodologia di analisi permette una notevole automazione, sgravando il trader dall'analisi manuale, sempre complessa e limitata

Il data mining in ambito finanziario rende possibile l'approfondimento di diversi fattori legati al business permettendo, tra le altre cose, di ridurre i costi, aumentare i ricavi o ottenere informazioni al marketing. La motivazione principale per cui viene utilizzato, però, è la previsione dei movimenti di mercato e la stima dei rischi. In origine, gli studi realizzati coinvolgevano dati raccolti a partire da un singolo mercato o indice; con il tempo, però, è nata la necessità di analizzare dati proveniente da molte più fonti, ricercando l'eventuale legame presente tra i vari mercati, anche collocati in aree geografiche differenti. In questo contesto assume un'importanza di un certo rilievo il trading intraday. Questa strategia è una delle più comuni e consiste nel comprare e vendere azioni a breve termine, tipicamente nella stessa giornata. Negli ultimi anni, in letteratura, sono stati proposti diversi sistemi automatici di trading per scegliere l'azione sui cui cercare un guadagno. La maggior parte di essi si basa sul generare segnali (ad esempio, *"compra l'azione X se il suo prezzo diminuisce dello 0.5%"*“) o sul predire se una determinata azione aumenterà o diminuirà

il suo prezzo. Un aspetto interessante e, soprattutto, ancora poco esplorato è la generazione di segnali di trading che dicano, in maniera specifica, su quale azione (o quale gruppo di azioni) effettuare compravendita affinché il guadagno sia il massimo possibile.
Partendo dal contesto fino a qua descritto, questa tesi è la continuazione di un lavoro di ricerca, concluso nel 2016, realizzato dal *"Database and Data Mining Group"* del Politecnico di Torino [3]. Analizzando i prezzi storici e utilizzando *algoritmi di regressione* e di *analisi delle sequenze*, tale studio si proponeva di scoprire su quali azioni fosse conveniente investire. La sperimentazione qui descritta, invece, ha avuto come obiettivo quello di utilizzare, per lo stesso scopo, tecniche di classificazione; inoltre, si è cercato di capire se il riconoscimento e l'utilizzo di pattern grafici derivanti dall'analisi tecnica standard potesse aiutare e migliorare l'efficacia della predizione. In aggiunta, il lavoro di ricerca qui presentato è partito da un'idea opposta a quanto presente in letteratura: solitamente si è sempre cercato di arricchire i dati con nuove informazioni, invece in questo caso l'obiettivo, grazie all'utilizzo di tali pattern grafici, è stato quello di ridurre l'insieme di azioni da considerare per la costruzione del modello e quindi per le successive predizioni. Per scegliere i titoli da inserire nel portafoglio di investimento è stato utilizzato il valore di confidenza generato dalle tecniche supervisionate utilizzate per questa tesi. Nello specifico si è deciso di utilizzare i seguenti classificatori: alberi di decisione, SVM, classificatore bayesiano e reti neurali. Per effettuare un numero di test soddisfacente, gli esperimenti sono stati eseguiti su due indici (FTSE MIB e S&P500), su tre anni (2011, 2013, 2015) e simulando due posizioni (short e long).

Questa tesi è composta da sette capitoli.
Il *capitolo 2*, che segue questa introduzione, tratta il data mining in linea generale, entrando poi nel dettaglio nelle tecniche di classificazione; vengono quindi approfonditi gli algoritmi utilizzati nei nostri esperimenti.
Nel *capitolo 3* viene trattato il trading: si parte dalla storia arrivando fino alle tecniche più moderne, descrivendo infine lo stato dell'arte nell'ambito del trading nel settore finanziario.
Il *capitolo 4* introduce ad alto livello il framework sviluppato, descrivendo brevemente ogni step che lo compone. Il *capitolo 5* segue con l'analisi dei software e delle varie fasi coinvolte in ogni singolo esperimento. Viene presentato Rapidminer, la raccolta e il preprocessing dei dati, la metodologia utilizzata per la predizione e la tecnica scelta per generare i risultati.
Il *capitolo 6* si occupa di presentare gli esperimenti realizzati.
Nel *capitolo 7* vengono presentati i risultati ottenuti, distinguendoli per mercato, e viene analizzata l'influenza degli algoritmi e dei settaggi sull'esito delle simulazioni.
Infine, il *capitolo 8* riepiloga i risultati ottenuti e di suggerisce eventuali sviluppi futuri.

Capitolo 2

Il data mining

Il data mining, nato a partire dagli anni '90, affonda le proprie origini in diversi settori: tra questi ci sono la statistica, il campo del machine learning e quello delle basi dati. Ben prima della sua nascita, i matematici e gli statistici hanno iniziato ad interessarsi ai dati e in particolare si sono occupati dei cosiddetti algoritmi di clustering; il primo di essi, vale a dire il *k-means*, è stato pubblicato nel 1967, decisamente molti anni prima della nascita del concetto di data mining.

Gli algoritmi di machine learning si basano sul fatto che il dataset sia sufficientemente piccolo in modo tale da effettuare tutte le attività direttamente in memoria, dopo averlo caricato completamente. Quest'ipotesi è tutt'ora valida e si cerca quindi di ridurre i dati per rendere possibile quanto appena descritto; questo permetterebbe di utilizzare tutti quegli algoritmi, ormai abbastanza consolidati, che però lavorano sfruttando la ram.

Il settore della basi dati fornisce il proprio contributo in termini di filtraggio dei dati; infatti, quando si effettua una query, non c'è l'aspettativa che tutto il database venga caricato in memoria prima di effettuare l'interrogazione. Esistono infatti delle tecniche, come ad esempio quelle per leggere parzialmente i dati, che diventano utili anche nel dominio del data mining.

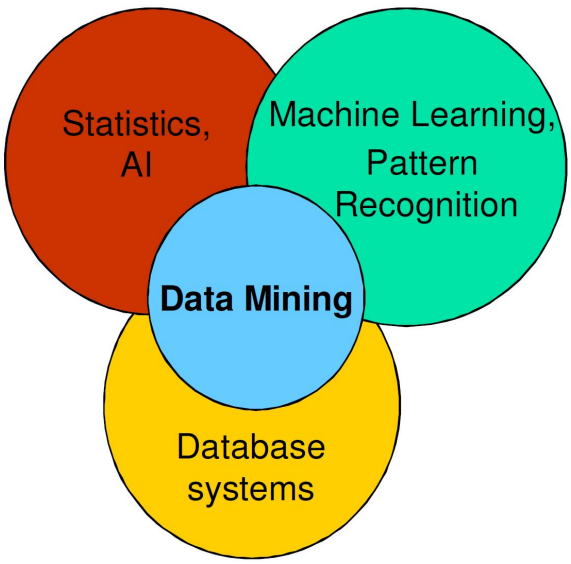

Figura 2.1: Data Mining come intersezione di più discipline[1]

L'area del data mining risulta quindi essere, al giorno d'oggi, un area molto ricca anche dal punto di vista dell'attività di ricerca; i contributi arrivano da tutte le aree sopra descritte. I campi di applicazioni di queste tecniche sono molto vasti e spaziano dalla ricerca in campo medico, all'analisi delle performance di un atleta, passando per il campo economico e finanziario.

La maggior parte delle società e degli enti possiedono basi dati enormi contenenti informazioni eterogenee e di natura diversa tra di loro. Nonostante questi dati siano potenzialmente una fonte di informazioni molto utili, a causa della loro natura diventa difficile andare a rilevare ciò che serve veramente. L'informazione risulta "nascosta" e non immediatamente visibile; l'analisi richiede una grande quantità di tempo e, per questo motivo, molte volte la maggior parte dei dati non viene analizzata.

Sotto un certo punto di vista, il data mining può essere riassunto come l'estrazione, a partire dai dati disponibili, di informazioni che sono:

- Implicite
- Precedentemente sconosciute
- Potenzialmente utili

Questa estrazione è totalmente automatica ed è realizzata grazie ad algoritmi sviluppati specificatamente. I dati estratti vengono rappresentati attraverso dei modelli astratti, chiamati *pattern*, e possono rappresentare, ad esempio, le correlazioni che esistono tra le informazioni presenti nel dataset in nostro possesso. In qualsiasi campo venga applicato, il data mining presenta moltissime potenzialità; tra queste troviamo, senza alcun dubbio, il miglioramento dell'efficienza, della sicurezza e dei guadagni. Alcuni scenari di utilizzo potrebbero essere, ad esempio, l'analisi del comportamento e degli interessi degli utenti di un sito di e-commerce o di un social network, la cosiddetta analisi del paniere, fino ad arrivare ad analisi nel campo della biologia.

Come avviene però l'analisi dei dati?
In letteratura, questo processo viene chiamato *Knowledge Discovery Process* (spesso abbreviato in KDD). Con il KDD si cerca di cerca di estrarre una nuova conoscenza a partire dai dati; uno degli step di questa ricerca, probabilmente il più importante, è il data mining. Entrando nel dettaglio, nel KDD si possono identificare le seguenti fasi:

- **Selezione**: primo step, dove avviene la scelta dei dati pertinenti all'operazione di analisi che vogliamo realizzare. Vengono applicate alcune regole in modo tale che soltanto l'informazione che rispetta determinate specifiche venga presa in considerazione. Solamente i dati con un certa rilevanza a livello di significato vengono estratti e mantenuti. L'output di questa fase sono i cosiddetti *target data*. Questa fase è molto importante poiché i dati provenienti dal mondo reale sono spesso di bassa qualità; se non la si migliorasse non ci sarebbe mai la possibilità ottenere dei buoni pattern.
- **Pre-processing**: la pre-elaborazione è utile per ripulire i dati. Vengono quindi eliminate tutte quelle porzioni di informazioni che risultano inutili.

[1]Immagine tratta da: P. P. Tan, M. Steinbach e V. V. Kumar. *Introduction to Data Mining*

Una delle operazioni più comuni è la rimozione del rumore e delle eccezioni (outliers).

- **Trasformazione**: grazie alla trasformazione è possibile preparare i dati per il data mining successivo. Le informazioni vengono aggregate o sintetizzate affinché lo step successivo possa elaborarle nella maniera più corretta.
- **Data Mining**: questa è la fase più importante dell'intero processo. Qui vengono ricercati e generati i pattern che verranno utilizzati successivamente per trarre delle conclusioni.
- **Interpretazione e valutazione**: i pattern generati precedentemente hanno la necessità di essere interpretati. Molto spesso l'analista che li genera non è sempre la stessa persona che è in grado di renderli significativi; per questo è necessario avere, all'interno del proprio team, un esperto del dominio appartenente ai dati. Sarà quest'ultimo che potrà assegnare un significato ai risultati ottenuti e capire la loro utilità.

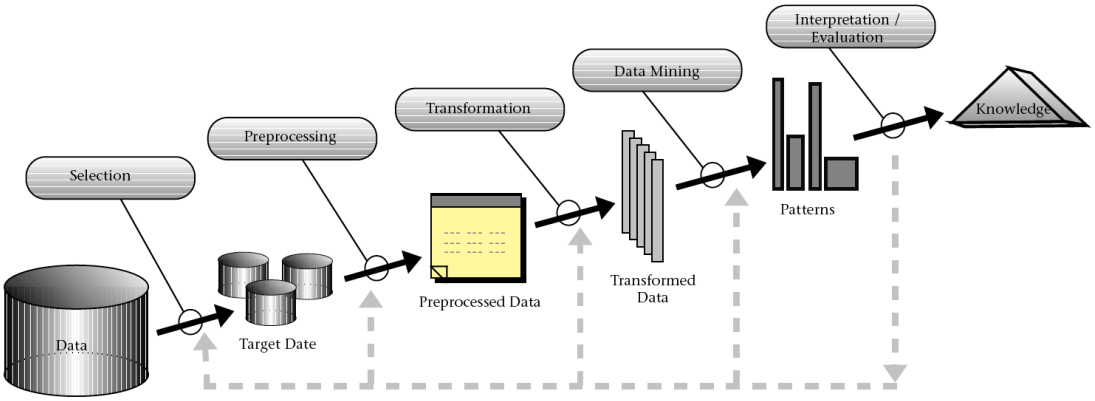

Figura 2.2: Knowledge Discovery Process[2]

Dopo questa panoramica, il prossimo paragrafo andrà a descrivere, nel dettaglio, le varie tecniche utilizzate nel data mining.

[2]Immagine tratta da: `http://www.infovis-wiki.net/index.php?title=Knowledge_Discovery_in_Databases_(KDD)`

2.1 Le tecniche di data mining

Le tecniche di data mining possono essere divise in due macro-categorie:

- **Metodi descrittivi**: si pongono come obiettivo quello di dare una descrizione al dataset a nostra disposizione. Cercano quindi di estrarre un modello interpretabile che descriva i dati. Un esempio, può essere quello della classificazione dei clienti di un'azienda raggruppando tra loro quelli che hanno un comportamento simile; in questo caso si parla di *tecnica descrittiva* in quanto non conosco a priori cosa fare, ma cerco solamente di creare dei gruppi (*clustering*).

- **Metodi predittivi**: cercano di effettuare una predizione generando informazioni che non si conoscono. I sistemi predittivi devono essere capaci di assegnare una categoria, detta *etichetta di classe*, a dei nuovi dati che non la possiedono. Per fare questo partono da dei dataset che hanno questa etichetta e creano un modello che permetta di fare l'assegnazione. Tra queste tecniche troviamo quelle di classificazione.

Cercando di effettuare un'ulteriore distinzione, è possibile identificare altre due macro-categorie: le **tecniche supervisionate** e le **tecniche non supervisionate**. La prima categoria è formata da tutti quegli algoritmi che utilizzano come dataset di addestramento un insieme di coppie input-output; questo significa che il valore ricercato è già conosciuto a propri e viene utilizzato per la creazione del modello. Questi metodi sono solitamente più veloci e precisi e, in linea generale, in seguito a quello che viene solitamente definito come *learning by examples*, il sistema è in grado di generare un output per qualsiasi nuovo input. Questa tipologia di tecniche inoltre è l'unica che genera, per ogni previsione effettuata, anche un valore di *confidenza*; tale numero esprime la probabilità, in percentuale, che i dati appartengano effettivamente alla classe predetta. Questo output si è dimostrato fondamentale per questo studio in quanto ha permesso di scegliere quali, tra le numerose predizioni, inserire nel portafoglio di investimento. Le tecniche non supervisionate invece non presentano i risultati nel training set, in quanto le classi non sono note a priori ma devono essere costruite ed apprese dinamicamente. Questi algoritmi vengono utilizzati per dividere gli input in gruppi (cluster) basandosi solamente sulle loro proprietà statistiche. Per ognuno di questi cluster viene generata un'etichetta che viene fornita in output, anche nel caso il numero di oggetti appartenenti alla classe ricercata sia basso.

Entrando più nel dettaglio e sfruttando la suddivisione appena descritta, i principali metodi di data mining sono:

- **Tecniche non supervisionate:**
  - **Clustering**. L'obiettivo, in questo caso, è quello di creare dei gruppi (cluster) ognuno dei quali contenente oggetti simili tra loro; vengono inoltre identificate le eccezioni, o *outliers*, cioè quegli elementi che non possono essere inseriti in uno di questi sottoinsiemi. Tra i principali algoritmi di clustering possiamo trovare gli algoritmi partizionali (k-means), gli algoritmi gerarchici e quelli density-base (DBSCAN).

  - **Regole di associazione**. Con queste tecniche si cerca di creare una correlazione tra dati (ad esempio, analizzare le ricevute di un supermercato per capire quali oggetti vengono acquistati spesso insieme). Fra le varie applicazioni c'è l'analisi dei carrelli (*market basket analysis*), la proposta di oggetti correlati (*cross-selling*) o persino l'ordinamento della merce a seconda del legame tra i vari prodotti.
  - **Sequenze ordinate**. Le metodologie che utilizzano le sequenze ordinate (sequence mining) concentrano la loro attività nel ricercare successioni di dati ricorrenti a partire da grandi dataset. Uno degli utilizzi più classici è quello di estrarre da un insieme di dati tutte le sequenze più frequenti; in questo caso, viene definito *supporto* la frequenza di occorrenza di una determinata sequenza. L'obiettivo diventa quindi quello di trovare tutte le sequenze il cui supporto sia sopra una determinata soglia.
  - **Serie temporali**. In questo contesto viene definito il cosiddetto Time Serie Database (TSDB) che rappresenta una collezione di dati nella quale assume una aspetto fondamentale la variabile "tempo". Dato un fenomeno, le serie temporali cercano di interpretarlo e di prevedere il suo andamento futuro.
  - **Rilevazione di eccezioni**. Si parla di eccezione (*outlier*) nel caso esista un elemento che appare errato rispetto gli altri dati. Un esempio reale di utilizzo di queste tecniche potrebbe essere la rilevazione di intrusi all'interno di una rete.

- **Tecniche supervisionate:**
  - **Classificazione**. La classificazione, come vedremo, ha come obiettivo la predizione di un'etichetta di classe. Alcuni esempi potrebbero essere quello di prevedere se un tumore è benigno o maligno, prevedere se le transazioni di una carta di credito sono fraudolente o anche, semplicemente, un filtro anti-spam.
  - **Regressione**. Con la regressione ci si pone l'obiettivo di predire un valore continuo; per questo motivo, l'utilizzo più comune è quello della previsione di un preciso valore numerico. Un esempio potrebbe essere quello della predizione del costo di un prodotto o di un servizio, date altre variabili e informazioni.

Per poter confrontare e comparare, in maniera oggettiva, le tecniche sopra descritte sono stati introdotti diversi indici di qualità. Tra questi troviamo:

- **Accuratezza**. Definisce quanto il modello è preciso a seconda della quantità di errori che vengono effettuati. E' utile ad esprimere la qualità della predizione.
- **Efficienza**. Valuta il tempo impiegato per la costruzione del modello e per la classificazione. Con questo indice è possibile valutare le performance delle tecniche di classificazione.
- **Scalabilità**. Esprime come gli algoritmi varino a seconda della dimensione del dataset di training, del numero di attributi o di entrambi.

- **Robustezza**. Valuta quanto il rumore e la presenza di eventuali dati mancanti influenzino la classificazione.

- **Interpretabilità**. Esprime quanto il modello è facilmente comprensibile da un utente e se fornisce informazioni utili a comprendere come è stato costruito.

Nonostante la continua ricerca porti sempre a nuovi traguardi, restano aperti ancora parecchi problemi riguardanti il data mining; uno dei più importanti è sicuramente la scalabilità nel caso di un volume enorme di dati, ma non da meno sono le problematiche riguardati la complessità delle strutture dati e la qualità delle informazioni.

Data la natura di questa tesi, andiamo ora ad approfondire le tecniche di classificazione.

2.2 La classificazione

L'obiettivo degli algoritmi di classificazione è la predizione di un'etichetta di classe. Dato un dataset e un insieme di etichette di classe, lo scopo è quello di analizzare le caratteristiche di queste classi in modo tale da assegnarne una a dei dati che non la possiedono. Il processo inizia sempre con la costruzione di un modello, utile ad avere una rappresentazione astratta del dataset di input; questo modello sarà poi utilizzato per la predizione delle cosiddette *label*.

Per poter creare il modello, i dati etichettati vengono utilizzati come input e divisi in due gruppi: il primo, denominato *training set*, sarà quello utilizzato per costruire il modello; il secondo, definito come *test set*, sarà utilizzato per validare il modello. Se il modello assegna le corrette etichette di classe si dice che è ben formato (*well-formed*) e potrà quindi essere utilizzato per effettuare predizioni analizzando nuovi dati.

Un classico esempio di sistema di classificazione è il filtro anti-spam; quest'ultimo assegna un etichetta (*spam/non spam*) alle e-mail ricevute basandosi sul modello costruito in precedenza.

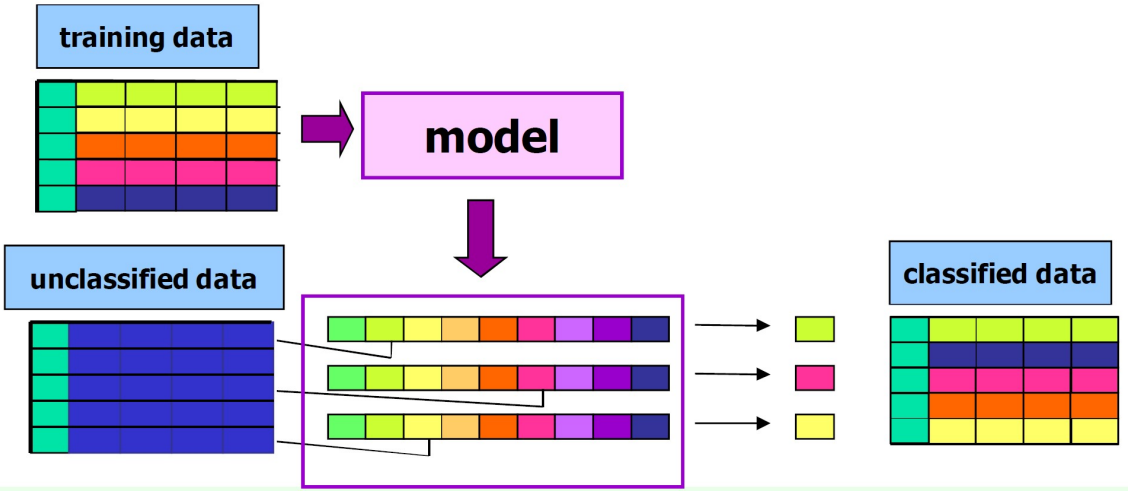

Figura 2.3: Il processo di classificazione[3]

Esistono differenti tecniche di classificazione ciascuna con le proprie caratteristiche; bisogna quindi trovare il giusto compromesso tra i vari indici di qualità e scegliere, di volta in volta, l'algoritmo migliore per quella determinata necessità che si va ad affrontare. Ognuna di queste tecniche è sempre formata da due fasi:

- **Induzione**: creazione del modello partendo dal training set ed utilizzando uno degli algoritmi di classificazione.
- **Deduzione**: step nel quale viene applicato il modello al test set, in modo da valutarne l'efficacia.

Considerando quanto appena esposto, per le sperimentazioni di questa tesi, sono stati scelti i seguenti quattro classificatori:

- Alberi decisionali (*Decision Tree*)
- SVM (*Support Vector Machine*)
- Classificatore bayesiano (*Naïve Bayes*)
- *Reti neurali*

2.2.1 Alberi decisionali

Questa tecnica è la più semplice tra quelle di classificazione e permette la generazione di un modello interpretabile. Si basa sulla creazione di un albero a nodi, mediante la suddivisione ripetuta dei record in sottoinsiemi omogenei rispetto l'attributo di classe. Ogni nodo intermedio dell'albero contiene un *attributo di split*, ovvero l'attributo della classe utilizzato per dividere l'albero e per costruire i diversi cammini; i nodi foglia, invece, contengono l'attributo di classe. Dato un dataset è possibile costruire diversi alberi decisionali, rendendo quindi importante la scelta del migliore tra questi. Esistono diversi algoritmi che realizzano la fase di induzione e uno dei più antichi risulta essere l'*algoritmo di Hunt*.

La struttura generale di questo algoritmo è la seguente:
definito come D_t l'insieme dei dati di training che raggiungono un nodo t, gli step sono:

1. se D_t contiene record che appartengono tutti alla stessa classe y_t, allora t è un nodo foglia etichettato come y_t;
2. se D_t è vuoto, allora t è un nodo foglia etichettato con la classe di default y_d (la classe di default è quella con la più alta probabilità);
3. se D_t contiene elementi che appartengono a più classi, allora viene ulteriormente diviso in dataset più piccoli.

Questa procedura viene ripetuta ricorsivamente per ogni sottoinsieme di dati.

La fase di *induzione* utilizza una strategia di tipo *Greedy*: la scelta dell'attributo di split viene effettuata localmente non scegliendo quindi l'ottimo in maniera globale.

[3]Immagine tratta da: E. Baralis. "Classification fundamentals"

Gli aspetti più importanti e problematici di questo algoritmo sono:

- Struttura dello split, vale a dire scegliere quale metodologia di suddivisione sia la migliore (albero binario o albero n-ario)
- Scelta del miglior attributo di suddivisione
- Condizione di stop dell'algoritmo

Struttura dello split

La scelta della condizione di split dipende dalla tipologia dell'attributo e varia a seconda del dataset; tipicamente, però, è possibile distinguere tra "*binary split*" e "*multi-way split*". Nel caso di split binario, o 2-way split, ogni nodo padre ha al più due nodi figli; nell'altro caso, l'albero presenta nodi che possono avere da uno a n figli. Sia nel caso di attributi nominali che ordinali, il multi-way split utilizza un numero di partizioni pari al numero di valori distinti tra loro; lo split binario, invece, divide i dati in due sottoinsiemi, dando molta importanza alla ricerca del corretto partizionamento. In caso di attributi continui, la situazione è più complessa ed è possibile procedere in diversi modi: è possibile utilizzare la *discretizzazione* in modo tale da creare un attributo di tipo discreto, dividendo gli attributi continui in intervalli; questa modalità può essere statica, quando viene realizzata una sola volta all'inizio del processo, o dinamica, dove la discretizzazione viene realizzata, in caso di necessità, durante la fase di induzione. Un'altra possibilità è quella di scegliere un unico valore su cui effettuare la suddivisione. Per realizzare questo criterio è necessario considerare tutte le possibili divisioni e trovare la migliore tra queste; per questo motivo, è molto più intensivo a livello computazionale.

Selezione del miglior attributo

Il modo più comune di scegliere l'attributo di split è quello di utilizzare un attributo che generi nodi omogenei. Vengono scartati tutti quegli attributi che genererebbero troppi sottoinsiemi, in quanto diventerebbero inutili; utilizzando questi attributi il rischio diventerebbe quello di creare un modello che funzioni solo con il dataset di training. Affinchè sia possibile creare nodi omogenei è stato necessario introdurre la misurazione dell'impurità di un determinato nodo. Solitamente, viene calcolata la purezza prima e dopo lo split, con l'obiettivo di scegliere la soluzione che garantisca un guadagno maggiore. Dato che il guadagno viene calcolato come differenza tra purezza prima e dopo lo split, nel caso di un guadagno elevato risulterà che il grado di impurità della partizione ottenuta sarà basso. I metodi maggiormente utilizzati per valutare la qualità di una suddivisione sono l'*indice di eterogeneità di Gini* e l'*entropia*.

Condizione di stop dell'algoritmo

Solitamente, l'algoritmo termina quando tutti i record del nodo appartengono alla stessa classe (*nodo puro*) o quando non sono presenti attributi all'interno del sottoinsieme. In realtà, esistono altre due condizioni per quali l'elaborazione può terminare: tutti i dati hanno attributi simili o sopraggiunge la cosiddetta *early termination*, "terminazione anticipata".

La conclusione anticipata dell'algoritmo sopraggiunge a causa delle tipiche problematiche legate alla classificazione: nel caso in cui il modello sia troppo preciso e dettagliato e siano stati creati moltissimi nodi in modo da ottenere un partizionamento puro, il rischio è quello che il modello non funzioni su dataset diversi da quello di training. Quando accade ciò il numero di errori cresce e di conseguenza diminuisce l'accuratezza. Questo fenomeno è chiamato *overfitting*. Può capitare anche il problema opposto, denominato *underfitting*, dove il modello è troppo semplice e il numero di errori in fase di test è elevato.

Per evitare l'overfitting esistono due soluzioni:

- **Pre-pruning**. Con questa tecnica si definiscono dei criteri che facciano fermare la creazione dell'albero prima che si possano verificare fenomeni di *overfitting*. Per fare ciò, si effettua il cosiddetto *look-ahead*: si effettua una previsione su ciò che accadrà e se si considera conveniente proseguire verrà effettuato lo split, in caso contrario ci si fermerà. Ci sono due possibilità per stoppare l'algoritmo: il numero di istanze è inferiore ad una determinata soglia oppure un'ulteriore split non comporta un miglioramento della purezza del nodo.

- **Post-pruning**. In questo caso viene effettuata un'elaborazione sul modello in seguito alla sua costruzione. L'albero viene scandito a partire dai nodi foglia (*bottom-up*) e vengono realizzati dei tagli per ogni sotto-albero. Se il taglio comporta un miglioramento dell'accuratezza viene mantenuto; in caso contrario ne viene valutato un altro.

Vantaggi e svantaggi

Gli alberi decisionali risultano poco costosi da costruire e, dovendo solamente visitare l'albero, garantiscono una veloce classificazione dei dati non ancora etichettati. Oltre tutto sono in grado di fornire un modello facilmente interpretabile. L'accuratezza, nel caso di dataset semplici e piccoli, risulta comparabile alle altre tecniche di classificazione.

Uno dei principali svantaggi nell'utilizzo degli alberi è la loro forte penalizzazione nel caso di dati mancanti; nel caso in cui qualche attributo sia assente, diventa difficile visitare l'albero e assegnare l'etichetta di classe se la suddivisione è realizzata proprio su quell'attributo mancante. Oltre tutto, se anche nel training set non è presente qualche attributo la costruzione del modello diventa decisamente complessa.

2.2.2 SVM

Le *Supported Vector Machine*, spesso abbreviate con SVM, permettono di realizzare sia tecniche di regressione sia tecniche di classificazione. Hanno come obiettivo quello di trovare un iperpiano che separi i dati; tra tutti quelli disponibili, viene scelto l'iperpiano che massimizza il margine, ovvero la distanza tra i punti più vicini al piano scelto.

Osservando la figura 2.4, si può facilmente notare come l'iperpiano B1 sia migliore di B2 avendo un margine superiore. Dovendo trovare il margine migliore, in questo caso, il problema della generazione del modello è definito sotto forma di problema di ottimizzazione.

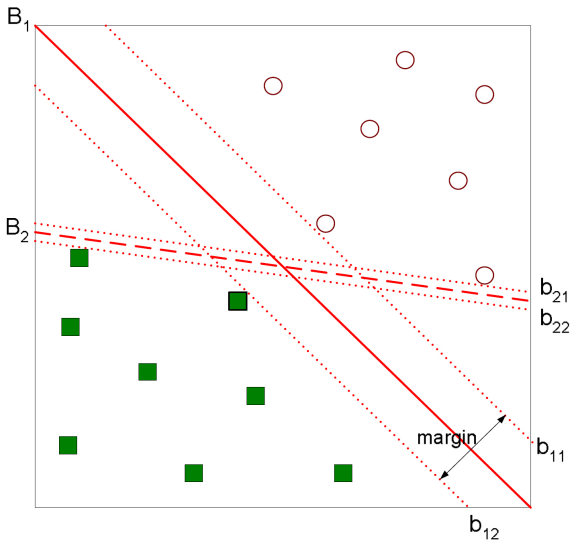

Figura 2.4: Iperpiano e margine nelle SVM[4]

Nel caso in cui l'iperpiano di separazione non sia lineare, è possibile trasformare i dati passando a uno spazio multidimensionale; le coordinate verranno quindi modificate di conseguenza. Per ogni problema non lineare possono esistere però diverse trasformazioni possibili; nella tecnica SVM esistono, a questo scopo, diverse tipologie di *kernel*, ognuno dei quali è in grado di risolvere un problema di natura differente (lineare, polinomiale, esponenziale ecc...). Per utilizzare questo classificatore, non si ha la necessità di conoscere la natura del problema, ma è possibile effettuare le sperimentazioni testando i diversi kernel e misurando, in seguito, la qualità del risultato.

Vantaggi e svantaggi

Tra i vantaggi delle SVM troviamo il fatto che sono in grado di risolvere problemi anche non lineari. Le loro prestazioni sono buone in caso di problemi complessi, ma non è detto che con problemi più semplici siano sempre in grado di ottenere delle buone performance. In alcuni casi, infatti, si ottiene un livello di accuratezza pari a quello degli alberi decisionali. Sono particolarmente indicati, per esempio, per l'elaborazione dei testi.

I principali svantaggi sono due: la creazione del modello può non essere breve e il modello non è interpretabile. Anche in questo caso si deve arrivare ad un compromesso: per guadagnare in accuratezza si perde di interpretabilità.

[4]Immagine tratta da: P. P. Tan, M. Steinbach e V. V. Kumar. *Introduction to Data Mining*

2.2.3 Classificazione bayesiana

Tecnica, di tipo statistico, utilizzata per calcolare la probabilità che un elemento possa appartenere ad una determinata classe. Questo classificatore si basa sull'applicazione del *teorema di Bayes*.

Supponiamo che C e X siano due variabili casuali. E' possibile definire, utilizzando la probabilità condizionata e la probabilità a priori in una sola variabile, la probabilità congiunta di trovarle entrambe come:

$$P(C, X) = P(C|X) \cdot P(X) \tag{2.1}$$

$$P(C, X) = P(X|C) \cdot P(C) \tag{2.2}$$

Uguagliando le formule 2.1 e 2.2 è possibile ottenere:

$$P(C|X) = P(X|C) \cdot \frac{P(C)}{P(X)} \tag{2.3}$$

che risulta essere la definizione del teorema di Bayes.

Applicando la formula 2.3 nel campo della classificazione, è possibile esprimere le etichette di classe attraverso la variabile C, mentre la variabile X rappresenta i valori degli attributi di un record.

Vogliamo calcolare quindi, dato un record X, per ogni classe C la probabilità di C dato X e sceglieremo così la classe che ci ha restituito la probabilità più alta. Grazie al teorema di Bayes, sarà possibile sostenere che quella determinata classe è quella con il legame più alto con quel determinato record. Fatte queste premesse, ci rendiamo conto di come $P(X)$ risulti essere costante per tutte le classi e calcolarlo diventerebbe inutile: verrà quindi tralasciato nel calcolo. $P(C)$ risulta essere la probabilità a priori di trovare la classe C, quindi basterà calcolare, sul training set il valore N_c/N, cioè il rapporto tra il numero di record che contengono quell'etichetta di classe e il numero totale dei record stessi. Infine, per calcolare $P(X|C)$ entra in gioco l'ipotesi Naïve tipica di questo classificatore. Ipotizzando infatti l'indipendenza statistica degli attributi $x_1, \ldots, x_k$ è possibile effettuare una notevole semplificazione ed il calcolo della probabilità diventa:

$$P(x_1, \ldots, x_k|C) = P(x_1|C) \cdot P(x_2|C) \cdot \ldots \cdot (x_k|C) \tag{2.4}$$

Grazie al risultato ottenuto con la formula 2.4) sarà possibile assegnare, anche se sotto una forte ipotesi, un'etichetta di classe ricorrendo a dei semplici calcoli.

Vantaggi e svantaggi

L'algoritmo Naïve Bayes risulta essere facile da costruire e particolarmente utile nel caso di dataset ampi.

Il modello risulta interpretabile se si va ad analizzare il modo con cui vengono assegnate le varie etichette; è possibile, ad esempio, identificare i termini dominanti nel calcolo e comprendere quali sono gli attributi che pilotano la scelta dell'etichetta di classe. Non sono però in grado di descrivere le classi solamente analizzando il modello.

Il vantaggio principale nell'utilizzo del classificatore bayesiano è che il modello costruito è incrementale. E' molto semplice infatti aggiungere nuovi record etichettati

e modificare il modello poiché questa operazione richiede solamente la modifica delle probabilità coinvolte. Oltretutto, il training set può anche essere stato eliminato senza che questa possibilità venga meno: è il modello stesso ad essere incrementale. Questa caratteristica diventa molto importante, ad esempio, nella creazione di un filtro anti-spam; dopo aver costruito il modello si può continuare ad evolverlo ogni volta che viene etichettata un nuova tipologia di mail.

Il principale svantaggio è che la qualità del modello dipende da quanto l'ipotesi naïve sia vera. L'accuratezza dell'algoritmo può facilmente essere influenzata dall'assunzione dell'indipendenza degli attributi ma, nel caso tutto ciò trovi riscontro nella realtà, questa tecnica di classificazione risulta paragonabile agli alberi decisionali e alle reti neurali.

2.2.4 Reti neurali

Metodo proposto all'inizio degli anni '80 ispirandosi al funzionamento del cervello umano; i neuroni vengono assimilati a delle piccole unità di elaborazione, mentre le sinapsi vengono considerate come se fossero la rete di connessione che permette di scambiarsi informazioni.

Analizzando questa tecnica, si può notare come il modello sia realizzato attraverso una rete composta di nodi. Possono essere presenti due o più livelli: quello formato dai *nodi di ingresso*, uno o più livelli con i cosiddetti *nodi nascosti*, e un livello con i *nodi di uscita*; quest'ultimo livello è quello che fornirà l'etichetta di classe. Dal punto di vista strutturale ogni nodo di un certo livello è connesso in maniera completa a tutti i nodi del livello successivo (figura 2.5).

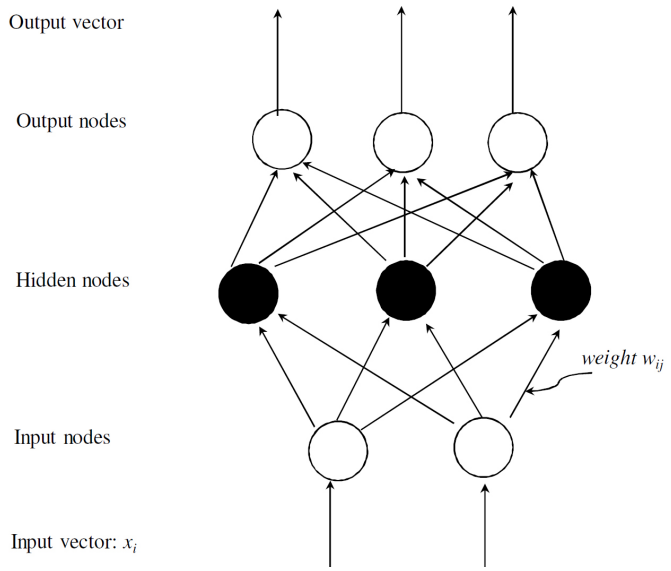

Figura 2.5: Rete neurale[5]

In caso di problema binario, la rete sarà formata soltanto da un nodo di uscita; nel caso invece siano presenti due o più etichette di classe, il numero di nodi deve essere pari al numero di quest'ultime.

[5]Immagine tratta da: Han Kamber. *Data mining; Concepts and Techniques.* 2006

Oltre ad assegnare l'etichetta, i *nodi output* forniscono anche un valore che esprime la percentuale di possibilità di appartenenza, da parte di un record, a quella determinata classe. In uscita non si avrà quindi un valore continuo (ad esempio, uno o zero), ma una percentuale che indica una *"probabilità"*. Il numero dei nodi di ingresso, invece, è dipendente dal numero e dal tipo di attributi e, in linea più generale, dalla struttura del dataset. Nel caso un attributo sia continuo è sufficiente un singolo nodo per modellarlo; se l'attributo è di tipo discreto il numero di *nodi input* deve essere pari al numero di valori presenti nel dominio dell'attributo stesso.

Poiché, spesso, gli attributi continui presentano intervalli diversi tra di loro, le reti neurali, solitamente, ricevono in ingresso dei valori normalizzati; in questo modo, il dominio di variazione viene ricondotto ad uno solo potendo così confrontare tra di loro diversi attributi.

I nodi più interessanti, dal punto di vista della costruzione del modello, risultano essere i nodi nascosti. Questa tipologia di nodo, come è visibile nella figura 2.6, è caratterizzata da un *vettore di pesi* e da un valore denominato *offset*. Attraverso questi due dati, viene calcolato un valore che viene poi normalizzato attraverso l'applicazione di quella che viene definita *funzione di attivazione*.

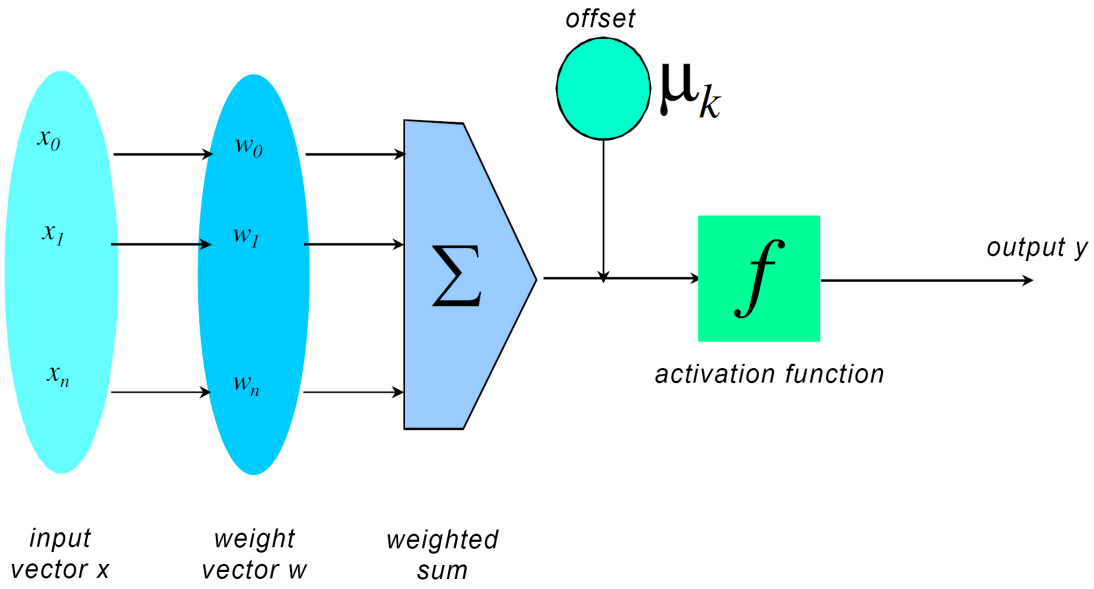

Figura 2.6: Struttura di un nodo nascosto[6]

Ma qual è il processo che viene eseguito in un nodo nascosto?
L'input di tale nodo è formato da un vettore; questi dati, vengono moltiplicati, attraverso un prodotto scalare, con il vettore dei pesi. Al risultato di questa operazione verrà poi applicato il valore di offset, prima di essere normalizzato grazie alla funzione di attivazione. Questo processo viene realizzato all'interno di ciascun dei nodi nascosti e all'interno dei nodi di uscita.

La definizione dell'offset e del valore dei pesi è molto importante per la realizzazione di una rete neurale che garantisca una buona qualità sui dati di training. L'approccio più semplice è un approccio iterativo sul training set. All'inizio del processo avrò un vettore dei pesi e un offset scelti in maniera totalmente casuale.

Utilizzando questi valori di partenza, il training set, un record alla volta, viene propagato all'interno della rete. In questo modo si ottiene una prima predizione che verrà poi comparata con l'effettiva etichetta di classe di ogni input. Diventa quindi possibile valutare la qualità della predizione e calcolare lo scostamento rispetto il valore reale. Questo errore appena ottenuto viene quindi propagato all'indietro (*back propagation*) e utilizzato per correggere il vettore dei pesi e l'offset di ogni nodo. Tale algoritmo viene eseguito per tutte le istanze del training set e ripetuto numerose volte (si definisce *epoca* il completo passaggio dell'intero training set all'interno del modello). La condizione di terminazione può essere:

- Percentuale di accuratezza superiore a una certa soglia. In questo caso il rischio è di raggiungere l'*overfitting*.
- Variazione dei parametri sotto una determinata soglia. Situazione che si verifica quando il variare l'offset e il vettore dei pesi non comporta alcun miglioramento: si è quindi raggiunto l'ottimo locale.
- E' stato raggiunto il numero di epoche stabilito.

Vantaggi e svantaggi

Questo classificatore permette la creazione di un modello di qualità elevata, grazie al fatto che, ad ogni epoca, gli *outliers* e i dati ricchi di rumore vengono via via messi da parte e marginalizzati. Risulta quindi essere una tecnica molto robusta. Il rovescio della medaglia è che, data questa natura, il processo di training risulta lento, poco scalabile e difficilmente configurabile; servono numerose esecuzioni prima di trovare i parametri migliori. Nonostante la lentezza durante la fase induttiva, il processo di deduzione risulta molto efficiente. Un altro punto a favore di questa tecnica è il fatto che consente di predire un valore continuo, supportando quindi nativamente anche la regressione.

Il modello non risulta interpretabile; questo tipo di classificazione è quindi da utilizzare quando si è alla ricerca di un'elevata accuratezza ma non serve conoscere come avviene la decisione dell'etichetta di classe.

[6]Immagine tratta da: Han Kamber. *Data mining; Concepts and Techniques*. 2006

Capitolo 3

Il trading

Con il termine *trading*, cioè commercio, si intende l'andamento di vendita o di acquisto di un titolo in Borsa. Questa attività ha come obiettivo quello di realizzare dei guadagni sul mercato finanziario e, ad oggi, è quasi totalmente realizzata online. L'attività di compravendita nella borsa valori è sempre avvenuta: prima con telefono e foglio in mano ed ora attraverso dei software sviluppati ad hoc.

Detto questo, si può facilmente comprendere come il campo del data mining e, più in generale, quello dell'informatica abbiano iniziato ad influenzare e partecipare in questo settore [32].

3.1 Cenni di storia

Con il termine *Borsa*, in economia, si intende un mercato organizzato e periodico, dove intermediari specializzati trattano contratti d'acquisto e vendita per determinati tipi di merci, servizi o strumenti finanziari, sotto l'osservanza di speciali norme [34].

Già a partire dall'antica Grecia e dall'antica Roma erano presenti figure assimilabili agli attuali operatori di borsa; l'*Agorà* e il *Forum*, ad esempio, possono essere considerati come il primo esempio di mercato borsistico, dato che racchiudevano diverse attività di tipo finanziario.

Le borse nascono, a partire dai mercati e dalle fiere medioevali, come luogo dove avveniva la compravendita di merci e venivano fissati i prezzi dei principali prodotti in base alla legge della domanda e dell'offerta. A partire dal XIV e XV secolo le borse iniziano a diventar permanenti nelle principali città europee e nacquero così le prime *Borse valori*, dove avveniva la compravendita di valuta, il cambio di assegni e la compravendita di assicurazioni. È proprio in questo periodo che si inizia a parlare di *"Borsa"*; questo termine, infatti, deriva dalle riunioni, tenute a Bruges presso l'Hotel de Bourses, per determinare il valore delle merci intorno al 1400. In questi anni inizia il cammino delle borse per passare da una gestione privata ad una pubblica, disciplinata da regole forti.

Le borse valori iniziano a diventare sempre più diffuse ed importanti con lo sviluppo, a partire dal XVII secolo, delle società per azioni e dei titoli di stato. Ma la loro vera e propria affermazione avviene con la seconda rivoluzione industriale e con lo sviluppo del capitalismo. In quel periodo, la sempre più grande necessità di disporre di grandi somme di denaro, ha permesso il definitivo sviluppo della borsa;

infatti, attraverso le società per azioni diventò più facile raccogliere grandi quantità di capitali.

È nella seconda metà del XIX secolo che, in tutte le principali economie mondiali, vengono approvate leggi che disciplinano il mercato azionario. Grazie a questa regolamentazione la borsa di Londra diventa la prima borsa mondiale, per poi essere scalzata, a partire dalla fine della prima guerra mondiale, da quella di New York [18].

Figura 3.1: Wall Street: la borsa di New York[1]

3.2 L'intraday trading

Con trading intraday si intendono tutte quelle operazioni atte a ricavare un profitto nel brevissimo tempo; infatti, il trading intraday, solitamente implica l'apertura e la chiusura dell'operazione entro pochissime ore o al più entro la stessa giornata. Detto questo, si può capire come questa modalità risulti tra le più difficili e complesse; la stessa azione può avere delle notevoli oscillazioni di prezzo durante la stessa giornata e la scelta dell'operazione da eseguire diventa uno nei nodi cardine di questa attività.

Per poter operare cercando di limitare i rischi, diventa necessario studiare il mercato e le sue dinamiche di prezzo; cosa fondamentale sarà impostare uno *stop loss*, affinché si riesca a ridurre al minimo le perdite [33]. La scelta della strategia operativa coinvolge, tra le altre cose, l'analisi tecnica. La valutazione di valori statistici e l'individuazione di pattern grafici può aiutare il trader nella scelta corretta; analizzare tutti questi dati però obbliga a dover operare su un limitato numero di azioni, per ovvi limiti "umani". È in questo scenario che entrano in gioco tutti quegli strumenti informatici che sono in grado di effettuare le operazioni appena descritte in maniera automatica. Oltre a questo, l'evoluzione tecnologica permette di effettuare

[1]Immagine tratta da: *Financial Planning for Professional Athletes: An Inside Look*. URL: `http://proathletewealthadvisor.com/financial-planning-professional-athletes-inside-look/`

delle predizioni analizzando il passato, l'andamento attuale e persino l'andamento di altri mercati. Si può quindi cercare di capire come e se quel determinato titolo sia influenzato da altri.

3.3 Applicazione di modelli di data mining in ambito finanziario

Uno dei problemi principali dell'operare in borsa è che il mercato azionario non è lineare, è difficilmente modellabile ed effettuare predizioni che siano accurate è quasi impossibile [37].

A partire dalla metà degli anni '90, come visibile nella figura 3.2, c'è stata una rapida evoluzione della tecnologia riguardante gli hard disk; con questo miglioramento è emersa la possibilità di memorizzare una quantità di dati impensabile fino a poco tempo prima e di conseguenza gli analisti si sono trovati impreparati a gestire una quantità così elevata di informazioni. E' chiaro quindi come i metodi automatici iniziassero a diventare fondamentali.

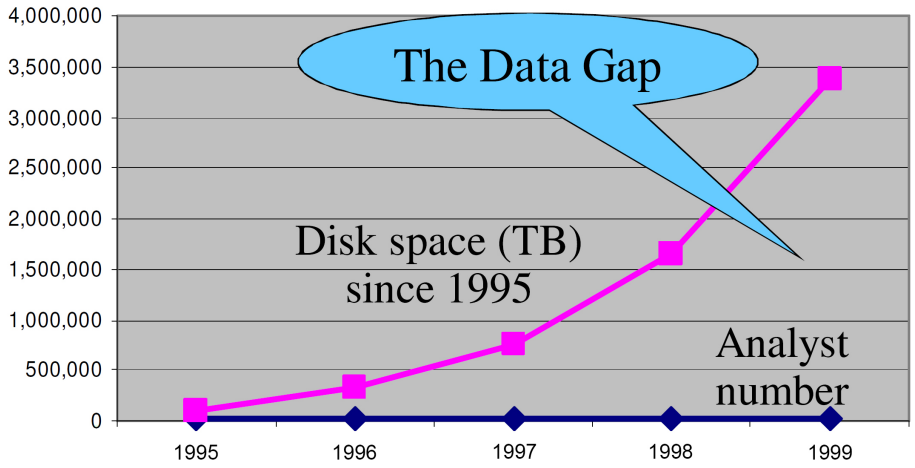

Figura 3.2: Il trend di crescita dei dati[2]

A partire dal 1999, è stata introdotta la possibilità di effettuare il *trading online* attraverso delle piattaforme fornite da intermediari, chiamati *broker* [32], Questa innovazione ha reso più facile operare in borsa potendo effettuare compravendita semplicemente possedendo un computer o, addirittura, uno smartphone o tablet. Il trading è diventato quindi alla portata di tutti e questo fattore, unitamente alla continua evoluzione tecnologica, ha portato allo sviluppo di numerosi sistemi automatici o semi-automatici; tra questi, hanno preso sempre più piede i cosiddetti *Automatic Trading Systems*. Questi sistemi sono applicazioni che creano ed effettuano ordini sui mercati in maniera totalmente automatica, basandosi su strategie e regole configurate al loro interno. Per fare ciò, la maggior parte di questi tool utilizza metodi derivati dall'analisi tecnica classica, ma possono supportare anche input da altre fonti. Nonostante all'interno di questi sistemi siano disponibili numerosi

[2]Immagine tratta da: R. R. Grossman, C. Kamath e V. Kumar. *Data Mining for Scientific and Engineering Applications*

indicatori, la maggior parte dei trader sceglie di lavorare a stretto contatto con un programmatore, in modo tale da sviluppare strategie personalizzate: questo maggior sforzo consente un maggior grado di libertà e può portare a migliori risultati. Dopo la configurazione, il sistema inizia a monitorare i mercati alla ricerca di titoli da comprare o vendere secondo le regole settate al suo interno; quando ne trova uno che soddisfa i criteri piazza un ordine praticamente istantaneo ed effettua l'operazione. Il maggior beneficio ottenibile utilizzando questi software deriva dal fatto che sono in grado di analizzare e processare notevoli quantità di informazioni ad una velocità impensabile per un essere umano; inoltre, è possibile testare le regole configurate su dati storici prima di iniziare il vero e proprio trading *back-testing*.[1] Di contro, viene eliminata tutta quella parte di analisi e valutazione dei rischi propria del giudizio umano. Se da un lato, non considerare le emozioni può sembrare un notevole punto a favore, dall'altro può creare notevoli problematiche come successo con il 2010 Flash Crash.[9]

3.4 Lo stato dell'arte: tecnologie e tecniche

A seconda del campo di analisi, dell'obiettivo e della natura dei dati, durante questi anni sono state studiate e sperimentate moltissime tecniche. La completa comprensione del mercato azionario e il tentativo di effettuare predizioni hanno due origini: la prima è il *settore economico* ed è fortemente legata all'applicazione di modelli matematici; la seconda deriva dall'implementazione di algoritmi derivanti dal campo dell'analisi dati (*data science*) e dal settore informatico più in generale. Entrando nello specifico di quest'ultimo, si può dire che la maggior parte delle sperimentazioni ha l'obiettivo di utilizzare tecniche di data mining per migliorare l'analisi nel settore finanziario e la predizione dei trend. Inizialmente queste tecniche sono state viste con sospetto ma, a partire dagli anni novanta, lo scetticismo iniziale è andato via via a scomparire una volta che si è dimostrata la loro efficacia.

Il data mining si è contrapposto ad un altro grande metodo di analisi di dati: l'*econometria*. Secondo Breiman, come descrive nel suo "Statistical modeling: The two cultures", esistono due culture contrapposte per modellare i dati: la prima, l'econometria appunto, utilizza la statistica tradizionale; la seconda sfrutta gli algoritmi di ottimizzazione propri del machine learning. L'econometria effettua la propria analisi prendendo in considerazione anche l'ambiente di sperimentazione; il data mining invece evita di fare delle ipotesi sul comportamento di quest'ultimo o sul modello [4]. Riassumendo, si può dire che l'analisi tecnica è basata su procedure che hanno origine nell'ambito economico; l'analisi attraverso questi metodi è di natura grafica e fortemente basata sul lavoro manuale del trader. Le tecniche di data mining invece permettono di automatizzare l'estrazione di informazione, rimuovendo l'ipotesi che il ricercatore conosca in maniera approfondita le dinamiche del settore finanziario. A causa della rimozione di questa ipotesi, almeno inizialmente, il data mining non ha trovato l'approvazione da parte degli economisti. Sebbene nel passato questi due filoni di analisi fossero totalmente slegati, al giorno d'oggi, diversi studi, tra cui quello realizzato da Vairan nel 2014, hanno provato a intersecare l'analisi tecnica con i metodi forniti dal data mining. Questi metodi si sono rivelati molto utili per elaborare la grande mole di dati e, grazie a tale risultato, ci si è resi conto di come sia importante creare una collaborazione tra gli econometristi e gli informatici [35].

In questa tesi, come descritto in seguito, è stata realizzata una sperimentazione anche in quest'ottica.

Concentriamoci ora sulle tecniche di machine learning.
Grazie a queste tecniche è possibile, partendo da dei data set di input, creare dei pattern; utilizzando questi ultimi, l'obiettivo è quello di trovare l'espressione in grado di generare le informazioni originarie. Uno degli ultimi studi, che centra in maniera precisa questa tematica, è quello condotto da Milosevic nel 2016 [21]. Questa ricerca ha avuto come obiettivo quello di utilizzare tecniche di data mining per predire l'andamento dei prezzi nel futuro. Per la creazione del modello sono stati scelti i seguenti algoritmi di classificazione: SVM con ottimizzazione minima sequenziale, alberi decisionali, alberi casuali, foreste casuali, reti bayesiane e Naïve Bayes. La simulazione è stata in grado, nel 76.5% dei casi, di predire correttamente se un azione sarebbe salita o meno di almeno il 10%. Lo studio appena descritto è stato realizzato considerando le azioni appartenenti ai maggiori indici tra cui l'S&P1000, il FTSE 100 e l'S&P Europe 500[3]. La scelta dei mercati da analizzare e la loro influenza sul risultato è una delle tematiche principali su cui sono stati eseguiti numerosi studi. Sotto questo aspetto, possiamo citare anche la ricerca effettuata da Haur Koh, Phua e Zhu. dove sono stati scelti come indici il DAX, il NASDAQ, il DJIA e l'HSI[4].

In letteratura sono presenti numerosi lavori che illustrano come il data mining possa essere interconnesso al campo finanziario. Nelle prossime sezioni, verranno quindi esposte le principali tecniche di financial data mining su cui è stata effettuata attività di ricerca; nello specifico verranno trattati gli algoritmi di particolare interesse per la sperimentazione effettuata in questa tesi.

3.4.1 Gli studi basati sulle reti neurali

Il primo tentativo eseguito per potenziare le performance dei metodi di associazione è stato quello di migliorare la lettura dei dati; a questo scopo, sono state scelte le reti neurali dal momento che sono in grado di leggere molti record contemporaneamente.

Come già discusso precedentemente, un modello realizzato attraverso una tecnica di data mining è privo di qualsiasi ipotesi iniziale. Trascurando persino l'ipotesi che le azioni e le variabili economiche abbiano un legame lineare, la necessità diventa quella di utilizzare un classificatore che permetta un'analisi non lineare [20][25]. Le reti neurali possono soddisfare tale necessità dato che riescono a determinare le relazioni tra le variabili durante il processo di costruzione del modello. Questa caratteristica diventa particolarmente utile nelle aree finanziarie come ad esempio quella degli investimenti o della predizione del prezzo di un'azione; oltretutto, non avendo la necessità di alcuna ipotesi a priori, questo modello rispetta la teoria dei mercati efficienti. L'EMH, ovvero l'*Efficient Market Hypothesis*, sostiene che il prezzo attuale di un titolo rispecchia completamente tutte le relative informazioni [7][6].

[3]S&P1000: indice creato da Standard & Poor's comprendente le maggiori mille compagnie statunitensi; S&P Europe 500: indice creato da Standard & Poor's formato dalle principali 500 azioni europe; FTSE 100: indice azionario delle 100 società più capitalizzate quotate al *London Stock Exchange*.

[4]DAX: segmento della Borsa di Francoforte; NASDAQ: indice dei principali titoli tecnologici della borsa americana; DJIA: il più noto indice azionario della borsa di New York; HSI: indice di Hang Seng.

In pratica, le variazioni nei prezzi sarebbero dovute solamente alla disponibilità di nuove informazioni [5].

In letteratura sono disponibili numerose ricerche riguardanti le reti neurali. Tra le più recenti e significative troviamo il lavoro realizzato da Hargreaves e Hiao nel 2012. In questo studio sono state confrontate cinque diverse strategie di trading: una scelta dal trader, due realizzate con un albero decisionale (CHAID e C5.0), una che utilizza una rete neurale e l'ultima che utilizzava un algoritmo di regressione lineare. Nel dettaglio, lo scopo della ricerca è stato quello di estrarre le sei azioni con la più alta probabilità di aumentare il proprio prezzo. Per confrontare i risultati ottenuti sono state calcolate, per ogni strategia, la *sensibilità*[5] e la *specificità*[6]. Osservando i valori ottenuti, la rete neurale ha ottenuto la più alta sensibilità (93%), insieme al classificatore C5.0 (98%), ma contemporaneamente anche la più bassa specificità (31%). Considerando però che l'obiettivo è quello di classificare correttamente le azioni che hanno un trend rialzista, bisogna evidenziare che il valore più importante è la sensibilità. Queste conclusioni ci obbligano ad effettuare una continua attività di ricerca nell'ambito delle reti neurali, esistendo ancora infinite configurazioni da esplorare [11].

I vantaggi e gli svantaggi di utilizzare una rete neurale sono stati una tematica su cui diversi ricercatori hanno effettuato i loro studi. La gestione di un portafoglio di investimento è lo scenario ideale per l'utilizzo di questa tecnica. Inoltre, la possibilità di automatizzare tutto il processo ha reso decisamente interessante questo modello; persino le più grandi compagnie potrebbero non essere in grado di analizzare manualmente un numero elevato di azioni. Una rete neurale, oltretutto, presenta una forte adattabilità rendendola perfetta nel caso in cui si decida di investire molto spesso, ad esempio ogni giorno [36][38]. Il principale problema delle reti neurali è la difficile interpretabilità: il risultato della modellazione infatti risulta spesso oscuro, rendendo impossibile un'interpretazione da parte di una persona reale [13].

3.4.2 Gli studi basati su alberi di decisione

Nonostante le reti neurali forniscano un buon livello di accuratezza, il modello che creano non è facilmente comprensibile ed interpretabile da un utente in carne ed ossa. Per questo motivo, da sempre, sono state proposte in contrapposizione delle soluzioni che possano fornire delle regole facili da comprendere. Per di più, se le regole sono facilmente esprimibili possono essere riutilizzate anche per altri task.

Per tutte queste motivazioni, negli ultimi quaranta anni c'è stato un continuo studio ed implementazione degli alberi binari; questa tipologia di tecnica ha avuto notevole successo, oltre che nel campo finanziario, anche in molti altri campi. Tra le applicazioni di questi metodi troviamo, ad esempio, la valutazione dell'idoneità ad una carta di credito, la predizione di eventuali rate in ritardo o il controllo di eccessive richieste di rimborso nel caso di un'assicurazione sanitaria [27].

Nel 2005, Vityaev e Kovalerchuk hanno realizzato uno studio concentrandosi sui decision tree e, in particolare, hanno posto enfasi ai casi *borderline* [36]. Infatti, alla fine degli anni ottanta, American Express UK ha effettuato dei test per automatizzare l'accettazione o meno di prestiti; il sistema è risultato possedere ottime

[5]Sensibilità: cogliere tutti i fenomeni oggetti d'interesse

[6]Specificità: classificare correttamente tutti i fenomeni osservati.

prestazioni a patto di non essere tra i casi "al limite". In quei casi è stato necessario l'intervento e l'analisi umana.

Grazie alla sperimentazione appena descritta si è iniziato a porre attenzione al problema dei casi borderline e la ricerca sopra citata ha cercato di concentrarsi su questa tematica. Entrando nello specifico, i ricercatori hanno implementato un sistema che cercasse di predire la direzione (salita o discesa) di un set di azioni appartenenti all'indice S&P500. Come input hanno scelto di utilizzare dieci anni di dati storici per la fase di training e quattro anni per quella di test. L'obiettivo è risultato essere la predizione, utilizzando i precedenti dieci giorni storici, dell'andamento di una determinata azione. La sfida dietro questo lavoro è stata quella di trovare un albero decisionale che potesse effettuare una predizione corretta ma che avesse anche delle dimensioni ragionevoli. Il modello finale, utilizzando l'algoritmo C4.5, ha prodotto un'accuratezza media del 55.6% ma le dimensioni finali dell'albero sono risultate essere decisamente grandi. Questo ha portato a concludere che, nel caso si abbiano numerose informazioni da elaborare, l'utilizzo degli alberi decisionali è troppo limitato e restrittivo [36].

Negli ultimi anni, sono state realizzate numerose altre ricerche con lo scopo di migliorare la generazione di segnali automatici di trading grazie all'utilizzo degli alberi decisionali. Tra queste troviamo, senza dubbio, quella sviluppata da Isern-Deya, Miro-Julia e Fiol-Roig nel 2010 e quella sviluppata da Thakur, Kamley e Jaloree nel 2014.

Il primo studio ha analizzato i dati giornalieri tra il 1995 e il 1998, utilizzando come attributi i valori di chiusura e di apertura, i valori di minimo e massimo di giornata e il volume delle operazioni. I dati raccolti sono stati quindi convertiti in pattern tipici dell'analisi tecnica quali MACD, EMA(C), EMA(V), K e BB[7]. Durante la creazione dell'albero, è stato notato come gli attributi relativi al MACD e a K facciano parte della radice dell'albero, confermando le idee degli analisti sulla loro validità ed importanza come indicatori. La fase di test è stata eseguita utilizzando dei dati con caratteristiche simili a quelli di addestramento; su queste informazioni è stato poi applicato il modello creato precedentemente e quindi analizzati i risultati ottenuti. È risultata un'accuratezza compresa tra il 40% e il 50%, mentre il profitto ottenuto è stato pari all'88% nel caso di albero a due etichette e pari al 118% in caso di albero a tre etichette. Nonostante il risultato ottenuto non abbia mostrato un valore di accuratezza sorprendente, il guadagno medio è risultato essere di buon livello; questo dimostra come l'analisi dei mercati finanziari sia un campo molto imprevedibile e del tutto sorprendente[14].

Il secondo lavoro di ricerca menzionato prima, ha cercato di utilizzare diversi indicatori per la costruzione dell'albero decisionale: Thakur, Kamley e Jaloree, infatti, hanno utilizzato la variazione percentuale del valore delle azioni. Per la sperimentazione, sono stati presi in considerazione i dati storici dei titoli elencati nell'indice SENSEX[8]. Dopo la fase di pre-processing, è stato costruito il modello considerando la variazione percentuale del prezzo di chiusura, del prezzo di apertura, del minimo

[7]EMA(C) e EMA(V) rappresentano la media mobile esponenziale, rispettivamente, del prezzo di chiusura e del volume. Il MACD identifica invece la convergenza e la divergenza di medie mobili. K sfrutta una singola linea di media mobile. Le BB, cioè le bande di Bollinger, sono basate sulla volatilità di un titolo.

[8]Il SENSEX, anche chiamato BSE 30, è l'indice delle prime 30 compagnie della *Bombay Stock Exchange*.

e del massimo di giornata rispetto il giorno precedente. Utilizzando l'algoritmo ID3, è stata ottenuta un'accuratezza di circa il 90% [30].

3.4.3 Il confronto tra le performance

Dopo tutti gli studi fin qui descritti, inizia a diventare importante avere un reale confronto tra le varie tecniche di classificazione.

Recentemente, Paliyawan ha confrontato le performance di vari classificatori, analizzando le azioni presenti nel SET[9]. L'analisi ha considerato una fascia temporale di venti anni, utilizzando la variazione percentuale dei cinque giorni precedenti e predicendo l'andamento per date successive: il giorno seguente, sei giorni dopo e ventuno giorni dopo. Le tecniche applicate sono state le reti neurali, gli alberi decisionali, la classificazione Naïve Bayes e il k-nearest neighbor (k-NN). I risultati sperimentali hanno visto come miglior risultato, per la predizione a breve termine, quello ottenuto dal K-NN, seguito dalle reti neurali e dal Naïve Bayes; l'albero decisionale è risultato all'ultimo posto. Considerando invece le due previsioni a medio termine è stato il metodo Naïve Bayes a primeggiare, seguito dal Decision Tree e da K-NN. Riducendo il numero di classi e considerando solo le classi "up" e "down", sono state le reti neurali ad ottenere le migliori performance nel caso di previsione del giorno successivo; con predizioni a medio-termine, invece, sono stati gli alberi decisionali ad avere l'accuratezza più elevata. Effettuati questi primi test, i ricercatori, con l'obiettivo di ottenere la migliore tecnica, hanno analizzato un insieme di probabilità; tra queste ne troviamo quattro che l'autore chiama *"Right Buying"*, *"Wrong Buying"*, *"Right Sell"* e *"Opportunity Loss"*. Questo scenario però è risultato inaffidabile; la probabilità di effettuare un acquisto errato è risultata maggiore rispetto quella di fare un buon acquisto, rischiando quindi di incorrere in suggerimenti totalmente errati. In conclusione, questa ricerca termina eleggendo l'albero decisionale come migliore tecnica; infatti, oltre a garantire buone performance permette di ottenere un modello e dei risultati facilmente interpretabili, utili per trovare dei pattern in grado di aiutare i trader [23].

In questo capitolo si è visto come, in questi ultimi anni, diversi studi abbiano avuto il financial data mining come argomento chiave. Moltissime ricerche sono partite dalle stesse considerazioni alla base di questa tesi, dimostrando come questa tematica sia un interessante argomento di ricerca.

[9]L'indice SET, cioè *Stock Exchange of Thailand*, è l'indice di mercato thailandese.

Capitolo 4

Strategia d'investimento basata su tecniche di classificazione

Questo capitolo illustrerà ad alto livello il trading system sviluppato. Ogni fase, rappresentata da un blocco in figura 4.1, verrà descritta dal punto di vista delle informazioni in input, di quelle in output e in linea generale dal punto di vista delle funzionalità.

Il processo inizia dal *download dei dati* grazie alle API fornite da Yahoo Finance!. Questo step riceve in input la data di inizio e la data di fine della fascia temporale desiderata; con queste informazioni effettua una richiesta al servizio e scarica i relativi dati, producendo un unico dataset. A seconda dei parametri configurati, è possibile generare in output dei file contenenti tutte le informazioni ottenute o solamente una parte. Nel caso la piattaforma non fornisca in maniera completa i valori delle azioni, il tool gestice queste situazioni inserendo il valore null all'interno del dataset: questa anomalia viene poi gestita dallo step successivo.

In seguito, nella fase di *preparazione dei dati* e in quella di *generazione del dataset di train*, vengono utilizzate le tecniche che verranno descritte nei capitoli successivi per la produzione del training set. Ricevendo in ingresso i dataset scaricati nello step precedente, viene costruito un dataset di train per ogni giorno da predire. Questo file è formato da undici record dove l'ultimo rappresenta proprio il giorno sul quale si vuole effettuare la predizione. La discretizzazione dei valori viene qui eseguita prendendo in considerazione la variazione percentuale rispetto al giorno precedente; a seconda della tipologia scelta, il valore continuo viene trasformato in valore discreto inserendolo in una delle fasce configurate. Il processo inoltre, nel caso rilevi un titolo che presenta valori nulli, lo inserisce in una lista di azioni da non considerare. Successivamente questa lista verrà scandita e utilizzata per far si che Rapidminer non consideri queste azioni; in aggiunta, in Rapidminer stesso, è presente un filtro che elimina gli attributi con valori nulli.

Nel caso si scelga di utilizzare il filtro basato sui pattern grafici, prima della creazione del modello, è presente la fase di *pattern recognition*. Questo step esamina il training set generato dallo step precedente e calcola le figure dell'analisi tecnica presenti al suo interno; in caso ne rilevi una o più opposte alla strategia di investimento scelta, quel determinato titolo viene inserito nella lista delle azioni da non considerare. In output viene quindi prodotto un dataset di dimensioni ridotte, che sarà il dataset di train finale per quella predizione.

Una volta ottenuto il training set definitivo, quest'ultimo viene dato in input

al classificatore affinché generi il modello. Gli algoritmi scelti, come già discusso, sono stati gli alberi decisionali, l'SVM, le reti bayesiane e le reti neurali. Il modello generato è stato poi utilizzato per effettuare la predizione del singolo giorno coinvolto in quella determinata esecuzione.

Terminata la *generazione delle predizioni* per tutte le azioni presenti nel dataset, queste informazioni vengono aggregate generando in output un unico insieme di dati. Su questo insieme viene effettuata la *scelta dei titoli* basandosi su due parametri: il valore e la confidenza della predizione. Nel portafoglio di investimento verranno inserite solamente le previsioni che presentano un trend concorde con la strategia scelta e un valore di confidenza massimo.

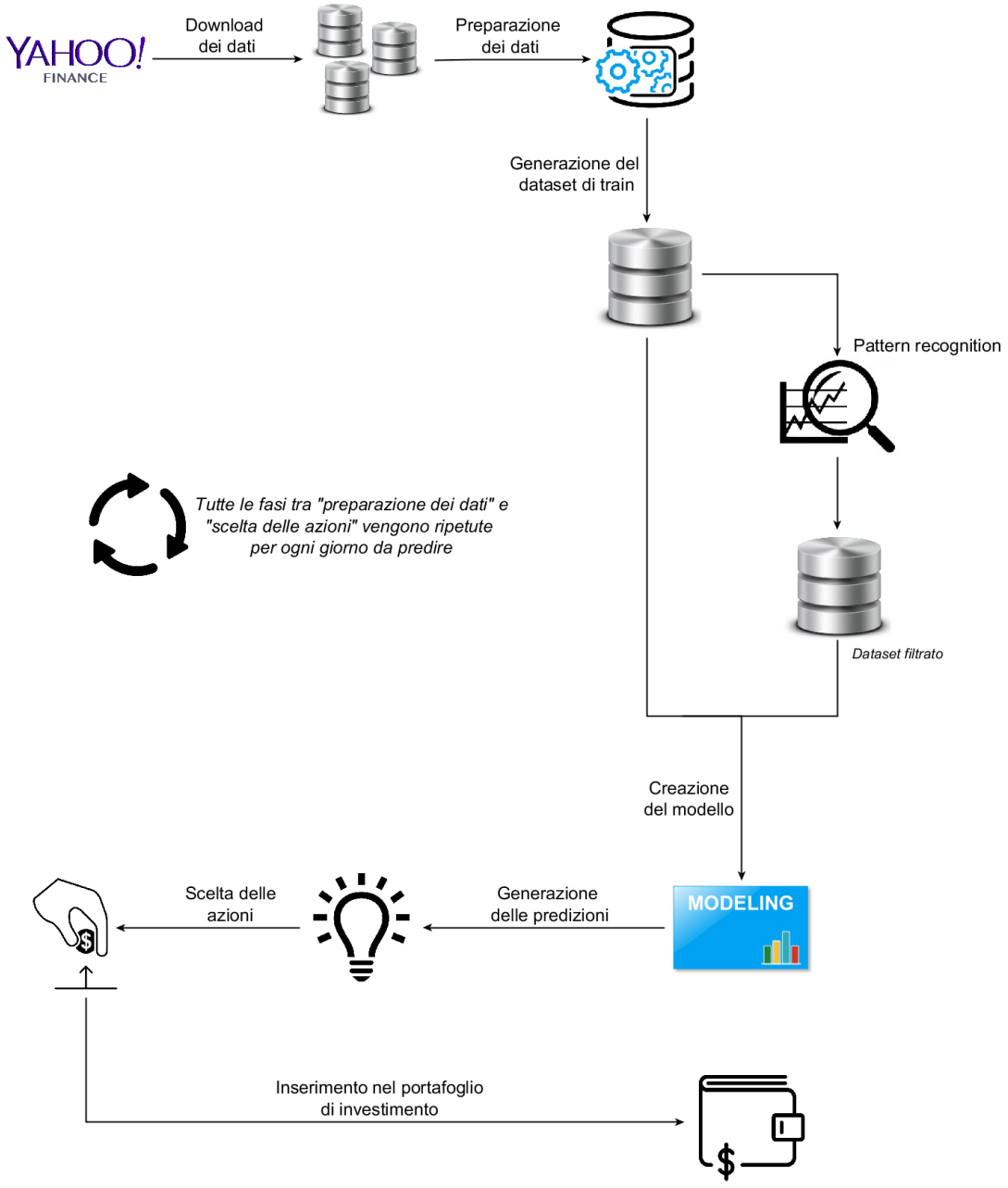

Figura 4.1: Diagramma a blocchi del framework sviluppato

Capitolo 5

Implementazione del sistema di trading basato su tecniche di classificazione

Per effettuare tutte le sperimentazioni sono stati utilizzati principalmente due applicazioni:

- *Rapidminer*: utilizzato per l'analisi dei dati e la creazione del modello
- Tool sviluppato in C# utile per la fase di pre-processing ed elaborazione dei risultati.

In questo capitolo si andranno ad analizzare tutte le varie fasi coinvolte nel lavoro di ricerca focalizzandosi sulle principali tematiche e sui problemi incontrati.

5.1 Il software di data mining: Rapidminer

Rapidminer è un software di data science, uscito nel 2006, che fornisce un ambiente idoneo per la ricerca e la sperimentazione nel campo del machine learning, del deep learning e del text mining. È molto utilizzato per applicazioni commerciali e nell'ambito della ricerca scientifica supportando tutti gli step del machine learning. La versione free è rilasciata sotto licenza AGPL ma presenta una limitazione in termini di data set (10000 righe al massimo) e in termini di utilizzo dei processori logici (al massimo uno). Quest'ultima limitazione è stata quella più vincolante per questo studio; i tempi di elaborazione e di creazione del modello sono risultati, in diversi casi, molto lunghi probabilmente a causa di tale vincolo.

Grazie alla sua struttura non è necessario, salvo particolari necessità, scrivere codice; infatti permette, tramite un'interfaccia grafica, la creazione e l'esecuzione di qualsiasi tipo di progetto (vedi figura 5.1). Oltre tutto, i file *rmp* generati hanno una struttura identica all'xml, rendendo eventuali modifiche facili da effettuare.

Le funzionalità, grazie all'utilizzo di plugins, possono essere facilmente espanse. Esiste infatti un marketplace dove è possibile trovare numerosi add-on; tra questi sono stati installati quelli appartenenti alla libreria Weka[1]. Per lo scopo di questa ricerca, Rapidminer non è stato utilizzato solamente per la creazione e l'applicazione

[1]Weka, acronimo di "Waikato Environment for Knowledge Analysis", è un insieme di algoritmi di machine learning utili per il data mining.

dei modelli, ma si è ma si è reso utile anche per la fase di pre e post processing dei dati.

Per tutti i test è stato utilizzato Rapidminer 5.3.015.

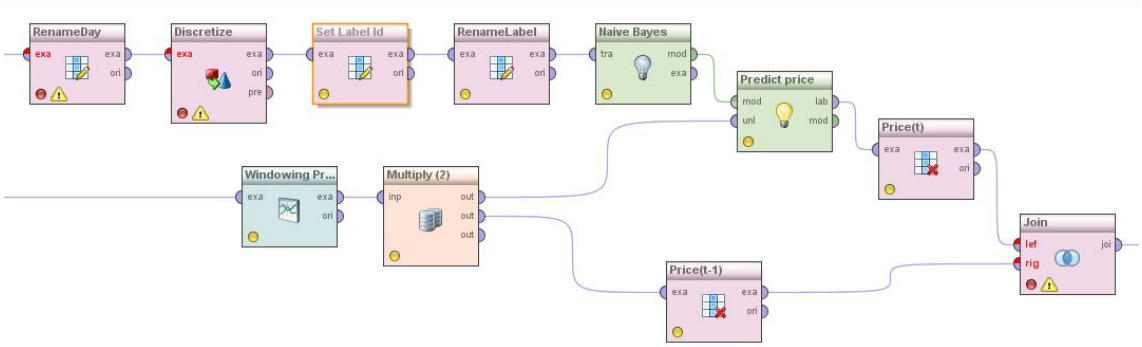

Figura 5.1: Esempio di processo Rapidminer

5.2 Il download dei dati: Yahoo! Finance

Per la creazione dei dataset sono stati utilizzati i dati forniti dal sito Yahoo! Finance[2]: grazie alle API fornite da questo servizio è possibile scaricare i dati storici per qualsiasi azione. Queste librerie sono già state utilizzate per realizzare lo studio sviluppato da Baralis et al. nel 2017 ma, a partire da Maggio dello stesso anno, Yahoo! ha effettuato alcuni cambiamenti all'interfaccia rendendo necessarie delle modifiche a quanto già sviluppato. Una delle modifiche più impattanti è stata l'inserimento nell'URL di richiesta di una stringa alfanumerica, denominata *crumb*. L'indirizzo, attualmente, è nella seguente forma:

https://query1.finance.yahoo.com/v7/finance/download/ISP.MI?
period1=1508167575&period2=1510849575&interval=1d&
events=history&crumb=L4viTB2tACO

Osservando l'esempio sono facilmente individuabili tutti i campi necessari al corretto scaricamento dei dati. L'URL è quindi composto da:

- **Simbolo dell'azione**: in questo caso è quello di Intesa Sanpaolo (ISP.MI).
- **period1** e **period2**: esprimono, in unix-time[3] la fascia temporale desiderata.
- **interval**: rappresenta la frequenza desiderata (giornaliera, settimanale, mensile)
- **crumb**: stringa alfanumerica di 11 caratteri introdotta recentemente. Svolge la funzione di token di sicurezza ed è strettamente legato ad un cookie.

[2]https://it.finance.yahoo.com/

[3]Nei sistemi operativi Unix e Unix-like, il tempo è rappresentato come offset in secondi rispetto alla mezzanotte del 1 gennaio 1970.

La problematica principale è stata quindi capire come generare il crumb in maniera corretta; per ogni download è infatti necessario, per non incorrere in un errore, possedere la corretta coppia cookie-crumb. Il problema è stato superato integrando, all'interno del codice sviluppato, una parte scritta da Lee [19]. Grazie alla funzione *getCrumb()* l'applicazione calcola la stringa necessaria al download in modo da costruire correttamente l'URL (figura 5.2). I dati così scaricati saranno nel formato *comma-separated values* e conterranno, per ogni giorno compreso nel range di date scelte, sei valori: prezzo di chiusura e di apertura, minimo e massimo di giornata, la chiusura aggiustata[4] e il volume. Questi record saranno poi utilizzati o meno a seconda del test che si andrà ad eseguire.

```
private static string getCrumb(string html)
{

    string crumb = null;
    try
    {
        if (regex_crumb == null)
            regex_crumb = new Regex("CrumbStore\":{\"crumb\":\"(?<crumb>.+?)\"}",
                RegexOptions.CultureInvariant | RegexOptions.Compiled);

        MatchCollection matches = regex_crumb.Matches(html);

        if (matches.Count > 0)
        {
            crumb = matches[0].Groups["crumb"].Value;
            if (crumb.Length != 11)
                crumb = crumb.Replace("\\u002F", "/");
        }
        else
            Debug.Print("Regex no match");

        matches = null;
    }
    catch (Exception ex)
    {
        Debug.Print(ex.Message);
    }
    GC.Collect();
    return crumb;
}
```

Figura 5.2: Funzione getGrumb()

Nonostante sia possibile utilizzare il tool sviluppato per ottenere, ad ogni esecuzione, nuovi dati, la scelta per cui si è optato è stata quella di lavorare su informazioni presenti in memoria. I data set sono stati creati all'inizio dello studio e utilizzati poi nelle varie sperimentazioni. Questo ci ha permesso di migliorare i tempi di simulazione e di evitare eventuali problemi di download che avrebbero potuto rendere i

[4]Con chiusura aggiustata si intende il prezzo ottenuto considerando eventuali distribuzioni di capitali avvenute prima dell'apertura del giorno successivo.

risultati non comparabili tra loro. Si ha così la garanzia che il dataset sarà comune ad ogni esecuzione potendo così confrontare i dati ottenuti.

5.3 Riconoscimento di pattern di analisi tecnica

Volendo integrare all'interno dello studio le funzioni proprie dell'analisi tecnica si è voluto cercare delle soluzioni sviluppate ad-hoc. La scelta è ricaduta su una libreria open-source in grado di calcolare numerosi indicatori e pattern tipici del mercato finanziario. TA-Lib mette a disposizione 200 indicatori tipici dell'analisi tecnica e il riconoscimento di pattern grafici [29]. Questi ultimi sono comunemente utilizzati dai trader per generare manualmente segnali di trading e utilizzano i grafici a candele giapponesi (*candlestick*) per identificare delle figure ricorrenti e predire l'andamento dei prezzi. Ad esempio, in figura 5.3, è possibile vedere il pattern *three line strike*, che segnala la continuazione del trend in corso.

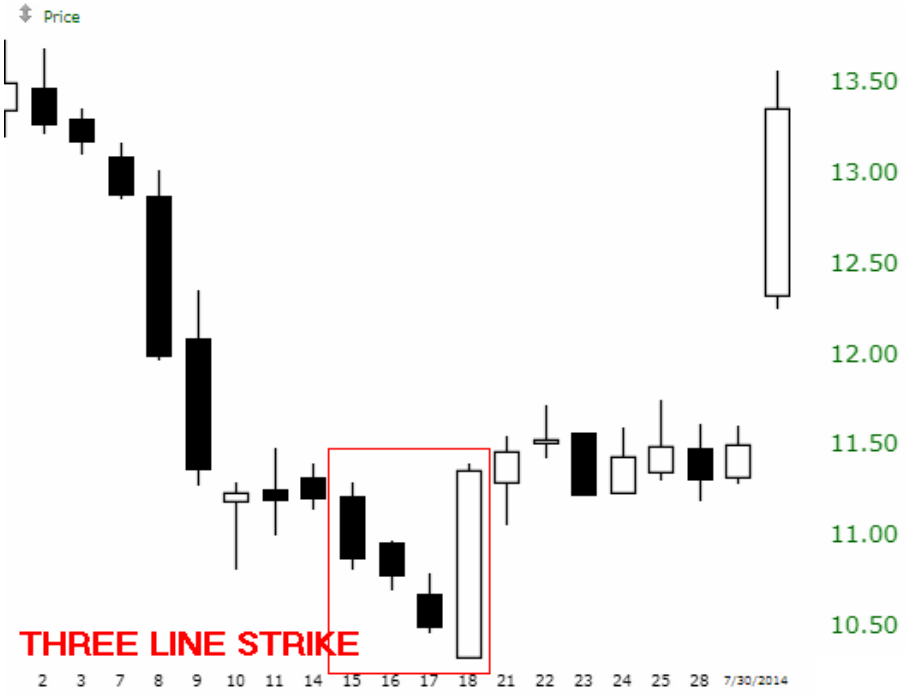

Figura 5.3: Pattern ribassista *"Bearish Tree Line Strike"*[5]

Grazie alla funzionalità di *pattern recognition* fornita da questa libreria è stato possibile, dopo una prima fase di sperimentazione, introdurre un filtro ai dati. L'obiettivo è stato quello di identificare i pattern contrari alla direzione di investimento desiderata ed eliminare tutte le azioni che li contenevano. Tutti i metodi forniti dalla libreria hanno in comune il fatto di ricevere, come parametri, cinque array contenenti ciascuno una tipologia diversa di prezzo. La tipica chiamata ad una di queste funzioni è visibile in figura 5.4.

[5]Immagine tratta da: *The 5 Most Powerful Candlestick Patterns.* URL: `https://www.investopedia.com/articles/active-trading/092315/5-most-powerful-candlestick-patterns.asp`

```
ta_lib.Cdl2Crows(startIdx, endIdx, open, high, low, close, out outBegIdx, out outNBElement, taOut);
```

Figura 5.4: Metodo per il calcolo del pattern *"Two Crows"*

In output viene fornito l'array *taOut* che potrà contenere, per ogni giorno di input, il valore 100, 0 o -100. Nel caso il risultato sia positivo è stato riconosciuto un pattern rialzista, nel caso opposto è presente un pattern ribassista. Utilizzando questo vettore è stato possibile filtrare i dati in input.

5.4 Preparazione dei dati

La prima vera e propria fase di lavorazione dei dati è il *pre-processing*. Prima di iniziare ad illustrare questa fase, conviene fare un breve spiegazione sui file di dati che vengono gestiti ed elaborati. Tutto il processo di predizione è basato su tre file:

- *prezzi.csv* contenente il dataset di input del classificatore.
- *list.csv* con all'interno l'elenco delle azioni da elaborare e sulle quali costruire il modello.
- *portfolio.csv* che avrà al suo interno la predizione generata.

Questi file, nel caso si decida di elaborare una fascia temporale ampia e quindi predire più di un singolo giorno, verranno aggiornati ad ogni iterazione.

Il tool sviluppato, dopo aver letto il file di configurazione, inizia ad elaborare il file di input e, per ogni giorno presente all'interno della fascia temporale richiesta, esegue i seguenti step:

1. Ricerca all'interno del dataset del giorno da predire. Partendo da quest'ultimo analizza l'input alla ricerca di dieci giorni precedenti in cui la borsa abbia generato operazioni.
2. Se trova dieci giorni validi, crea un file che verrà utilizzato per la costruzione del modello simile a quello visibile in figura 5.5. Nel caso non sia disponibile una storicità del genere, il processo considera quel determinato giorno come non predicibile e passa ad elaborare il giorno successivo.
3. Poichè i dati forniti da Yahoo! Finance possono presentare dei valori nulli, l'applicazione passa quindi ad analizzare il dataset appena generato eliminando dal file list.csv tutte quelle azioni da non considerare per questo motivo. Non è necessario eliminarle anche dal file creato nello step precedente, poiché Rapidminer elabora soltanto le azioni presenti nella lista in input.

La fase di pre-processing appena descritta viene eseguita per qualsiasi configurazione e scenario di utilizzo. L'obiettivo infatti è quello di effettuare una predizione sfruttando una storicità di dieci giorni. L'unica differenza sarà il numero di attributi che comporrà il dataset.

Nel caso venga utilizzato il filtro basato sui pattern grafici, all'interno del tool è presente un successivo step di modifica dei dati. Il dataset generato allo stadio precedente viene quindi fatto passare attraverso un'ulteriore funzione che esegue

Date	A2A.MI	AGL.MI	ATL.MI	AZM.MI
2013-03-28	0.463	3.617	12.320	12.620
2013-04-02	0.469	3.595	12.430	12.380
2013-04-03	0.472	3.564	12.090	12.230
2013-04-04	0.486	3.597	12.200	12.110
2013-04-05	0.492	3.626	12.310	12.170
2013-04-08	0.499	3.613	12.290	11.990
2013-04-09	0.514	3.674	12.160	12.680
2013-04-10	0.529	3.840	12.500	13.430
2013-04-11	0.535	3.799	12.610	13.410
2013-04-12	0.548	3.815	12.560	13.340
2013-04-15	0.551	3.789	12.350	13.240

Figura 5.5: Esempio di dataset generato dal primo step di preprocessing

il seguente processo: per ogni azione presente nel file ricerca il primo pattern tra quelli desiderati; se lo trova, dato che la ricerca ha come scopo quello di trovare quelli contro-tendenza, l'azione viene inserita in una lista di esclusioni denominata *stocksToRemove*. In caso negativo, si ricerca il secondo pattern, poi il terzo e così via. Il processo continua fino ad aver ricercato tutti e 61 i pattern scelti per questo studio. Al termine dell'algoritmo è presente un ulteriore step che elimina dal file list.csv tutte quelle azioni che sono state memorizzate durante lo step precedente. Per eliminare un'azione è sufficiente quindi che contenga anche soltanto un pattern opposto rispetto il trend ricercato.

Dopo aver creato correttamente i file di input, il tool si occupa di effettuare un'esecuzione via riga di comando di Rapidminer. Per effettuare una corretta esecuzione, il file rmp del processo da eseguire viene editato a seconda dei parametri configurati.

La successiva fase di pre-processing è realizzata da Rapidminer. Ricevuto il dataset in input, il processo sviluppato si occupa di effettuare un'operazione di *windowing* sui dati. Per ogni azione considerata, utilizzando una finestra "scorrevole" composta da due elementi, crea un'ulteriore attributo corrispondente al prezzo del giorno precedente. Considerando i dati presenti in figura 5.5, il dataset pronto da utilizzare per la creazione del modello sarà realizzato come in figura 5.6. Come è facilmente visibile, l'attributo suffissato con -1 rappresenta il valore del giorno precedente a quello rappresentato dal record.

Realizzato il dataset appena descritto, è stato necessario introdurre un passo di discretizzazione dei dati. Infatti, per definizione, i classificatori utilizzano valori discreti per la creazione dei modelli. A questo scopo, nel processo rapidminer è stato inserito un operatore di classificazione. Calcolata la variazione percentuale rispetto al giorno precedente si è deciso di effettuare le seguenti sperimentazioni:

1. Discretizzazione $[-\infty, 0][0, \infty]$. In questa modalità, l'azione viene considerata in rialzo per tutte quelle percentuali sopra lo 0%, e in ribasso nel caso opposto.

Date	A2A.MI-1	A2A.MI-0	AGL.MI-1	AGL.MI-0	ATL.MI-1	ATL.MI-0
2013-04-02	0.463	0.469	3.617	3.595	12.320	12.430
2013-04-03	0.469	0.472	3.595	3.564	12.430	12.090
2013-04-04	0.472	0.486	3.564	3.597	12.090	12.200
2013-04-05	0.486	0.492	3.597	3.626	12.200	12.310
2013-04-08	0.492	0.499	3.626	3.613	12.310	12.290
2013-04-09	0.499	0.514	3.613	3.674	12.290	12.160
2013-04-10	0.514	0.529	3.674	3.840	12.160	12.500
2013-04-11	0.529	0.535	3.840	3.799	12.500	12.610
2013-04-12	0.535	0.548	3.799	3.815	12.610	12.560

Figura 5.6: Dataset realizzato attraverso la fase di windowing

2. Discretizzazione $[-\infty, -0.5][0.5, \infty]$. Azione in rialzo se sopra lo 0.5%, in ribasso se sotto il -0.5%. In questo caso, si avrà la fascia indicante un andamento costante più ampia.

3. Discretizzazione $[-\infty, -1][1, \infty]$. Opzione nella quale il range che esprime un valore costante è il più ampio tra tutti; si avranno azioni in rialzo se il loro valore avrà un incremento pari o superiore all'1%, azioni in ribasso se perdono più dell'1% del loro valore. Un esempio di questa discretizzazione è visibile in figura 5.7.

Nella fase iniziale della sperimentazione sono state testate tutte e tre le tipologie di discretizzazione, per poi scegliere di utilizzare solo l'ultima descritta. Terminata la fase di pre-processing, è possibile quindi generare il modello.

Date	label	A2A.MI-1	A2A.MI-0
2013-04-02	=	0.463	0.469
2013-04-03	+	0.469	0.472
2013-04-04	+	0.472	0.486
2013-04-05	+	0.486	0.492
2013-04-08	+	0.492	0.499
2013-04-09	+	0.499	0.514
2013-04-10	+	0.514	0.529
2013-04-11	+	0.529	0.535
2013-04-12	=	0.535	0.548

Figura 5.7: Esempio di discretizzazione di tipo 3

5.5 Addestramento di modelli di classificazione

Il classificatore riceve un dataset realizzato mediante i processi descritti nella sezione precedente ed in seguito effettua la costruzione del modello. Questo processo viene eseguito per ogni azione considerata e, per ognuna di essa, viene effettuata una predizione. Saranno poi gli step successivi ad effettuare un'aggregazione dei dati.

Questa fase è realizzata attraverso Rapidminer che permette di cambiare il tipo di classificatore utilizzato semplicemente sostituendo un blocco. Ogni classificatore è stato utilizzato con i parametri di default che fornisce il software. Nello specifico, prendendo in considerazione la nomenclatura presente in Rapidminer, sono stati utilizzati: *"Decision Tree"*, *"SVM (Support Vector Machine (LibSVM)"*, *"Naive Bayes"* e *"Neural Net"*. Le relative configurazioni sono descritte nelle tabelle 5.1 e 5.2.

Il modello generato, come visibile in figura 5.1, viene utilizzato come input per l'operatore Rapidminer *"Apply Model"*. Quest'ultimo, ricevendo in ingresso anche i dati da etichettare, effettua la predizione.

Decision Tree

Parametro	Valore
criterion	gain_ratio
minimal size for split	4
minimal leaf size	2
minimal gain	0.1
maximal depth	20
confidence	0.25

SVM

Parametro	Valore
svm type	C-SVC
kernel type	Poly
degree	3
gamma	0.0
coef()	0.0
C	0.0
epsilon	0.001
calculate confidence	disabilitato

Tabella 5.1: Configurazioni dei classificatori: Decision Tree e SVM

Naïve Bayes

Parametro	Valore
laplace correction	abilitato

Neural Net

Parametro	Valore
training cycle	500
learning rate	0.3
momentum	0.2
error epsilon	$1.0 \cdot 10^{-5}$

Tabella 5.2: Configurazioni dei classificatori: Naïve Bayes e Neural Net

5.6 La predizione e la scelta dei risultati

Come descritto precedentemente, la predizione viene effettuata mediante l'applicazione del modello a dei dati non etichettati.

Per effettuare la predizione e generare il portafoglio di investimento, tutte le predizioni generate vengono aggregate creando un output simile alla figura 5.8.

Date	confidence(=)	confidence(-)	confidence(+)	Prediction	DayBefore	Stock
2013-04-15	0	0	1	+	7.295	REC.MI
2013-04-15	1	0	0	=	0.551	A2A.MI
2013-04-15	1	0	0	=	3.789	AGL.MI
2013-04-15	0	1	0	-	12.350	ATL.MI
2013-04-15	0	1	0	-	13.240	AZM.MI
2013-04-15	0	1	0	-	4.965	BPE.MI
2013-04-15	0	1	0	-	11.070	BZU.MI

Figura 5.8: Esempio di dati generati dalla classificazione

Ogni predizione è accompagnata da tre valori di confidenza: uno per la classe positiva, uno per quella negativa e l'ultimo per quella che rappresenta un andamento costante. La scelta delle azioni da consigliare è realizzata in base a questi attributi e, a seconda dello scenario per cui si sta effettuando l'esperimento (short o long), si valuterà la confidenza positiva o negativa. In entrambi i casi viene calcolato il valore massimo dell'attributo scelto e, tra tutte le azioni che appartengono a quella determinata classe, vengono scelte quelle che hanno un valore di *confidence* pari al più alto possibile.

Prendendo d'esempio i dati presenti in figura 5.8 e ipotizzando uno scenario long, la ricerca dovrà essere indirizzata verso le azioni appartenenti alla classe positiva. Tra queste troviamo *Recordati S.p.A.* (REC.MI) che presenta $confidence(+) = 1$; questo valore, almeno in questo caso, è pari al massimo tra tutte le confidenze positive. Nel portafoglio di investimento verrà quindi inserita questa azione.

Capitolo 6

Gli esperimenti

Come descritto nei capitoli precedenti, l'obiettivo di questa tesi è stato quello di effettuare una sperimentazione il più completa possibile. In quest'ottica, i test sono risultati molteplici e tutti sono stati caratterizzati dalla variazione di quattro caratteristiche: lo scenario di investimento, il mercato analizzato, il classificatore utilizzato e gli attributi presenti in input. In questo capitolo, si andrà quindi a descrivere più nel dettaglio ognuna di queste caratteristiche.

6.1 Progettazione della campagna sperimentale

Le principali suddivisioni identificate per questo studio sono basate sul mercato e sulla strategia di investimento.

Per quanto riguarda gli scenari di investimento le opzioni possibili sono state due: posizione long e posizione short. Queste due strategie sono tra le più comuni ed entrambe sono applicabile al trading intraday. La strategia long è sicuramente la più conosciuta e consiste nell'acquistare titoli azionari nella speranza che il loro prezzo aumenti; se questo accade, il trader venderà le azioni acquistate traendone un guadagno. In pratica si scommette sul rialzo. Contrariamente al pensiero comune, si può ricavare del profitto anche puntando al ribasso di un titolo. Scegliendo la posizione short, infatti, attraverso il meccanismo delle vendite allo scoperto, si spera che un titolo diminuisca il proprio valore. In altre parole, l'investitore venderà delle azioni che non possiede prendendole in prestito dalla propria banca sulla base della loro quotazione di mercato. Il trader con questa operazione si impegna a restituirle entro un determinato periodo di tempo. Se durante questo periodo il valore dei titoli scelti cala, sarà possibile comprare il numero di azioni da restituire, pagandole un prezzo inferiore e ottenendo così un profitto. Il guadagno sarà dato dalla differenza tra il prezzo di vendita e il prezzo di riacquisto. Le simulazioni effettuate hanno quindi considerato entrambe le strategie di trading; in caso di posizione long si è considerato di aver ottenuto un guadagno quando l'azione suggerita ha aumentato il proprio valore rispetto al giorno precedente; in caso di investimento short invece è stato ipotizzato di aver realizzato un profitto se la predizione coinvolge un titolo che ha avuto una diminuzione di valore [24].

Per quel che concerne invece il mercato sono stati stati scelti tre scenari possibili: mercato crescente, decrescente e misto. Questa decisione è dovuta al fatto che anche l'andamento del mercato può portare a conclusioni differenti. In questo ottica sono stati scelti tre diversi periodi:

- **2011**. Periodo di recessione. Il mercato ha avuto un andamento ribassista. La posizione ideale in questo periodo sarebbe stata quella short (fig. 6.1 e 6.2).
- **2013**. Come mostrato in figura 2, il 2013 è stato un anno nel quale i titoli hanno avuto un trend al rialzo. In questo periodo la strategia di investimento più idonea è quella long (fig. 6.3 e 6.4).
- **2015**. Il mercato ha avuto continui alti e bassi, non potendo quindi individuare il profilo di investimento migliore (fig. 6.5 e 6.6).

Oltre a queste due grandi suddivisioni, un'altra variabile è stata data dall'indice scelto e quindi dai relativi titoli; per effettuare dei test il più possibile omogenei sono stati scelti il FTSE MIB e l'S&P500.

Il **FTSE MIB**, il cui acronimo sta per *Financial Times Stock Exchange Milano Indice di Borsa*, è il principale indice di benchmark dei mercati azionari italiani e rappresenta circa l'80% della capitalizzazione. Raccoglie le 40 società italiane, anche con sede legale all'estero, che hanno la maggior capitalizzazione e liquidità [10]. L'**S&P500** (Standard & Poor's 500) invece segue l'andamento delle principali 500 società statunitensi; fanno parte di questo paniere le azioni delle aziende comprate e vendute al New York Stock Exchange, all'American Stock Exchange e al Nasdaq. I due mercati considerati, come è facilmente intuibile, sono governati da logiche di mercato molto diverse tra loro. Scegliere su quale investire è sicuramente una scelta difficile; sia dal punto di vista geografico sia da quello numerico presentano notevoli differenze ed optare per uno o l'altro potrebbe comportare differenze sui risultati ottenuti.

Con le varie casistiche appena descritte vengono quindi a crearsi numerose possibilità di test. Avere queste diverse opzioni fa sì che la valutazione sperimentale diventi abbastanza completa. Considerando solo quanto appena descritto, Si avranno quindi sei diverse casistiche che andranno a coprire vari scenari sia dal punto di vista geografico, sia da quello numerico e persino dal punto di vista dell'andamento dei titoli.

6.2 I classificatori utilizzati

Come descritto nei capitoli precedenti, i classificatori utilizzati sono stati gli alberi decisionali, l'SVM, le reti bayesiane e le reti neurali. Ognuno di questi classificatori è stato utilizzato con le configurazioni di default (visibili nella sezione 5.5). Per ognuna di queste tecniche di classificazione sono stati creati due processi Rapidminer: uno per lo scenario di tipo short e uno per quello long. Questa soluzione è stata adottata in modo tale da permettere una più veloce gestione delle simulazioni: è bastato cambiare il processo da eseguire affinché cambiasse la strategia di investimento e/o la tecnica di modellazione. Arrivati a questo punto, combinando i classificatori con le opzioni descritte nelle sezioni precedenti si arriva ad ottenere 24 configurazioni di test realizzabili. Il numero di possibilità inizia quindi a diventare interessante ma, come vedremo nel successivo paragrafo, è destinato ad aumentare introducendo altre varianti.

6.3 I dataset analizzati

Una delle tematiche affrontate è stata quella di capire come e se la variazione degli attributi presenti nei dataset potesse influenzare il risultato.

Si è quindi scelto di effettuare quattro variazioni su questo tema: la prima, la più semplice, considera come attributo solamente il prezzo di chiusura dell'azione; nella seconda, viene aggiunto anche il volume di operazioni eseguite; nella terza e nella quarta soluzione sono stati utilizzati tutti i valori necessari alla costruzione dei grafici a candele giapponesi (prezzo di chiusura e apertura combinati con il minimo e il massimo di giornata). Quest'ultima tipologia di input è stata utilizzata per testare il filtraggio descritto nei precedenti capitoli.

Per realizzare queste quattro configurazioni, prima di iniziare con le simulazioni, sono state scaricate da Yahoo! Finance tutte le informazioni necessarie. Sono stati quindi creati tre file csv per ogni combinazione anno-mercato, per un totale di dodici dataset. A seconda del test da eseguire, il file di configurazione veniva quindi modificato impostando il file di input e i parametri corretti.

6.4 Simulazione d'investimento intraday

Come già ampiamente descritto, le tecniche di investimento possibili sono due: comprare al rialzo o comprare al ribasso. In entrambi i casi l'obiettivo è quello di guadagnare, ma il titolo scelto dovrà avere un andamento diverso a seconda del tipo di strategia scelto. Per questo motivo, il calcolo del guadagno varierà.
In caso di posizione long il guadagno sarà pari a:

$$\text{Profitto}_{long} = \frac{V_t - V_{t-1}}{V_{t-1}} \cdot 100$$

In caso di posizione short invece varrà:

$$\text{Profitto}_{short} = \frac{V_{t-1} - V_t}{V_t} \cdot 100$$

Per effettuare il calcolo, il tool sviluppato per la tesi elabora le predizioni effettuate analizzando i dati scaricati da Yahoo! Finance. Utilizzando i prezzi presenti all'interno di questi data set e configurando la strategia di investimento desiderata, l'applicazione produrrà in output un file contenente i profitti ottenuti. Nello specifico, ipotizzando che l'azione ISP.MI dal giorno precedente abbia aumentato il proprio prezzo dell'1.5%, in caso di posizione long è stato considerato un guadagno pari a questo valore, in caso di posizione short invece è stato registrata una perdita.

Partendo da quanto calcolato, sono stati quindi prodotti altri tre guadagni teorici, in modo tale di avvicinarsi il più possibile a degli scenari reali: viene ipotizzato l'inserimento di uno stop loss pari all'1% e viene prevista la possibilità di un costo di transazione pari allo 0.15%. Si avranno quindi quattro possibili valori di profitto, ma bisogna considerare che quello più realistico sarà sicuramente quello ottenuto utilizzando lo stop loss e applicando le commissioni.

6.5 L'hardware utilizzato

Per eseguire questo studio sono state utilizzate due configurazioni:

- Il primo computer è dotato di CPU Intel i7 4790 a 3.60 GHz e 16 GB di memoria ram.; il sistema operativo è Windows 10 Pro a 64 bit. Questa è la macchina dove è stato sviluppato tutto il codice e sono stati eseguiti la maggior parte degli esperimenti.

- La seconda macchina è configurata con una CPU Intel i7 870 a 2.96 Ghz, 4 GB di ram e Windows 7 Professional 64 bit. Questo computer è stato utilizzato per parallelizzare l'esecuzione dei test.

La principale limitazione riscontrata è stata relativa alla memoria ram e alla versione di Rapidminer.

Ovviamente un quantitativo maggiore di memoria rende le elaborazioni più veloci e la differenza in tal senso è stata evidente; monitorando l'utilizzo infatti è stato possibile notare come il processo di creazione del modello cercasse di allocare quasi tutta la memoria disponibile.

Rapidminer, inoltre, con la versione free del suo software inserisce una limitazione al numero massimo di processori logici utilizzabili; purtroppo non è stato possibile verificarlo ma sicuramente una versione dotata di licenza avrebbe reso più performante le simulazioni.

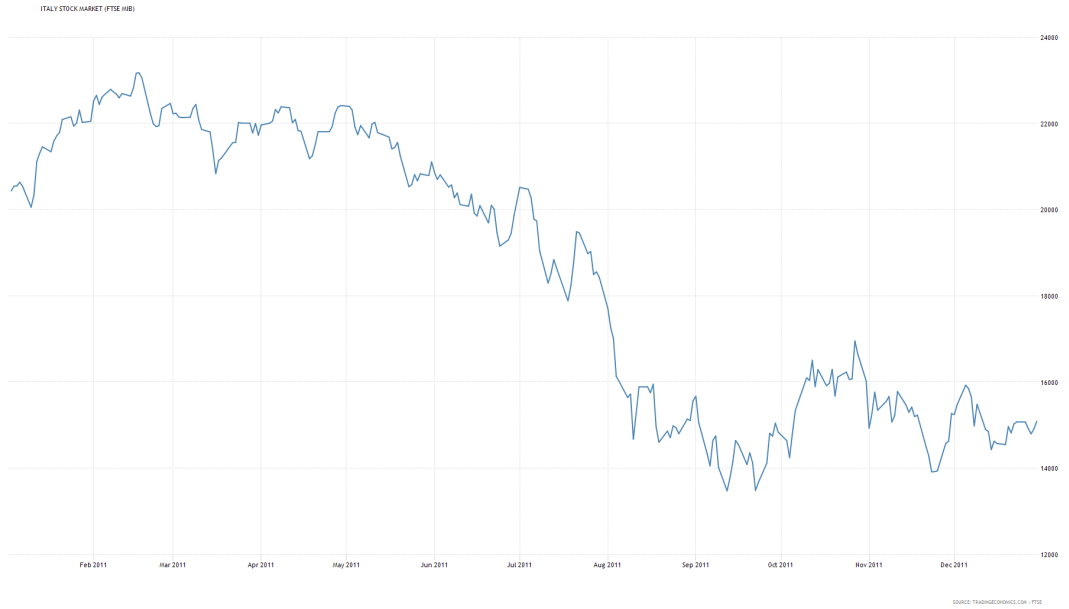

Figura 6.1: Andamento dell'indice FTSEMIB nel 2011[1]

Figura 6.2: Andamento dell'indice S&P500 nel 2011[2]

[1]Immagine tratta da: *Italy Stock Market (FTSE MIB)*. URL: `https://it.tradingeconomics.com/italy/stock-market`

[2]Immagine tratta da: *S&P 500 Index - 90 Year Historical Chart*. URL: `http://www.macrotrends.net/2324/sp-500-historical-chart-data`

Figura 6.3: Andamento dell'indice FTSEMIB nel 2013[3]

Figura 6.4: Andamento dell'indice S&P500 nel 2013[4]

[3]Immagine tratta da: *Italy Stock Market (FTSE MIB)*. URL: `https://it.tradingeconomics.com/italy/stock-market`

[4]Immagine tratta da: *S&P 500 Index - 90 Year Historical Chart*. URL: `http://www.macrotrends.net/2324/sp-500-historical-chart-data`

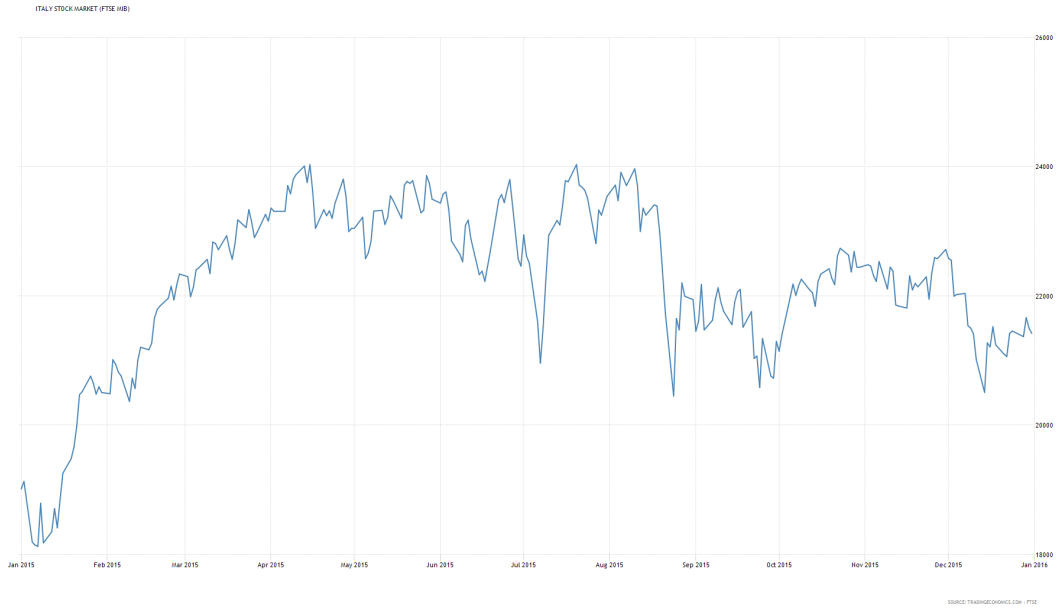

Figura 6.5: Andamento dell'indice FTSEMIB nel 2015[5]

Figura 6.6: Andamento dell'indice S&P500 nel 2015[6]

[5]Immagine tratta da: *Italy Stock Market (FTSE MIB)*. URL: `https://it.tradingeconomics.com/italy/stock-market`

[6]Immagine tratta da: *S&P 500 Index - 90 Year Historical Chart*. URL: `http://www.macrotrends.net/2324/sp-500-historical-chart-data`

Capitolo 7

I risultati

Questo capitolo descriverà quanto ottenuto attraverso le varie sperimentazioni. Per ogni test sono stati calcolati, come già accennato, quattro profitti teorici variando la presenza o meno di stop loss e costi di commissione. Per semplicità, nelle sezioni che seguono verrà preso in considerazione solamente il guadagno realizzato con l'applicazione di uno stop loss pari all'1% e la presenza di un costo di commissione pari allo 0.15%. I valori sono espressi in termini di guadagno medio giornaliero sull'intero anno. Come vedremo, i risultati sono stati abbastanza costanti confermando le prove eseguite.

7.1 FTSE MIB: anni 2011 e 2013

Le simulazioni sul mercato italiano sono state le prime ad essere effettuate. La decisione di partire da questo indice è dovuta principalmente alle sue dimensioni ridotte; questa caratteristica ha permesso infatti di testare tecniche di classificazione molto velocemente, potendo quindi eliminare, sin dall'inizio, quegli algoritmi che non portavano a buoni risultati. Sotto questo aspetto, è stato quindi possibile effettuare due prove che però non sono state replicate né in altre fasce temporali né su altri mercati.

Il primo classificatore testato è stato l'albero decisionale; gli anni presi in considerazione sono stati il 2011 e il 2013, ipotizzando entrambi gli scenari di investimento descritti in questa tesi. Utilizzando questa tecnica si è deciso di iniziare applicando tutte e tre le metodologie di discretizzazione descritte nel Capitolo 5. I risultati ottenuti, mostrati nella tabella 7.1, evidenziano come utilizzare una discretizzazione più restrittiva, e quindi considerare in rialzo solo le azioni che hanno ottenuto un incremento di valore sopra all'1%, dia guadagni migliori. Oltre tutto si nota chiaramente come all'aumento della soglia per cui considerare l'azione in rialzo corrisponda anche un aumento del profitto medio. Per queste motivazioni è stato scelto di scartare le prime due forme di discretizzazione e sfruttare quella tra -1 e 1 per i test seguenti.

Tipo di discretizzazione	Anno 2011		Anno 2013	
	Short	Long	Short	Long
Discretizzazione 1 - $[-\infty, 0][0, \infty]$	0.60	0.56	0.47	0.69
Discretizzazione 2 - $[-0.5, 0][0, 0.5]$	0.79	0.85	0.62	0.81
Discretizzazione 3 - $[-1, 0][0, 1]$	1.04	1.09	0.81	1.07

Tabella 7.1: Risultati percentuali medi delle tecniche di discretizzazione

Dopo questo primo test, si è voluto testare una tecnica di regressione concatenata ad una di classificazione. Quest'idea è nata prendendo spunto dallo studio eseguito da Baralis et al. nel quale venivano sfruttate proprio queste tecniche per la generazione dei segnali di trading [3]. I risultati ottenuti con l'albero decisionale e discretizzazione di tipo 3 sono stati quindi utilizzati come input per una SVM, utilizzata in questo caso per la regressione. Sono stati simulati due scenari:

1. il dataset di input è formato da tutte le azioni originarie ma l'SVM crea il modello solo per quelle predette dal classificatore;

2. l'SVM utilizza un input formato dai soli titoli generati dall'albero decisionale, sia come file list.csv sia come veri e propri prezzi da utilizzare nella costruzione del modello. E' molto più selettivo rispetto al precedente.

Anche in questo caso, come visibile nella tabella 7.2, i profitti ottenuti, se confrontati con quanto ottenuto precedentemente, non sono aumentati. Sfruttare un algoritmo di regressione in cascata all'albero decisionale non ha portato a miglioramenti e si è deciso quindi di scartare anche questa tecnica.

	Anno 2011		Anno 2013	
	Short	Long	Short	Long
SVM - Scenario 1	0.60	0.56	0.47	0.69
SVM - Scenario 2	0.60	0.56	0.47	0.69

Tabella 7.2: Risultati percentuali medi ottenuti con l'algoritmo di regressione SVM in cascata all'albero decisionale

Esauriti questi primi due test si è configurato lo scenario idoneo per il test di altri classificatori. Sfruttando la discretizzazione di tipo 3 si è passati quindi ad eseguire le simulazioni con l'SVM, questa volta in modalità classificazione, Naïve Bayes e con le reti neurali. I risultati ottenuti sono visibili nella tabella 7.3.

	Anno 2011		Anno 2013	
	Short	Long	Short	Long
Decision Tree	1.04	1.09	0.81	1.07
SVM	1.29	1.15	0.87	1.07
Naïve Bayes	1.72	1.56	1.04	1.47
Neural Net	1.63	1.48	1.40	1.73

Tabella 7.3: FTSE MIB - Risultati percentuali medi con l'input formato dai soli prezzi di chiusura

Con queste prime simulazioni, è possibile iniziare ad intravedere quali sono gli algoritmi che danno i risultati migliori: le reti bayesiane e quelle neurali si contendono il primato senza però esserci un vincitore assoluto.

I test appena descritti sono stati eseguiti con un dataset composto da soltanto il prezzo di chiusura giornaliero di ogni azione; considerato ciò si è voluto aumentare il numero di attributi e verificare la loro influenza sui risultati. L'obiettivo è stato quello di comprendere se i valori utilizzati nell'analisi tecnica tradizionale portassero realmente ad un beneficio, o se i metodi utilizzati tutti i giorni dai trader non fossero così ottimali.

Il primo passo è stato inserire il volume delle operazioni all'interno dell'input fornito al classificatore. I risultati, visibili nella tabella 7.4 hanno evidenziato, in linea generale, un peggioramento delle prestazioni, con il degrado più elevato ottenuto dalla SVM; solamente con le reti neurali, in tre casi su quattro, l'inserimento di questo attributo ha portato a un incremento del guadagno.

	Anno 2011		Anno 2013	
	Short	Long	Short	Long
Decision Tree	1.09	1.09	0.73	0.96
SVM	0.62	0.46	0.27	0.41
Naïve Bayes	1.57	1.48	1.05	1.17
Neural Net	1.58	1.57	1.43	1.97

Tabella 7.4: FTSE MIB - Risultati percentuali medi ottenuti inserendo il volume nel dataset di training

Si è passati quindi a considerare come attributi i quattro valori utilizzati per la creazione dei candlestick: prezzo di chiusura, prezzo di apertura, minimo e massimo di giornata. Quest'idea è nata dalla considerazione che i trader, nella compravendita di tutti i giorni, utilizzano proprio questa tipologia di grafici per effettuare le loro considerazioni. La dimensione del dataset si è quindi quadruplicata portando ad notevole un incremento delle tempistiche di elaborazione. I risultati ottenuti (tabella 7.5) hanno mostrato come questi quattro valori aiutino realmente alla creazione del modello. In linea generale, si è ottenuto un miglioramento delle performance da

parte di tutti i classificatori; solamente in due casi, le prestazioni hanno subito un piccolo degrado.

	Anno 2011		Anno 2013	
	Short	Long	Short	Long
Decision Tree	1.25	1.24	0.95	1.23
SVM	1.34	1.23	0.89	1.06
Naïve Bayes	1.87	1.73	1.12	1.33
Neural Net	1.81	1.63	1.65	1.83

Tabella 7.5: FTSE MIB - Risultati percentuali medi ottenuti includendo apertura, chiusura, minimo e massimo nel dataset di input

Dimostrata l'efficacia nell'utilizzo di attributi tipici dell'analisi tecnica, si è voluto effettuare un ulteriore passo avanti. L'obiettivo è stato quello di sfruttare i pattern grafici proprio di questo campo per filtrare le azioni prima fornirle al classificatore. Utilizzando una libreria di pattern recognition si è cercato di applicare quest'idea. La metodologia è stata quella descritta nel Capitolo 5 e i risultati ottenuti sono stati abbastanza sorprendenti. Oltre a migliorare notevolmente i tempi di elaborazione, i profitti ottenuti hanno giovato di questa scelta garantendo un incremento di performance su tutti i classificatori, ad eccezione della rete neurale. Come visibile nella tabella 7.6, i guadagni ottenuti, nei migliori casi, si attestano intorno al 2%, valore decisamente alto per la tipologia di sperimentazione.

	Anno 2011		Anno 2013	
	Short	Long	Short	Long
Decision Tree	1.58	1.73	1.27	1.40
SVM	2.14	1.69	1.48	1.33
Naïve Bayes	2.64	2.20	1.79	1.64
Neural Net	1.75	1.48	1.37	1.58

Tabella 7.6: FTSE MIB - Risultati percentuali medi realizzati filtrando i dati

Completata la prima batteria di test, l'obiettivo è stato quello di comprendere se con un mercato decisamente più ampio e con continue variazioni ed influenze dovute all'esterno i risultati fossero comparabili. La scelta, come descritto durante questa tesi, è ricaduta sul mercato statunitense e nello specifico sull'indice S&P500. Vediamo ora i risultati ottenuti in questo contesto.

7.2 S&P500: anni 2011 e 2013

Partendo da quanto realizzato per l'indice FTSE MIB, si è voluto eseguire i medesimi test anche per il mercato statunitense. Dato che le dimensioni del dataset sono più di dieci volte superiori a quanto precedentemente testato, si è cercato di partire dalle simulazioni meno esose in termini di tempi. Oltretutto, si è dovuto accettare un compromesso: a causa della notevole richiesta computazionale, sono stati scartati i test con il classificatore basato sulle reti neurali. Infatti, le prime prove hanno stimato dei tempi di esecuzione non idonei al tipo di studio e quindi si è scelto di non eseguirli, considerando anche i risultati ottenuti sul mercato italiano.

Si è quindi cominciato testando il filtro basato sui pattern di analisi tecnica. I risultati, visibili in tabella 7.7, hanno confermato quanto concluso nei precedenti test; il classificatore migliore è risultato essere il Naïve Bayes.

	Anno 2011		Anno 2013	
	Short	Long	Short	Long
Decision Tree	1.52	1.48	1.16	1.52
SVM	1.90	1.42	1.20	1.90
Naïve Bayes	2.16	1.79	1.52	2.16

Tabella 7.7: S&P500 - Risultati percentuali medi con l'input formato dai soli prezzi di chiusura

I guadagni previsti hanno lo stesso ordine di grandezza di quanto visto con i precedenti test; l'aumento delle numero di titoli, anche se in maniera netta, non ha, almeno in questo caso, influenzato particolarmente la predizione. Vediamo ora se è possibile verificare lo stesso trend anche per le altre situazioni di test.

Dopo aver eseguito i test più rapidi, si è passati ad eseguire quelli più pesanti variando il dataset di input. Si è quindi creato il modello sfruttando solamente i prezzi di chiusura, poi aggiungendo il volume e infine considerando tutti gli altri attributi già descritti; purtroppo, per la stessa motivazione descritta precedentemente, è stato necessario tralasciare il test con l'albero decisionale e l'input più grande. Questa rinuncia non ha influenzato particolarmente l'analisi dei dati in quanto l'albero decisionale, come visto precedentemente, è risultato uno tra i classificatori meno performanti. I risultati ottenuti, visibili nelle tabelle 7.8 e 7.9, sono stati in linea con quanto visto con le simulazioni sulla Borsa di Milano: anche in questo caso l'inserimento del volume nel dataset non ha portato miglioramenti, mentre l'utilizzo degli attributi tipici dell'analisi tecnica ha reso i risultati più interessanti.

Quanto ottenuto sull'indice S&P500 ha quindi confermato ciò che è emerso nel caso del principale indice italiano. In linea generale, i guadagni più alti sono stati ottenuti grazie all'utilizzo di tutti quegli attributi che i trader utilizzano giornalmente nella loro attività; l'introduzione poi di un filtro ha reso l'elaborazione decisamente più veloce e ha permesso al classificatore di ridurre l'errore migliorando le performance.

	Anno 2011		Anno 2013	
	Short	Long	Short	Long
Decision Tree	0.86	0.99	0.68	0.86
SVM	0.35	0.47	0.11	0.35
Naïve Bayes	1.44	1.25	1.07	1.44

Tabella 7.8: S&P500 - Risultati percentuali medi ottenuti inserendo il volume nel dataset di training

	Anno 2011		Anno 2013	
	Short	Long	Short	Long
SVM	1.20	1.03	0.82	1.20
Naïve Bayes	1.62	1.42	1.27	1.62

Tabella 7.9: S&P500 - Risultati percentuali medi ottenuti includendo apertura, chiusura, minimo e massimo nel dataset di input

7.3 I test sul 2015

Per completare e rendere il più possibile attendibile lo studio, si è passati ad eseguire gli esperimenti considerando l'anno 2015. Questo periodo, a differenza degli altri due testati precedentemente, non ha avuto un andamento dei prezzi ben definito: i titoli hanno avuto un trend misto, senza che prevalesse quello ribassista o quello rialzista.

Sfruttando questa fascia temporale e utilizzando entrambi i mercati testati in precedenza, è stato possibile rendere complete le simulazioni. In questo modo, i vari classificatori saranno stati testati su due mercati di dimensione totalmente differente considerando strategie di investimento e fasce temporali con trend diversi tra loro.

Le modalità di test sono state le medesime di quelle utilizzate negli altri due anni presi in considerazione; per il mercato italiano la batteria di simulazioni è stata completa, per quello statunitense è stato tralasciato il classificatore basato sulle reti neurali e i test con l'albero decisionale e il dataset più ampio.

I risultati ottenuti in precedenza, visibili nelle tabelle 7.10 e 7.11, sono stati confermati anche in questo scenario. Il profitto medio ottenuto è in linea con le simulazioni precedenti e persino l'andamento dei vari classificatori è assimilabile.

Utilizzare un periodo dove non è stato identificato un trend ben preciso non ha portato a perdita di performance. Sotto quest'ottica, bisogna considerare il fatto che i precedenti test hanno garantito degli ottimi risultati sia con strategia long su periodi ribassisti sis con strategia short su periodi rialzisti. Il risultato qui ottenuto conferma quindi le aspettative, non avendo rilevato un degrado dei risultati nel caso la strategia di investimento non fosse la più idonea per il periodo considerato.

FTSE MIB - ANNO 2015

Attributi in input	Classificatore	Short	Long
Prezzo di chiusura	Decision Tree	0.58	0.93
	SVM	0.86	0.92
	Naïve Bayes	1.21	1.24
	Neural Net	2.02	1.35
Prezzo di chiusura e volume	Decision Tree	0.50	0.85
	SVM	0.32	0.38
	Naïve Bayes	1.07	1.11
	Neural Net	1.35	1.52
Prezzo di apertura, chiusura, minimo e massimo	Decision Tree	0.79	1.06
	SVM	0.88	0.95
	Naïve Bayes	1.22	1.31
	Neural Net	1.56	1.49
Prezzo di apertura, chiusura, minimo e massimo con utilizzo del filtro	Decision Tree	1.24	1.35
	SVM	1.33	1.36
	Naïve Bayes	1.73	1.68
	Neural Net	1.14	1.15

Tabella 7.10: FTSE MIB - Risultati percentuali medi per l'anno 2015

S&P500 - ANNO 2015

Attributi in input	Classificatore	Short	Long
Prezzo di chiusura	Decision Tree	0.89	0.88
	SVM	0.90	0.97
	Naïve Bayes	1.39	1.32
Prezzo di chiusura e volume	Decision Tree	0.79	0.76
	SVM	0.29	0.26
	Naïve Bayes	1.24	1.22
Prezzo di apertura, chiusura, minimo e massimo	SVM	0.91	0.98
	Naïve Bayes	1.34	1.36
Prezzo di apertura, chiusura, minimo e massimo con utilizzo del filtro	Decision Tree	1.32	1.35
	SVM	1.38	1.27
	Naïve Bayes	1.77	1.68

Tabella 7.11: S&P500 - Risultati percentuali medi per l'anno 2015

7.4 Gli effetti degli algoritmi e degli attributi

Analizzando nel loro insieme i risultati prodotti (tabelle presenti in Appendice) si può vedere come, per tutte le casistiche di test, l'andamento di performance dei classificatori sia stato costante. Per ogni dataset in input, le tecniche migliori sono risultate sempre essere le reti neurali e quelle bayesiane, con queste ultime che riescono ad ottenere il profitto più alto in assoluto nel caso di utilizzo del filtro.

Esaminando quanto ottenuto è possibile inoltre notare come il peggior risultato sia stato ottenuto mediante l'utilizzo delle SVM abbinato all'inserimento del volume di operazioni nel dataset. Questa tecnica, insieme agli alberi decisionali, si è dimostrata essere quella con le performance peggiori soprattutto quando il numero di attributi ha iniziato a crescere. È possibile quindi sostenere che l'aumento delle dimensioni del training set peggiori le prestazioni di questi classificatori. Questo trend è invece risultato opposto nel caso delle altre due tecniche utilizzate. Le performance delle reti neurali crescono all'aumentare del numero di attributi; è possibile notare infatti come il guadagno migliore, considerando solo questo classificatore, è stato ottenuto nel caso del dataset più ampio. Anche la tecnica Naïve Bayes migliora le proprio prestazioni con il crescere del training set, ma si è visto come la selezione degli attributi mediante i pattern grafici abbia reso particolarmente efficace questo metodo di classificazione.

In conclusione possiamo confermare quanto visto in maniera teorica nei capitoli precedenti. Le tecniche che hanno portato ad un profitto maggiore sono state quelle che, data la loro natura, hanno beneficiato dell'aumento di attributi. Questi ultimi forniscono sicuramente informazioni utili alla creazione di un modello performante, ma se il classificatore non gode di un'ottima scalabilità, come visto, si ottiene solamente un degrado delle prestazioni.

7.5 I tempi di elaborazione

I tempi di elaborazione sono sicuramente un punto su cui fare attenzione. Per il tipo di strategia di investimento di questo studio diventa importante essere in grado di effettuare una predizione nel più breve tempo possibile e, considerando ciò, sono stati collezionati i tempi di elaborazione di ciascuna configurazione di test. Come visibile nella tabella A.13, i tempi, nel caso dell'indice italiano, sono risultati abbastanza costanti per le tecniche di classificazione più semplici, con le reti neurali totalmente distaccate. Questa tecnica, come già visto, risulta essere molto robusta a discapito però di un processo training molto lento. L'applicazione del filtro ha portato a un risparmio di tempo ma senza eccellere particolarmente in tal senso: le dimensioni ridotte dell'input non hanno permesso infatti un notevole taglio sotto questo aspetto.

Utilizzando invece i titoli presenti sul mercato statunitense i tempi di elaborazione perdono di costanza. Le dimensioni notevoli del dataset rendono l'efficacia della tecnica di classificazione molto importante e l'hardware utilizzato la fa da padrone in tal senso. Le configurazioni hardware utilizzate in questo studio non sono di livello professionale ed elaborazioni così lunghe possono venire influenzate dallo stato dell'elaboratore o da altri task che vengono eseguiti in background.

In linea generale, è possibile sostenere che le caratteristiche proprie di un classificatore, in termini di tempi di creazione del modello, entrano in gioco all'aumentare delle dimensioni del training set; con input ridotti, le caratteristiche di scalabilità

non influenzano il risultato, portando ad avere tempi simili indipendentemente dalla tecnica utilizzata.

Attributi in input	Classificatore	FTSE MIB	S&P500
Prezzo di chiusura	Decision Tree	3,65	270,05
	SVM	3,65	14,48
	Naïve Bayes	3,59	13,26
	Neural Net	7,88	
Prezzo di chiusura e volume	Decision Tree	4,47	1315,26
	SVM	3,94	20,64
	Naïve Bayes	3,53	20,65
	Neural Net	19,50	
Prezzo di apertura, chiusura, minimo e massimo	Decision Tree	5,34	0,00
	SVM	3,45	70,56
	Naïve Bayes	3,48	69,79
	Neural Net	57,87	
Prezzo di apertura, chiusura, minimo e massimo con utilizzo del filtro	Decision Tree	3,32	18,73
	SVM	3,28	6,14
	Naïve Bayes	3,23	6,16
	Neural Net	3,35	

Tabella 7.12: Tempi di esecuzione medi per la predizione di un singolo giorno

7.6 Il filtraggio dei dati

Come descritto in questo capitolo, il filtraggio basato sui pattern tipici dell'analisi tecnica ha portato a notevoli risultati. È stato possibile evidenziare un una riduzione tempi di elaborazione ma è dal punto di vista dei profitti che l'incremento di performance è stato notevole.

Le tempistiche di modellazione, utilizzando la tecnica descritta nel Capitolo 5, sono state ridotte di circa il 50%, dimostrando come il filtro utilizzato abbia una grande capacità di selezione dei titoli. È possibile infatti notare nella tabella 7.13, come il valore medio di filtraggio, ottenuto nelle varie casistiche di test, si attesti intorno al 70%, valore ben superiore alle aspettative.

L'utilizzo di questo metodo ha confermato come le strategie normalmente utilizzate dai trader siano realmente efficaci; i pattern grafici individuati sui grafici a candele giapponesi danno realmente indicazioni sull'andamento dei titoli e, integrati con tecniche di data mining, possono diventare fondamentali nella scelta del giusto investimento.

	Anno 2011		Anno 2013		Anno 2015	
Mercato	Short	Long	Short	Long	Short	Long
FTSE MIB	67.52	69.93	72.06	69.93	70.97	67.31
S&P500	70.87	64.14	72.64	66.82	71.95	67.35

Tabella 7.13: Percentuali di filtraggio medie ottenute con l'utilizzo dei pattern di analisi tecnica

Capitolo 8

Conclusioni e sviluppi futuri

In seguito alle sperimentazioni eseguite in questa tesi, è ora possibile effettuare delle considerazioni sullo studio svolto. In questa sezione verranno riassunte e concentrate tutte le conclusioni che è stato possibile trarre; inoltre, verranno discussi eventuali sviluppi futuri che potrebbero portare ad ulteriori conferme o aprire nuove strade.

8.1 Conclusioni e considerazioni finali

L'obiettivo di questo lavoro di ricerca è stato quello di capire se le tecniche di classificazione possano essere utilizzate, non soltanto per predire la direzione del mercato (salita o discesa), ma anche per generare segnali di trading intraday (acquista/vendi un determinato sottoinsieme di azioni). A tale scopo, sono stati analizzati i livelli di confidenza delle predizioni effettuate, separatamente per ciascun titolo, e in base a questi valori è stato deciso quali segnali di trading generare. In aggiunta, si è cercato di valutare se il riconoscimento di pattern grafici derivanti dall'analisi tecnica possa essere efficace per filtrare le feature di partenza utilizzate per l'addestramento del classificatore. Anzichè arricchire la collezione di dati con nuovi indicatori e statistiche (problema comunemente affrontato in letteratura), l'analisi si è focalizzata su come ridurre l'insieme di azioni da considerare per il training del classificatore e quindi per la successiva generazione di segnali di trading.

Per realizzare quanto appena descritto, è stata considerata una storicità dei prezzi di dieci giorni; il classificatore, sfruttando questi dati, ha costruito un modello sul quale ha effettuato una predizione che è stata poi inserita nel portafoglio di investimento. Ottenuti i titoli consigliati è stato effettuato un confronto con i dati reali e sono state verificate l'affidabilità e le prestazioni della tecnica scelta.

I test eseguiti confermano che la generazione di segnali di trading, basati su predizioni ad alta confidenza generate da tecniche di classificazione, produce risultati promettenti e mediamente superiori a tecniche di regressione tradizionali. Inoltre, l'integrazione di un filtro basato sul riconoscimento di pattern di analisi tecnica consente di ridurre significativamente il rumore nei dati di train aumentando di conseguenza la precisione delle previsioni effettuate. I classificatori che garantiscono le migliori performance, in linea generale, si sono dimostrati essere le reti neurali e quelle bayesiane; l'albero decisionale e l'SVM infatti non sono mai riusciti a garantire profitti più alti rispetto alle altre due tecniche.

Valutando l'influenza dell'analisi tecnica, è possibile quindi confermare come le strategie utilizzate dai trader forniscano un importante aiuto nella scelta del cor-

retto investimento; i risultati ottenuti infatti hanno dimostrato che l'introduzione di valori caratteristici di quel settore rende la costruzione del modello più efficace e conseguentemente porta a guadagni più alti. Le prestazioni migliori, in linea generale, sono state ottenute utilizzando in input il prezzo di chiusura, il prezzo di apertura, il minimo e il massimo di giornata; inserendo poi un'altra caratteristica dell'analisi tecnica, quali i pattern grafici, e filtrando in base ad essi è stato possibile notare un ulteriore incremento di performance. Con quest'ultimo metodo sono stati raggiunti valori sorprendenti, arrivando a superare il 2% medio giornaliero. L'attributo invece che non ha reso più performante il sistema è stato il volume giornaliero di operazioni; i test eseguiti sfruttando questo valore non hanno evidenziato nessun miglioramento ma, per alcuni classificatori, ha addirittura fatto registrare i peggiori profitti di tutta la sperimentazione. Riassumendo i punti chiave di questo studio, è possibile dire che:

- La variazione del mercato analizzato non comporta una cambiamento di performance. La principale differenza sta nei tempi di creazione del modello: un numero di titoli più alto comporta ovviamente una fase di induzione più lenta.
- Nel caso di training set ridotto, la maggior parte degli algoritmi impiega lo stesso tempo. Le proprietà di scalabilità di un classificatore non entrano in gioco in questo caso.
- Intersecare l'analisi tecnica e gli algoritmi di data mining ha portato a un miglioramento delle prestazioni. Ampliare il dataset con alcuni attributi utilizzati dai trader ha reso il modello più accurato.
- Le predizioni non sono influenzate né dalla strategia di investimento scelta né dal trend di mercato che caratterizza la fascia temporale del test. Investire contro-tendenza ha garantito comunque ottime prestazioni.

In ogni caso, bisogna notare come nessuna simulazione, se opportunamente configurata, abbia generato delle perdite; infatti questo risultato è reso possibile dall'applicazione dello stop loss ed è quindi un'ulteriore conferma di come gli automatic trading systems siano realmente efficaci se in mani esperte.

In conclusione, è possibile dire che, almeno a livello teorico, sfruttando il sistema sviluppato in questa tesi, sarebbe possibile, considerando un periodo lungo un anno, garantire un guadagno medio giornaliero pari a circa il 2%. Questo valore, come già espresso, è decisamente sorprendente e ad oggi non esistono soluzioni di pubblico dominio con queste prestazioni.

8.2 Sviluppi futuri ed eventuali approfondimenti

Partendo da quanto sviluppato, è sicuramente possibile approfondire alcune tematiche ricercando performance ancora migliori.

È stato dimostrato come una discretizzazione più restrittiva abbia reso di più rispetto ad una dove la fascia di uguaglianza fosse meno ampia; in quest'ottica, diventa interessante testare altre forme di discretizzazione tra cui, ad esempio, quella dove un titolo è considerato in rialzo solo se ha una crescita superiore al 2%. Il contro di questo metodo è che se la discretizzazione diventa troppo poco permissiva c'è il

rischio di non generare alcuna predizione; diventa quindi fondamentale effettuare un corretto trade-off tra l'effettiva capacità di produrre risultati e le prestazioni.

Un altro punto interessante da approfondire è sicuramente quello relativo alle configurazioni dei classificatori; come descritto nel Capitolo 5, la scelta è stata quella di utilizzare i parametri di default, ma esistono un moltitudine di possibilità. Tenendo bene a mente l'importanza e l'esigenza di generare la singola previsione in un tempo breve, variare questi parametri potrebbe portare ad ottenere risultati ancora più importanti. Anche in questo caso è necessario effettuare un compromesso: in questo caso tra prestazioni e tempi di predizione.

Entrando nel settore dell'analisi tecnica ed avendo evidenziato l'efficacia dell'applicazione dei pattern grafici alle tecniche di data mining, una sperimentazione da poter approfondire è sicuramente quella relativa al filtraggio dei titoli. In questa tesi, è stato scelto di eliminare le azioni che presentavano un pattern contro tendenza rispetto quello ricercato; l'idea potrebbe essere invece quella di utilizzare solamente le azioni che presentano un indicatore concorde alla strategia di investimento scelta. Considerando che il filtro, utilizzato come in questo studio, rimuove circa il 70% dei titoli, esiste il rischio concreto di generare pochissime predizioni. Se questo rischio è considerato accettabile, quest'opzione diventa sicuramente interessante da testare.

Concludendo, si potrebbe provare ad eseguire una sperimentazione unendo i dataset già utilizzati in questa tesi. Nonostante si sia visto come il volume non apporti un miglioramento ai risultati, l'idea potrebbe essere quella di unire questo valore con i quattro utilizzati per la costruzione dei diagrammi a candele. Diventerebbe interessante capire se arricchire il dataset che ha prodotto il miglior risultato con quest'altra informazione possa generare una perdita di prestazioni oppure no.

Appendice A

Appendice

Di seguito sono riportati integralmente tutti i risultati ottenuti con le varie configurazioni di test. In questo contesto, con la variabile C_t si intende il costo di commissione applicato ad ogni transazione.

È presente inoltre, al fondo di questa appendice, una tabella riepilogativa dei tempi di elaborazione impiegati. Tutti i dati sono espressi in secondi e rapportati all'esecuzione sull'elaboratore 1 descritto nel Capitolo 6.

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.67	0.52	1.19	1.04
	SVM	0.43	0.28	1.44	1.29
	Naïve Bayes	0.87	0.72	1.87	1.72
	Neural Net	-0.93	-1.08	1.78	1.63
Prezzo di chiusura e volume	Decision Tree	0.60	0.45	1.24	1.09
	SVM	0.14	-0.01	0.77	0.62
	Naïve Bayes	0.77	0.62	1.72	1.57
	Neural Net	-0.95	-1.10	1.73	1.58
Prezzo di apertura. chiusura. minimo e massimo	Decision Tree	0.94	0.79	1.40	1.25
	SVM	0.82	0.67	1.49	1.34
	Naïve Bayes	1.53	1.38	2.02	1.87
	Neural Net	0.62	0.47	1.96	1.81
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	1.47	1.32	1.73	1.58
	SVM	1.95	1.80	2.29	2.14
	Naïve Bayes	2.61	2.46	2.79	2.64
	Neural Net	1.65	1.50	1.90	1.75

Tabella A.1: FTSE MIB - Anno 2011 - Strategia short

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.47	0.32	1.24	1.09
	SVM	0.59	0.44	1.30	1.15
	Naïve Bayes	1.15	1.00	1.71	1.56
	Neural Net	0.73	0.58	1.63	1.48
Prezzo di chiusura e volume	Decision Tree	0.60	0.45	1.22	1.07
	SVM	0.07	-0.08	0.61	0.46
	Naïve Bayes	1.16	1.01	1.63	1.48
	Neural Net	1.30	1.15	1.72	1.57
Prezzo di apertura. chiusura. minimo e massimo	Decision Tree	0.84	0.69	1.39	1.24
	SVM	0.80	0.65	1.38	1.23
	Naïve Bayes	1.54	1.39	1.88	1.73
	Neural Net	1.41	1.26	1.78	1.63
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	1.64	1.49	1.88	1.73
	SVM	1.62	1.47	1.84	1.69
	Naïve Bayes	2.22	2.07	2.35	2.20
	Neural Net	1.33	1.18	1.60	1.45

Tabella A.2: FTSE MIB - Anno 2011 - Strategia long

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.45	0.30	0.96	0.81
	SVM	-0.43	-0.58	1.02	0.87
	Naïve Bayes	-2.52	-2.67	1.19	1.04
	Neural Net	-0.07	-0.22	1.55	1.40
Prezzo di chiusura e volume	Decision Tree	0.13	-0.02	0.88	0.73
	SVM	-1.24	-1.39	0.42	0.27
	Naïve Bayes	-2.31	-2.46	1.20	1.05
	Neural Net	1.40	1.25	1.58	1.43
Prezzo di apertura. chiusura. minimo e massimo	Decision Tree	0.66	0.51	1.10	0.95
	SVM	-0.41	-0.56	1.04	0.89
	Naïve Bayes	-1.83	-1.98	1.27	1.12
	Neural Net	-0.56	-0.71	1.80	1.65
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	1.30	1.15	1.42	1.27
	SVM	1.47	1.32	1.63	1.48
	Naïve Bayes	1.86	1.71	1.94	1.79
	Neural Net	1.37	1.22	1.52	1.37

Tabella A.3: FTSE MIB - Anno 2013 - Strategia short

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.56	0.41	1.22	1.07
	SVM	-0.02	-0.17	1.22	1.07
	Naïve Bayes	0.75	0.60	1.62	1.47
	Neural Net	1.41	1.26	1.88	1.73
Prezzo di chiusura e volume	Decision Tree	0.56	0.41	1.11	0.96
	SVM	-0.01	-0.16	0.56	0.41
	Naïve Bayes	0.44	0.29	1.32	1.17
	Neural Net	1.97	1.82	2.12	1.97
Prezzo di apertura. chiusura. minimo e massimo	Decision Tree	0.50	0.35	1.38	1.23
	SVM	0.02	-0.13	1.21	1.06
	Naïve Bayes	0.60	0.45	1.48	1.33
	Neural Net	1.88	1.73	1.98	1.83
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	0.89	0.74	1.55	1.40
	SVM	-0.04	-0.19	1.48	1.33
	Naïve Bayes	0.99	0.84	1.79	1.64
	Neural Net	0.67	0.52	1.73	1.58

Tabella A.4: FTSE MIB - Anno 2013 - Strategia long

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	-0.81	-0.96	0.73	0.58
	SVM	-0.11	-0.26	1.01	0.86
	Naïve Bayes	-0.65	-0.80	1.36	1.21
	Neural Net	1.22	1.07	2.17	2.02
Prezzo di chiusura e volume	Decision Tree	-0.37	-0.52	0.65	0.50
	SVM	-0.21	-0.36	0.47	0.32
	Naïve Bayes	-0.17	-0.32	1.22	1.07
	Neural Net	0.62	0.47	1.50	1.35
Prezzo di apertura. chiusura. minimo e massimo	Decision Tree	-0.35	-0.50	0.94	0.79
	SVM	-0.18	-0.33	1.03	0.88
	Naïve Bayes	-0.03	-0.18	1.37	1.22
	Neural Net	1.49	1.34	1.71	1.56
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	1.22	1.07	1.39	1.24
	SVM	1.23	1.08	1.48	1.33
	Naïve Bayes	1.76	1.61	1.88	1.73
	Neural Net	1.09	0.94	1.29	1.14

Tabella A.5: FTSE MIB - Anno 2015 - Strategia short

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.38	0.23	1.08	0.93
	SVM	0.45	0.30	1.07	0.92
	Naïve Bayes	0.79	0.64	1.39	1.24
	Neural Net	0.82	0.67	1.50	1.35
Prezzo di chiusura e volume	Decision Tree	0.48	0.33	1.00	0.85
	SVM	0.02	-0.13	0.53	0.38
	Naïve Bayes	0.82	0.67	1.26	1.11
	Neural Net	1.46	1.31	1.67	1.52
Prezzo di apertura. chiusura. minimo e massimo	Decision Tree	0.72	0.57	1.21	1.06
	SVM	0.47	0.32	1.10	0.95
	Naïve Bayes	0.89	0.74	1.46	1.31
	Neural Net	0.97	0.82	1.64	1.49
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	0.70	0.55	1.50	1.35
	SVM	1.15	1.00	1.51	1.36
	Naïve Bayes	1.20	1.05	1.83	1.68
	Neural Net	-0.15	-0.30	1.30	1.15

Tabella A.6: FTSE MIB - Anno 2015 - Strategia long

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.56	0.41	1.04	0.89
	SVM	0.71	0.56	1.31	1.16
	Naïve Bayes	1.23	1.08	1.68	1.53
Prezzo di chiusura e volume	Decision Tree	0.57	0.42	1.01	0.86
	SVM	0.19	0.04	0.50	0.35
	Naïve Bayes	1.10	0.95	1.59	1.44
Prezzo di apertura. chiusura. minimo e massimo	SVM	0.75	0.60	1.35	1.20
	Naïve Bayes	1.35	1.20	1.77	1.62
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	1.46	1.31	1.67	1.52
	SVM	1.77	1.62	2.05	1.90
	Naïve Bayes	2.13	1.98	2.31	2.16

Tabella A.7: S&P500 - Anno 2011 - Strategia short

Attributi in input	Classificatore	$Stop\ loss = 0\%$		$Stop\ loss = 1\%$	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.78	0.63	1.20	1.05
	SVM	0.78	0.63	1.17	1.02
	Naïve Bayes	1.17	1.02	1.51	1.36
Prezzo di chiusura e volume	Decision Tree	0.74	0.59	1.14	0.99
	SVM	0.26	0.11	0.62	0.47
	Naïve Bayes	1.08	0.93	1.40	1.25
Prezzo di apertura. chiusura. minimo e massimo	SVM	0.80	0.65	1.18	1.03
	Naïve Bayes	1.29	1.14	1.57	1.42
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	1.44	1.29	1.63	1.48
	SVM	1.38	1.23	1.57	1.42
	Naïve Bayes	1.79	1.64	1.94	1.79

Tabella A.8: S&P500 - Anno 2011 - Strategia long

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.66	0.51	0.87	0.72
	SVM	0.62	0.47	0.92	0.77
	Naïve Bayes	1.15	1.00	1.34	1.19
Prezzo di chiusura e volume	Decision Tree	0.62	0.47	0.83	0.68
	SVM	0.07	-0.08	0.26	0.11
	Naïve Bayes	0.99	0.84	1.22	1.07
Prezzo di apertura. chiusura. minimo e massimo	SVM	0.68	0.53	0.97	0.82
	Naïve Bayes	1.24	1.09	1.42	1.27
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	1.20	1.05	1.31	1.16
	SVM	1.23	1.08	1.35	1.20
	Naïve Bayes	1.59	1.44	1.67	1.52

Tabella A.9: S&P500 - Anno 2013 - Strategia short

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.95	0.80	1.08	0.93
	SVM	0.93	0.78	1.10	0.95
	Naïve Bayes	1.39	1.24	1.51	1.36
Prezzo di chiusura e volume	Decision Tree	0.82	0.67	0.95	0.80
	SVM	0.29	0.14	0.41	0.26
	Naïve Bayes	1.20	1.05	1.33	1.18
Prezzo di apertura. chiusura. minimo e massimo	SVM	0.96	0.81	1.12	0.97
	Naïve Bayes	1.40	1.25	1.51	1.36
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	1.31	1.16	1.40	1.25
	SVM	1.29	1.14	1.41	1.26
	Naïve Bayes	1.72	1.57	1.78	1.63

Tabella A.10: S&P500 - Anno 2013 - Strategia long

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.81	0.66	1.04	0.89
	SVM	0.68	0.53	1.05	0.90
	Naïve Bayes	1.28	1.13	1.54	1.39
Prezzo di chiusura e volume	Decision Tree	0.72	0.57	0.94	0.79
	SVM	0.25	0.10	0.44	0.29
	Naïve Bayes	1.11	0.96	1.39	1.24
Prezzo di apertura. chiusura. minimo e massimo	SVM	0.69	0.54	1.06	0.91
	Naïve Bayes	1.23	1.08	1.49	1.34
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	1.38	1.23	1.47	1.32
	SVM	1.35	1.20	1.53	1.38
	Naïve Bayes	1.80	1.65	1.92	1.77

Tabella A.11: S&P500 - Anno 2015 - Strategia short

Attributi in input	Classificatore	*Stop loss* = 0%		*Stop loss* = 1%	
		$C_t = 0\%$	$C_t = 0.15\%$	$C_t = 0\%$	$C_t = 0.15\%$
Prezzo di chiusura	Decision Tree	0.84	0.69	1.03	0.88
	SVM	0.93	0.78	1.12	0.97
	Naïve Bayes	1.33	1.18	1.47	1.32
Prezzo di chiusura e volume	Decision Tree	0.72	0.57	0.91	0.76
	SVM	0.23	0.08	0.41	0.26
	Naïve Bayes	1.21	1.06	1.37	1.22
Prezzo di apertura. chiusura. minimo e massimo	SVM	0.94	0.79	1.13	0.98
	Naïve Bayes	1.38	1.23	1.51	1.36
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	Decision Tree	1.42	1.27	1.50	1.35
	SVM	1.32	1.17	1.42	1.27
	Naïve Bayes	1.77	1.62	1.82	1.67

Tabella A.12: S&P500 - Anno 2015 - Strategia long

Attributi in input	Classificatore	Anno 2011		Anno 2013		Anno 2015	
		Short	Long	Short	Long	Short	Long
Prezzo di chiusura	DecisionTree	3.95	4.00	4.19	3.25	3.26	3.25
	LibSVM	3.76	3.81	4.14	4.00	3.13	3.04
	Naïve Bayes	3.67	3.69	3.90	3.95	3.17	3.18
	Neural Net	7.67	7.62	8.48	8.33	7.64	7.56
Prezzo di chiusura e volume	DecisionTree	3.95	4.00	4.19	3.25	3.26	3.25
	LibSVM	3.76	3.81	4.14	4.00	3.13	3.04
	Naïve Bayes	3.67	3.69	3.90	3.95	3.17	3.18
	Neural Net	7.67	7.62	8.48	8.33	7.64	7.56
Prezzo di apertura. chiusura. minimo e massimo	DecisionTree	3.95	4.00	4.19	3.25	3.26	3.25
	LibSVM	3.76	3.81	4.14	4.00	3.13	3.04
	Naïve Bayes	3.67	3.69	3.90	3.95	3.17	3.18
	Neural Net	7.67	7.62	8.48	8.33	7.64	7.56
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	DecisionTree	3.95	4.00	4.19	3.25	3.26	3.25
	LibSVM	3.76	3.81	4.14	4.00	3.13	3.04
	Naïve Bayes	3.67	3.69	3.90	3.95	3.17	3.18
	Neural Net	7.67	7.62	8.48	8.33	7.64	7.56

Tabella A.13: FTSE MIB - Tempi di esecuzione medi per la singola predizione

Attributi in input	Classificatore	Anno 2011		Anno 2013		Anno 2015	
		Short	Long	Short	Long	Short	Long
Prezzo di chiusura	DecisionTree	201.72	347.35	249.11	292.47	260.50	269.18
	LibSVM	13.41	13.95	17.33	13.71	14.56	13.90
	Naïve Bayes	13.61	13.29	13.54	13.18	14.17	11.75
Prezzo di chiusura e volume	DecisionTree	1565.22	1179.81	1252.52	1420.00	1337.91	1136.12
	LibSVM	24.05	23.00	18.52	17.36	19.13	21.78
	Naïve Bayes	22.95	20.40	20.01	21.51	19.42	19.58
Prezzo di apertura. chiusura. minimo e massimo	LibSVM	71.81	69.55	72.88	73.34	69.22	66.57
	Naïve Bayes	69.46	71.42	72.02	72.30	67.07	66.49
Prezzo di apertura. chiusura. minimo e massimo con utilizzo del filtro	DecisionTree	17.75	18.99	18.53	18.27	20.31	18.54
	LibSVM	7.53	6.92	5.52	6.14	4.85	5.86
	Naïve Bayes	6.48	7.37	5.49	3.54	8.69	5.38

Tabella A.14: S&P500 - Tempi di esecuzione medi per la singola predizione

Bibliografia

[1] *Automatic Trading Systems*. URL: https://www.investopedia.com/articles/trading/11/automated-trading-systems.asp.

[2] E. Baralis. "Classification fundamentals".

[3] E. Baralis et al. "Discovering profitable stocks for intraday trading". In: *Information Sciences 405* (2017).

[4] L. Breiman. "Statistical modeling: The two cultures". In: *Statistical Science* (2001).

[5] *Efficient Market Hypothesis - EMH*. URL: http://www.performancetrading.it/Documents/GfAnalisi/GfA_aEfficientMarket.htm.

[6] E. Fama. "Random walks in stock market prices". In: *Financial Analysts Journal, 21* (1965).

[7] E. Fama. "The behaviour of stock market prices". In: *Journal of Business, 38* (1965).

[8] *Financial Planning for Professional Athletes: An Inside Look*. URL: http://proathletewealthadvisor.com/financial-planning-professional-athletes-inside-look/.

[9] *Flash Crash*. URL: https://www.investopedia.com/terms/f/flash-crash.asp.

[10] *FTSE MIB*. URL: http://www.borsaitaliana.it/borsa/indici/indici-in-continua/dettaglio.html?indexCode=FTSEMIB.

[11] C. Hargreaves e Y. Hiao. "Prediction of stock performance using analytical techniques". In: (2012).

[12] C. Haur Koh, P. K. H. Phua e X. Zhu. "Forecasting stock index increments using neural networks with trust region methods". In: *Proceedings of the International Joint Conference on Neural Networks, volume 1* (2003).

[13] J.P. Ignizio e L. Burke. "A practical overview of neural networks". In: *Journal of Intelligent Manufacturing, 8* (1997).

[14] A.P. Isern-Deya, M. Miro-Julia e G. Fiol-Roig. "Decision trees in stock market analysis: Construction and validation". In: *Proceedings of the 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems - Volume Part I, 2010* (2010).

[15] *Italy Stock Market (FTSE MIB)*. URL: https://it.tradingeconomics.com/italy/stock-market.

[16] Han Kamber. *Data mining; Concepts and Techniques*. 2006.

[17] *Knowledge Discovery in Databases (KDD)*. URL: http://www.infovis-wiki.net/index.php?title=Knowledge_Discovery_in_Databases_(KDD).

[18] *La borsa valori e la sua storia*. URL: http://gianfrancomarini.blogspot.it/2017/02/la-borsa-valori-e-la-sua-storia.html.

[19] D. Lee. *YahooFinanceAPI*. URL: https://github.com/dennislwy/YahooFinanceAPI.

[20] T.C. Mills. "Nonlinear time series models in economics". In: *Journal of Economic Surveys, 5(3)* (1990).

[21] N. Milosevic. "Equity forecast: Predicting long term stock price movement using machine learning". In: *School of Computer Science, University of Manchester, UK* (2016).

[22] P. P. Tan, M. Steinbach e V. V. Kumar. *Introduction to Data Mining*.

[23] P. Paliyawan. "Stock market direction prediction using data mining classification". In: *ARPN Journal of Engineering and Applied Sciences, 10(3)* (2015).

[24] *Posizioni long e short*. URL: http://www.valoreazioni.com/approfondimenti-finanziari/posizioni-long-e-short.html.

[25] M.B. Priestley. "Non-linear and non-stationary time series analysis". In: *Applied Statistics, 39(2)* (1988).

[26] R. R. Grossman, C. Kamath e V. Kumar. *Data Mining for Scientific and Engineering Applications*.

[27] J.W. Shavlik e M.W. Craven. "Understanding time-series networks: A case study in rule extraction". In: *International Journal of Neural Systems, 4* (1997).

[28] *S&P 500 Index - 90 Year Historical Chart*. URL: http://www.macrotrends.net/2324/sp-500-historical-chart-data.

[29] *TA-Lib : Technical Analysis Library*. URL: http://ta-lib.org/.

[30] R.S. Thakur, S. Kamley e S. Jaloree. "An association rule mining model for nding the interesting patterns in stock market dataset". In: *International Journal of Computer Applications* (2014).

[31] *The 5 Most Powerful Candlestick Patterns*. URL: https://www.investopedia.com/articles/active-trading/092315/5-most-powerful-candlestick-patterns.asp.

[32] *Trading: il significato spiegato in parole semplici*. 2017. URL: https://www.binaryoptioneurope.com/2017/04/10/trading-significato-spiegato-con-parole-semplici-in-italiano.

[33] *Trading intraday: significato e consigli pratici. La guida*. URL: https://www.money.it/Trading-intraday-significato-consigli-guida.

[34] *Treccani: definizione di borsa*. URL: http://www.treccani.it/enciclopedia/borsa.

[35] H.R. Vairan. "Big data: New tricks for econometrics". In: *Journal of Economic Perspectives* (2014).

[36] E. Vityaev e B. Kovalerchuk. “Data mining for financial applications”. In: *he Data Mining and Knowledge Discovery Handbook* (2005).

[37] Y.F. Wang. “Mining stock price using fuzzy rough system”. In: *Expert Systems with Application* (2003).

[38] P. Zhang. “Neural networks for data mining”. In: *Data Mining and Knowledge Discovery Handbook, 2nd ed.* (2010).