



UNIVERSITÀ DEGLI STUDI DI UDINE

Dipartimento di Scienze Matematiche, Informatiche e Fisiche

Corso di Laurea Triennale in Internet of Things, Big Data & Web

Tesi di Laurea

INTELLIGENZA ARTIFICIALE IN FINANZA

Relatore:

Prof. GIUSEPPE SERRA

Laureando:

FELIPE NICODEMO RIVA

ANNO ACCADEMICO 2022-2023

Ai miei genitori e a tutta la mia famiglia

Indice

Introduzione	xi
1 AI e Finanza	xii
1.1 Introduzione dell'intelligenza artificiale nei mercati azionari	xii
1.2 Alcune attuali e future applicazioni dell'intelligenza artificiale in finanza	xiii
2 Obiettivo della tesi	xvi
I Trading Azionario	1
1 Le basi del trading azionario	3
1.1 Il mercato azionario e le sue componenti	3
1.2 Tipologie di azioni e le loro caratteristiche	4
1.3 Strategie di trading per principianti	5
1.4 Gestione del rischio nel trading azionario	6
1.5 Analisi tecnica e analisi fondamentale	7
1.6 Indicatori di mercato e dati economici	8
1.7 Leggi e regolamenti	9
II Finanza e Machine Learning	11
2 La Finanza Normativa	13
2.1 Incertezza e Rischio	13
2.1.1 Definizioni	13
2.1.2 Esempio Numerico	17
2.2 Teoria dell'utilità attesa (Expected Utility Theory EUT)	18
2.2.1 Assiomi e teoria normativa	18
2.2.2 Preferenze di un agente	19
2.2.3 Utility Functions	20

2.2.4	Funzioni di utilità attesa	20
2.2.5	Avversione al rischio	21
2.2.6	Esempio numerico	21
2.3	Mean Variance Portfolio Theory (MVP)	24
2.3.1	Presupposti e risultati	25
2.3.2	Distribuzione Normale Implicitamente Assunta	25
2.3.3	Statistiche di portafoglio	26
2.3.4	Sharpe Ratio	27
2.3.5	Esempio Numerico	27
2.4	Capital Asset Pricing Model (CAPM)	34
2.4.1	Presupposti e risultati	35
2.4.2	Esempio Numerico	37
2.5	Conclusioni	42
3	Finanza guidata dai dati	45
3.1	Metodo scientifico	45
3.2	Econometria finanziaria e Regressione	46
3.3	Disponibilità dei Dati	49
3.3.1	Dati Alternativi	50
3.4	Teorie normative rivisitate	56
3.4.1	Utilità attesa e realtà	56
3.4.2	Previsioni del comportamento basati sui dati	62
3.4.3	Teoria della Media-Varianza del Portafoglio (MVP)	62
3.4.4	Capital Asset Pricing Model	69
3.5	Sfatare gli assunti centrali	72
3.5.1	Rendimenti a distribuzione normale	73
3.5.2	Relazioni Lineari	82
3.6	Conclusioni	84
4	Finanza AI-First	87
4.1	Mercati Efficienti	88
4.1.1	Sostenitori della teoria dei mercati efficienti	91
4.1.2	Detrattori della teoria dei mercati efficienti	92
4.2	Previsioni di mercato basate sui dati dei rendimenti	97
4.3	Previsione di mercato con più features	102
4.4	Previsione del mercato Intraday	105
4.5	Conclusioni	107

III Scoprire e Sfruttare le Inefficienze	109
5 Reti neurali profonde (DNNs)	113
5.1 I Dati	113
5.2 Previsione di Base	114
5.3 Normalizzazione	116
5.4 Dropout	117
5.5 Regolarizzazione	118
5.6 Bagging	121
5.7 Ottimizzatori	122
5.8 Conclusioni	122
6 Backtest Vettorializzato	123
6.1 Backtesting di una strategia basata sulle SMA	124
6.2 Backtesting di una strategia giornaliera basata su DNN	130
6.3 Backtesting di una strategia intraday basata su DNN	133
6.4 Conclusioni	136
IV Metodi di IA & ML e applicazioni in Finanza	139
7 IA & ML nella finanza: Panoramica	141
7.1 Machine Learning, Deep Learning, Intelligenza Artificiale e Data Science	142
7.2 Tipologie di apprendimento automatico	144
7.2.1 Supervisionato	144
7.2.2 Non Supervisionato	145
7.2.3 Natural Language Processing	148
7.3 Reti Neurali Artificiali (ANNs)	149
7.4 Applicazioni dell'Intelligenza Artificiale nella finanza	149
7.4.1 Trading Algoritmico	150
7.4.2 Intercettazione delle frodi	152
7.4.3 Previsione del riciclaggio di denaro	153
7.4.4 Analisi dei documenti	154
7.4.5 Gestione del rischio	155
7.4.6 Chatbot	156
7.4.7 Sottoscrizione di prestiti e assicurazioni	157
7.4.8 Applicazioni future dell'IA in finanza	158
7.5 Conclusioni	159

8 Natural Language Processing (NLP)	161
8.1 NLP: Teoria e concetti	163
8.1.1 Preelaborazione	164
8.2 Rappresentazione delle features	169
8.2.1 Bag of Words - word count	170
8.2.2 TF-IDF	171
8.2.3 Word embedding	173
8.3 Inferenza	175
8.3.1 Esempio di apprendimento supervisionato	176
8.3.2 Esempio di apprendimento non supervisionato	177
8.4 Caso di studio: NLP e strategie di trading basate sull'analisi del sentimento	178
8.4.1 Costruire una strategia di trading basata sull'analisi del sentimento	179
8.4.2 Conclusioni del caso di studio	204
8.5 Caso di studio: Document Summarization (Sintesi del documento)	205
8.5.1 Utilizzo dell'NLP per il Document Summarization	205
8.6 Conclusione	212
9 CNNs per serie temporali finanziarie e immagini satellitari	213
9.1 Come le CNN imparano dai dati di tipo reticolare	214
9.1.1 Dalla codifica manuale all'apprendimento dei filtri dai dati	215
9.1.2 Come operano gli elementi di uno strato convolutivo	217
9.1.3 L'evoluzione delle architetture CNN - innovazioni chiave	221
9.2 CNNs per immagini satellitari	222
9.2.1 LeNet5 - La prima CNN con applicazioni industriali	223
9.2.2 AlexNet - riaccendere la ricerca sull'apprendimento profondo	227
9.2.3 Transfer Learning - Apprendimento più rapido con meno dati	230
9.2.4 Transfer Learning con VGG16	232
9.2.5 Classificare le immagini satellitari con il Transfer Learning	240
9.3 Le CNN per i dati delle serie temporali	242
9.3.1 CNN-TA - clustering di serie temporali in formato 2D	243
9.4 Conclusioni	252
V Regolamentazione e Prospettive future	255
10 Regolamentazione dell'IA in Finanza	257
10.1 Regtech	257

10.1.1 Regtech in numeri	259
10.2 Suptech	262
10.3 Rischio Sistemico dovuto all'IA in finanza	264
10.4 Considerazioni Ethiche	266
10.4.1 Razzismo e sessismo nell'IA	267
10.4.2 Raccomandazioni pratiche	267
10.4.3 Regolamentazione dell'IA e del ML in finanza	268
10.4.4 Etica dell'intelligenza artificiale in finanza	270
11 Competizione basata sull'intelligenza artificiale	275
11.1 Istruzione e formazione	275
11.2 Lotta per le risorse	276
11.3 Impatto sul mercato	278
11.4 Scenari competitivi	279
12 Singolarità finanziaria	283
12.1 Nozioni e definizioni	283
12.1.1 Forme di Intelligenza	284
12.2 Qual'è il rischio?	285
12.3 Percorsi verso la singolarità finanziaria	289
12.4 Scenari prima e dopo	290
12.5 Star Trek o Star Wars	291
12.6 Conclusioni	292
13 Conclusioni	293
13.1 Ottimismo consapevole	294
13.2 Citazioni risorse web	299
Bibliografia	301
Sitografia	303

Introduzione

“I programmi di trading algoritmico di oggi sono relativamente semplici e fanno un uso limitato dell’intelligenza artificiale. La situazione è destinata a cambiare.”

—Murray Shanahan (2015)

L’intelligenza artificiale (AI) è diventata un aspetto sempre più importante della vita moderna e ha il potenziale per rivoluzionare vari settori. Oggi giorno è applicata all’interno di numerose discipline scientifiche, dall’automazione industriale all’ingegneria biomedica, dal marketing alla chirurgia medica. Nel settore economico, ad esempio, i maggiori utilizzi dell’intelligenza artificiale rientrano nell’area del marketing e nell’utilizzo di social network e piattaforme online. Come detto, le possibili applicazioni sono veramente infinite e una tra queste è proprio l’oggetto di studio di questa tesi: l’utilizzo dell’intelligenza artificiale nella finanza. Per molti anni la finanza, per estrarre evidenze dai dati, ha fatto affidamento a tecniche statistiche standard e a modelli costruiti su assunzioni lontane dalla realtà. Da sempre gli investitori, quando sono chiamati a scegliere l’allocazione ottimale di risorse tra le asset classes, analizzano le news dei giornali, i reports degli analisti o gli indicatori economici, facendo prevalentemente affidamento sulle proprie intuizioni e giudizi. La finanza comportamentale ha recentemente contribuito a mettere in evidenza come queste valutazioni, distorte e guidate dalle emozioni, abbiano generato, nei modelli di asset allocation, portafogli poco realistici. Negli ultimi decenni, lo sviluppo tecnologico raggiunto dai computer insieme all’aumentata dimensione dei dataset e alle nuove conoscenze nell’analisi dei dati, hanno permesso agli investitori di integrare le strategie d’investimento con strumenti avanzati, consentendo di prendere decisioni più razionali e indirizzate dai dati.

1 AI e Finanza

Gli esperti affermano che l'intelligenza artificiale e la finanza sembrano fatte l'una per l'altra in molti modi. Le tecniche che facilitano l'identificazione di uno schema che l'occhio umano potrebbe altrimenti non rilevare sono una delle funzioni fondamentali dell'IA. Poiché la finanza sembra essere essenzialmente quantitativa, è difficile non correlarla con tali funzioni (analisi dei dati, previsione, codifica degli errori) che l'IA può facilmente svolgere. Secondo un attore del settore, "l'intelligenza artificiale è nel trading ciò che il fuoco era per gli uomini delle caverne". In altre parole, il trading di azioni basato su AI è stato un punto di svolta per gli investitori moderni.

Ma come si è arrivati a questo?

1.1 Introduzione dell'intelligenza artificiale nei mercati azionari

L'inizio dell'intelligenza artificiale nel mercato azionario è iniziato a livello teorico negli anni '60 con Robert Shlaifer. Nel 1959, Shlaifer scrisse un libro fondamentale intitolato "Probabilità e statistiche per le decisioni aziendali". Le conseguenze della sua pubblicazione hanno visto un aumento della popolarità della ricerca nel campo della statistica nel mondo degli affari. A livello pratico, gli anni '80 hanno visto la crescita delle reti neurali artificiali (ANN), che dovevano essere incorporate per fornire agli strumenti finanziari un migliore potere predittivo. Forse il primo, o almeno uno dei primi pochi programmi guidati dall'intelligenza artificiale che presumibilmente prevedeva il mercato azionario era il "sistema esperto Protrader". Il sistema esperto Protrader è stato progettato dallo studente K.C Chen della School of Business della California State University e da Ting-peng Lian dell'Università dell'Illinois. Con la loro esperienza, Chen e Lian hanno predetto con successo il famoso calo di 87 punti del Dow Jones Industrial Average del 1986 (sebbene alcuni documenti storici affermino che i loro risultati positivi potrebbero essere stati la conseguenza di un errore di overfitting). Le principali funzioni del sistema esperto Protrader includevano il monitoraggio dei premi sul mercato, la determinazione della strategia di investimento ottimale, l'esecuzione delle transazioni quando erano più vantaggiose e la modifica della base di conoscenza del sistema attraverso un meccanismo di apprendimento automatico. La maggior parte delle IA predittive delle odierni società di brokeraggio sono modellate su questo sistema nascente. Dagli anni '80, il ruolo dell'IA nella finanza si è evoluto da sperimentale a essenziale. Sebbene gli esseri umani continuino a svolgere un ruolo enorme nella trading equation, il ruolo dell'IA è cresciuto a un livello significativo. Anche le aree di investimento che sembrano essere riluttanti a

incorporare una maggiore automazione nel loro funzionamento, come gli hedge funds, operano con strumenti di analisi basati sull'intelligenza artificiale per costruire portafogli e ricevere i loro suggerimenti di investimento. In aggiunta a ciò, il ramo dell'intelligenza artificiale noto come "apprendimento automatico (ML)" sembra evolversi a un ritmo ancora più rapido, con le istituzioni finanziarie che ne stanno diventando tra le prime a adottarlo. Una volta che gli statistici di Wall Street hanno appreso che potevano applicare l'apprendimento automatico a molteplici aspetti della finanza, come le applicazioni di trading di investimenti, hanno sfruttato efficacemente il potere di elaborare milioni di dati in tempo reale. Uno dei principali modi in cui l'apprendimento automatico ha influito sul trading moderno è la sua capacità di identificare modelli di trading complessi su vasta scala in tempo reale su più mercati. Quando questo tipo di apprendimento automatico è combinato con la potenza di elaborazione dei big data ad alta velocità dell'intelligenza artificiale, il software di trading odierno può fornire ai propri clienti una chiara valutazione continua del rischio, previsioni di mercato e opzioni di investimento. L'apprendimento automatico non viene utilizzato solo per elaborare numeri. Un'azienda con sede a Chicago utilizza l'intelligenza artificiale utilizzando il riconoscimento vocale e la tecnologia di elaborazione del linguaggio naturale per ridurre il tempo necessario per vagliare dati finanziari, conversioni e note. Utilizzando la piattaforma dell'azienda, i professionisti finanziari utilizzano l'intelligenza artificiale per accedere a informazioni di mercato, note e società di tendenza in tempo reale.

Ricapitolando, l'intelligenza artificiale è un ingrediente essenziale per il trading e la finanza di oggi. Sta sostituendo attivamente gli statistici umani con le sue enormi capacità di elaborazione dei dati. Quando l'elaborazione dei big data si combina con la capacità dell'intelligenza artificiale di separare questi dati in modo "intelligente", gli investitori ottengono un sostituto guidato dalla macchina per un consulente finanziario umano.

1.2 Alcune attuali e future applicazioni dell'intelligenza artificiale in finanza

- **Trading Algoritmico**

Il trading algoritmico è l'uso di algoritmi per condurre operazioni di trading in maniera automatica. Le sue origini risalgono agli anni '70, il trading algoritmico (a volte chiamato Automated Trading System) prevede l'uso di istruzioni di trading preprogrammate per prendere decisioni

di trading estremamente rapide e obiettive. L'intelligenza artificiale è destinata a spingere il trading algoritmico a nuovi livelli. Non solo è possibile impiegare e adattare strategie più avanzate in tempo reale, ma le tecniche basate sull'apprendimento automatico possono offrire molte più strade per ottenere una speciale conoscenza e visione dei movimenti del mercato.

- **Gestione del portafoglio e robo-consulenti**

Le società di gestione patrimoniale stanno esplorando le potenziali soluzioni di intelligenza artificiale per migliorare le loro decisioni di investimento e l'utilizzo dei loro dati storici. Un esempio è l'utilizzo dei robo-consulenti, algoritmi costruiti per calibrare un portafoglio finanziario in base agli obiettivi e alla tolleranza al rischio dell'utente. Inoltre, forniscono una guida finanziaria automatizzata e servizi agli investitori finali e ai clienti. Un utente inserisce i propri obiettivi finanziari (ad esempio, andare in pensione all'età di 65 anni con 250.000 euro di risparmi), l'età, il reddito e le attività finanziarie attuali. Il robo-consulente ripartisce quindi gli investimenti tra le asset classes e strumenti finanziari per raggiungere gli obiettivi dell'utente. Il sistema calibra poi in base alle variazioni degli obiettivi dell'utente e alle variazioni in tempo reale del mercato, con l'obiettivo di trovare ciò che si adatta meglio agli obiettivi originali dell'utente. I robo-consulenti hanno guadagnato una posizione significativa tra i consumatori che non hanno bisogno di un consulente umano per sentirsi a proprio agio negli investimenti.

- **Rilevamento delle frodi**

Le frodi sono un problema enorme per le istituzioni finanziarie e uno dei motivi principali per sfruttare l'apprendimento automatico in finanza. Attualmente esiste un rischio significativo per la sicurezza dei dati a causa dell'elevata potenza di calcolo, dell'uso frequente di internet e alla crescente quantità di dati aziendali memorizzati online. Mentre i precedenti sistemi di rilevamento delle frodi finanziarie dipendevano in larga misura da complessi e robusti insiemi di regole, i moderni sistemi di rilevamento delle frodi vanno oltre il seguire una lista di controllo dei fattori di rischio: apprendono attivamente e si calibrano a nuove potenziali (o reali) minacce di sicurezza. L'apprendimento automatico è ideale per combattere le transazioni finanziarie fraudolente. Questo perché i sistemi di ML sono in grado di analizzare enormi insiemi di dati, rilevare attività insolite e segnalarle istantaneamente. Dato l'incalcolabile numero

di modi in cui la sicurezza può essere violata, i sistemi di apprendimento automatico saranno una necessità assoluta nei prossimi giorni a venire.

- **Gestione del rischio**

Le tecniche di apprendimento automatico stanno trasformando il nostro approccio alla gestione del rischio. Tutti gli aspetti della comprensione e del controllo del rischio vengono rivoluzionati grazie alla crescita di soluzioni basate sull'apprendimento automatico. Gli esempi vanno dal decidere quanto una banca dovrebbe prestare a un cliente, al migliorare la conformità e ridurre il rischio di modello.

- **Previsione del prezzo degli asset**

La previsione dei prezzi degli asset è considerata l'area più frequentemente discussa e più sofisticata in ambito finanziario. La previsione dei prezzi degli asset consente di comprendere i fattori che guidano il mercato e di speculare sulla performance degli asset. Tradizionalmente, la previsione dei prezzi degli asset veniva effettuata analizzando report finanziari passati e le performance di mercato per determinare la posizione da assumere per uno specifico titolo o asset class. Tuttavia, con l'enorme aumento della quantità di dati finanziari, gli approcci tradizionali per l'analisi e le strategie di selezione dei titoli vengono integrati con tecniche basate sull'apprendimento automatico.

- **Sentiment Analysis**

L'analisi del sentimento implica l'esame di enormi volumi di dati non strutturati, come video, trascrizioni, foto, file audio, post sui social media, articoli e documenti aziendali per determinare il sentimento del mercato. L'analisi del sentimento è fondamentale per tutte le aziende nel mondo del lavoro odierno ed è un ottimo esempio di applicazione dell'apprendimento automatico in finanza. L'uso più comune dell'analisi del sentimento nel settore finanziario è l'analisi delle notizie finanziarie, in particolare, prevedere i comportamenti e le possibili tendenze dei mercati. Il mercato azionario si muove in risposta ad una miriade di fattori umani, la speranza quindi è che l'apprendimento automatico sia in grado di replicare e migliorare l'intuizione umana sull'attività finanziaria scoprendo nuove tendenze e segnali. La maggior parte delle future applicazioni dell'apprendimento automatico saranno nella comprensione dei social media, nelle tendenze delle notizie e di altre fonti di dati relative

alla previsione del sentimento dei clienti verso gli sviluppi di mercato. Non si limiterà alla previsione dei prezzi e degli scambi azionari.

- **Riciclaggio di denaro**

Un rapporto delle Nazioni Unite stima che l'ammontare di denaro riciclato nel mondo ogni anno è pari al 2%-5% del PIL mondiale. Le tecniche di intelligenza artificiale possono analizzare i dati interni, pubblici e transazionali provenienti dalla rete di un cliente nel tentativo di individuare segnali di riciclaggio di denaro.

Queste sono solo alcune delle innumerevoli applicazioni che l'intelligenza artificiale trova nel complesso campo della finanza.

In sintesi, come con altri campi, l'intelligenza artificiale cambierà la finanza e il modo in cui gli attori dei mercati finanziari operano, fondamentalmente e per sempre.

2 Obiettivo della tesi

L'obiettivo centrale del presente lavoro di ricerca è quello di mostrare come i modelli e le teorie normative finanziarie più popolari, che hanno guidato la finanza per decenni, non hanno quasi nessuna prova a loro sostegno nel mondo finanziario reale; di come grazie ai big data e agli algoritmi di intelligenza artificiale (Machine Learning e Deep Learning) non sono più necessarie teorie e modellizzazioni del comportamento umano. In aggiunta, questo elaborato ha come obiettivo quello di presentare le varie tipologie di machine learning e di intelligenza artificiale nonché alcune delle loro applicazioni più importanti nel settore finanziario. Questo per mostrare come l'intelligenza artificiale sia diventata cruciale per il funzionamento dei mercati finanziari e della finanza e di come l'intelligenza artificiale sta diventando la tecnologia preferita delle banche e altre istituzioni finanziarie per migliorare l'analisi finanziaria e per snellire i processi. Si vuole inoltre illustrare quali sono le regolamentazioni dell'intelligenza artificiale nella finanza, e fornire una prospettiva sulle conseguenze che potrebbe avere l'adozione diffusa dell'intelligenza artificiale nel contesto finanziario.

La trattazione dell'IA in finanza è suddivisa nelle seguenti parti:

- **Parte I - Trading Azionario**

Questa prima parte passa in rassegna alcuni termini e concetti utili per la comprensione delle basi del trading azionario.

- **Parte II - Finanza e Machine Learning**

La seconda parte tratta le teorie della finanza tradizionale e normativa, di come il campo finanziario sia stato trasformato dalla finanza guidata dai dati (data-driven finance) e dall'apprendimento automatico (ML). Insieme, la finanza guidata dai dati e l'apprendimento automatico, danno origine ad un approccio alla finanza model-free (priva di modelli) e incentrata sull'intelligenza artificiale.

- **Parte III - Scoprire e sfruttare le inefficienze**

L'obiettivo principale di questa parte è applicare le reti neurali per scoprire inefficienze statistiche nei mercati finanziari (dati). Questa parte, inoltre, si occupa di identificare e sfruttare le inefficienze economiche per le quali le inefficienze statistiche sono in generale un prerequisito.

- **Parte IV - Metodi di IA & ML e applicazioni in finanza**

In questa parte viene presentata una breve introduzione delle varie tipologie di machine learning, del natural language processing (NLP) e dei più diffusi algoritmi di deep learning. Dopodiché, si procede con l'approfondimento di due delle metodologie di IA più importanti da utilizzare nel settore finanziario ovvero: l'NLP e le reti neurali convolutive (CNNs).

- **Parte V - Regolamentazione e prospettive future**

La sesta parte riguarda la regolamentazione dell'IA in finanza e delle conseguenze derivanti dalla concorrenza basata sull'intelligenza artificiale nel settore finanziario. Discute anche della possibilità di una singolarità finanziaria, nel momento in cui gli agenti di intelligenza artificiale dominerebbero tutti gli aspetti della finanza così come la conosciamo.

Dopo aver trattato queste cinque fasi di approfondimento, verrà esposta nella conclusione finale un resoconto dell'intero elaborato sull'IA nel settore della finanza.

Parte I

Trading Azionario

Capitolo 1

Le basi del trading azionario

“Non litigate con il mercato, perché è come il tempo: anche se non è sempre buono, ha sempre ragione.”

—Kenneth Walden

Il trading azionario è il processo di compra e vendita di azioni di società quotate in borsa. In sostanza, è un modo per i privati e le istituzioni di investire e possedere una parte di queste società. Il trading azionario esiste da secoli e nel tempo è diventato sempre più accessibile e conveniente per le persone che vi partecipano. Oggi, con l'avvento delle piattaforme di trading online e delle app per i dispositivi mobili, chiunque abbia una connessione a internet può acquistare e vendere azioni comodamente da casa.

Ma come funziona il trading azionario?

1.1 Il mercato azionario e le sue componenti

Comprendere il mercato azionario e le sue componenti è il primo e il più importante passo da compiere nel trading azionario. Il mercato azionario è una piattaforma in cui vengono acquistate e vendute azioni delle società quotate in borsa. Il mercato azionario funge da barometro dell'economia di un Paese e riflette la salute finanziaria complessiva delle società quotate. Il mercato azionario si divide in due categorie principali: il mercato primario, dove vengono emessi e scambiati nuovi titoli, ed il mercato secondario, dove vengono acquistati e venduti titoli precedentemente emessi, già in circolazione. Per comprendere appieno il mercato azionario, è fondamentale conoscerne le componenti principali:

- **Le azioni:** Un'azione rappresenta una quota di proprietà di una società. Quando si acquista un'azione, si possiede una piccola parte dell'azienda e si ha diritto a una quota dei suoi profitti e *assets*¹.
- **Borse:** Una borsa di azioni è un mercato in cui vengono scambiate le azioni. Le due borse più importanti del mondo sono il New York Stock Exchange (NYSE) e il NASDAQ.
- **Indici:** Un indice è una misura statistica delle variazioni di un portafoglio (insieme) di azioni, che riflette le performance del mercato azionario nel suo complesso. Gli indici più noti sono lo S&P 500 (Standard & Poor's 500), il Dow Jones Industrial Average ed il NASDAQ Composite.
- **Broker:** Un broker è una persona o una società che acquista e vende azioni per conto dei clienti. Essi agiscono come intermediari tra acquirenti e venditori, eseguendo le transazioni e offrendo consulenza ai clienti sui titoli migliori da acquistare o vendere.
- **Volume di scambi:** Il volume degli scambi si riferisce al numero di azioni scambiate in un determinato periodo di tempo. Si tratta di un indicatore chiave del livello di interesse per un particolare titolo o per l'intero mercato.

Comprendendo queste componenti, otteniamo una solida base per poter esplorare meglio il mondo del trading azionario.

1.2 Tipologie di azioni e le loro caratteristiche

Nel mondo della finanza, le azioni sono un'opzione di investimento popolare che offre l'opportunità di possedere un pezzo di una società e di beneficiare della sua crescita. Esistono vari tipi di azioni con caratteristiche diverse ed è importante che gli investitori comprendano queste differenze per prendere decisioni di investimento informate. Le principali tipologie di azioni sono:

- **Azioni ordinarie (comuni):** Sono il tipo di azione più comune e rappresentano la proprietà di una società. I possessori di azioni ordinarie hanno diritto di voto sulle questioni societarie e possono ricevere dividendi, che sono pagamenti effettuati agli azionisti. Gli azionisti di azioni ordinarie sono anche gli ultimi in ordine di tempo a ricevere un risarcimento nel caso in cui una società fallisca.

¹Nel trading finanziario, il termine asset si riferisce a tutto quello che viene scambiato sul mercato finanziario, come azioni, bond, valute o materie prime.

- **Azioni privilegiate:** Sono un tipo di azioni che generalmente offrono un dividendo fisso e hanno la precedenza sulle azioni ordinarie in caso di fallimento. Gli azionisti privilegiati non hanno diritto di voto e non hanno diritto allo stesso potenziale di crescita degli azionisti ordinari.
- **Azioni di crescita:** Sono quelle di società per le quali si prevede una crescita più rapida rispetto alla società media. Queste azioni di solito non pagano dividendi, ma reinvestono gli utili nell'espansione dell'attività. I titoli di crescita sono spesso associati a un rischio elevato, in quanto sono vulnerabili alle fluttuazioni del mercato, ma offrono anche un potenziale di rendimento elevato.
- **I titoli Value:** Sono quelli di società sottovalutate dal mercato e quindi considerate buone opportunità di investimento. Questi titoli hanno in genere un rapporto prezzo/guadagno più basso e possono pagare dividendi, che possono costituire una fonte di reddito stabile per gli investitori.
- **I titoli Blue Chip:** Sono quelli di società consolidate, finanziariamente stabili e con una lunga storia di successo. Queste società sono considerate investimenti a basso rischio e in genere pagano dividendi, il che le rende popolari tra gli investitori conservatori.
- **Penny Stocks:** sono titoli con un prezzo per azione estremamente basso. Si tratta di titoli che vengono scambiati a valori inferiori ai 5 dollari per ogni singola azione. Gli investitori che comprano queste azioni generalmente scommettono su una società di piccole dimensioni che crescerà nel tempo, con il grosso elemento di rischio che questo azzardo può comportare, ma anche con la possibilità di rendimenti stratosferici.

In conclusione, la comprensione dei diversi tipi di azioni e delle loro caratteristiche è un passo importante per chi vuole investire nel mercato azionario. Considerando i potenziali rischi e benefici di ogni tipo di azione, gli investitori possono prendere decisioni informate e beneficiare del potenziale di crescita delle azioni.

1.3 Strategie di trading per principianti

Le strategie di trading giocano un ruolo fondamentale per il successo di un qualsiasi trader azionario. Come principiante, è essenziale sviluppare una strategia di trading ben studiata per massimizzare i rendimenti e minimizzare i rischi. Sebbene non esista una strategia valida per tutti, esistono diverse

strategie che si sono dimostrate efficaci per i principianti. Ecco alcune strategie di trading che i principianti possono prendere in considerazione:

- **Investimento a lungo termine:** Questa strategia prevede l'acquisto di azioni con l'intenzione di mantenerle per un lungo periodo, in genere diversi anni. L'obiettivo è quello di superare le fluttuazioni del mercato a breve termine e beneficiare del potenziale di crescita a lungo termine dell'azienda. Questa strategia richiede pazienza, una buona conoscenza dell'azienda e un portafoglio ben diversificato
- **Mediazione del costo del dollaro:** Questa strategia prevede l'investimento di un importo fisso in un titolo a intervalli regolari, indipendentemente dal prezzo del titolo. Nel tempo, questa strategia può portare a un prezzo medio di acquisto più basso, riducendo il rischio complessivo dell'investimento.
- **Value Investing:** Questa strategia prevede l'acquisto di titoli sottovalutati sul mercato e l'attesa che il mercato riconosca il loro reale valore. Questa strategia richiede un'ampia ricerca e una forte comprensione dei dati finanziari della società.
- **Investimenti per la crescita:** Questa strategia prevede l'investimento in società che hanno un elevato potenziale di crescita, indipendentemente dalla loro valutazione attuale. L'obiettivo è quello di trarre vantaggio dal potenziale di crescita dell'azienda, piuttosto che dai suoi guadagni attuali.
- **Momentum Investing:** Questa strategia consiste nell'investire in titoli che stanno registrando una buona performance e nel cavalcare il momentum dei loro aumenti di prezzo. Questa strategia richiede un attento monitoraggio delle tendenze di mercato e un tempo di risposta rapido ai cambiamenti delle condizioni di mercato.

Con l'aiuto dell'IA i principianti possono semplificare la loro ricerca, analizzare le tendenze del mercato e prendere decisioni informate in base alle loro strategie di trading. Tuttavia, è bene ricordare che nessuna strategia di trading è infallibile, ed è essenziale rivalutare e regolare costantemente la propria strategia in base alle condizioni di mercato e agli obiettivi finanziari personali.

1.4 Gestione del rischio nel trading azionario

La gestione del rischio è una componente fondamentale del trading azionario, in quanto aiuta i trader a minimizzare le potenziali perdite e a massimizzare

i rendimenti. Come vedremo nei seguenti capitoli, vi sono tecniche di IA con le quali i trader possono comprendere e gestire meglio il rischio. Il primo passo per la gestione del rischio è **identificare** le varie tipologie di rischio coinvolti in un determinato investimento. Tra questi ci sono i rischi di mercato, come le variazioni delle condizioni di mercato, e i rischi specifici, come la situazione finanziaria di una società. Uno dei migliori modi per gestire il rischio nel trading azionario è quello di **diversificare** il proprio portafoglio investendo in una serie di titoli di diversi settori e mercati. Questo aiuta a ridurre l'impatto della performance di un particolare titolo sul portafoglio complessivo. Un altro aspetto importante della gestione del rischio è la **definizione dei limiti di rischio** che si è disposti ad assumere. A tal fine si possono impostare ordini di stop-loss (ferma la perdita), i quali vendono automaticamente le vostre azioni se il prezzo del titolo scende al di sotto di una certa soglia. La gestione del rischio consiste anche nel **monitorare** regolarmente il portafoglio, in quanto vi aiuta a rimanere informati sulle tendenze del mercato e sui cambiamenti nella performance dei vostri investimenti.

In conclusione, la gestione del rischio è una parte essenziale del trading azionario e gli strumenti di IA possono aiutare i trader a gestire in maniera più efficace il rischio, portandogli a prendere decisioni di investimento di maggior successo.

1.5 Analisi tecnica e analisi fondamentale

L'analisi tecnica e l'analisi fondamentale sono due strumenti molto importanti a disposizione dei trader, in quanto forniscono preziose indicazioni sull'andamento dei singoli titoli e del mercato nel suo complesso. Entrambi gli strumenti hanno dei punti di forza e dei punti deboli, l'approccio migliore consiste nell'utilizzare una combinazione di entrambi gli strumenti.

L'analisi tecnica consiste nell'analizzare i dati di mercato passati, come il prezzo ed il volume delle azioni, per prevedere le future tendenze del mercato e identificare le opportunità di acquisto e di vendita di azioni. Questo approccio si basa sull'idea che le tendenze di mercato, una volta stabilite, tendono a persistere e possono essere utilizzate per fare previsioni sul futuro comportamento del mercato.

L'analisi fondamentale, invece, prevede l'analisi della salute finanziaria di una azienda, compresi gli utili, i ricavi, gli attivi e i passivi, per determinarne il valore ed il potenziale di crescita. L'analisi fondamentale su una azienda si basa sull'idea che i risultati finanziari di una società si rifletta sul prezzo delle sue azioni e possa essere utilizzato per prendere decisioni di investimento.

Sia l'analisi tecnica che l'analisi fondamentale possono fornire indicazioni preziose sul mercato azionario, ma devono essere utilizzate congiuntamente per prendere decisioni di investimento consapevoli.

L'importanza dell'analisi tecnica:

- Offre una rappresentazione visiva delle tendenze di mercato
- Può fornire indicazioni sul sentimento del mercato
- Aiuta a identificare le potenziali opportunità di mercato

L'importanza dell'analisi fondamentale:

- Fornisce una visione completa della salute finanziaria di un'azienda
- Aiuta a identificare i titoli sottovalutati e sopravvalutati
- Può fornire indicazioni sul potenziale di crescita di un'azienda

Combinando l'analisi tecnica e l'analisi fondamentale i trader possono prendere decisioni informate su quando acquistare o vendere un titolo.

Gli strumenti e tecniche basate sull'IA, anche in questo processo, possono assistere i trader fornendo dati di mercato in tempo reale, analizzando le tendenze del mercato e offrendo spunti di investimento.

1.6 Indicatori di mercato e dati economici

In qualità di trader azionario, è essenziale avere una comprensione approfondita degli indicatori di mercato e dei dati economici che influenzano i prezzi delle azioni. Grazie a una buona informazione è possibile prendere decisioni di investimento consapevoli e raggiungere i propri obiettivi finanziari. A seguire analizzeremo i vari indicatori di mercato e i dati economici che sono necessari da conoscere come trader azionario.

Gli **indicatori di mercato** sono strumenti utilizzati per misurare lo stato di salute dell'economia e del mercato azionario. Questi indicatori aiutano i trader a capire l'andamento attuale del mercato e a fare previsioni sui movimenti futuri. Alcuni indicatori di mercato più comunemente utilizzati sono:

- Dow Jones Industrial Average (DJIA). Il DJIA è un indice del mercato azionario che tiene conto della performance di 30 grandi società quotate al NYSE. È un barometro del mercato azionario ed è considerato uno degli indicatori di mercato più seguiti.

- S&P 500. Lo S&P 500 è un indice del mercato azionario che tiene conto dell'andamento di 500 grandi società quotate al NYSE. È un indicatore su base ampia ed è considerato una delle migliori misure sulla salute generale del mercato.
- NASDAQ Composite. Il NASDAQ Composite è un indice del mercato azionario che tiene conto della performance di tutte le società quotate al NASDAQ. È un buon indicatore della performance delle aziende tecnologiche ed è molto seguito dagli investitori del settore.

I **dati economici**, invece, si riferiscono alle informazioni che forniscono indicazioni sullo stato di salute dell'economia. Questi dati comprendono statistiche su disoccupazione, inflazione, spesa dei consumatori e prodotto interno lordo (PIL). Analizzando i dati economici, i trader possono fare previsioni sui futuri movimenti del mercato. Alcuni dei dati economici più importanti da monitorare sono:

- Prodotto Interno Lordo (PIL). Il PIL è il valore di tutti i beni e servizi prodotti in un Paese. Un PIL in crescita indica che l'economia del Paese è in aumento, mentre un PIL in calo indica che l'economia è in contrazione.
- Indice dei Prezzi al Consumo (CPI). Il CPI misura il prezzo medio di un cestino di beni e servizi acquistati dai consumatori. È un buon indicatore dell'inflazione e aiuta i trader a capire come cambiano i prezzi di beni e servizi nel corso del tempo.
- Tasso di Disoccupazione. Il tasso di disoccupazione è la percentuale della forza lavoro che non ha un lavoro. È un buon indicatore della salute del mercato del lavoro ed è osservato attentamente dai trader. Un tasso di disoccupazione basso indica che l'economia è forte, mentre un tasso di disoccupazione elevato indica che l'economia è debole.

Tenendo d'occhio questi indicatori e dati, i trader possono prendere migliori decisioni di investimento.

1.7 Leggi e regolamenti

Il trading azionario è un settore altamente regolamentato, con numerose leggi e regolamenti volti a proteggere gli investitori e a mantenere la stabilità del mercato. È essenziale avere una buona comprensione di queste leggi e regolamenti per guidare le decisioni di investimento. Ecco alcune delle principali leggi e normative che regolano il trading azionario:

- La legge sui titoli (Securities Act) del 1993 richiede alle società di registrare i loro titoli con la Securities and Exchange Commission (SEC) e di fornire informazioni dettagliate sulla società, sui suoi risultati finanziari e sui rischi associati all'investimento. Questa legge contribuisce a garantire che gli investitori abbiano accesso a informazioni accurate e complete sulle società in cui investono.
- La legge sugli scambi di titoli (Securities Exchange Act) del 1934 ha istituito la SEC come ente regolamentatore principale dell'industria dei titoli e delle borse. Questa legge richiede alle società pubbliche di presentare rapporti e informazioni periodiche alla SEC, oltre a stabilire regole sull'insider trading e altre norme in materia di abusi di mercato.
- La legge Sarbanes-Oxley del 2002 è stata promulgata in risposta agli scandali contabili societari dei primi anni 2000. Questa legge ha stabilito nuovi requisiti per l'informativa finanziaria e la revisione contabile, oltre a inasprire le sanzioni per le frodi sui titoli e altri tipi di abuso di mercato.
- La legge Dodd-Frank sulla riforma di Wall Street e la protezione dei consumatori del 2010. Detta legge è stata emanata in risposta alla crisi finanziaria del 2008. Tale legge ha stabilito nuove norme per l'industria finanziaria, tra cui nuove regole per la negoziazione dei derivati e una maggiore protezione dei consumatori.
- Le regole FINRA (Financial Industry Regulatory Authority). La FINRA è un'organizzazione privata, senza scopo di lucro, che vigila, supervisiona il settore dei titoli. La FINRA dispone di una serie di norme e regolamenti che regolano la condotta delle società affiliate e dei broker. Queste regole includono norme sull'integrità del mercato, la tutela dei clienti e la responsabilità finanziaria.

Oltre alle leggi e ai regolamenti federali, ogni Stato ha una propria serie di norme che disciplinano il commercio di titoli. È importante conoscere le varie normative per assicurarsi di essere in regola e per evitare potenziali problemi legali.

Dopo questa breve panoramica sul mondo del trading azionario abbiamo acquisito una comprensione a livello base delle principali nozioni nel campo finanziario. Tuttavia, anche se basilari, tali nozioni potranno esserci utili per capire gli argomenti che verranno affrontati nei capitoli successivi.

Passiamo ora alla prossima parte della tesi che ha come argomenti centrali la teoria finanziaria tradizionale e normativa, la finanza basata sui dati ed il machine learning nella finanza.

Parte II

Finanza e Machine Learning

Capitolo 2

La Finanza Normativa

“Il CAPM (Capital Asset Pricing Model) si basa su molte ipotesi non realistiche. Ad esempio, l’ipotesi che gli investitori si preoccupino solo della media e della varianza dei rendimenti di un periodo del portafoglio è estrema.”

—Eugene Fama and Kenneth French (2004)

Questo capitolo presenta importanti e popolari teorie e modelli finanziari che sono stati considerati pietre miliari della finanza per decenni. Vengono trattati, tra gli altri, la teoria della mean-variance portfolio (MVP) e il modello del capital asset pricing (CAPM). Generazioni di economisti, analisti finanziari, gestori patrimoniali, trader, banchieri, contabili e altri ancora si sono formati su queste teorie. In questo senso, è lecito affermare che la finanza come disciplina teorica e pratica è stata in gran parte plasmata da queste teorie.

2.1 Incertezza e Rischio

La teoria finanziaria si occupa di investimenti, trading e valutazioni in presenza di incertezza e rischio. Questa sezione introduce, a livello un po' formale, le nozioni centrali relative a questi argomenti. L'attenzione si concentra sui concetti fondamentali della teoria della probabilità che costituisce l'ossatura della finanza quantitativa.

2.1.1 Definizioni

Spazio di probabilità

Nella teoria della probabilità, uno spazio di probabilità o una tripla di probabilità (Ω, F, P) è un costrutto matematico che fornisce un modello formale

di un processo casuale o “esperimento”. Ad esempio, si può definire uno spazio di probabilità che modella il lancio di un dado. Uno spazio di probabilità è composto da tre elementi:

1. Uno spazio campione, Ω , che è l’insieme di tutti i possibili risultati.
2. Uno spazio degli eventi, F , che è un insieme di eventi, un evento è un insieme di risultati nello spazio campione.
3. Una funzione di probabilità, P , che assegna a ciascun evento nello spazio degli eventi una probabilità, che è un numero compreso tra 0 e 1.

Nell’esempio del lancio di un dado, consideriamo lo spazio campionario come $\{1, 2, 3, 4, 5, 6\}$. Per lo spazio degli eventi, potremmo semplicemente usare l’insieme di tutti i sottoinsiemi dello spazio campionario, che conterebbe eventi semplici come $\{5\}$ (il dado cade sul numero 5), ma anche eventi complessi come $\{2, 4, 6\}$ (il dado cade su un numero pari). Infine, per la funzione di probabilità, mappereemo ogni evento al numero di esiti di quell’evento diviso per 6 - quindi, per esempio 5 sarebbe mappato a $1/6$, e $\{2, 4, 6\}$ verrebbe mappato a $3/6 = 1/2$. Quando viene condotto un esperimento, immaginiamo che la “natura” “selezioni” un singolo risultato, ω , dallo spazio campionario Ω . Tutti gli eventi nello spazio degli eventi F che contengono il risultato selezionato ω vengono chiamati “eventi verificati”. Questa “selezione” avviene in modo tale che, se l’esperimento fosse ripetuto molte volte, il numero di occorrenze di ciascun evento, come frazione del numero totale di esperimenti, tenderebbe molto probabilmente alla probabilità assegnata a quell’evento dalla funzione di probabilità P . In breve, uno spazio di probabilità è uno spazio di misura tale che la misura dell’intero spazio è uguale a uno.

La definizione estesa è la seguente: uno spazio di probabilità è una tripla (Ω, F, P) costituita da:

- Lo spazio campionario Ω , un insieme arbitrario non vuoto
- La σ -algebra $F \subseteq 2^\Omega$ – un insieme di sottoinsiemi di Ω , detti eventi, tale che:
 - F contiene lo spazio campionario: $\Omega \in F$
 - F è chiuso rispetto ai complementi: se $A \in F$ allora anche $\Omega/A \in F$
 - F è chiuso rispetto a unioni numerabili:
se $A_i \in F$ per $i = 1, 2, \dots$, allora anche $(\cup_{i=1}^{\infty} A_i) \in F$

- Il corollario delle due proprietà precedenti e la legge di De Morgan è che F è chiuso anche rispetto alle intersezioni numerabili:

se $A_i \in F$ per $i = 1, 2, \dots$, allora anche $(\cap_{i=1}^{\infty} A_i) \in F$

- La misura di probabilità $P : F \rightarrow [0, 1]$ - una funzione su F tale che:

- * P è numerabile additivo: se $A_{i=1}^{\infty} \subseteq F$ è una raccolta numerabile di insiemi disgiunti allora $P(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$
- * La misura dell'intero spazio campionario è uguale a uno: $P(\Omega) = 1$

La teoria della probabilità discreta ha bisogno solo di spazi campionari numerabili Ω . Le probabilità possono essere attribuite a punti di Ω dalla funzione di probabilità di massa $p : \Omega \rightarrow [0, 1]$ tale che $\sum_{\omega \in \Omega} p(\omega) = 1$. Tutti i sottoinsiemi di Ω possono essere trattati come eventi (quindi, $F = 2^{\Omega}$ è l'insieme potenza). La misura di probabilità assume la forma semplice:

$$P(A) = \sum_{\omega \in \Omega} p(\omega) \text{ per ogni } A \subseteq \Omega$$

La più grande σ -algebra $F = 2^{\Omega}$ descrive l'informazione completa. In generale, una σ -algebra $F \subseteq 2^{\Omega}$ corrisponde a una partizione finita o numerabile $\Omega = B_1 \cup B_2 \cup \dots$, la forma generale di un evento $A \in F$.

Esempio: Se l'esperimento consiste in un solo lancio di una moneta, il risultato è testa o croce: $\Omega = H, T$. La σ -algebra $F = 2^{\Omega}$ contiene $2^2 = 4$ eventi: $\{H\}$ (“testa”), $\{T\}$ (“croce”), $\{\}$ (“né testa né croce”) e $\{H, T\}$ (“sia testa che croce”); in altre parole, $F = \{\{\}, \{H\}, \{T\}, \{H, T\}\}$. C’è una probabilità del 50% che esca testa e del 50% che esca croce; quindi, la misura delle probabilità in questo esempio è:

$$P(\{\}) = 0, P(\{H\}) = 0.5, P(\{T\}) = 0.5, P(\{H, T\}) = 1$$

Dopo questa breve digressione sullo spazio di probabilità, torniamo a parlare di incertezza e rischio, e a dare ulteriori definizioni utilizzando le nozioni sullo spazio delle probabilità.

Assumiamo un’economia per la quale l’attività è osservata solo in due momenti: oggi, $t=0$, e un anno dopo, $t=1$. Le teorie finanziarie che discuteremo più avanti si basano in gran parte su un’economia così *statica*¹.

¹In un’economia dinamica, l’incertezza si risolverebbe gradualmente nel tempo, ad esempio, in ogni giorno tra oggi e un anno dopo.

A $t=0$, non c'è alcuna incertezza. A $t=1$, l'economia può assumere un numero finito S di possibili stati o risultati $\omega \in \Omega = \{\omega_1, \omega_2, \dots, \omega_S\}$. Ω è lo spazio campionario e ha cardinalità pari a S , $|\Omega| = S$.

L'insieme potenza $F = 2^\Omega$ è la σ -algebra più grande, mentre l'insieme $F = \{\emptyset, \Omega\}$ è la σ -algebra più piccola di Ω .

Una σ -algebra è un modello per gli eventi osservabili in un'economia. In questo contesto, un singolo stato dell'economia $\omega \in \Omega$ può essere interpretato come un evento atomico. Uno stato $\omega \in \Omega$ può essere parte di un evento appartenente allo spazio degli eventi, $E \in F$.

Come è stato detto prima, quando viene condotto un esperimento, immaginiamo che la "natura" "selezioni" un singolo stato, ω , dallo spazio campionario Ω . Tutti gli eventi nello spazio degli eventi F che contengono il risultato selezionato ω vengono chiamati "eventi verificati". Dunque, una funzione di probabilità assegna un valore reale $0 \leq p_\omega \equiv P(\{\omega\}) \leq 1$ allo stato $\omega \in \Omega$, o un numero reale $0 \leq P(E) \leq 1$ all'evento $E \in F$.

Se le probabilità di tutti gli stati sono note, vale che $P(E) = \sum_{\omega \in E} p_\omega$.

Si veda l'esempio dell'evento "cade su un numero pari" nell'esperimento del dado regolare menzionato sopra. In tal caso, le probabilità dei singoli stati appartenenti all'evento "cade su un numero pari" sono $\{2\}$, $\{4\}$ e $\{6\}$ e ciascuno ha probabilità di $1/6$, mentre che l'evento $\{2, 4, 6\}$ ha probabilità $1/2$ che la somma delle probabilità dei singoli stati. Uno spazio di probabilità (Ω, F, P) è la rappresentazione formale dell'incertezza nel modello economico. Se la misura di probabilità P è fissata, si dice che l'economia è a rischio. Se è nota a tutti gli agenti dell'economia si dice che l'economia ha un'informazione simmetrica. Dato uno spazio di probabilità (Ω, F, P) , una variabile casuale è una funzione:

$$S : \Omega \rightarrow R_+, \omega \mapsto S(\omega), \text{ che è } F - \text{misurabile}$$

Questo implica che per ogni $E \in \{[a, b] : a, b \in R, a < b\}$ si ha quanto segue:

$$S^{-1}(E) \equiv \{\omega \in \Omega : S(\omega) \in E\} \in F$$

Se $F = 2^\Omega$ l'attesa (expectation) di una variabile casuale è definita come segue:

$$E^P(S) = \sum_{\omega \in \Omega} P(\omega) \cdot S(\omega)$$

Oppure è definito da:

$$E^P(S) = \sum_{E \in F} P(E) \cdot S(E)$$

In generale, si assume che un'economia finanziaria sia perfetta.

Questo significa che non ci sono costi di transazione, gli assets disponibili hanno prezzi fissi e sono disponibili in quantità infinite, tutto avviene alla velocità della luce, e gli agenti hanno una completa informazione simmetrica.

2.1.2 Esempio Numerico

Si assuma una semplice economia statica sotto rischio (Ω, F, P) per la quale vale quanto segue:

- $\Omega \equiv \{u, d\}$
- $F = 2^\Omega$
- $P \equiv \{P(\{u\}) = 1/2, P(\{d\}) = 1/2\}$

Traded Assets

Nell'economia vengono scambiati due assets. Il primo è un'asset rischioso, titolo o stock, con un prezzo certo a oggi di $S_0 = 10$ e un payoff (guadagno) incerto di domani sotto forma di variabile casuale:

$$S_1 = \begin{cases} S_1^u = 20 & \text{se } \omega = u \\ S_1^d = 5 & \text{se } \omega = d \end{cases}$$

Il secondo è un asset senza rischio, l'obbligazione (bond), con un determinato prezzo a oggi di $B_0 = 10$ e un determinato payoff di domani pari a:

$$B_1 = \begin{cases} B_1^u = 11 & \text{se } \omega = u \\ B_1^d = 11 & \text{se } \omega = d \end{cases}$$

Formalmente, il modello economico può essere scritto come $M^2 = (\{\Omega, F, P\}, A)$ dove A rappresenta gli assets commerciali sotto forma di vettore di prezzi $M_0 = (S_0, B_0)^T$ oggi, e la matrice dei payoff (rendimenti) di mercato di domani è la seguente:

$$M_1 = \begin{pmatrix} S_1^u & B_1^u \\ S_1^d & B_1^d \end{pmatrix}$$

2.2 Teoria dell'utilità attesa (Expected Utility Theory EUT)

La teoria dell'utilità attesa (EUT) è una pietra miliare della teoria finanziaria. Sin dalla sua formulazione negli anni '40, è stata uno dei paradigmi centrali per modellare il processo decisionale in condizioni di incertezza. Praticamente ogni testo introduttivo sulla teoria finanziaria e sulla teoria degli investimenti fornisce un resoconto della EUT.

L'EUT è una teoria assiomatica, che risale al lavoro seminale di von Neumann e Morgenstern (1944). Axiomatica significa che i principali risultati della teoria possono essere dedotti solo da un piccolo numero di assiomi.

2.2.1 Assiomi e teoria normativa

Su Wolfram MathWorld si trova la seguente definizione di assioma:

“Un assioma è una proposizione considerata auto-evidentemente vera senza prove.”

L'EUT si basa generalmente su un piccolo insieme di assiomi principali che riguardano le preferenze di un agente quando si trova a dover scegliere sotto incertezza.

Anche se la definizione di assioma suggerisce diversamente, non tutti gli assiomi sono considerati “auto-evidentemente veri senza prove” da tutti gli economisti. Von Neumann e Morgenstern (1944, p.25) commentano così la scelta degli assiomi:

Una scelta di assiomi non è un compito puramente oggettivo. Di solito ci si aspetta che raggiunga uno scopo definito — qualche specifico teorema o teoremi devono essere derivabili dagli assiomi — e in questa misura il problema è esatto e oggettivo.

Ma oltre a questo ci sono sempre altri importanti desiderata di natura meno esatta: gli assiomi non dovrebbero essere troppo numerosi, il loro sistema deve essere il più semplice e trasparente possibile, e ogni assioma dovrebbe avere un significato intuitivo immediato in base al quale la sua adeguatezza possa essere giudicata direttamente.

In tal senso, un insieme di assiomi costituisce una teoria normativa del (delle parti del) mondo che deve essere modellato dalla teoria. L'insieme degli assiomi raccoglie l'insieme minimo di ipotesi o assunti, che dovrebbero essere soddisfatte a priori e non attraverso qualche dimostrazione formale o simili. Prima di elencare l'insieme di assiomi che portano alla EUT ecco alcune pa-

role sulle preferenze di un agente quando si trova di fronte ad una scelta in condizioni di incertezza.

2.2.2 Preferenze di un agente

Un singolo agente si trova di fronte a opzioni chiamate lotterie. Dati alcuni risultati che si escludono a vicenda, una lotteria è uno scenario in cui ogni risultato si verificherà con una data probabilità, sommando tutte le probabilità dà uno. Ad esempio, per due risultati A e B, $L = 0.25A + 0.75B$ Denota uno scenario in cui $P(A) = 25\%$ è la probabilità che si verifichi A e $P(B) = 75\%$ (si verificherà esattamente uno di essi) è la probabilità che si verifichi B. Più in generale, per una lotteria con molti possibili esiti A_i , scriviamo:

$L = \sum p_i A_i$ con la somma delle probabilità p_i uguale a 1. I risultati di una lotteria possono essere essi stessi lotterie tra altri risultati e l'espressione espansa è considerata una lotteria equivalente:

$$0.5(0.5A + 0.5B) + 0.5C = 0.25A + 0.25B + 0.5C$$

Se la lotteria M è preferita alla lotteria L, scriviamo $M \succ L$, o equivalentemente, $L \prec M$. Se l'agente è indifferente tra L e M, scriviamo la relazione di indifferenza $L \sim M$. Se M è preferito o visto con indifferenza rispetto a L, scriviamo $L \preceq M$. Date queste descrizioni, un possibile insieme di assiomi che portano all'EUT è il seguente:

- Completezza

L'agente può classificare tutte le lotterie l'una rispetto all'altra. Almeno una delle seguenti deve valere: $M \succ L, L \prec M \text{ o } M \sim L$.

- Transitività

Se esiste una terza lotteria N allora se $L \succ M$ e $M \succ N$ segue che $L \succ N$.

- Continuità

Se $L \succ M \succ N$, allora esiste una probabilità $p \in [0, 1]$ tale che: $pL + (1 - p)N \sim M$.

- Indipendenza

Da $L \sim M$ segue che $pL + (1 - p)N \sim pM + (1 - p)N$. Analogamente, da $L \succ M$ segue che $pM + (1 - p)N \succ pL + (1 - p)N$.

- Dominanza

Se $N_1 = p_1L + (1 - p_1)N$ e $C_2 = p_2 + (1 - p_2)N$, da $L \succ N$ e $N_1 \succ N_2$ segue che $p_1 > p_2$.

2.2.3 Utility Functions

Una funzione di utilità è un modo per rappresentare le preferenze di un agente in modo matematico e numerico, in quanto una funzione di utilità assegna un valore numerico ad un certo risultato A. Esiste una funzione U che assegna a ciascun risultato A un numero reale U(A) tale che per due lotterie qualsiasi, $L \prec M$ se e solo se $U(L) < U(M)$ dove $U(L)$ è dato da $U(p_1A_1 + \dots + p_nA_n) = p_1U(A_1) + \dots + p_nU(A_n)$. Se la funzione U rappresenta le preferenze di un agente, allora sono vere le seguenti relazioni:

- $L \succ M \rightarrow U(L) > U(M)$ (preferenza forte)
- $L \succeq M \rightarrow U(L) \geq U(M)$ (preferenza debole)
- $L \prec M \rightarrow U(L) < U(M)$ (non preferenza forte)
- $L \preceq M \rightarrow U(L) \leq U(M)$ (non preferenza debole)
- $L \sim M \rightarrow U(L) = U(M)$ (indifferenza)

Una funzione di utilità U è determinata solo fino a una trasformazione lineare positiva. Pertanto, se U rappresenta le preferenze, allora anche $V = a + bU$ con $a, b > 0$ lo fa. Riguardo alle funzioni di utilità, Von Neumann e Morgenstern (1944, p.25) riassumono come segue: “*Così vediamo: se esiste una tale valutazione numerica dell'utilità, allora è determinata fino ad una trasformazione lineare. Cioè, l'utilità è un numero fino a una trasformazione lineare.*”

2.2.4 Funzioni di utilità attesa

Von Neumann e Morgenstern (1944) dimostrano che se le preferenze di un agente soddisfano i cinque assiomi precedenti, allora esiste una funzione di utilità attesa della forma:

$$U : X \rightarrow R_+, x \mapsto E^P(u(x)) = \sum_{\omega}^{\Omega} P(\omega)u(x(\omega))$$

La funzione di utilità attesa U applica prima una funzione u al payoff $x(\omega)$ in un certo stato e poi usa la probabilità che si verifichi un dato stato $P(\omega)$ per pesare le singole utilità. L'utilità di Bernoulli $u : R \rightarrow R, x \mapsto u(x)$ è una funzione monotona crescente, indipendente dallo stato, ad esempio: $u(x) = \ln(x)$, $u(x) = x$, o $u(x) = x^2$.

Nel caso particolare di utilità lineare di Bernoulli $u(x) = x$, l'utilità attesa è semplicemente il valore atteso del payoff che dipende dallo stato, $U(x) = E^P(x)$.

2.2.5 Avversione al rischio

In finanza, il concetto di avversione al rischio è importante. La misura più comunemente utilizzata dell'avversione al rischio è la misura dell'avversione assoluta al rischio (ARA) di Arrow-Pratt che risale a Pratt (1964).

Si supponga che un agente sia Bernoulli indipendente dallo stato, e che la funzione di utilità sia $u(x)$.

Allora la misura Arrow-Pratt dell'ARA è definita come segue:

$$ARA(x) = -\frac{u''(x)}{u'(x)}, x \geq 0$$

In base a questa misura si possono distinguere i seguenti tre casi:

$$ARA(x) = -\frac{u''(x)}{u'(x)} \begin{cases} > 0 & \text{averso al rischio} \\ = 0 & \text{neutrale al rischio} \\ < 0 & \text{amante del rischio} \end{cases}$$

Nelle teorie e nei modelli finanziari, l'avversione al rischio e la neutralità del rischio in generale sono considerati come casi appropriati. Nel gioco d'azzardo, probabilmente si possono trovare anche agenti che amano il rischio.

Consideriamo le tre funzioni di Bernoulli precedentemente menzionate $u(x) = \ln(x)$, $u(x) = x$ e $u(x) = x^2$. È facilmente verificabile che esse modellano agenti avversi al rischio, neutrali al rischio e amanti del rischio rispettivamente. Si consideri ad esempio $u(x) = x^2$:

$$-\frac{u''(x)}{(u'(x))} = \frac{-2}{2x}, \text{ con } x > 0 \rightarrow \text{amante del rischio}$$

2.2.6 Esempio numerico

L'applicazione dell'EUT è facilmente illustrata in Python. Si assuma l'esempio di modello economico della sezione precedente $M^2 = (\{\Omega, F, P\}, A)$. Si supponga che un'agente con preferenze decida in base all'EUT tra diversi payoff futuri. La Bernoulli Utility dell'agente è data da $u(x) = \sqrt{x}$. Nell'esempio, il payoff A_1 risultante dal portafoglio ϕA è preferito al payoff D_1 risultante dal portafoglio ϕD .

Il seguente codice illustra tale applicazione:

```
1 import numpy as np
2
3 #prezzi del titolo e del bond oggi
4 S0 = 10
```

```
5 B0 = 10
6
7 #incertezza del payoff del titolo e del bond domani
8 S1 = np.array((20, 5))
9 B1 = np.array((11, 11))
10
11 #vettore dei prezzi
12 M0 = np.array((S0, B0))
13 M0
14 array([10, 10])
15 #matrice dei payoff
16 M1 = np.array((S1, B1)).T
17 M1
18 array([[20, 11],
19         [ 5, 11]])
20 #funzione Bernoulli Utility di avversione al rischio
21 def u(x):
22     return np.sqrt(x)
23
24 #due portafogli con pesi diversi
25 phi_A = np.array((0.75, 0.25))
26 phi_D = np.array((0.25, 0.75))
27
28 np.dot(M0, phi_A) == np.dot(M0, phi_D)
29 True
30 #payoff incerto di un portafoglio
31 A1 = np.dot(M1, phi_A)
32 A1
33 array([17.75, 6.5 ])
34 #payoff incerto dell'altro portafoglio
35 D1 = np.dot(M1, phi_D)
36 D1
37 array([13.25, 9.5 ])
38 #la misura di probabilita'
39 P = np.array((0.5, 0.5))
40
41 #funzione utilita' attesa
42 def EUT(x):
43     return np.dot(P, u(x))
44
45 #i valori di utilita' per i due payoffs incerti
46 EUT(A1), EUT(D1)
47 (3.381292321692286, 3.3611309730623735)
```

Il questo contesto, il problema è quello di ricavare un portafoglio ottimale (cioè che massimizzi l'utilità attesa) dato un budget fisso dell'agente $w \geq 0$. Il seguente codice Python modella questo problema. Dal budget disponibile, l'agente investe circa il 60% nell'asset (titolo) rischioso e circa il 40% nell'asset (bond) senza rischio. I risultati sono principalmente determinati dalla particolare forma della funzione di utilità di Bernoulli:

```

1 from scipy.optimize import minimize
2
3 #budget dell'agente
4 w = 10
5
6 #vincolo di budget da utilizzare con minimize
7 cons = {'type': 'eq', 'fun': lambda phi: np.dot(M0, phi) - w}
8
9 #la funzione di utilita' attesa definita sui portafogli
10 def EUT_(phi):
11     x = np.dot(M1, phi)
12     return EUT(x)
13
14 opt = minimize(lambda phi: -EUT_(phi),#minimizzando -EUT_(phi)
15 si massimizza EUT_(phi)
16         x0 = phi_A, #l'ipotesi iniziale per l'ottimizzazione
17         constraints = cons) #il vincolo del budget applicato
18
19 #risultati ottimali, includendo il portafoglio ottimale sotto x
20 opt
21 message: Optimization terminated successfully
22 success: True
23 status: 0
24     fun: -3.385015999493397
25     x: [ 6.112e-01 3.888e-01]
26     nit: 4
27     jac: [-1.692e+00 -1.693e+00]
28     nfev: 12
29     njev: 4
30
31 #L'ottimale (la piu' alta) utilita' attesa data dal budget di w = 10
32 EUT_(opt['x'])
33 3.385015999493397

```

2.3 Mean Variance Portfolio Theory (MVP)

La teoria della media-varianza del portafoglio (MVP), di Markowitz (1952), è un'altra pietra miliare della teoria finanziaria. Si tratta di una delle prime teorie di investimento in condizioni di incertezza che si concentravano su misure statistiche per la costruzione di portafogli di investimento azionario.

MVP astrae completamente, ad esempio, dai *fondamentali*² di un'azienda che potrebbero guidare la sua performance azionaria, oppure da ipotesi sulla competitività futura di una azienda che potrebbero essere importanti per le prospettive di crescita di una azienda.

In sostanza, gli unici dati di input che contano sono le serie temporali dei prezzi delle azioni e le statistiche da esse derivate, come il rendimento medio (storico) annualizzato e la varianza (storica) annualizzata dei rendimenti.

Attraverso i concetti presentati in detta teoria, gli investitori possono trarre guide pratiche nella costruzione di portafogli di investimento che massimizzino il loro rendimento atteso sulla base di un dato livello di rischio.

Definiamo alcuni dei termini di base che useremo nel contesto della MVP:

- **Rischio:** nel contesto della MVP, il rischio può essere definito come la varianza o la deviazione del rendimento dell'investimento rispetto al livello atteso.
- **Rendimento:** questa è la ricompensa che un investitore guadagna investendo/impegnando il proprio capitale in un determinato bene/titolo.
- **Portafoglio:** un portafoglio è una raccolta di assets come azioni, proprietà, obbligazioni, valute, ecc.
- **Insieme di opportunità:** è l'insieme di portafogli disponibili che un investitore può scegliere in base alle sue combinazioni di rischio e rendimento.
- **Diversificazione:** è il processo di miscelazione di diversi assets all'interno di un portafoglio per garantire che il rischio sia attenuato. In questo caso, la performance negativa di un determinato asset/titolo all'interno del portafoglio è bilanciata dalla performance positiva di altri asset all'interno del portafoglio.

²I fondamentali includono le informazioni qualitative e quantitative di base che contribuiscono al benessere finanziario o economico di un'azienda, di un titolo o di una valuta e la loro successiva valutazione finanziaria. I fondamentali forniscono un metodo per impostare il valore finanziario di una società, titolo o valuta.

In generale, l'MVP cerca di spiegare un metodo per la costruzione di un portafoglio che generi un rendimento massimo per un dato livello di rischio o un rischio minimo per un rendimento dichiarato. L'investitore cerca di trovare l'equilibrio ottimale tra il rendimento e il rischio connesso all'investimento.

Sebbene l'MVP abbia alcuni limiti, continua ad essere una pietra miliare per i gestori di portafoglio applicando tecniche statistiche per illustrare agli investitori i vantaggi della diversificazione.

2.3.1 Presupposti e risultati

Il presupposto centrale della MVP, secondo Markowitz (1952) è che gli investitori si preoccupino solo dei rendimenti attesi e della varianza di tali rendimenti:

Consideriamo la regola secondo cui l'investitore considera (o dovrebbe considerare) il rendimento atteso una cosa desiderabile e la varianza del rendimento un elemento indesiderabile. Questa regola ha molti punti di forza sia come massima che come ipotesi sul comportamento d'investimento. Il portafoglio con il massimo rendimento atteso non è necessariamente quello con la minima varianza. C'è un tasso al quale l'investitore può ottenere il rendimento atteso assumendo la varianza o ridurre la varianza rinunciando al rendimento atteso.

Questo approccio alle preferenze degli investitori è molto diverso dall'approccio che definisce le preferenze e la funzione di utilità di un agente sui payoffs.

MVP presuppone piuttosto che le preferenze e la funzione di utilità di un agente possano essere definite dal momento primo e secondo dei rendimenti attesi da un portafoglio di investimenti.

2.3.2 Distribuzione Normale Implicitamente Assunta

In generale, la teoria MVP, che si concentra solo sul rischio e sul rendimento del portafoglio in un certo periodo, non è compatibile con l'EUT standard. Un modo per risolvere questo problema è quello di assumere che i rendimenti degli asset rischiosi siano distribuiti normalmente (seguono una distribuzione normale), in modo tale che il momento primo e secondo (media e varianza) sono sufficienti a descrivere l'intera distribuzione dei rendimenti di un asset.

Questo è un aspetto che non si osserva quasi mai nei dati finanziari reali. L'altro modo è quello di assumere una particolare funzione di utilità quadratica di Bernoulli, come mostrato nella prossima sezione.

2.3.3 Statistiche di portafoglio

Si ipotizzi un'economia statica $M^N = (\{\Omega, F, P\}, A)$, per la quale l'insieme di assets negoziabili A è costituito da N assets rischiosi, A_1, A_2, \dots, A_N . Con A_n^0 che indica il prezzo fisso dell'n-esimo asset oggi e A_n^1 che indica il suo payoff tra un anno, il vettore dei rendimenti netti (semplici) dell'n-esimo asset è definito come segue:

$$r^n = \frac{A_1^n}{A_0^n} - 1$$

Per tutti gli stati futuri che hanno la stessa probabilità di verificarsi, il *rendimento atteso* dell'asset n è dato da:

$$u^n = \frac{1}{|\Omega|} \sum_{\omega}^{\Omega} r^n(\omega)$$

Di conseguenza il *vettore dei rendimenti attesi* è dato da:

$$u = (u^1, u^2, \dots, u^N)^T$$

Un portafoglio (vettore) $\phi = (\phi^1, \phi^2, \dots, \phi^N)^T$ con $\phi^n \geq 0$ e $\sum_{n=1}^N \phi^n = 1$, assegna i pesi a ciascun asset del portafoglio. Il *rendimento atteso del portafoglio* è quindi dato dal prodotto del vettore dei pesi del portafoglio e del vettore dei rendimenti attesi:

$$u^{phi} = \phi \cdot u$$

Definiamo ora la *covarianza* tra gli asset n e m con la seguente formula:

$$\sigma_{mn} = \sum_{\omega}^{\Omega} (r^m(\omega) - u^m)(r^n(\omega) - u^n)$$

La matrice di covarianza è quindi data da:

$$\sum = \begin{matrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{matrix}$$

La *varianza attesa del portafoglio* è a sua volta data dal doppio prodotto riga per colonna:

$$\varphi^{phi} = \phi^T \cdot \sum \cdot \phi$$

La volatilità attesa del portafoglio è quindi la seguente:

$$\sigma^{phi} = \sqrt{\phi^{phi}}$$

2.3.4 Sharpe Ratio

Sharpe (1966) introduce una misura per giudicare la performance corretta per il rischio di fondi comuni e altri portafogli, o anche di singoli asset rischiosi. Nella sua forma più semplice, mette in relazione il rendimento (atteso, realizzato) di un portafoglio con la sua volatilità (attesa, realizzata). Formalmente, lo Sharpe ratio può essere definito come segue:

$$\pi^{phi} = \frac{u^{phi}}{\sigma^{phi}}$$

Se r rappresenta il tasso a breve privo di rischio, il premio di rischio o rendimento in eccesso di un portafoglio phi rispetto a un'alternativa priva di rischio è definita da $\pi^{phi} - r$.

In un'altra versione dello Sharpe ratio, questo premio di rischio è il numeratore:

$$\pi^{phi} = \frac{u^{phi} - r}{\sigma^{phi}}$$

Se lo short rate senza rischio r è relativamente basso, le due versioni non producono risultati numerici troppo diversi se si applica lo stesso tasso a breve senza rischio (risk-less short rate). In particolare, quando si classificano diversi portafogli in base allo Sharpe ratio, le due versioni dovrebbero generare lo stesso ordine di classifica, a parità di altre condizioni.

2.3.5 Esempio Numerico

Tornando al modello statico di economia M^2 le nozioni di base dell'MVP possono essere facilmente illustrate con l'uso di Python.

Statistiche di portafoglio

In primo luogo, ecco la derivazione del *rendimento atteso del portafoglio*:

```

1 import numpy as np
2 S0 = 10
3 B0 = 10
4 S1 = np.array((20, 5))
5 B1 = np.array((11, 11))
6 M0 = np.array((S0, B0))

```

```

7 M1 = np.array((S1, B1)).T
8 P = np.array((0.5, 0.5))
9
10 #vettore del rendimento dell'asset rischioso
11 rS = S1 / S0 - 1
12 rS
13 array([ 1. , -0.5])
14 #Vettore del rendimento dell'asset senza rischio
15 rB = B1 / B0 - 1
16 rB
17 array([0.1, 0.1])
18 #funzione di rendimento atteso
19 def mu(rX):
20     return np.dot(P, rX)
21
22 #rendimenti attesi degli asset negoziati
23 mu(rS)
24 0.25
25 #rendimenti attesi degli asset negoziati
26 mu(rB)
27 0.10000000000000009
28 #matrice dei rendimenti per gli asset negoziati
29 rM = M1 / M0 - 1
30 rM
31 array([[ 1. , 0.1],
32         [-0.5, 0.1]])
33 #vettore del rendimento atteso
34 mu(rM)
35 array([0.25, 0.1 ])

```

In secondo luogo, la varianza e la volatilità, così come la matrice di covarianza:

```

1 #funzione della varianza
2 def var(rX):
3     return ((rX - mu(rX))**2).mean()
4
5 var(rS)
6 0.5625
7 var(rB)
8 0.0
9 #funzione della volatilita'
10 def sigma(rX):
11     return np.sqrt(var(rX))

```

```

12
13 sigma(rS)
14 0.75
15 sigma(rB)
16 0.0
17 #la matrice di covarianza
18 np.cov(rM.T, aweights=P, ddof=0)
19 array([[0.5625, 0. ],
20         [0. , 0. ]])

```

In terzo luogo, il rendimento atteso del portafoglio, la varianza attesa del portafoglio, e la volatilità attesa del portafoglio, illustrati per un portafoglio equamente ponderato:

```

1 phi = np.array((0.5, 0.5))
2
3 #il rendimento atteso del portafoglio
4 def mu_phi(phi):
5     return np.dot(phi, mu(rM))
6
7 mu_phi(phi)
8 0.17500000000000004
9 #la varianza attesa del portafoglio
10 def var_phi(phi):
11     cv = np.cov(rM.T, aweights=P, ddof=0)
12     return np.dot(phi, np.dot(cv, phi))
13 var_phi(phi)
14 0.140625
15
16
17 #la volatilita' attesa del portafoglio
18 def sigma_phi(phi):
19     return var_phi(phi) ** 0.5
20
21 sigma_phi(phi)
22 0.375

```

Insieme di opportunità di investimento

Sulla base di una simulazione Monte Carlo per i pesi del portafoglio ϕ , si può visualizzare l'insieme delle opportunità di investimento nello spazio volatilità-rendimento (figura 2.1).

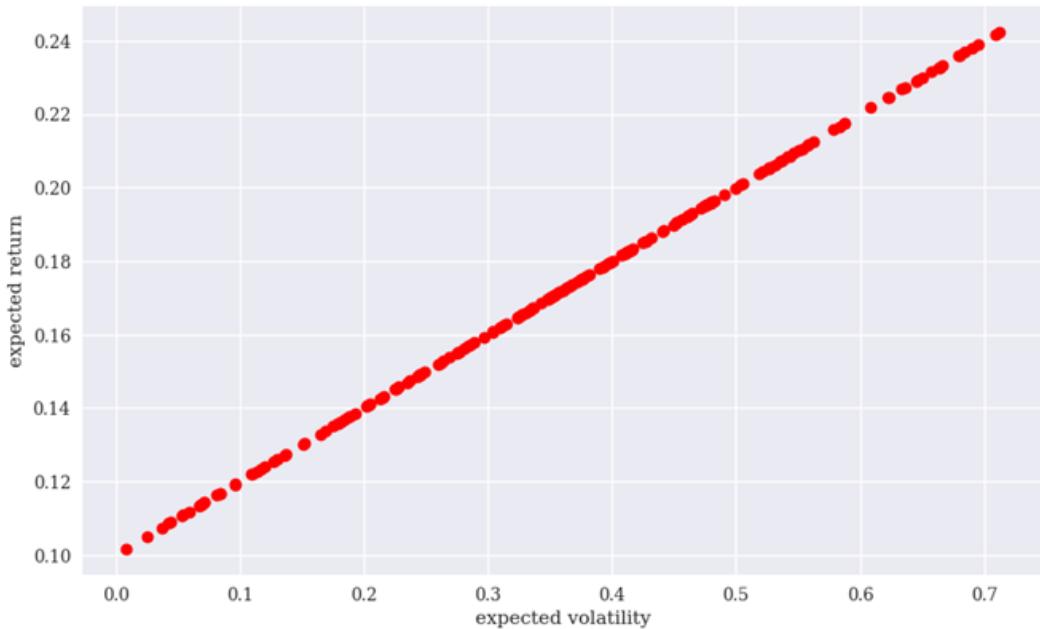


Figura 2.1: Volatilità e rendimento attesi del portafoglio simulato (asset rischioso)

Poiché ci sono solo un asset rischioso e un asset privo di rischio, l'insieme delle opportunità è una linea retta. Consideriamo ora il caso di un'economia statica a tre stati M^3 in cui $\Omega = \{u, m, d\}$. I tre stati sono ugualmente probabili, $P=\{1/3,1/3,1/3\}$. L'insieme degli asset negoziabili è composto da due assets rischiosi S e T con un prezzo fisso si $S_0 = T_0 = 10$ e payoff incerti, rispettivamente:

$$S_1 = (20, 10, 5)^T$$

$$T_1 = (1, 12, 13)^T$$

Sulla base di queste ipotesi, il codice Python che segue ripete la simulazione Monte Carlo e possiamo osservare i risultati nella figura 2.2. Con due asset rischiosi, il ben noto “proiettile” MVP diventa visibile.

¹ #nuove misure di probabilita' per i tre stati

² P = np.ones(3)/3

³ P

⁴ array([0.33333333, 0.33333333, 0.33333333])

⁵ S0 = 10

⁶ S1 = np.array((20, 10, 5))

```

7 T0 = 10
8 T1 = np.array((1, 12, 13))
9
10 M0 = np.array((S0, T0))
11 M0
12 array([10, 10])
13 M1 = np.array((S1, T1)).T
14 M1
15 array([[20, 1],
16         [10, 12],
17         [ 5, 13]])
18 rM = M1 / M0 -1
19 rM
20 array([[ 1. , -0.9],
21         [ 0. , 0.2],
22         [-0.5, 0.3]])

```

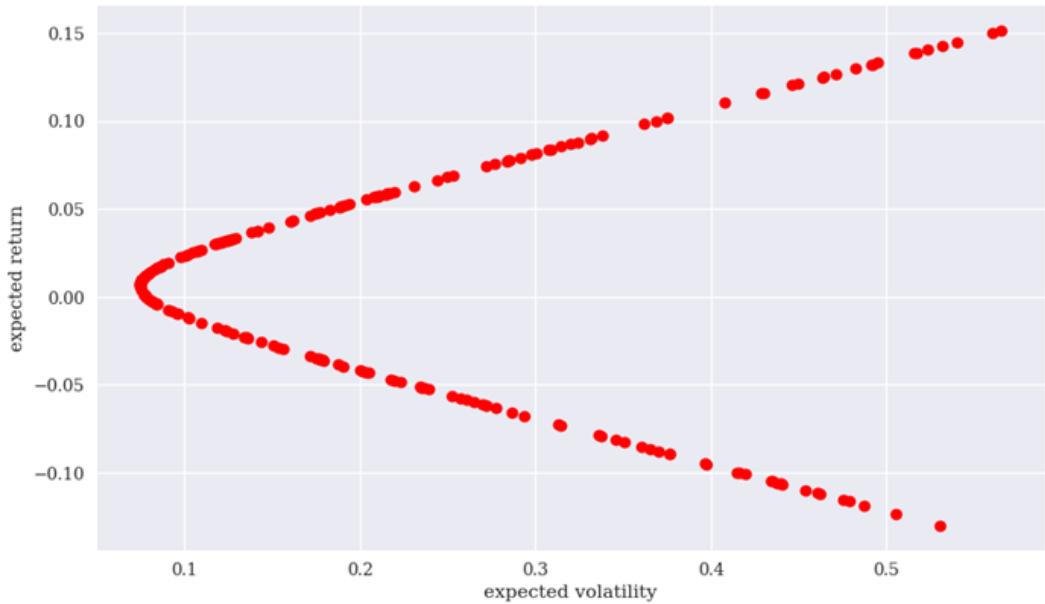


Figura 2.2: Volatilità e rendimento attesi di portafogli simulati (due assets rischiosi)

Volatilità minima e Sharpe Ratio massimo

Successivamente, la derivazione dei portafogli a volatilità minima (varianza minima) e a Sharpe ratio massimo. La figura ?? mostra la posizione dei due portafogli nello spazio rischio-rendimento.

Sebbene l'asset rischioso T abbia un rendimento atteso negativo, ha un peso significativo nel Sharpe ratio massimo.

Questo è dovuto agli effetti di diversificazione che abbassano il rischio del portafoglio più di quanto si riduca il rendimento atteso del portafoglio:

```

1 from scipy.optimize import minimize
2 cons = {'type': 'eq', 'fun': lambda phi: np.sum(phi) - 1}
3 bnds = ((0, 1), (0, 1))
4
5 #riduce al minimo la volatilità attesa del portafoglio
6 min_var = minimize(sigma_phi, (0.5, 0.5),
7                     constraints=cons, bounds=bnds)
8
9 min_var
10 message: Optimization terminated successfully
11 success: True
12 status: 0
13     fun: 0.07481322946903253
14     x: [ 4.651e-01 5.349e-01]
15     nit: 4
16     jac: [ 7.427e-02 7.529e-02]
17     nfev: 13
18     njev: 4
19
20 #definisce la funzione Sharpe ratio,
21 #ipotizzando un short rate di 0
22 def sharpe(phi):
23     return mu_phi(phi) / sigma_phi(phi)
24
25 #massimizza lo Sharpe ratio minimizzando
26 #il suo valore negativo
27 max_sharpe = minimize(lambda phi: -sharpe(phi), (0.5, 0.5),
28                         constraints=cons, bounds=bnds)
29 max_sharpe
30 message: Optimization terminated successfully
31 success: True
32 status: 0
33     fun: -0.27216540990230487
34     x: [ 6.673e-01 3.327e-01]
35     nit: 9
36     jac: [ 1.205e-04 -2.417e-04]
37     nfev: 29
38     njev: 9

```

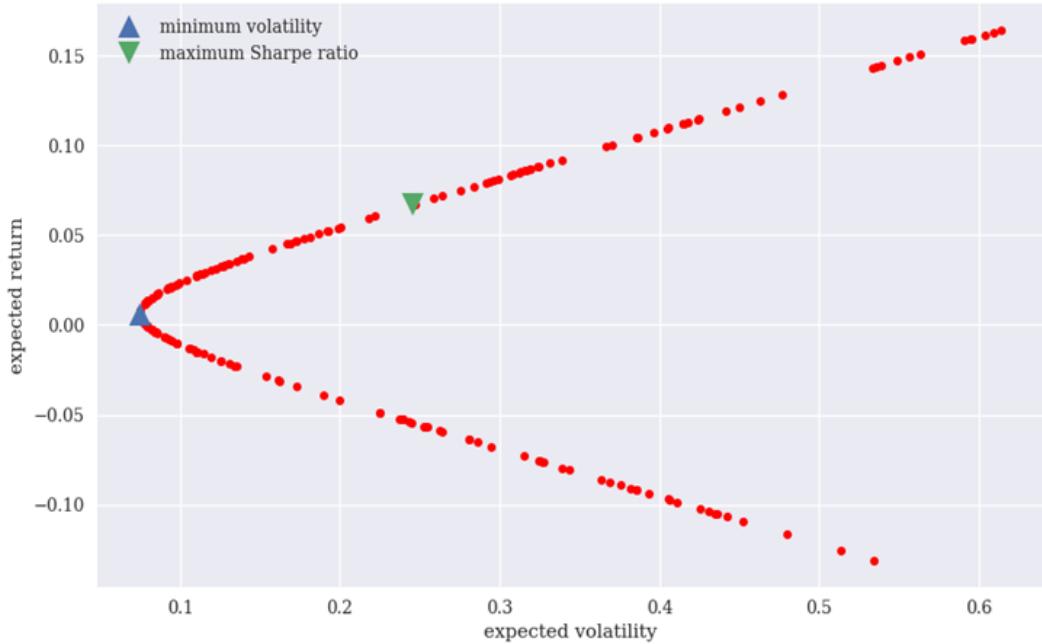


Figura 2.3: Portafogli a volatilità minima e Sharpe ratio massimo

Frontiera efficiente

Un *portafoglio efficiente* è quello che presenta il massimo rendimento atteso (rischio) dato il rischio atteso (rendimento). Nella figura 2.3, tutti i portafogli che hanno un rendimento atteso inferiore al portafoglio a rischio minimo sono *inefficienti*. Ricaviamo i portafogli efficienti nello spazio rischio-rendimento e li tracciamo come si vede nella figura 2.4. L'insieme di tutti i portafogli efficienti è chiamato *frontiera efficiente*, e gli agenti sceglieranno solo un portafoglio che si trova sulla frontiera efficiente:

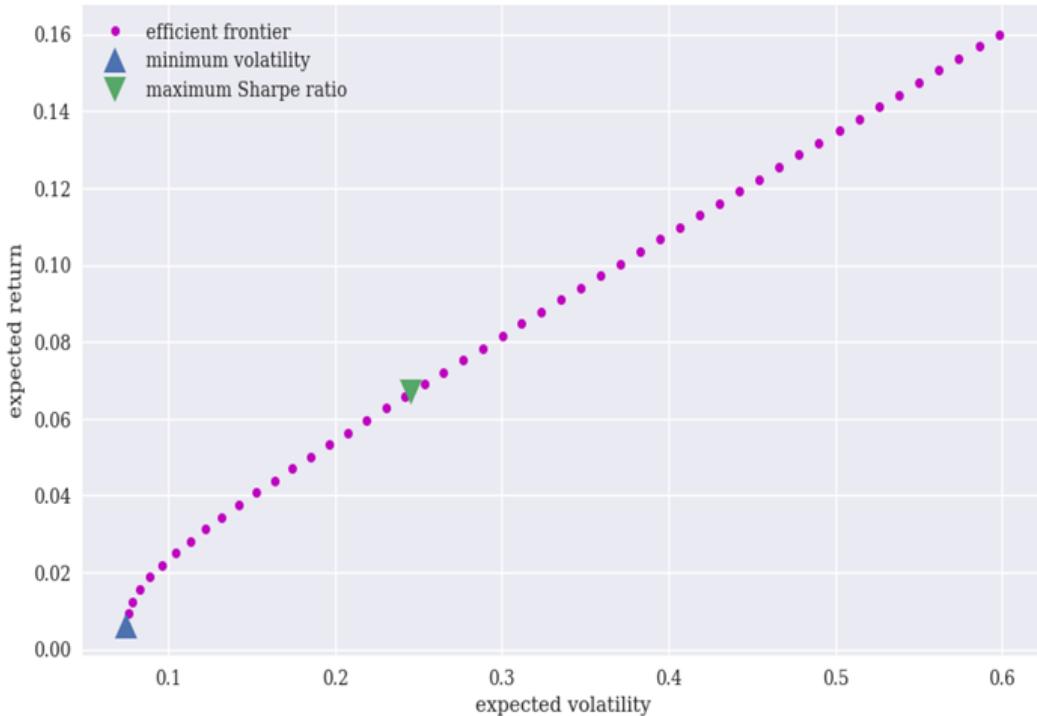


Figura 2.4: Frontiera efficiente

2.4 Capital Asset Pricing Model (CAPM)

Il Capital Asset Pricing Model (CAPM) è uno dei modelli più ampiamente documentati e applicati in finanza. Il suo principio fondamentale è quello di mettere in relazione in modo lineare il rendimento atteso di un singolo titolo con il rendimento atteso del portafoglio di mercato, di solito approssimato da un ampio indice azionario come l'S&P 500. Il modello risale al lavoro pionieristico di Sharpe (1964) e Lintner (1965). Jones (2012) descrive il CAPM in relazione al MVP come segue:

La teoria del mercato dei capitali (capital market theory) è una teoria positiva in quanto ipotizza come gli investitori si comportano piuttosto che, come dovrebbero comportarsi, come nel caso della moderna teoria del portafoglio, (MVP).

È ragionevole considerare la teoria del mercato dei capitali come un'estensione della teoria del portafoglio, ma è importante comprendere che la MVP non si basa sulla validità, o meno, della teoria del mercato dei capitali.

Il modello di equilibrio specifico che interessa a molti investitori è noto come capital asset pricing model, tipicamente indicato come CAPM. Esso ci consente

di valutare il rischio rilevante di un singolo titolo e di valutare la relazione tra il rischio e i rendimenti attesi dall'investimento. Il CAPM è interessante come modello di equilibrio per la sua semplicità e le sue implicazioni.

2.4.1 Presupposti e risultati

Si assuma il modello economico statico della sezione precedente $M^N = (\{\Omega, F, P\}, A)$, con N assets scambiabili e tutte le ipotesi semplificative. Nel CAPM, si ipotizza che gli agenti investono in base alla MVP, preoccupandosi solo delle statistiche di rischio e di rendimento degli assets rischiosi in un periodo.

In un equilibrio del mercato dei capitali (capital market equilibrium), tutti gli assets disponibili sono detenuti da tutti gli agenti e i mercati sono “clear” (compensati), ovvero, l’offerta di tutto ciò che viene scambiato è equiparata alla domanda in modo che non vi sia eccesso di offerta o di domanda.

Poiché si ipotizza che gli agenti siano identici in quanto utilizzano il MVP per formare i loro portafogli efficienti, ciò implica che tutti gli agenti devono detenere lo stesso portafoglio efficiente (in termini di composizione) poiché l’insieme di assets scambiabili è lo stesso per ogni agente. In altre parole, il portafoglio di mercato (insieme di assets scambiabili) deve trovarsi sulla frontiera efficiente. Se così non fosse, il mercato non potrebbe essere in equilibrio.

Qual è il meccanismo per ottenere l’equilibrio del mercato dei capitali?

I prezzi odierni degli assets scambiabili sono il meccanismo che assicura che i mercati siano “clear” (compensati). Se gli agenti non chiedono (domanda) una quantità sufficiente di un asset negoziabile, il suo prezzo deve diminuire. Se la domanda è superiore dell’offerta, il prezzo deve aumentare. Se i prezzi sono fissati correttamente, la domanda e l’offerta si equivalgono per ogni asset commerciabile.

Il CAPM presuppone l’esistenza di (almeno) un asset senza rischio in cui ogni agente può investire qualsiasi importo e che guadagna il tasso privo di rischio di r . Ogni agente deterrà quindi una combinazione del portafoglio di mercato e dell’asset privo di rischio in equilibrio, cosa nota come teorema di separazione dei due fondi. L’insieme di tutti i portafogli di questo tipo è denominato *capital market line (CML)*.

La figura 2.5 mostra il CML schematicamente. I portafogli a destra del portafoglio di mercato sono realizzabili solo se gli agenti possono *vendere allo scoperto*³ (short selling) l’asset privo di rischio e prendere così a prestito denaro:

³La vendita allo scoperto (in lingua inglese short selling, o semplicemente short), è un’operazione finanziaria che consiste nella vendita di titoli non direttamente posseduti dal venditore, ma presi in prestito dietro il versamento di un corrispettivo, con l’intento di ot-

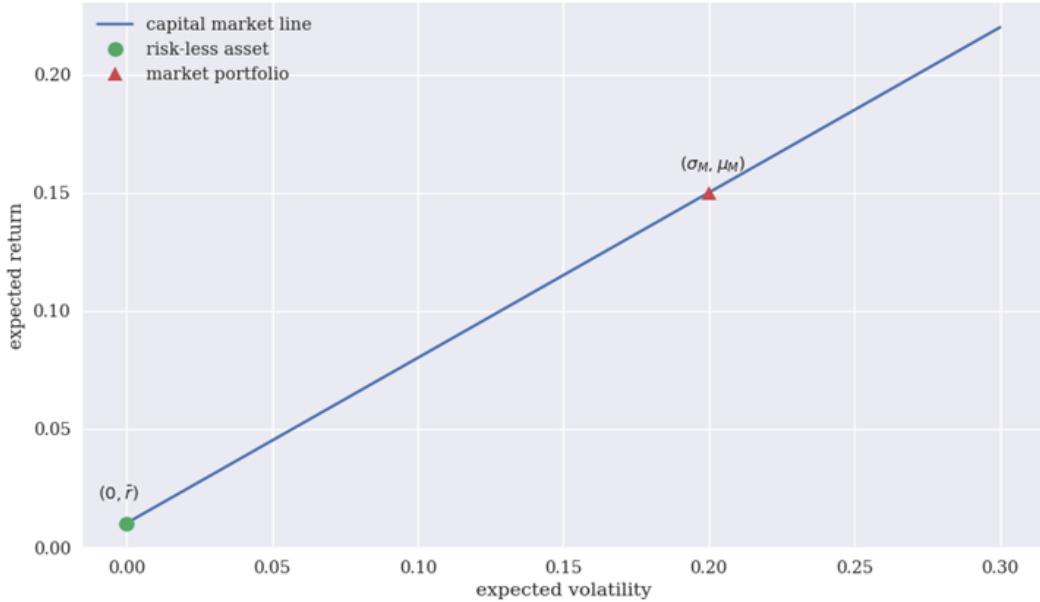


Figura 2.5: Capital Market Line (CML)

Se σ_M, μ_M sono la volatilità e il rendimento atteso del portafoglio di mercato, la retta del mercato dei capitali che mette in relazione il rendimento atteso del portafoglio μ con la volatilità attesa σ è definita come segue:

$$\mu = r + \frac{\mu_M - r}{\sigma_M} \sigma$$

La seguente espressione è chiamata *prezzo di mercato del rischio* (*market price of risk*):

$$\frac{\mu_M - r}{\sigma_M}$$

Esprime la quantità di rendimento atteso in equilibrio necessario affinché un agente sopporti un'unità di rischio in più. Il CAPM mette quindi in relazione il rendimento atteso di un qualsiasi asset rischioso negoziabile $n = 1, 2, \dots, N$ al rendimento atteso del portafoglio di mercato come segue:

$$\mu^n = r + \beta_n(\mu_M - r)$$

In questo caso, β_n è definito dalla covarianza del portafoglio di mercato con l'asset rischioso n divisa per la varianza del portafoglio di mercato stesso:

tenere un profitto a seguito di un movimento ribassista in una borsa valori. La vendita allo scoperto è un'operazione finanziaria di tipo prettamente speculativo e orientata verso un orizzonte temporale di brevissimo periodo.

$$\beta_n = \frac{\sigma_{M,n}}{\sigma_M^2}$$

Quando $\beta_n = 0$, il rendimento atteso secondo la formula del CAPM è il tasso privo di rischio. Quanto più alto è β_n , tanto più alto sarà il rendimento atteso per l'asset rischioso. β_n misura il rischio non diversificabile. Questo tipo di rischio è chiamato anche rischio di mercato o rischio sistemico. Secondo il CAPM, si tratta dell'unico rischio per il quale un agente dovrebbe essere ricompensato con un rendimento atteso superiore.

2.4.2 Esempio Numerico

Si ipotizzi un modello economico statico con tre possibili stati futuri $M^3 = (\{\Omega, F, P\}, A)$ con la possibilità di prendere e concedere prestiti a un tasso privo di rischio pari a $r = 0.0025$. I due asset rischiosi S e T sono disponibili in quantità pari a 0.8 e 0.2 rispettivamente.

Capital Market Line

La figura 2.6 mostra la frontiera efficiente, il portafoglio di mercato, l'attività senza rischio e la capital market line risultante nello spazio rischio-rendimento:

```

1 S0 = 10
2 S1 = np.array((20, 10, 5))
3 T0 = 10
4 T1 = np.array((1, 12, 13))
5 P = np.ones(3)/3
6 M0 = np.array((S0, T0))
7 M1 = np.array((S1, T1)).T
8 rM = M1 / M0 - 1
9
10 #funzione di rendimento atteso
11 def mu(rX):
12     return np.dot(P, rX)
13 #il rendimento atteso del portafoglio
14 def mu_phi(phi):
15     return np.dot(phi, mu(rM))
16 phi_M = np.array((0.8, 0.2))
17 mu_M = mu_phi(phi_M)
18 mu_M
19 0.1066666666666666
20 #la varianza attesa del portafoglio

```

```

21 def var_phi(phi):
22     cv = np.cov(rM.T, aweights=P, ddof=0)
23     return np.dot(phi, np.dot(cv, phi))
24
25 #la volatilita' attesa del portafoglio
26 def sigma_phi(phi):
27     return var_phi(phi) ** 0.5
28
29 sigma_M = sigma_phi(phi_M)
30 sigma_M
31 0.39474323581566567

```

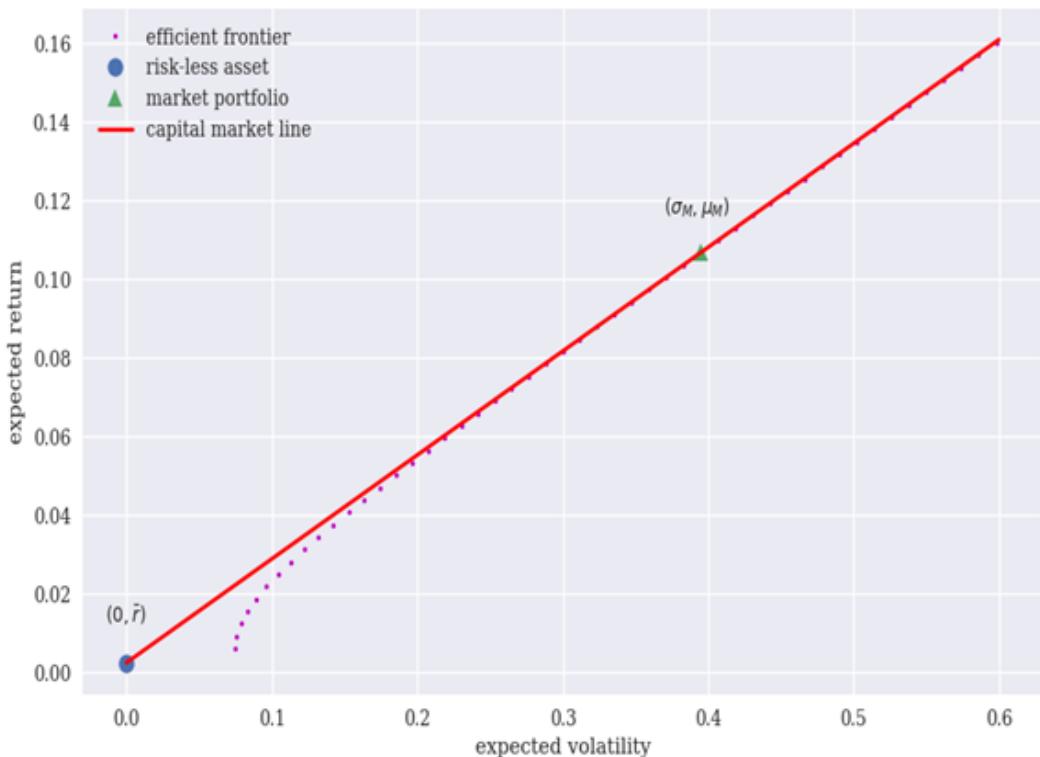


Figura 2.6: Capital Market Line con due assets rischiosi

Portafoglio ottimale

Si ipotizzi un agente con una funzione di utilità attesa definita sui payoff futuri come segue:

$$U : X \rightarrow \mathbb{R}_+, x \mapsto E^P(u(x)) = E^P(x - \frac{b}{2}x^2)$$

In questo caso, $b > 0$. Dopo alcune trasformazioni, la funzione di utilità attesa può essere espressa su combinazioni di rischio-rendimento:

$$U : R_+ \times R_+ \rightarrow R, (\sigma, \mu) \mapsto \mu - \frac{b}{2}(\sigma^2 + \mu^2)$$

Quale combinazione di portafoglio sceglierrebbe l'agente sul CML?

Una semplice massimizzazione dell'utilità, implementata in Python, fornisce la risposta. A tal fine fissiamo il parametro $b = 1$:

```

1 #la funzione di utilita' nello spazio rischio—rendimento
2 def U(p):
3     mu, sigma = p
4     return mu - 1 / 2 * (sigma ** 2 + mu ** 2)
5
6 #la condizione che il portafoglio sia sulla CML
7 cons = {'type': 'eq', 'fun': lambda p: p[0] -
8         (r + (mu_M - r) / sigma_M * p[1])}
9
10 opt = minimize(lambda p: -U(p), (0.1, 0.3),
11                 constraints=cons)
12
13 opt
14 message: Optimization terminated successfully
15 success: True
16 status: 0
17     fun: -0.034885186826739426
18     x: [ 6.744e-02 2.461e-01]
19     nit: 2
20     jac: [-9.326e-01 2.461e-01]
21     nfev: 6
22     njev: 2

```

Curve di differenza

Un'analisi visiva può illustrare il processo decisionale ottimale dell'agente. Fissando un livello di utilità per l'agente, è possibile tracciare curve di indifferenza nello spazio rischio-rendimento. Un portafoglio ottimale si trova quando una curva di indifferenza è tangente alla CML. Qualsiasi altra curva di indifferenza (che non tocchi la CML o che la tagli due volte) non può identificare un portafoglio ottimale.

In primo luogo, ecco un codice Python simbolico che trasforma la funzione di utilità nello spazio rischio-rendimento in una relazione funzionale tra μ e σ per un livello di utilità fisso v e un valore fisso del parametro b .

La figura 2.7 mostra due curve di indifferenza. Ogni combinazione (σ, μ) su tale curva di indifferenza produce la stessa utilità; l'agente è indifferente tra questi portafogli:

```
from sympy import *
init_printing(use_unicode=False, use_latex=False)

#definisce i simboli di SymPy
mu, sigma, b, v = symbols('mu sigma b v')

#risolve la funzione di utilita' per u
sol = solve('mu - b / 2 * (sigma ** 2 + mu ** 2) - v', mu)

sol
[-----, -----]
[  / 2 2      / 2 2
 1 - \b - sigma - 2*b*v + 1 , \b - sigma - 2*b*v + 1 + 1 ]
          b           b

#sostituisce i valori numerici per b, v
u1 = sol[0].subs({'b': 1, 'v': 0.1})
u1
-----, 2
1 - \b 0.8 - sigma
u2 = sol[0].subs({'b': 1, 'v': 0.125})
u2
-----, 2
1 - \b 0.75 - sigma
#genera funzioni richiamabili dalle equazioni risultanti
f1 = lambdify(sigma, u1)
f2 = lambdify(sigma, u2)

#specifica i valori per sigma su cui valutare le funzioni
sigma_ = np.linspace(0.0, 0.5)
```

```
#valuta le funzioni richiamabili per i
#due livelli diversi di utilita'
u1_ = f1(sigma_)
u2_ = f2(sigma_)
plt.figure(figsize=(10, 6))
plt.plot(sigma_, u1_, label='v=0.1')
plt.plot(sigma_, u2_, '—', label='v=0.125')
plt.xlabel('expected volatility')
plt.ylabel('expected return')
plt.legend();
```

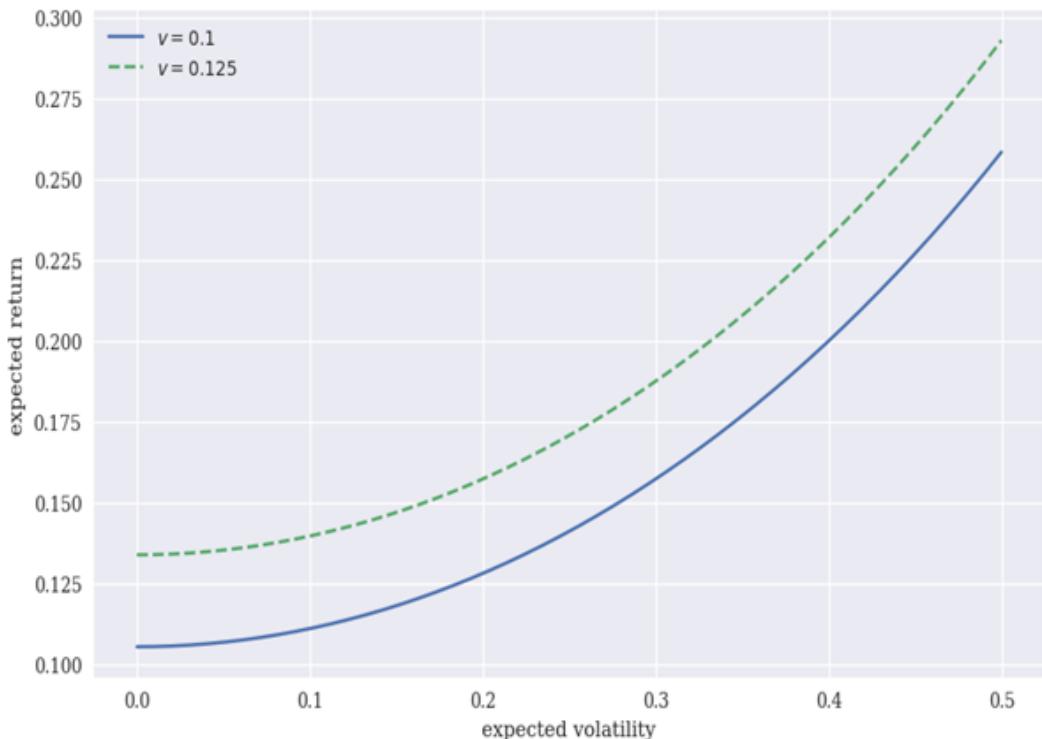


Figura 2.7: Curve di indifferenza nello spazio rischio-rendimento

In una fase successiva, le curve di indifferenza devono essere combinate con la CML per scoprire visivamente qual è la scelta ottimale del portafoglio dell'agente.

Utilizzando i risultati dell'ottimizzazione numerica, la figura 2.8 mostra il portafoglio ottimale – il punto in cui la curva di indifferenza è tangente alla CML.

La figura 2.8 mostra che l'agente sceglie effettivamente una miscela di portafoglio di mercato e l'asset privo di rischio:

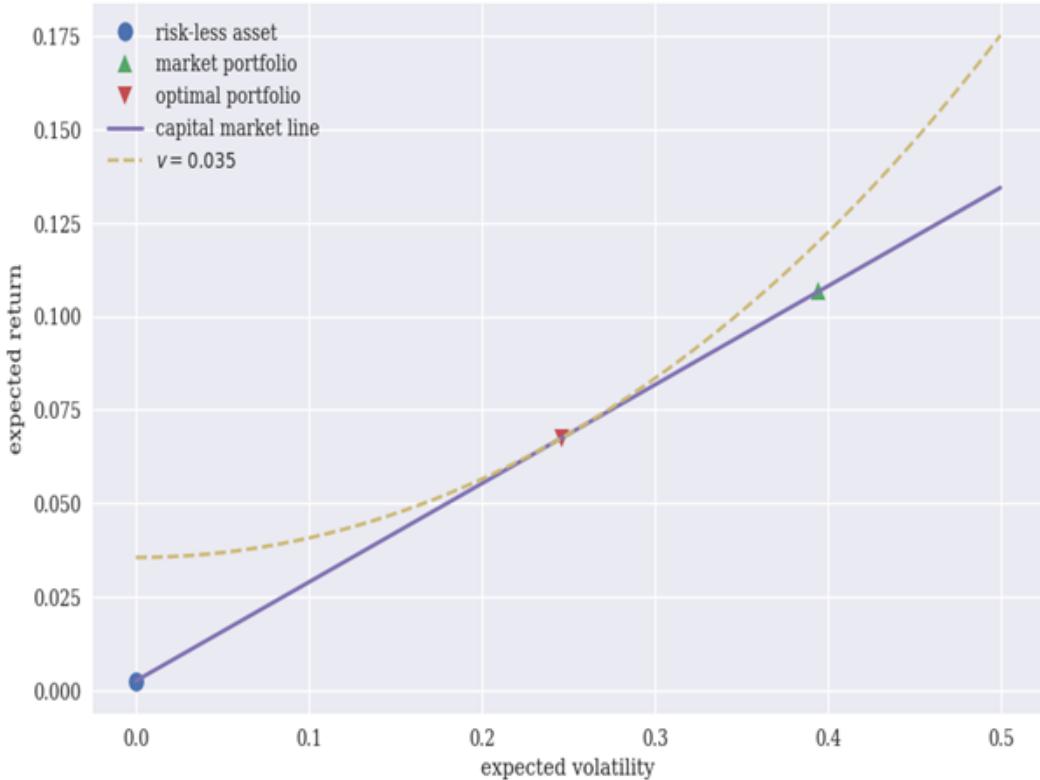


Figura 2.8: Portafoglio ottimale sul CML

Gli argomenti presentati in questa sottosezione sono solitamente trattati nell'ambito della *teoria del mercato dei capitali (CMT)*. Il CAPM fa parte di questa teoria e sarà illustrato con l'uso di dati reali di serie temporali finanziarie nel prossimo capitolo.

2.5 Conclusioni

Alcune delle prime teorie e dei primi modelli dagli anni '40 agli anni '70, in particolare quelli presentati in questo capitolo, sono ancora argomenti centrali dei libri di testo di finanza e sono ancora utilizzati nella pratica finanziaria. Uno dei motivi è che molte di queste teorie e modelli, per lo più normativi, esercitano un forte fascino intellettuale su studenti, accademici e professionisti. In qualche modo “sembrano semplicemente avere senso”.

Nonostante teorie e modelli come il MVP e il CAPM siano intellettualmente attraenti, facili da implementare e matematicamente eleganti, è sorprendente che siano ancora oggi così popolari, per alcune ragioni. In primo luogo, le teorie e i modelli popolari presentati in questo capitolo non hanno praticamente alcun supporto empirico significativo. In secondo luogo, alcune di queste teorie e modelli sono addirittura teoricamente incoerenti tra loro sotto diversi punti di vista. In terzo luogo, si sono registrati continui progressi sul fronte teorico e modellistico della finanza, tanto che sono disponibili teorie e modelli alternativi. In quarto luogo, la moderna finanza computazionale ed empirica può contare su fonti di dati quasi illimitate e su una potenza di calcolo quasi illimitata, rendendo sempre meno rilevanti modelli e risultati matematici concisi ed eleganti.

Il prossimo capitolo analizza alcune delle teorie e modelli introdotti in questo capitolo sulla base di dati finanziari reali. Mentre nei primi anni della finanza quantitativa i dati erano una risorsa scarsa, oggi anche noi studenti abbiamo accesso a una grande quantità di dati finanziari e a strumenti open source che consentono un'analisi completa delle teorie e dei modelli finanziari sulla base di dati reali. La finanza empirica è sempre stata un'importante disciplina sorella della finanza teorica. Tuttavia, la teoria finanziaria ha sempre guidato in larga misura la finanza empirica. La nuova area della finanza guidata dai dati potrebbe portare a un cambiamento duraturo nell'importanza relativa della teoria rispetto ai dati in finanza.

Capitolo 3

Finanza guidata dai dati

“Se l’intelligenza artificiale è la nuova elettricità, i big data sono il petrolio che alimenta i generatori.”

— Kai-Fu Lee (2018)

In questo capitolo si discutono gli aspetti centrali della finanza guidata dai dati. La finanza guidata dai dati è intesa come un contesto finanziario (teoria, modello, applicazione e così via) che si basa principalmente su intuizioni e informazioni ottenute dai dati.

3.1 Metodo scientifico

Il metodo scientifico si riferisce a un insieme di principi generalmente accettati che dovrebbero guidare qualsiasi progetto scientifico. Wikipedia definisce il metodo scientifico come segue:

Il metodo scientifico è un metodo empirico di acquisizione della conoscenza che ha caratterizzato lo sviluppo della scienza almeno sin dal XVII secolo. Comporta un’attenta osservazione, l’applicazione di un rigoroso scetticismo nei confronti di ciò che viene osservato, dato che i presupposti cognitivi possono distorcere il modo in cui si interpreta l’osservazione. Comporta la formulazione di ipotesi, per induzione, sulla base di tali osservazioni; la verifica sperimentale e di misurazione delle deduzioni tratte dalle ipotesi; e il perfezionamento (o l’eliminazione) delle ipotesi sulla base dei risultati sperimentali. Questi sono i principi del metodo scientifico, che si distingue da una serie definitiva di passi applicabili a tutte le imprese scientifiche.

In base a questa definizione, la finanza normativa, come discusso nel capitolo

precedente, si pone in netto contrasto con il metodo scientifico. Le teorie finanziarie normative si basano per lo più su presupposti e assiomi in combinazione con la deduzione come principale metodo analitico per arrivare ai loro risultati centrali.

- La teoria dell'utilità attesa (EUT) presuppone che gli agenti abbiano la stessa funzione di utilità a prescindere dallo stato del mondo in cui si svolge e che essi massimizzino l'utilità attesa in condizioni di incertezza.
- La teoria della media-varianza del portafoglio (MVP) descrive come gli investitori dovrebbero investire in condizioni di incertezza, ipotizzando che contino solo sul rendimento atteso e sulla volatilità attesa di un portafoglio in un certo periodo.
- Il Capital Asset Pricing Model (CAPM) ipotizza che solo il rischio di mercato non diversificabile spieghi il rendimento atteso e che il rendimento atteso e la volatilità attesa di un'azione in un periodo.

Ciò che caratterizza le teorie finanziarie normative sopra citate è sono state originariamente derivate sulla base di determinate ipotesi (presupposti, assunzioni) e assiomi utilizzando solo “carta e penna”, senza alcun ricorso a dati o a osservazioni del mondo reale. Da un punto di vista storico, molte di queste teorie sono state rigorosamente testate rispetto ai dati del mondo reale solo molto tempo dopo la loro data di pubblicazione. Ciò può essere spiegato principalmente dal fatto che soltanto successivamente si ha avuto una migliore disponibilità dei dati e maggiori capacità di calcolo. D'altronde, i dati e i calcoli sono gli ingredienti principali per l'applicazione dei metodi statistici nella pratica. La disciplina all'incrocio tra matematica, statistica e finanza che applica tali metodi ai dati dei mercati finanziari, è tipicamente chiamata econometria finanziaria, argomento della prossima sezione.

3.2 Econometria finanziaria e Regressione

Adattando la definizione fornita da Investopedia per l'econometria, si può definire l'econometria finanziaria come segue:

L'econometria (finanziaria) è l'applicazione quantitativa di modelli matematici e statistici utilizzando dati (finanziari) per sviluppare teorie finanziarie o per testare le ipotesi esistenti in finanza e di prevedere le tendenze future a partire dai dati storici. Sottopone i dati (finanziari) del mondo reale a prove statistiche e poi confronta e contrasta i risultati con la teoria o le teorie (finanziarie) in fase di test.

Uno dei principali strumenti dell'econometria finanziaria è la regressione, sia in forma univariata che multivariata. La regressione è anche uno strumento centrale dell'apprendimento statistico in generale. Qual è la differenza tra la matematica tradizionale e l'apprendimento statistico? Sebbene non esista una risposta in generale a questa domanda (dopo tutto, la statistica è un sottocampo della matematica), un semplice esempio dovrebbe sottolineare una differenza importante per il contesto di questa tesi. Il primo è il metodo matematico standard. Si supponga che una funzione matematica sia data come segue:

$$f : R \rightarrow R_+, \quad x \mapsto 2 + \frac{1}{2}x$$

Dati più valori di x_i , $i = 1, 2, \dots, n$, si possono ricavare i valori della funzione f applicando la definizione precedente:

$$y_i = f(x_i), \quad i = 1, 2, \dots, n$$

Il secondo è l'approccio adottato nell'apprendimento statistico. Mentre nell'esempio precedente, la funzione viene prima e poi vengono ricavati i dati, nell'apprendimento statistico tale sequenza è invertita. Qui i dati sono generalmente forniti e si deve trovare una relazione funzionale. In questo contesto, x è spesso chiamata variabile indipendente e y variabile dipendente. Di conseguenza, si considerino i seguenti dati:

$$(x_i, y_i), \quad i = 1, 2, \dots, n$$

Il problema è trovare, ad esempio, i parametri α, β , tali che:

$$\hat{f}(x_i) \equiv \alpha + \beta x_i = \hat{y}_i \approx y_i, \quad i = 1, 2, \dots, n$$

Un altro modo di scriverlo è includendo i valori residui ϵ_i , $i = 1, 2, \dots, n$:

$$\alpha + \beta + \epsilon_i = y_i, \quad i = 1, 2, \dots, n$$

Nel contesto della regressione ordinaria dei minimi quadrati (OLS), α, β sono scelti per minimizzare l'errore quadratico medio tra i valori approssimati \hat{y}_i e i valori reali y_i . Il problema di minimizzazione è quindi il seguente:

$$\min_{\alpha, \beta} \frac{1}{n} \sum_i^n (\hat{y}_i - y_i)^2$$

Nel caso della regressione OLS semplice, come descritto in precedenza, le soluzioni ottimali sono note in forma chiusa e sono le seguenti:

$$\begin{cases} \beta = \frac{Cov(x,y)}{Var(x)} \\ \alpha = \bar{y} - \beta \bar{x} \end{cases}$$

Dove, $\text{Cov}()$ indica la covarianza, $\text{Var}()$ la varianza e \bar{x} , \bar{y} i valori medi di x e y . Ciò ci consente di ricavare i parametri ottimali α, β . A seguito un semplice script in Python di come eseguire la regressione OLS:

```

1 def f(x):
2     return 2 + 1/2 * x
3
4 x = np.arange(-4, 5)
5 x
6 array([-4, -3, -2, -1, 0, 1, 2, 3, 4])
7
8 y = f(x)
9 y
10 array([0., 0.5, 1., 1.5, 2., 2.5, 3., 3.5, 4.])
11
12 beta = np.cov(x, y, ddof=0)[0,1] / x.var()
13 beta
14 0.4999999999999994
15
16 alpha = y.mean() - beta * x.mean()
17 alpha
18 2.0
19
20 y_ = alpha + beta * x
21
22 #controlla se i valori di y_ e di y sono numericamente uguali
23 np.allclose(y_, y)
24 True

```

L'applicazione della regressione OLS a un dato insieme di dati è in generale semplice. Ci sono anche altre ragioni per le quali la regressione OLS è diventata uno degli strumenti centrali dell'econometria finanziari. Tra questi vi sono i seguenti:

- **Vecchio di secoli:** L'approccio dei minimi quadrati, in particolare in combinazione con la regressione, è stato utilizzato per più di 200 anni.
- **Semplicità:** La matematica che sta alla base della regressione OLS è facile da comprendere e da implementare nella programmazione.
- **Scalabilità:** Non c'è praticamente alcun limite per quanto riguarda la dimensione dei dati a cui la regressione OLS può essere applicata.
- **Flessibilità:** La regressione OLS può essere applicata ad un'ampia gamma di problemi e insiemi di dati.

- **Velocità:** La regressione OLS è veloce da valutare, anche su insiemi di dati di grandi dimensioni.
- **Disponibilità:** Sono disponibili implementazioni efficienti in Python e in molti altri linguaggi di programmazione.

Tuttavia, per quanto l'applicazione della regressione OLS possa essere semplice e diretta, il metodo si basa su una serie di assunzioni, la maggior parte delle quali relative ai valori residui, che non sono sempre soddisfatte nella pratica.

- **Linearità:** Il modello è lineare nei suoi parametri, sia per i coefficienti che per i valori residui.
- **Indipendenza:** Le variabili indipendenti non sono perfettamente correlate tra loro.
- **Media Zero:** il valore medio dei residui è (prossimo) allo zero.
- **Nessuna Correlazione:** I valori residui non sono (fortemente) correlati con le variabili indipendenti.
- **Omoschedasticità:** La deviazione standard dei valori residui è (quasi) costante.

In pratica, è abbastanza semplice verificare la validità delle assunzioni, dato uno specifico insieme di dati.

3.3 Disponibilità dei Dati

Dagli anni '50 agli anni '90, e anche i primi anni duemila, la ricerca finanziaria teorica ed empirica si basava principalmente su insiemi di dati relativamente piccoli rispetto agli standard odierni, ed erano per lo più composti da dati di fine giornata (end-of-day EOD). La disponibilità dei dati è cambiata drasticamente nell'ultimo decennio, con sempre più dati di tipo finanziario disponibili in granularità, quantità e velocità sempre maggiori. Dietro questa tendenza c'è una storia familiare: spinti dalla crescita esplosiva di internet e delle reti mobili, i dati digitali continuano a crescere in modo esponenziale tra i progressi della tecnologia per elaborare, archiviare e analizzare nuove fonti di dati. La crescita esponenziale della disponibilità e della capacità di gestire dati digitali sempre più diversificati, a sua volta, è stata una forza critica dietro i drastici miglioramenti delle prestazioni dell'apprendimento automatico (ML) che sta guidando l'innovazione in tutti i settori, compreso quello degli investimenti.

La portata della rivoluzione dei dati è straordinaria: solo negli ultimi due anni è stato creato il 90% di tutti i dati oggi esistenti al mondo e si prevede che ognuna delle 7.7 miliardi di persone al mondo produrrà 1.7MB di nuove informazioni ogni secondo di ogni giorno. D'altra parte, nel 2012, solo lo 0.5% di tutti i dati è stato analizzato e utilizzato, mentre il 33% avrà un valore nel futuro. Il divario tra disponibilità e utilizzo dei dati è destinato a ridursi rapidamente man mano che gli investimenti globali nell'analisi sono impostati a superare i 210 miliardi, mentre il potenziale il potenziale di creazione di valore è di gran lunga superiore.

La seguente tabella 3.1 offre una panoramica delle categorie di dati che sono generalmente rilevanti in un contesto finanziario. Nella tabella, i dati strutturati si riferiscono a tipi di dati numerici che spesso sono presentati in strutture tabellari, mentre dati non strutturati si riferiscono a dati sotto forma di testo che spesso non hanno una struttura al di là di intestazioni o paragrafi. I dati alternativi si riferiscono a tipi di dati che in genere non sono considerati dati finanziari.

Time	Structured Data	Unstructured Data	Alternative Data
Historical	Prices, fundamentals	News, texts	Web, Social Media, Satellites
Streaming	Pieces, volumes	News, filings	Web, social media, satellites, Internet of Things

Tabella 3.1: Tipologie rilevanti di dati finanziari

A seguito approfondiremo soltanto i dati alternativi poiché sono quelli più interessanti e più attuali.

3.3.1 Dati Alternativi

In questa sezione, ci soffermeremo sulla recente comparsa di un'ampia gamma di fonti di dati molto più diversificate come carburante per strategie discrezionali e algoritmiche.

La loro eterogeneità e la loro novità hanno ispirato l'etichetta di “dati alternativi” e hanno creato un’industria di fornitori e servizi in rapida crescita.

Oggigiorno, le istituzioni finanziarie, e in particolare gli hedge funds, estraggono sistematicamente una serie di fonti di dati alternativi per ottenere un

vantaggio nel trading e negli investimenti. Un recente articolo di Bloomberg elenca le seguenti fonti di dati alternativi:

- Dati raccolti dal web (web-scraped data)
- Dati di crowdsourcing (crowd-sourced data)
- Carte di credito e sistemi POS (point-of-sales)
- Sentiment dei social media
- Tendenze di ricerca
- Traffico web
- Dati sulla catena di approvvigionamento
- Dati sulla produzione di energia
- Profili dei consumatori
- Immagini satellitari/dati geospaziali
- Installazioni di app
- Tracciamento di imbarcazioni oceaniche
- Dispositivi indossabili, droni, sensori di Internet of Things (IoT)

La rivoluzione dei dati alternativi

Il diluvio di dati guidato dalla digitalizzazione, dalla rete e dal crollo dei costi di archiviazione ha portato a profondi cambiamenti qualitativi nella natura delle informazioni disponibili per l'analisi predittiva, spesso sintetizzati dalle cinque V:

- **Volume:** La quantità di dati generati, raccolti e immagazzinati è di ordini di grandezza superiore, come risultato di attività online e offline, transazioni, registrazioni e di altre fonti. I volumi continuano a crescere insieme alla capacità di analisi e di archiviazione.
- **Velocità:** I dati vengono generati, trasferiti ed elaborati per diventare disponibili in tempo reale.

- **Varietà:** I dati sono organizzati in formati non più limitati a forme strutturate e tabellari, come i file CSV o le tabelle dei database aziendali. Invece, le nuove fonti producono formati semistrutturati, come JSON o HTML, e contenuti non strutturati, tra cui il testo grezzo, immagini e dati audio e video, aggiungendo nuove sfide per rendere i dati adatti agli algoritmi di ML.
- **Veridicità:** La diversità delle fonti e dei formati rende molto più difficile l'affidabilità del contenuto informativo dei dati.
- **Valore:** La determinazione del valore di nuovi insiemi di dati può richiedere molto più tempo e risorse, oltre che essere più incerto rispetto al passato.

Oggi gli investitori possono accedere in tempo reale a dati macro o specifici di una azienda che storicamente erano disponibili solo con una frequenza molto più bassa.

I casi di utilizzo delle nuove fonti di dati includono le seguenti:

- I dati sui prezzi online di un insieme rappresentativo di beni e servizi possono essere utilizzati per misurare l'inflazione.
- Il numero di visite o di acquisti nei negozi permette di stimare in tempo reale le vendite o l'attività economica di un'azienda o di un settore.
- Le immagini satellitari possono rivelare i raccolti agricoli, l'attività nelle miniere o sulle piattaforme petrolifere prima che queste informazioni siano disponibili altrove.

Con l'avanzare della standardizzazione e dell'adozione dei big data, le informazioni contenute nei dati convenzionali perderanno probabilmente gran parte del loro valore predittivo. Inoltre, la capacità di elaborare e integrare insiemi di dati diversi e di applicare il ML consente di ottenere informazioni più complesse.

In passato, gli approcci quantitativi si basavano su semplici euristiche per classificare le aziende utilizzando i dati storici per metriche come il rapporto prezzo/valore contabile, mentre gli algoritmi di ML sintetizzano nuove metriche e apprendono, adattano tali regole tenendo conto dell'evoluzione dei dati di mercato.

Queste nuove informazioni (insights) creano nuove opportunità per cogliere i classici temi d'investimento come valore, momentum, qualità e sentimento:

- **Momentum:** Attraverso il ML si riesce ad identificare l'esposizione degli asset ai movimenti dei prezzi di mercato, al sentimento del settore e ai fattori economici.
- **Valore:** Gli algoritmi di ML possono analizzare grandi quantità di dati economici e di settore, strutturati e non strutturati, oltre ai bilanci, per prevedere il valore intrinseco di un'azienda.
- **Qualità:** L'analisi sofisticata dei dati integrati consente di valutare le recensioni dei clienti o dei dipendenti, dell'e-commerce e del traffico delle app per identificare gli aumenti di quota di mercato o altri fattori di qualità degli utili.
- **Sentiment:** L'elaborazione e l'interpretazione in tempo reale delle notizie e dei contenuti dei social media permette agli algoritmi di ML di rilevare rapidamente il sentimento e di sintetizzare le informazioni provenienti da fonti diverse avendo un quadro generale più coerente.

In pratica, tuttavia, i dati contenenti segnali di valore spesso non sono liberamente disponibili e vengono tipicamente prodotti per scopi diversi da quello del trading. Di conseguenza, i set di dati alternativi richiedono di una valutazione approfondita, di un'acquisizione costosa, di una gestione attenta e di un'analisi sofisticata per estrarre i segnali negoziabili.

Il mercato dei dati alternativi

Si stima che il settore degli investimenti abbia speso dai 2 ai 3 miliardi di dollari per i servizi di dati nel 2018, e si prevede che questo numero sia destinati a crescere a un tasso di due cifre all'anno.

Questa spesa comprende l'acquisizione di dati alternativi, gli investimenti in tecnologie correlate e l'assunzione di personale qualificato.

Un'indagine di Ernst & Young mostra un'adozione significativa di dati alternativi nel 2017; il 43% dei fondi utilizzava dati web scraped, e quasi il 30% sperimentava con dati satellitari (Figura 3.1).

Sulla base dell'esperienza maturata finora, i gestori dei fondi ritengono che i dati ricavati dal web (web-scraping) e i dati delle carte di credito siano più perspicaci, a differenza dai dati satellitari e di geolocalizzazione, che circa il 25 percento considera meno informativi:

A causa della rapida crescita di questo nuovo settore, il mercato dei fornitori di dati è alquanto frammentato. J.P. Morgan elenca oltre 500 società specializzate sui dati, mentre AlternativeData.org ne elenca oltre 300. I fornitori svolgono numerosi ruoli, tra cui intermediari come consulenti, aggregatori,

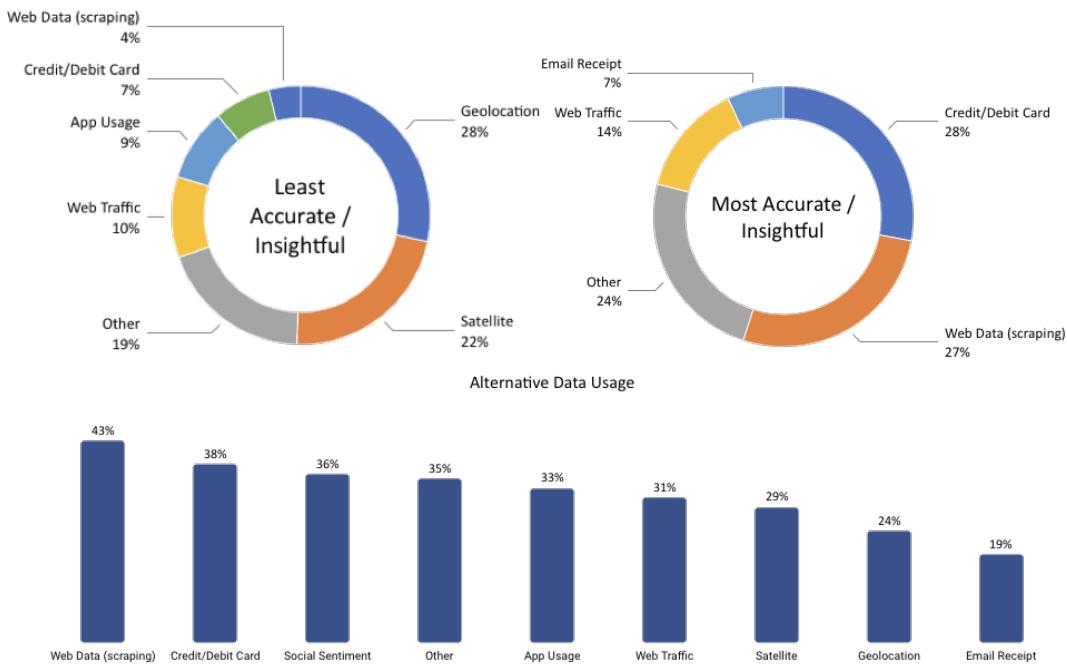


Figura 3.1: Utilità e utilizzo dei dati alternativi (Fonte: Ernst & Young, 2017)

e soluzioni tecnologiche; supporto al lato vendita forniscono dati in vari formati, che vanno dai dati grezzi a quelli semi-elaborati o a qualche forma di segnale estratto da una o più fonti.

Fornitori di dati e casi d'uso

AlternativeData.org (supportato dal provider YipitData) elenca diverse categorie che possono servire come approssimazione delle attività nei vari segmenti di fornitori di dati. L'analisi del sentimento è di gran lunga la categoria più ampia, mentre i dati satellitari e di geolocalizzazione sono cresciuti rapidamente negli ultimi anni:

I brevi esempi che seguono hanno lo scopo di illustrare l'ampia gamma di fornitori e di potenziali casi d'uso dei dati alternativi.

- **Social sentiment data**

L'analisi del sentimento sociale è più strettamente associata ai dati di Twitter. Gnip è stato un primo aggregatore di social media che forniva dati da numerosi siti tramite un'API ed è stato acquisito da Twitter nel 2014 per 134 milioni di dollari. I motori di ricerca sono un'altra fonte che

Categoria di prodotto	N° di fornitori
Social sentiment	48
Satellite	26
Geolocalizzazione	22
Dati e traffico Web	22
Infrastrutture e interfacce	20
Consulenti	18
Utilizzo carte di credito	14
Broker di dati	10
Dati pubblici	10
Utilizzo di app	7
E-mail e ricevute dei consumatori	6
Lato vendite	6
Meteo	4
Altro	87

Tabella 3.2: Fornitori dati

è diventata di rilievo quando i ricercatori hanno pubblicato, su Nature, che le strategie di investimento basate su Google Trends per termini come il debito potevano essere utilizzate per una strategia di trading redditizia per un periodo prolungato (Preis, Moate Stanley 2013).

- **Dati satellitari**

RC Metrics, fondata nel 2010, triangola dati geospaziali provenienti da satelliti, droni, e aerei, con particolare attenzione ai metalli e alle materie prime, nonché al settore immobiliare e industriale. L'azienda offre segnali, analisi predittive, avvisi e applicazioni per l'utente finale basate sui propri satelliti ad alta risoluzione. I casi d'uso includono la stima del traffico al dettaglio in alcune catene o immobili commerciali, nonché la produzione e lo stoccaggio di alcuni metalli comuni o dell'occupazione nei relativi luoghi di produzione.

- **Dati di geolocalizzazione**

Advan, fondata nel 2015, serve i clienti degli hedge fund con segnali derivati dai dati di traffico della telefonia mobile, mirando a 1.600 titoli di diversi settori nei Stati Uniti e nell'UE. L'azienda raccoglie i dati utilizzando app che installano codici di geolocalizzazione sugli smartphone con l'esplícito consenso dell'utente e tracciano la posizione utilizzando diversi canali (ad es. Wi-Fi, Bluetooth e il segnale cellulare) per una maggiore precisione. I casi d'uso comprendono le stime del traffico dei

clienti presso negozi fisici, che a loro volta possono essere utilizzate come input per i modelli di previsione dei ricavi delle società quotate in borsa.

- **Dati sulle ricevute e-mail**

Eagle Alpha fornisce, tra gli altri servizi, dati su un'ampia serie di transazioni online utilizzando le ricevute e-mail, che coprono più di 5000 rivenditori, compresi i dati delle transazioni a livello di SKU, suddivisi in 53 gruppi di prodotti. J.P. Morgan ha analizzato una serie di dati temporali, che coprono il periodo 2013-16, che copriva un gruppo costante di utenti attivi per tutto il periodo del campione. Il set di dati conteneva la spesa totale aggregata, il numero di ordini e il numero di acquirenti unici per periodo (Kolanovic e Krishnamachari, 2017).

In questa sezione è stato presentato un quadro generale sulla grande disponibilità di dati e delle nuove fonti di dati alternativi resi disponibili dalla rivoluzione dei big data.

Nella prossima sezione si occupa della rivisitazione di alcune delle principali teorie della finanza normativa resa possibile anche grazie a questa grande disponibilità di dati.

3.4 Teorie normative rivisitate

Nel capitolo 2 abbiamo introdotto alcune delle teorie finanziarie normative, come la teoria EUT o la teoria MVP.

Per molto tempo, gli studenti e gli accademici che hanno imparato e studiato tali teorie sono stati vincolati alla teoria stessa. Con tutti i dati finanziari disponibili, come abbiamo visto nella sezione precedente, in combinazione con un potente software open source per l'analisi dei dati, come Python, NumPy, pandas e così via, è diventato piuttosto semplice e diretto mettere le teorie finanziarie alla prova nel mondo reale.

Non sono più necessari piccoli team e gradi studi per farlo. Un semplice notebook, un accesso a internet e un ambiente Python standard sono sufficienti.

Questo è l'argomento del capitolo corrente. Prima di immergervi nella finanza guidata dai dati, a seguito discutiamo brevemente di alcuni paradossi nel contesto dell'EUT e di come le aziende modellino e prevedano il comportamento degli individui nella pratica.

3.4.1 Utilità attesa e realtà

In economia, il rischio descrive una situazione i cui possibili stati futuri e le probabilità che tali stati si verifichino sono noti in anticipo al decisore. Questa è

l’ipotesi standard in finanza e nel contesto dell’EUT. D’altro canto, l’ambiguità descrive situazioni in economia in cui le probabilità, o addirittura i possibili stati futuri, non sono noti in anticipo al decisore. L’incertezza racchiude le due diverse situazioni decisionali.

Esiste una lunga tradizione di analisi del comportamento decisionale concreto degli individui (“agenti”) in condizioni di incertezza. Innumerevoli studi ed esperimenti sono stati condotti per osservare e analizzare come gli agenti si comportano di fronte all’incertezza rispetto a quanto previsto dalle teorie come l’EUT. Per secoli, i paradossi hanno svolto un ruolo importante nella teoria e nella ricerca del processo decisionale.

Uno di questi paradossi, il paradosso di San Pietroburgo, ha dato origine all’invenzione delle funzioni di utilità e dell’EUT.

Daniel Bernoulli presentò suddetto paradosso e la sua soluzione nel 1738. Tale paradosso si basa sul seguente gioco del lancio della moneta G. Un agente si trova di fronte al gioco in cui una moneta regolare viene lanciata potenzialmente infinite volte. Se dopo il primo lancio esce “testa”, l’agente riceve un payoff apri a 1 (unità monetaria). Finché esce testa, la moneta viene lanciata di nuovo. In caso contrario, il gioco termina. Se esce testa una seconda volta, l’agente riceve un payoff aggiuntivo di 2. Se prevale una terza volta, il payoff aggiuntivo è di 4. Per la quarta volta è di 8. E così via. Si tratta di una situazione di rischio, poiché tutti i possibili stati futuri e le relative probabilità sono noti in anticipo.

Il payoff atteso di questo gioco è infinito. Ciò si evince dalla seguente somma infinita di cui ogni elemento è strettamente positivo:

$$E(G) = 1/2 \cdot 1 + 1/4 \cdot 2 + 1/8 \cdot 4 + 1/16 \cdot 8 + \dots = \sum_{k=1}^{\infty} \frac{1}{2^k} 2^{k-1} = \sum_{k=1}^{\infty} \frac{1}{2} = \infty$$

Tuttavia, di fronte a un gioco di questo tipo, un decisore in generale sarebbe disposto a pagare una somma finita solo per poter giocare. Una delle ragioni principali è che i payoff relativamente grandi si verificano solo con una probabilità relativamente piccola. Consideriamo come payoff potenziale $W = 511$:

$$W = 1 + 2 + 4 + 8 + 16 + 32 + 64 + 128 + 256 = 511$$

La probabilità di vincere tale payoff è piuttosto bassa. Per essere precisi, è di solo $P(\chi = W) = \frac{1}{512} = 0.001953125$. La probabilità di ottenere un tale payoff o uno minore, d’altra parte, è piuttosto alta: $P(\chi \leq W) = \sum_{k=1}^9 \frac{1}{2^k} = 0.998046875$

In altre parole, in 998 partite su 1.000 il payoff è minore o uguale a 511. Pertanto, un agente probabilmente non scommetterebbe molto di più di 511 per giocare a questo gioco. La via d'uscita da questo paradosso è l'introduzione di una funzione di utilità con utilità marginale positiva ma decrescente. Nel contesto del paradosso di San Pietroburgo, ciò significa che esiste una funzione $u : R_+ \rightarrow R$ che assegna a ogni payoff positivo x un valore reale $u(x)$. L'utilità marginale positiva ma decrescente si traduce formalmente come segue:

$$\frac{\partial_u}{\partial_x} > 0$$

$$\frac{\partial^2_u}{\partial^2_x} < 0$$

Come abbiamo visto nel capitolo 2, una di queste funzioni candidate è $\mu(x) = \ln(x)$ con:

$$\frac{\partial_u}{\partial_x} = \frac{1}{x}$$

$$\frac{\partial^2_u}{\partial^2_x} = -\frac{1}{x^2}$$

L'utilità attesa è quindi finita, come dimostra il calcolo della seguente somma infinita:

$$E(u(G)) = \sum_{k=1}^{\infty} \frac{1}{2^k} u(2^{k-1}) = \sum_{k=1}^{\infty} \frac{\ln(2^{k-1})}{2^k} = \left(\sum_{k=1}^{\infty} \frac{k-1}{2^k} \right) \ln(2) = \ln(2) < \infty$$

L'utilità attesa di $\ln(2) = 0.693147$ è ovviamente un numero piuttosto piccolo rispetto al payoff infinito atteso. Le funzioni di utilità di Bernoulli e l'EUT risolvono il paradosso di San Pietroburgo.

Altri paradossi, come il paradosso di Allais pubblicato in Allais(1953), riguardano la stessa EUT. Questo paradosso si basa su un esperimento con quattro giochi differenti che i soggetti del test devono selezionare. La seguente tabella 3.3 mostra i quattro giochi (A,B,C,D).

Allais riscontrò che per i 72 intervistati del campione, l'82% sceglieva B nella prima lotteria e l'83% C nella seconda lotteria. Tale risultato viola l'assioma di indipendenza dell'utilità attesa, poiché se il gioco B è preferito al gioco A allora per lo stesso agente il gioco D deve essere preferito a C.

Formalmente, negli esperimenti la maggioranza degli intervistati seleziona i giochi come segue: $B \succ A$ e $C \succ D$.

Probability	Game A	Game B	Game C	Game D
0.66	2,400	2,400	0	0
0.33	2,500	2,400	2,500	2,400
0.01	0	2,400	0	2,400

Tabella 3.3: Giochi nel paradosso di Allais

La preferenza $B \succ A$ conduce alle seguenti disuguaglianze, dove $u1 \equiv u(2400)$, $u2 \equiv u(2500)$, $u3 \equiv u(0)$:

$$u1 > 0.66 \cdot u1 + 0.33 \cdot u2 + 0.01 \cdot u3$$

$$0.34 \cdot u1 > 0.33 \cdot u2 + 0.01 \cdot u3$$

La preferenza $C \succ D$ a sua volta porta alle seguenti disuguaglianze:

$$0.33 \cdot u2 + 0.01 \cdot u3 > 0.33 \cdot u1 + 0.01 \cdot u1$$

$$0.34 \cdot u1 < 0.33 \cdot u2 + 0.01 \cdot u3$$

Queste disuguaglianze sono ovviamente in contraddizione tra loro e portano al paradosso di Allais.

Gli individui scelgono l'alternativa B certa nonostante A incerto abbia un valore atteso maggiore mostrando la preferenza per la certezza degli individui avversi al rischio.

Nella seconda coppia di prospettive l'individuo, avendo una probabilità decisamente bassa di avere un premio, preferisce un premio più alto con una probabilità minore rispetto ad un premio più basso con una probabilità maggiore, e dà quindi più credito al premio che alle probabilità. Una possibile spiegazione è che le persone in generale apprezzano la certezza più di quanto i modelli tipici, come l'EUT, prevedano. La maggior parte delle persone probabilmente preferirebbe ricevere 1 milione con certezza piuttosto che giocare a un gioco in cui si possono vincere 100 milioni con una probabilità del 5%, sebbene ci siano numerose funzioni di utilità adatte che in EUT farebbero scegliere all'individuo il gioco invece dell'importo certo. Un'altra possibile spiegazione risiede nell'inquadramento delle decisioni e nella psicologia. È risaputo che un maggior numero di persone accetterebbe un intervento chirurgico se ha il 95% di probabilità di successo piuttosto che il 5% di probabilità di morte. Cambiando semplicemente la formulazione potrebbe portare a un comportamento che non è coerente con le teorie decisionali come l'EUT.

Un altro famoso paradosso che affronta le carenze dell'EUT nella sua forma soggettiva, Savage (1954, 1972), è il paradosso di Ellsberg (1961). Si tratta dell'importanza dell'ambiguità in molte situazioni decisionali nel mondo reale. Un'ambientazione standard per questo paradosso comprende due urne diverse, entrambe contenenti esattamente 100 palline. Si sa che l'urna 1 contiene esattamente 50 palline nere e 50 rosse. Dell'urna 2 si sa solamente che contiene palline nere e rosse ma non in quale proporzione. I soggetti del test possono scegliere tra le seguenti opzioni di gioco:

- Scelta 1: red 1, black 1, o indifferente
- Scelta 2: red 2, black 2 o indifferente
- Scelta 3: red 1, red 1, o indifferente
- Scelta 4: black 1, black 2 o indifferente

In questo caso, “red 1”, ad esempio, significa che dall’urna 1 viene estratta una pallina rossa. In genere, un soggetto risponde come segue:

- Scelta 1: indifferente
- Scelta 2: indifferente
- Scelta 3: red 1
- Scelta 4: black 1

Questa serie di decisioni, che non è l'unica ad essere osservata ma è comune, esemplifica la cosiddetta avversione all'ambiguità. Poiché le probabilità per le palline rosse e nere, rispettivamente, non sono note per l'urna 2, i soggetti preferiscono una situazione di rischio anziché di ambiguità.

I due paradossi di Allais e Ellsberg dimostrano che i soggetti reali si comportano molto spesso in maniera opposta da quanto previsto dalle teorie decisionali ben consolidate in economia. In altre parole, gli esseri umani in qualità di decisorи non possono essere paragonati a macchine che raccolgono attentamente i dati e poi snocciolano i numeri per prendere una decisione in condizioni di incertezza, sia essa sotto forma di rischio o di ambiguità.

Il comportamento umano è più complesso di quanto suggeriscono la maggior parte, se non tutte, le teorie attualmente in vigore.

Quanto possa essere difficile e complesso spiegare il comportamento umano è chiaro dopo aver letto, per esempio, le 800 pagine del libro “Behave” di

Sapolsky (2018). Copre molteplici sfaccettature di questo argomento, dai processi biochimici alla genetica, dall’evoluzione umana alle tribù, dal linguaggio alla religione e molto altro ancora.

Se i paradigmi economici decisionali standard, come l’EUT, non spiegano bene i processi decisionali del mondo reale, quali alternative sono disponibili? Gli esperimenti economici che costituiscono la base per i paradossi di Allais e Ellsberg sono un buon punto di partenza per capire come si comportano i decisori in situazioni specifiche e controllate. Tali esperimenti e i loro risultati, a volte sorprendenti e paradossali, hanno infatti motivato molti ricercatori a proporre teorie e modelli alternativi che risolvano i paradossi. Il libro “L’esperimento nella storia dell’economia” di Fontaine e Leonard (2005) tratta del ruolo storico degli esperimenti in economia. Esiste, ad esempio, tutta una serie di letteratura che affronta i problemi derivanti dal paradosso di Ellsberg. Tale letteratura si occupa, tra gli altri argomenti, delle probabilità non additive, di integrali di Choquet e di euristiche decisionali come la massimizzazione del payoff minimo (“max-min”) o la minimizzazione della perdita massima (“min-max”). Questi approcci alternativi si sono dimostrati superiori all’EUT, almeno in alcuni scenari decisionali. Ma sono ben lontani dall’essere mainstream in finanza.

Che cosa si è dimostrato utile nella pratica?

Non troppo sorprendentemente, la risposta sta nei dati e negli algoritmi di apprendimento automatico. Internet, con i suoi miliardi di utenti, genera un tesoro di dati che descrivono il comportamento umano nel mondo reale, o ciò che a volte viene chiamato revealed preferences (preferenze rilevate). I big data generati sul web hanno una scala di molti ordini di grandezza superiore a quella che i singoli esperimenti possono generare. Aziende come Amazon, Facebook, Google e Twitter sono in grado di guadagnare miliardi di dollari registrando il comportamento degli utenti (cioè le loro preferenze rivelate) e capitalizzando, sfruttando le intuizioni generate da algoritmi di ML addestrati su questi dati. L’approccio ML predefinito adottato in questo contesto è l’apprendimento supervisionato. Gli algoritmi stessi sono in genere privi di teorie e di modelli; spesso si applicano varianti di reti neurali. Pertanto, quando le aziende oggi prevedono il comportamento dei loro utenti o clienti, il più delle volte viene utilizzato un algoritmo model-free di ML. Le teorie decisionali tradizionali come EUT o uno dei suoi successori in genere non svolgono alcun ruolo.

Questo rende sorprendente che tali teorie siano ancora, all’inizio degli 2020, una pietra miliare della maggior parte delle teorie economiche e finanziarie applicate nella pratica. Senza contare il gran numero di testi di finanza che trattano le teorie decisionali tradizionali. Se uno degli elementi costitutivi più

fondamentali della teoria finanziaria sembra violare un significativo supporto empirico o vantaggi pratici, che dire dei modelli finanziari che si basano su di esso? Tale quesito verrà trattato nelle sezioni successive.

3.4.2 Previsioni del comportamento basati sui dati

Le teorie decisionali economiche standard sono intellettualmente attraenti per molti, anche per coloro che, di fronte a una decisione concreta in condizioni di incertezza, si comporterebbero in contrasto con le previsioni delle teorie. D’altro canto, i big data e gli approcci di apprendimento automatico model-free e supervisionato si dimostrano utili e di successo nella pratica per prevedere il comportamento di utenti e clienti. In un contesto finanziario, questo potrebbe significare che non ci si dovrebbe preoccupare del perché e del come gli agenti finanziari decidono. Ci si dovrebbe piuttosto concentrare sulle loro preferenze indirettamente rilevate sulla base dei “features data” (nuove informazioni) che descrivono lo stato di un mercato finanziario e labels data (risultati) che riflettono l’impatto delle decisioni prese dagli agenti finanziari. Ciò porta a una visione del processo decisionale nei mercati finanziari basata sui dati invece che su una teoria o su un modello.

3.4.3 Teoria della Media-Varianza del Portafoglio (MVP)

Supponiamo che un investitore, che compie azioni basandosi sui dati, voglia applicare la teoria MVP per investire un portafoglio di titoli tecnologici e voglia aggiungere un exchange traded fund (ETF) relativo all’oro per diversificare. Probabilmente l’investitore accede ai dati storici sui prezzi tramite un’API a una piattaforma di trading o a un fornitore di dati. Per rendere riproducibile la seguente analisi, si basa su un file di dati CSV archiviato in una posizione remota. Il seguente codice Python recupera tale file di dati, seleziona un certo numero di simboli in base all’obiettivo dell’investitore e calcola i rendimenti logaritmici dai dati delle serie temporali dei prezzi. La figura 3.2 confronta le serie temporali normalizzate dei prezzi dei *simboli*¹ selezionati:

```

1 url = 'http://hilpisch.com/aiif_eikon_eod_data.csv'
2 raw = pd.read_csv(url, index_col=0, parse_dates=True).dropna()
3
4 #recuperiamo di dati storici di EOD, da una postazione remota
5 raw.info()

```

¹Un simbolo azionario o ticker è una serie univoca di lettere assegnate a un titolo a scopo commerciale. I simboli sono solo un modo abbreviato per descrivere le azioni di una società; quindi, non c’è alcuna differenza significativa tra quelli che hanno tre lettere e quelli che ne hanno quattro o cinque.

```

6 DatetimeIndex: 2516 entries , 2010-01-04 to 2019-12-31
7 Data columns (total 12 columns):
8 #   Column   Non-Null Count   Dtype  
9 -- 
10 0   AAPL.O    2516 non-null    float64 
11 1   MSFT.O    2516 non-null    float64 
12 2   INTC.O    2516 non-null    float64 
13 3   AMZN.O    2516 non-null    float64 
14 4   GS.N     2516 non-null    float64 
15 5   SPY      2516 non-null    float64 
16 6   .SPX     2516 non-null    float64 
17 7   .VIX     2516 non-null    float64 
18 8   EUR=     2516 non-null    float64 
19 9   XAU=     2516 non-null    float64 
20 10  GDX      2516 non-null    float64 
21 11  GLD      2516 non-null    float64 

22
23 #specifichiamo i simboli (RIC) in cui investire
24 symbols = [ 'AAPL.O' , 'MSFT.O' , 'INTC.O' , 'AMZN.O' , 'GLD' ]
25
26 #calcolaliamo i rendimenti logaritmici per tutte le serie temporali
27 rets = np.log(raw[symbols] / raw[symbols].shift(1)).dropna()
28
29 #tracciamo il grafico delle serie temporali finanziarie
30 #normalizzate per i simboli selezionati
31 (raw[symbols] / raw[symbols].iloc[0]).plot(figsize=(10, 6));

```

L'investitore orientato ai dati vuole innanzitutto stabilire una base di riferimento per la performance, data da un portafoglio equamente ponderato per l'intero periodo dei dati disponibili. A tal fine, vengono definite le funzioni per calcolare il rendimento del portafoglio, la volatilità del portafoglio e lo Sharpe ratio del portafoglio, dato un insieme di pesi per i simboli selezionati:

```

1 port_return(rets, weights)
2 0.156947646530181
3
4 port_volatility(rets, weights)
5 0.16106507848480675
6
7 port_sharpe(rets, weights)
8 0.9744362217225496

```

L'investitore vuole anche analizzare quali combinazioni di rischio e rendimento del portafoglio – e di conseguenza lo Sharpe ratio – sono approssimativamente possibili applicando la simulazione Monte Carlo per randomizzare



Figura 3.2: Serie temporali finanziarie normalizzate

i pesi del portafoglio. Sono escluse le vendite a breve termine (allo scoperto) e si presume che i pesi del portafoglio siano pari al 100%.

Implementiamo la simulazione e visualizziamo i risultati (vedi figura 3.3).

L'investitore guidato dai dati vuole effettuare un backtest della performance di un portafoglio creato all'inizio del 2011. La composizione ottimale del portafoglio è stata ricavata dai dati delle serie temporali finanziarie disponibili dal 2010. All'inizio del 2012, la composizione del portafoglio è stata adeguata in base ai dati disponibili del 2011, e così via. A tal fine, ricaviamo i *pesi del portafoglio per ogni anno rilevante che massimizziamo lo Sharpe ratio*:

```

1 opt_weights
2 {2010: array([ 0.366,  0.000,  0.000,  0.056,  0.578]), 
3  2011: array([ 0.543,  0.000,  0.077,  0.000,  0.380]), 
4  2012: array([ 0.324,  0.000,  0.000,  0.471,  0.205]), 
5  2013: array([ 0.012,  0.305,  0.219,  0.464,  0.000]), 
6  2014: array([ 0.452,  0.115,  0.419,  0.000,  0.015]), 
7  2015: array([ 0.000,  0.000,  0.000,  1.000,  0.000]), 
8  2016: array([ 0.150,  0.260,  0.000,  0.058,  0.533]), 
9  2017: array([ 0.231,  0.203,  0.031,  0.109,  0.426]), 
10 2018: array([ 0.000,  0.295,  0.000,  0.705,  0.000])}
```

Le composizioni ottimali di portafoglio ricavate per gli anni in questione illustrano che la teoria dell'MVP nella sua forma originale porta molto spesso a situazioni (relativamente) estreme, nel senso che uno o più assets non ven-

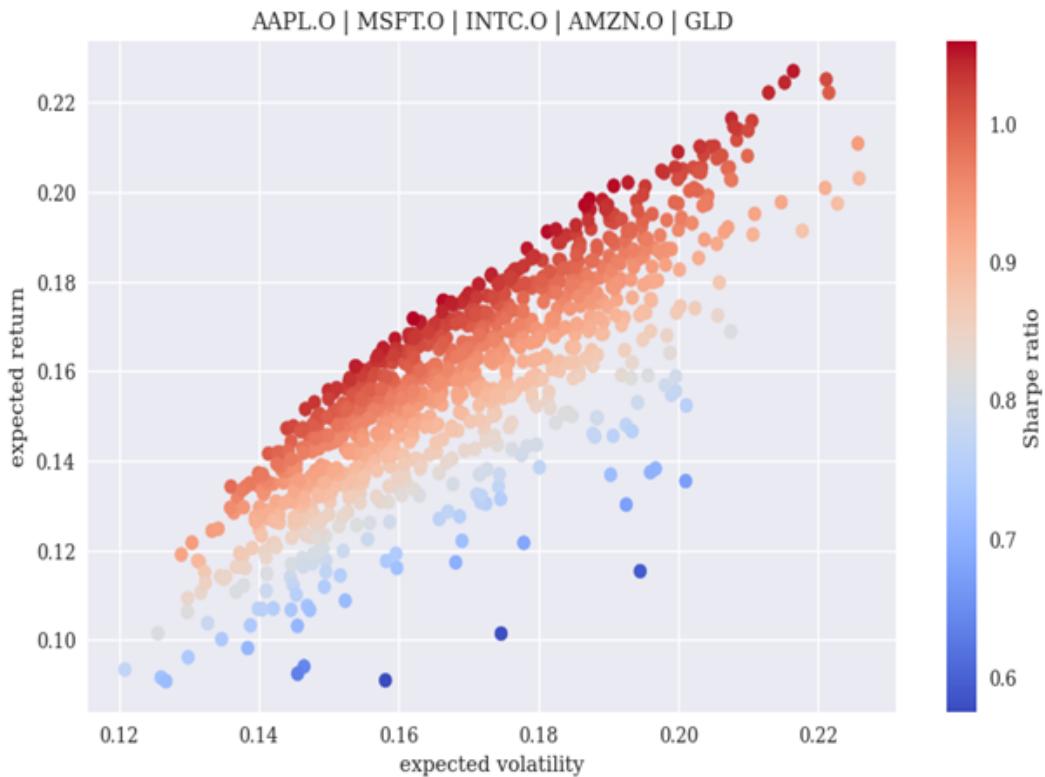


Figura 3.3: Volatilità, rendimenti e Sharpe ratios simulati del portafoglio

gono inclusi affatto o che addirittura un singolo asset costituisca il 100% del portafoglio. Naturalmente, ciò può essere evitato attivamente stabilendo, per esempio, un peso minimo per ogni asset considerato. I risultati indicano che questo approccio porta a ribilanciamenti significativi del portafoglio, guidati dalle statistiche e dalle correlazioni realizzate nell'anno precedente. Per completare il backtest, a seguito confrontiamo le statistiche di portafoglio attese (dalla composizione ottimale dell'anno precedente applicata ai dati dell'anno precedente) con le statistiche di portafoglio realizzate per l'anno in corso (dalla composizione ottimale dell'anno precedente applicata ai dati dell'anno in corso):

	res	epv	epr	esr	rpv	rpr
	rsr					
3	2011	0.157440	0.303003	1.924564	0.160622	0.133836
4	2012	0.173279	0.169321	0.977156	0.182292	0.161375
5	2013	0.202460	0.278459	1.375378	0.168714	0.166897
6	2014	0.181544	0.368961	2.032353	0.197798	0.026830

```

7 2015  0.160340  0.309486  1.930190  0.211368 -0.024560 -0.116194
8 2016  0.326730  0.778330  2.382179  0.296565  0.103870  0.350242
9 2017  0.106148  0.090933  0.856663  0.079521  0.230630  2.900235
10 2018  0.086548  0.260702  3.012226  0.157337  0.038234  0.243004
11 2019  0.323796  0.228008  0.704174  0.207672  0.275819  1.328147
12
13 res.mean()
14 epv      0.190920
15 epr      0.309689
16 esr      1.688320
17 rpv      0.184654
18 rpr      0.123659
19 rsr      0.838755

```

La figura 3.4 confronta le volatilità attese e realizzate del portafoglio per i singoli anni. La teoria MVP fa un buon lavoro nel prevedere la volatilità del portafoglio. Ciò è supportato anche da una correlazione relativamente alta tra le due serie temporali:

```

1 res[["epv", "rpv"]].corr()
2           epv      rpv
3 epv    1.000000  0.765733
4 rpv    0.765733  1.000000

```

Tuttavia, le conclusioni sono opposte quando si confrontano i rendimenti di portafoglio attesi con i rendimenti di realizzati (vedi Figura 3.5). La teoria MVP ovviamente non riesce a prevedere i rendimenti del portafoglio come è confermato dalla correlazione negativa tra le due serie temporali:

```

1 res[["epr", "rpr"]].corr()
2           epr      rpr
3 epr    1.000000 -0.350437
4 rpr   -0.350437  1.000000

```



Figura 3.4: Volatilità attesa e volatilità realizzata del portafoglio



Figura 3.5: Rendimenti di portafoglio attesi e realizzati

Conclusioni simili, o addirittura peggiori, devono essere tratte per quanto riguarda lo Sharpe ratio (vedi Figura 3.6). Per l'investitore data-driven (guidato dai dati) che mira a massimizzare lo Sharpe ratio del portafoglio, le previsioni della teoria sono generalmente molto lontane dai valori realizzati. La correlazione tra le due serie temporali è persino inferiore a quella dei rendimenti:

```

1 res[["esr", "rsr"]].corr()
2             esr      rsr
3 esr    1.000000 -0.698607
4 rsr   -0.698607  1.000000

```



Figura 3.6: Sharpe ratios di portafoglio previsti e realizzati

Il potere predittivo della teoria MVP

La teoria dell'MVP applicata ai dati del mondo reale rivela le sue carenze pratiche. Senza ulteriori vincoli, le composizioni e i ribilanciamenti ottimali del portafoglio possono essere estremi. Il potere predittivo di rendimento del portafoglio e dello Sharpe ratio è piuttosto negativo nell'esempio numerico, mentre il potere predittivo rispetto al rischio del portafoglio sembra accettabile.

Tuttavia, gli investitori sono generalmente interessati a misure di performance corrette per il rischio, come lo Sharpe ratio, e questa è la statistica per la quale la teoria MVP fallisce di più nell'esempio.

3.4.4 Capital Asset Pricing Model

Un approccio simile può essere applicato per mettere alla prova il CAPM nel mondo reale. Supponiamo che l'investitore tecnologico guidato dai dati di prima voglia applicare il CAPM per ricavare i rendimenti attesi per i quattro titoli tecnologici di prima. Il seguente codice Python ricava innanzitutto il beta per ogni titolo per un determinato anno, quindi calcola il rendimento atteso per il titolo nell'anno successivo, dato il suo beta e la performance del portafoglio di mercato. Il portafoglio di mercato è approssimato dall'indice azionario S&P 500:

```

1 #specifica il tasso a breve senza rischio
2 r = 0.005
3
4 #definisce il portafoglio di mercato
5 market = '.SPX'
6
7 AAPL.O
8 =====
9 2011 | beta: 1.052 | mu_capm: -0.000 | mu_real: 0.228
10 2012 | beta: 0.764 | mu_capm: 0.098 | mu_real: 0.275
11 2013 | beta: 1.266 | mu_capm: 0.327 | mu_real: 0.053
12 2014 | beta: 0.630 | mu_capm: 0.070 | mu_real: 0.320
13 2015 | beta: 0.833 | mu_capm: -0.005 | mu_real: -0.047
14 2016 | beta: 1.144 | mu_capm: 0.103 | mu_real: 0.096
15 2017 | beta: 1.009 | mu_capm: 0.180 | mu_real: 0.381
16 2018 | beta: 1.379 | mu_capm: -0.091 | mu_real: -0.071
17 2019 | beta: 1.252 | mu_capm: 0.316 | mu_real: 0.621
18
19 MSFT.O
20 =====
21 2011 | beta: 0.890 | mu_capm: 0.001 | mu_real: -0.072
22 2012 | beta: 0.816 | mu_capm: 0.104 | mu_real: 0.029
23 2013 | beta: 1.109 | mu_capm: 0.287 | mu_real: 0.337
24 2014 | beta: 0.876 | mu_capm: 0.095 | mu_real: 0.216
25 2015 | beta: 0.955 | mu_capm: -0.007 | mu_real: 0.178
26 2016 | beta: 1.249 | mu_capm: 0.113 | mu_real: 0.113
27 2017 | beta: 1.224 | mu_capm: 0.217 | mu_real: 0.321
28 2018 | beta: 1.303 | mu_capm: -0.086 | mu_real: 0.172

```

```

29 2019 | beta: 1.442 | mu_capm: 0.364 | mu_real: 0.440
30
31 INTC.O
32 =====
33 2011 | beta: 1.081 | mu_capm: -0.000 | mu_real: 0.142
34 2012 | beta: 0.842 | mu_capm: 0.108 | mu_real: -0.163
35 2013 | beta: 1.081 | mu_capm: 0.280 | mu_real: 0.230
36 2014 | beta: 0.883 | mu_capm: 0.096 | mu_real: 0.335
37 2015 | beta: 1.055 | mu_capm: -0.008 | mu_real: -0.052
38 2016 | beta: 1.009 | mu_capm: 0.092 | mu_real: 0.051
39 2017 | beta: 1.261 | mu_capm: 0.223 | mu_real: 0.242
40 2018 | beta: 1.163 | mu_capm: -0.076 | mu_real: 0.017
41 2019 | beta: 1.376 | mu_capm: 0.347 | mu_real: 0.243
42
43 AMZN.O
44 =====
45 2011 | beta: 1.102 | mu_capm: -0.001 | mu_real: -0.039
46 2012 | beta: 0.958 | mu_capm: 0.122 | mu_real: 0.374
47 2013 | beta: 1.116 | mu_capm: 0.289 | mu_real: 0.464
48 2014 | beta: 1.262 | mu_capm: 0.135 | mu_real: -0.251
49 2015 | beta: 1.473 | mu_capm: -0.013 | mu_real: 0.778
50 2016 | beta: 1.122 | mu_capm: 0.102 | mu_real: 0.104
51 2017 | beta: 1.118 | mu_capm: 0.199 | mu_real: 0.446
52 2018 | beta: 1.300 | mu_capm: -0.086 | mu_real: 0.251
53 2019 | beta: 1.619 | mu_capm: 0.408 | mu_real: 0.207

```

La figura 3.7 confronta il rendimento previsto (atteso) per un singolo titolo, dato il beta dell'anno precedente e la performance del portafoglio di mercato dell'anno in corso, con il rendimento realizzato del titolo per l'anno in corso. Ovviamente, il CAPM nella sua forma originale non si dimostra molto utile nel prevedere la performance di un'azione basandosi solo sul beta:

```

1 sym = 'AMZN.O'
2 res [ res [ 'symbol' ] == sym ]. corr ()
3
4 mu_capm    mu_real
5 mu_capm  1.000000 -0.004826
6 mu_real   -0.004826  1.000000

```



Figura 3.7: Rendimenti azionari previsti dal CAPM rispetto a quelli realizzati per un singolo titolo.

La figura 3.8 confronta le medie dei rendimenti azionari previsti dal CAPM con le medie dei rendimenti realizzati. Anche in questo caso, il CAPM non fa un buon lavoro. È facile che le previsioni del CAPM non variano molto in media per i titoli analizzati; sono comprese tra il 12.2% e il 14.4%. Tuttavia, i rendimenti medi realizzati dei titoli mostrano un'elevata variabilità; questi sono compresi tra il 9.4% e il 29.2%. La performance del portafoglio di mercato e il beta da soli non sono ovviamente in grado di spiegare i rendimenti osservati dei titoli (tecnologici):

	μ_{capm}	μ_{real}
symbol		
AAPL.O	0.110855	0.206158
AMZN.O	0.128223	0.259395
INTC.O	0.117929	0.116180
MSFT.O	0.120844	0.192655

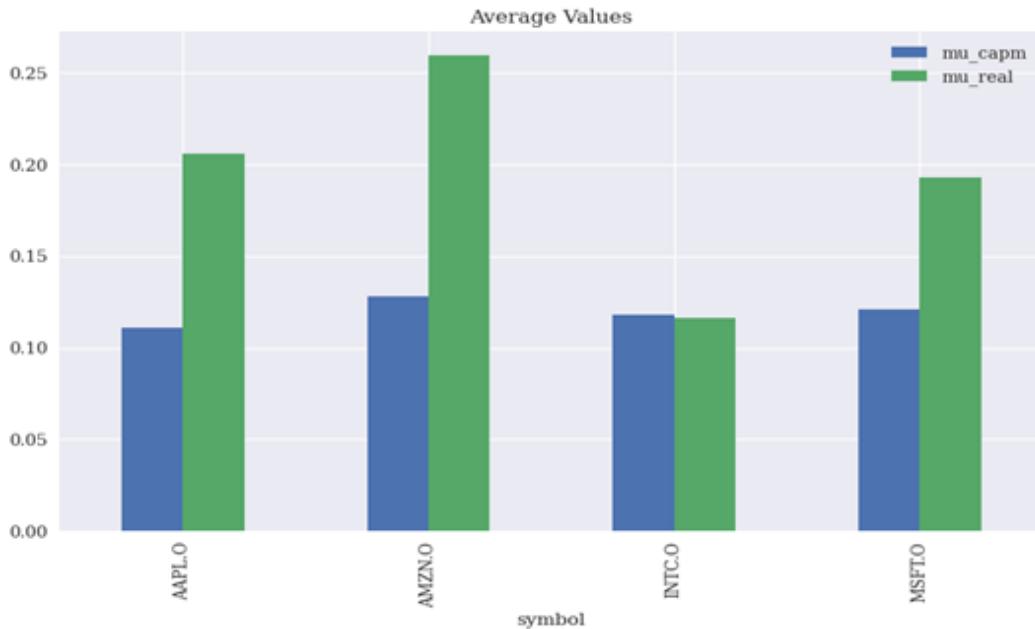


Figura 3.8: Rendimenti azionari medi previsti dal CAPM rispetto a quelli medi realizzati per più azioni

Il potere predittivo del CAPM

Il potere predittivo del CAPM per quanto riguarda la performance futura dei titoli, rispetto al portafoglio di mercato, è piuttosto basso o addirittura inesistente per alcuni titoli. Una delle ragioni è probabilmente il fatto che il CAPM si basa sugli stessi assunti (presupposti) centrali della teoria del MVP, ovvero che gli investitori si preoccupano solo del rendimento (atteso) e della volatilità (attesa) di un portafoglio e/o di un titolo.

3.5 Sfatare gli assunti centrali

La sezione precedente ha fornito una serie di esempi numerici e reali che mostrano come le teorie finanziarie normative più diffuse possano fallire nella pratica. Questa sezione sostiene che una delle ragioni principali è che gli assunti centrali di queste teorie finanziarie popolari non sono validi; cioè non descrivono la realtà dei mercati finanziari. I due assunti analizzati sono i *rendimenti normalmente distribuiti* e le *relazioni lineari*.

3.5.1 Rendimenti a distribuzione normale

Di fatto, solo una distribuzione normale è completamente specificata attraverso il suo primo (aspettativa) e secondo momento (deviazione standard).

Insieme di dati di esempio

A titolo illustrativo, si consideri un insieme di numeri generato casualmente normalmente distribuiti (distribuzione normale standard varianza 1, valore atteso 0). La Figura 3.9 mostra la tipica forma a campana dell'istogramma risultante:

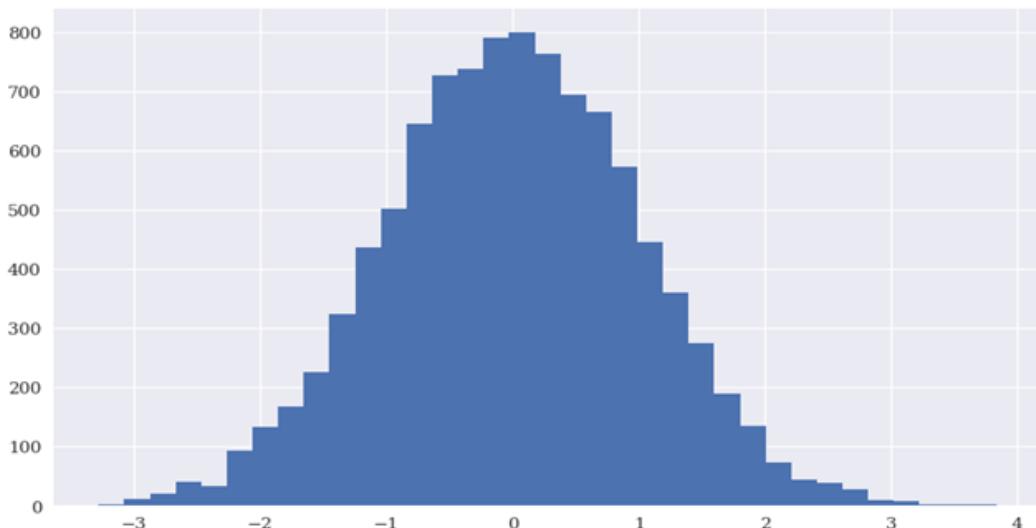


Figura 3.9: Numeri casuali normalmente distribuiti

Ora consideriamo un insieme di numeri casuali che condividono gli stessi valori del primo e del secondo momento, ma che hanno una distribuzione completamente diversa da quella illustrata dalla figura 3.9 (vedi figura 3.10). Sebbene i momenti siano gli stessi, questa distribuzione è composta solo da tre valori discreti:

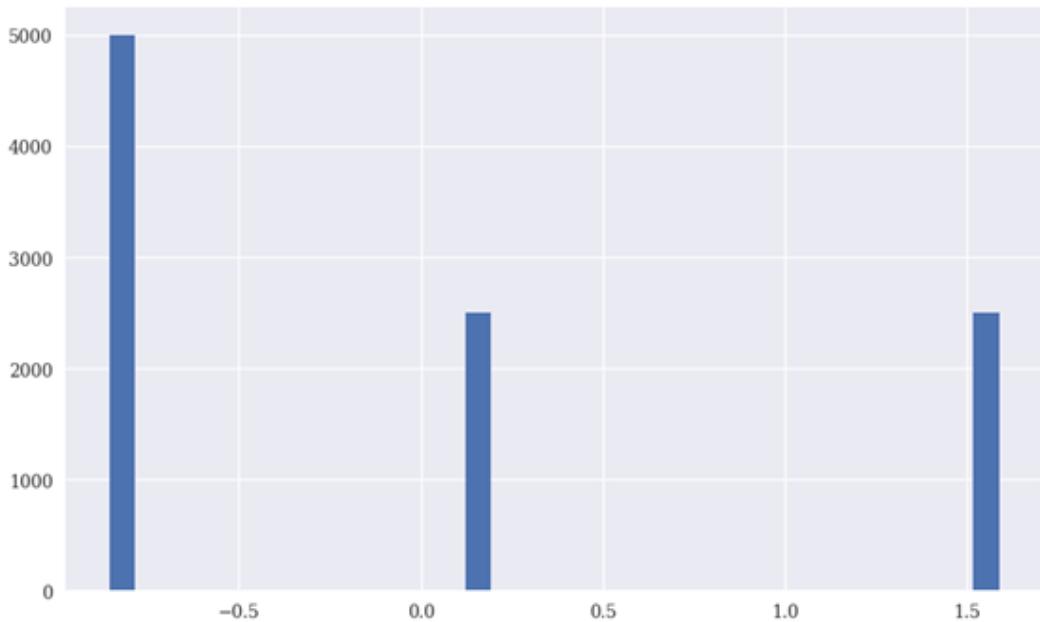


Figura 3.10: Distribuzione con primo e secondo momento a 0.0 e 1.0, rispettivamente

Primo e secondo momento: Il primo e il secondo momento di una distribuzione di probabilità descrivono completamente solo una distribuzione normale. Esistono infinite altre distribuzioni che potrebbero condividere i primi due momenti con una distribuzione normale pur essendo completamente diverse. In preparazione a un test sui rendimenti finanziari reali, usiamo funzioni in Python che consentono di visualizzare i dati in un istogramma e di aggiungere la *funzione di densità di probabilità (PDF)* di una distribuzione normale con i primi due momenti dei dati (valore atteso e deviazione standard). La figura 3.11 mostra quanto l'istogramma si approssimi alla PDF per i numeri casuali distribuiti normalmente:

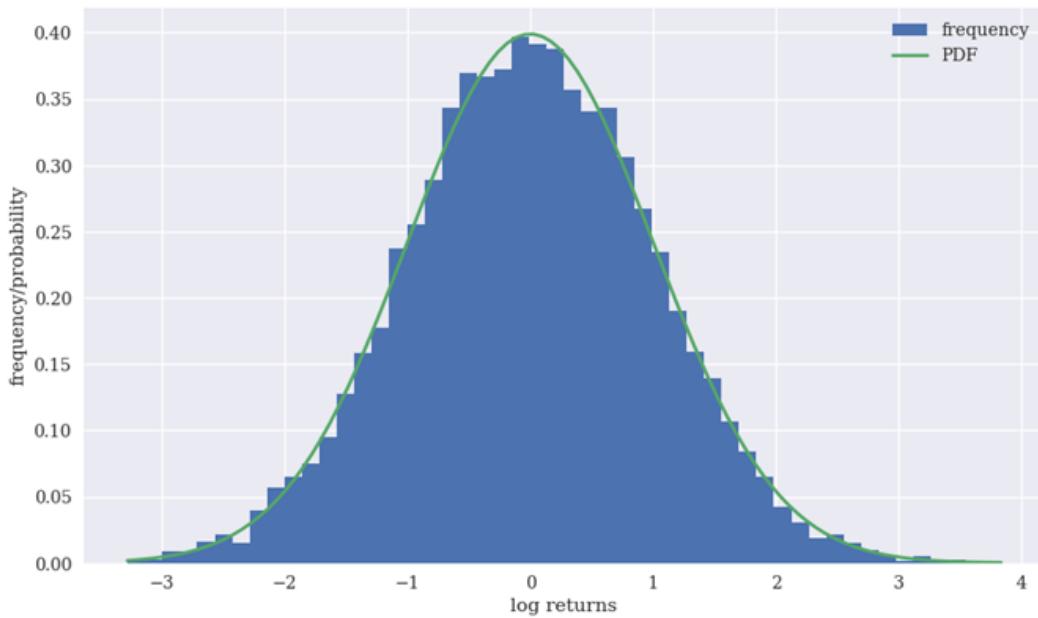


Figura 3.11: Istogramma e PDF per numeri distribuiti normalmente

La figura 3.12 di contro, illustra come la PDF della distribuzione normale non ha niente che vedere con i dati mostrati come istogramma:

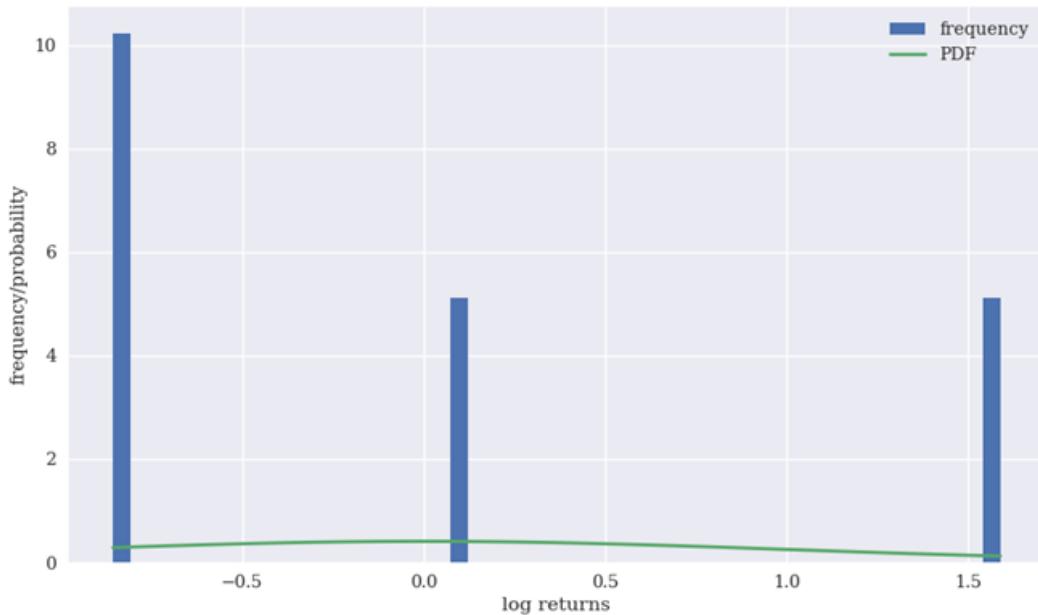


Figura 3.12: Istogramma e PDF normale per numeri discreti

Un altro modo per confrontare una distribuzione normale con i dati è il grafico Quantile-Quantile (Q-Q). Come mostra la figura 3.13, per i numeri normalmente distribuiti, i numeri stessi giacciono (per lo più) su una linea retta nel piano Q-Q:

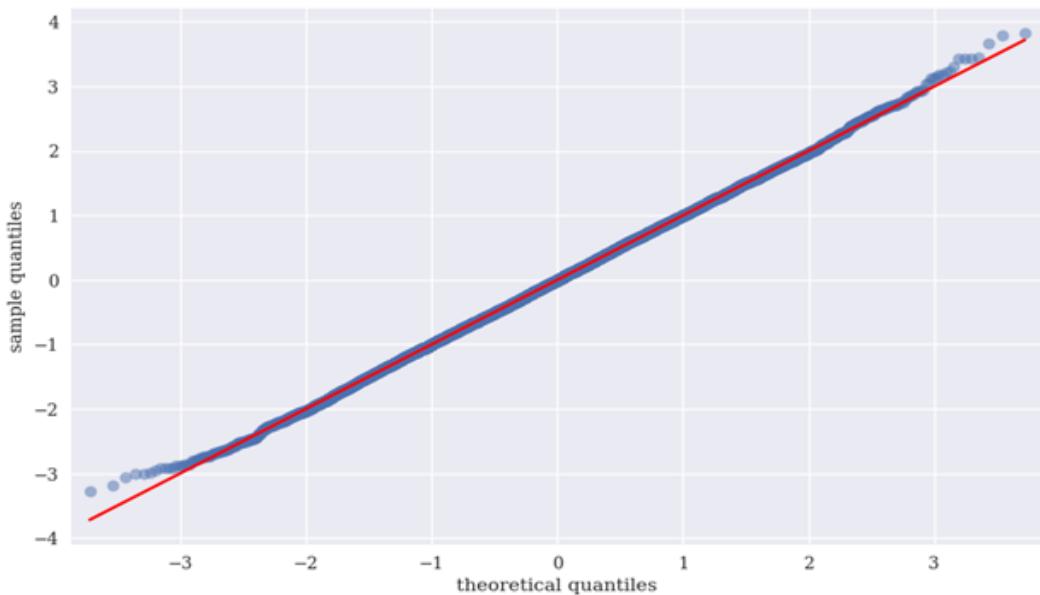


Figura 3.13: Grafico Q-Q per numeri con distribuzione normale standard

Anche in questo caso, il diagramma Q-Q mostrato nella figura 3.14 per i numeri discreti appare completamente diverso da quello della figura 3.13:

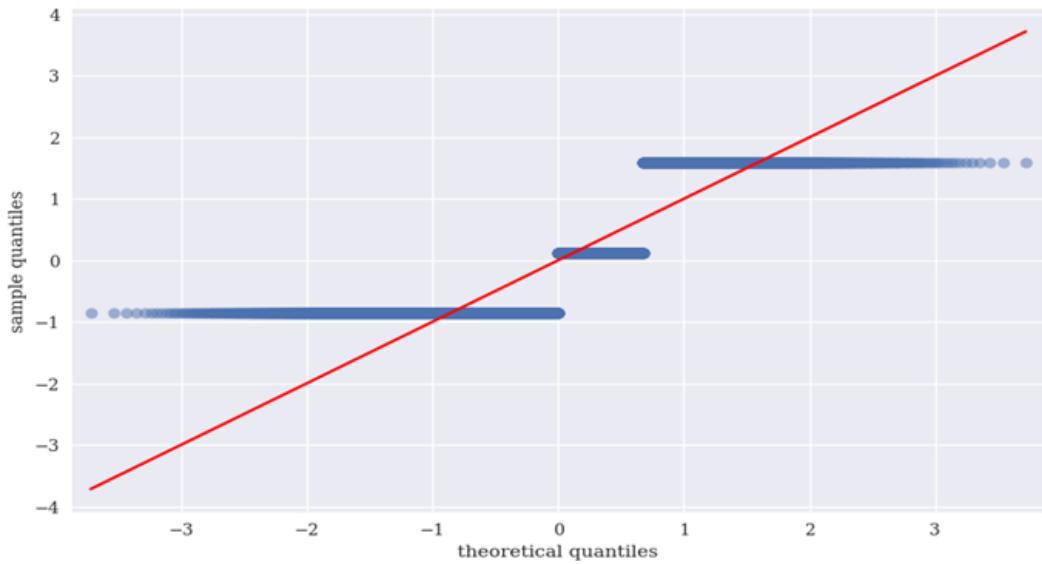


Figura 3.14: Grafico Q-Q per numeri discreti

Infine, si possono utilizzare i test statistici per verificare se un insieme di numeri è normalmente distribuito o meno. Di seguito implementiamo tre test:

- Test per la skew normale.
- Test per la curtosi normale.
- Test per la normale skew e curtosi combinate.

Un p-value inferiore a 0.05 è generalmente considerato un conto indicatore di normalità; in altre parole, l'ipotesi che i numeri siano distribuiti normalmente viene rifiutata. In questo senso, come nelle figure precedenti, i p-value per le due serie di dati parlano da soli:

```

1 RETURN SAMPLE STATISTICS
2
3 Skew of Sample Log Returns  0.016793
4 Skew Normal Test p-value   0.492685
5
6 Kurt of Sample Log Returns -0.024540
7 Kurt Normal Test p-value    0.637637
8
9 Normal Test p-value        0.707334
10
11

```

```

12
13 print_statistics(numbers)
14 RETURN SAMPLE STATISTICS
15 -----
16 Skew of Sample Log Returns 0.689254
17 Skew Normal Test p-value 0.000000
18 -----
19 Kurt of Sample Log Returns -1.141902
20 Kurt Normal Test p-value 0.000000
21 -----
22 Normal Test p-value 0.000000
23 -----

```

Rendimenti finanziari reali

Recuperiamo i dati EOD da una fonte remota, come fatto in precedenza nel capitolo, e calcola i rendimenti logaritmici per tutte le serie temporali finanziarie contenute nel dataset. La figura 3.15 mostra che i rendimenti logaritmici dell'indice azionario S&P 500, rappresentati come istogramma, presentano un picco molto più alto e code più piene rispetto alla normale PDF con il valore atteso e la deviazione standard del campione. Queste due intuizioni sono *fatti stilizzati*² perché possono essere osservati in modo coerente per diversi strumenti finanziari:

²Nelle scienze sociali , specialmente in economia, un fatto stilizzato è una rappresentazione semplificata di un risultato empirico. Un fatto empirico è spesso un'ampia generalizzazione che riassume dati che, sebbene generalmente veri, possono rivelarsi imprecisi nei dettagli.

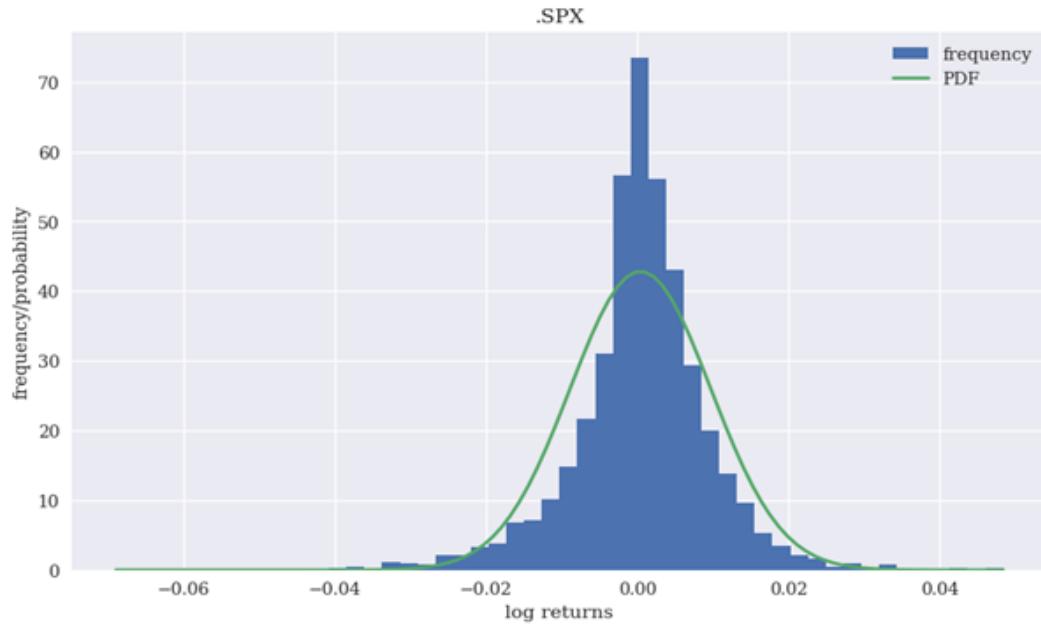


Figura 3.15: Distribuzione della frequenza e PDF normale per i rendimenti logaritmici dello S&P 500.

Si possono trarre evidenze simili considerando il grafico Q-Q per i rendimenti logaritmici dello S&P 500 nella figura 3.16. In particolare, il grafico Q-Q visualizza abbastanza bene le code “grasse” (punti al di sotto della linea retta a sinistra e al di sopra della linea retta a destra):

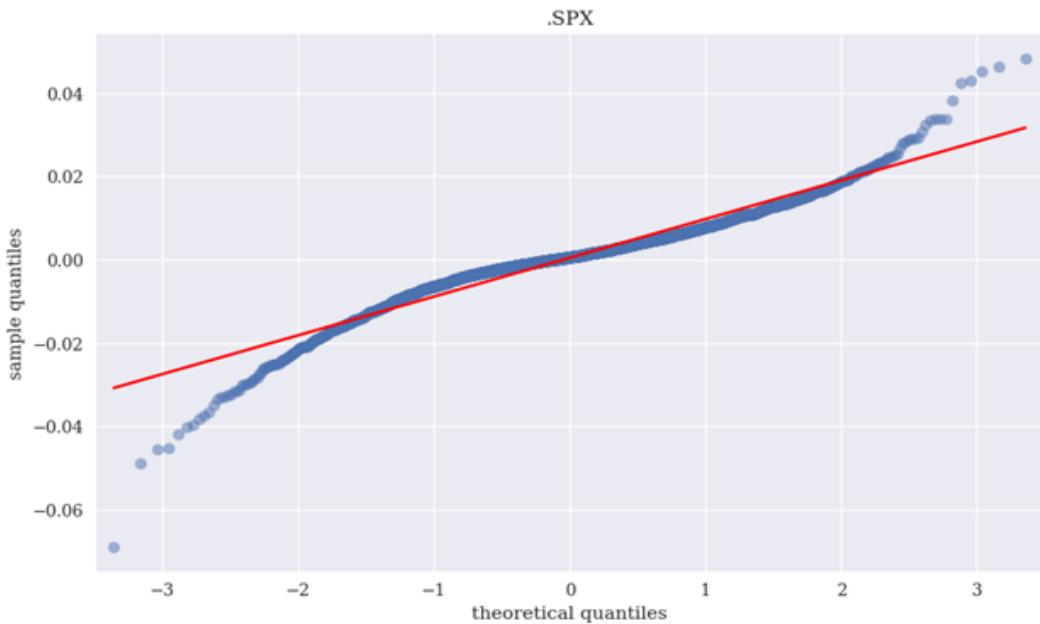


Figura 3.16: Grafico Q-Q per i rendimenti logaritmici dello S&P 500.

L’output del codice Python è il risultato dei test statistici sulla normalità dei rendimenti finanziari reali per una selezione di serie temporali finanziarie dall’insieme dei dati. I rendimenti finanziari reali non superano regolarmente tali test. Pertanto, è lecito concludere che l’assunto di normalità dei rendimenti finanziari difficilmente, se non per nulla, descrive la realtà finanziaria:

```

1 .SPX
2 =====
3 RETURN SAMPLE STATISTICS
4
5 Skew of Sample Log Returns -0.497160
6 Skew Normal Test p-value 0.000000
7
8 Kurt of Sample Log Returns 4.598167
9 Kurt Normal Test p-value 0.000000
10
11 Normal Test p-value 0.000000
12
13
14 AMZN.O
15 =====
16 RETURN SAMPLE STATISTICS
17

```

```
18 Skew of Sample Log Returns 0.135268
19 Skew Normal Test p-value 0.005689
20 -----
21 Kurt of Sample Log Returns 7.344837
22 Kurt Normal Test p-value 0.000000
23 -----
24 Normal Test p-value 0.000000
25 -----
26
27 EUR=
28 -----
29 RETURN SAMPLE STATISTICS
30 -----
31 Skew of Sample Log Returns -0.053959
32 Skew Normal Test p-value 0.268203
33 -----
34 Kurt of Sample Log Returns 1.780899
35 Kurt Normal Test p-value 0.000000
36 -----
37 Normal Test p-value 0.000000
38 -----
39
40 GLD
41 -----
42 RETURN SAMPLE STATISTICS
43 -----
44 Skew of Sample Log Returns -0.581025
45 Skew Normal Test p-value 0.000000
46 -----
47 Kurt of Sample Log Returns 5.899701
48 Kurt Normal Test p-value 0.000000
49 -----
50 Normal Test p-value 0.000000
51 -----
```

Assunzione di normalità

Sebbene l'assunto di normalità sia una buona approssimazione per molti fenomeni del mondo reale, come ad esempio in fisica, non è appropriata e può addirittura essere pericolosa quando si tratta di rendimenti finanziari. Quasi nessun campione di dati sui rendimenti finanziari supera i test di normalità statistica.

Oltre al fatto che si è dimostrata utile in altri ambiti, uno dei motivi principali per cui questa ipotesi si trova in così tanti modelli finanziari è che porta a modelli matematici, calcoli e prove eleganti e relativamente semplici.

3.5.2 Relazioni Lineari

Analogamente alla “onnipresenza” dell’assunzione di normalità nei modelli e nelle teorie finanziarie, le relazioni lineari tra le variabili sembrano essere un altro punto di riferimento molto diffuso. Questa sottosezione ne considera un importante, ovvero la relazione lineare assunta nel CAPM tra il beta di un titolo e il suo rendimento atteso (realizzato). In generale, quanto più alto è il beta, tanto più alto sarà il rendimento atteso in presenza di una performance di mercato positiva, in modo fisso e proporzionale al valore del beta stesso. Ricordiamo il calcolo dei beta, dei rendimenti attesi CAPM e dei rendimenti realizzati per una selezione di titoli tecnologici dalla sezione precedente. Questa volta, i valori beta vengono aggiunti anche all’oggetto DataFrame dei risultati.

La seguente analisi calcola il R^2 per la regressione lineare in cui il beta è la variabile indipendente e il *rendimento CAPM atteso*, data la performance del portafoglio di mercato, è la variabile dipendente. R^2 si riferisce al *coefficiente di determinazione* e misura la performance di un modello rispetto a un previsore di base sotto forma di semplice valore medio. La regressione lineare può solo spiegare circa il 10%, della variabilità nel rendimento CAPM atteso, un valore piuttosto basso, confermato anche dalla figura 3.17:

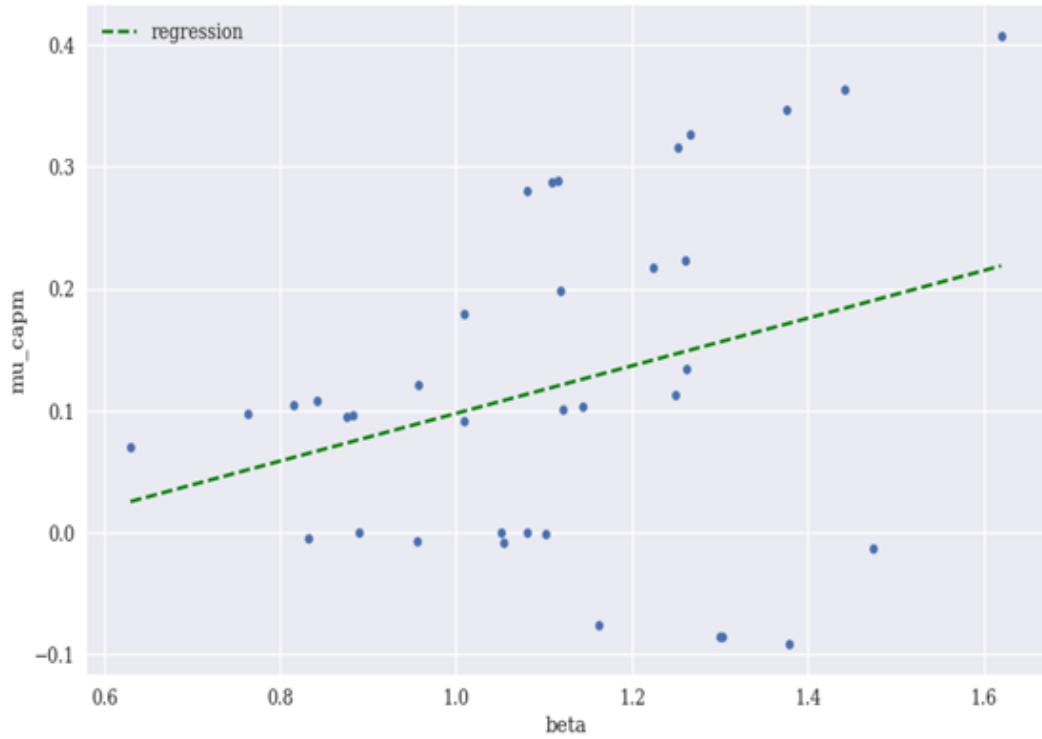


Figura 3.17: Rendimento atteso del CAPM rispetto al beta (inclusa la regressione lineare)

Per il rendimento realizzato, il potere esplicativo della regressione lineare è ancora più basso, con circa il 4.5% (vedi figura 3.18). Le regressioni lineari recuperano la relazione positiva tra beta e rendimenti azionari – “più alto è il beta, più alto è il rendimento data la performance (positiva) del portafoglio di mercato” – come indicato dalla pendenza positiva delle linee di regressione. Tuttavia, esse spiegano solo una piccola parte della variabilità complessiva osservata nei rendimenti azionari:

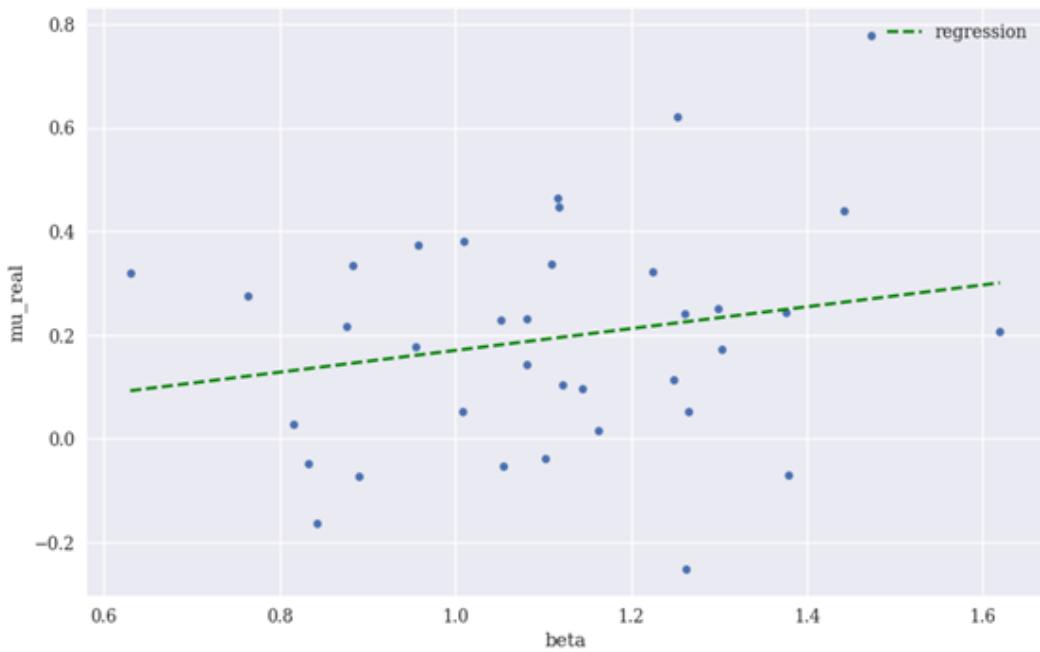


Figura 3.18: Rendimento atteso del CAPM rispetto al beta (inclusa la regressione lineare)

Come per le assunzioni di normalità, nel mondo fisico si possono spesso osservare relazioni lineari. Tuttavia, in finanza non esistono quasi mai casi in cui le variabili dipendono l'una dall'altra in modo chiaramente lineare. Da un punto di vista modellistico, le relazioni lineari portano, come l'assunto di normalità, a modelli matematici, calcoli e prove eleganti e relativamente semplici. Inoltre, lo strumento standard dell'econometria finanziaria, la regressione OLS, è adatto a trattare relazioni lineari nei dati. Queste sono le ragioni principali per cui la normalità e la linearità sono spesso scelte deliberatamente come elementi costitutivi dei modelli e delle teorie finanziarie.

3.6 Conclusioni

La scienza è stata guidata per secoli dalla generazione e dall'analisi rigorosa dei dati. Tuttavia, la finanza è stata caratterizzata da teorie normative basate su modelli matematici semplificati dei mercati finanziari, basati su assunzioni quali la normalità dei rendimenti e le relazioni lineari. La disponibilità quasi universale e completa di dati (finanziari) ha portato a uno spostamento dell'attenzione dall'approccio teorico alla finanza basata sui dati. Diversi esempi basati su dati finanziari reali illustrano che molti modelli e teorie finanziarie

popolari non possono sopravvivere al confronto con la realtà dei mercati finanziari. Per quanto eleganti, potrebbero essere troppo semplicistici per cogliere le complessità, la natura mutevole e la non linearità dei mercati finanziari.

Capitolo 4

Finanza AI-First

“Un calcolo computazionale prende le informazioni e le trasforma, implementando ciò che i matematici chiamano una funzione... Se siete in possesso di una funzione che prende in ingresso tutti i dati finanziari del mondo e produce le migliori azioni da acquistare, sarete presto estremamente ricchi.”

— Max Tegmark (2017)

Questo capitolo si propone di combinare la finanza guidata dai dati con l'apprendimento automatico (ML), vengono utilizzate le reti neurali per scoprire le inefficienze statistiche. In “Mercati Efficienti” (Efficient Markets) si discute *l'ipotesi di mercato efficiente* e si utilizza la regressione OLS (con il metodo dei minimi quadrati) per illustrarla sulla base dei dati delle serie temporali finanziarie. La “Previsione di mercato basata sui dati dei rendimenti” (Market Prediction Based on Returns Data) applica le reti neurali, assieme alla regressione OLS, per prevedere la direzione futura del prezzo di uno *strumento finanziario*¹ (“market direction”). L’analisi si basa solo sui dati dei rendimenti.

La “Previsione di mercato con più features” (Market Prediction with More Features) aggiunge altre caratteristiche (features) al mix, come i tipici indicatori finanziari. In questo contesto, i primi risultati indicano che le inefficienze statistiche potrebbero essere effettivamente presenti. Ciò viene confermato nella “Previsione di Mercato Intragionaliero” (Market Prediction Intraday), che lavora con i dati intraday rispetto ai dati di fine giornata (EOD).

Infine, nelle “Conclusioni” si discute dell’efficienza dei big data in combinazione con l’IA in alcuni settori e si sostiene che la finanza AI-first, finanza priva

¹Sono molti gli strumenti finanziari attraverso i quali si può decidere di investire il proprio denaro. Si tratta di mezzi di investimento di tipo diverso, che vanno dalle azioni alle obbligazioni, passando per titoli di Stato e derivati.

di teorie (theory-free), potrebbe rappresentare una via d'uscita dalle fallacie teoriche della finanza tradizionale.

4.1 Mercati Efficienti

L'efficient market hypothesis o EMH è una teoria economica e degli investimenti che tenta di spiegare il movimento dei mercati finanziari. È stata sviluppata negli anni '60 dall'economista Eugene Fama, il quale sosteneva che i prezzi di tutti i titoli sono assolutamente equi e rispecchiano il valore intrinseco degli asset in un dato momento.

Quando si parla di mercati efficienti, ci si riferisce a una situazione in cui tutte le decisioni dei partecipanti al mercato sono puramente razionali e si sta tenendo conto di tutte le informazioni disponibili. La teoria del mercato efficiente si basa su questi presupposti e sostiene che il prezzo di mercato sarà sempre accurato, poiché subirà delle variazioni immediate non appena vi siano eventuali nuove informazioni. La teoria EMH riconosce che i movimenti volatili si verificano a seguito di notizie impreviste ma che, una volta assimilate le relative informazioni, si ritorna al mercato efficiente.

Seguendo i principi di questa teoria, per i singoli trader, investitori e gestori di fondi sarebbe impossibile "battere" il mercato (con questo si intende il caso in cui si ottengano rendimenti superiori alla media di mercato). Questo perché non esisterebbero azioni *sopravalutate* o *sottovalutate*. Se consideriamo la forma forte della teoria dei mercati efficienti, essa toglie all'*analisi fondamentale* e all'*analisi tecnica*² la ragion d'essere, poiché non possono esistere informazioni in grado di produrre rendimenti consistenti oltre l'insider trading.

È questo il motivo per cui la teoria dei mercati efficienti è estremamente controversa e, benché abbia un notevole seguito, è anche oggetto di aspre critiche.

In parole povere, l'ipotesi dei mercati efficienti dice che i prezzi degli strumenti finanziari in un certo momento riflettono tutte le informazioni disponibili in quel momento. Se l'EMH fosse vera, non avrebbe senso discutere se il prezzo di un'azione sia troppo alto o troppo basso. Il prezzo di un'azione, in base all'EMH, è in ogni momento esattamente al livello appropriato in base alle informazioni disponibili. Sono stati compiuti molti sforzi per perfezionare e formalizzare l'idea di mercati efficienti sin dalla formulazione e dalle prime di-

²L'analisi fondamentale è, insieme all'analisi tecnica, uno dei metodi più importanti di analisi di mercato. Mentre i trader tecnici prendono tutte le informazioni, di cui hanno bisogno per operare, dai grafici, i trader fondamentali valutano i fattori al di fuori dei movimenti di prezzo dell'asset.

scussioni dell'EMH negli anni '60. Le definizioni presentate in Jensen (1978) sono ancora oggi utilizzate. Jensen definisce un mercato efficiente come segue:

Un mercato è efficiente rispetto a un insieme di informazioni θ_t se è impossibile ottenere profitti economici commercializzando sulla base di un insieme di informazioni θ_t . Per profitti economici intendiamo i rendimenti corretti per il rischio al netto di tutti i costi.

In questo contesto, Jensen distingue tre forme di efficienza di mercato:

- **Forma debole dell'EMH**

In questo caso, l'insieme di informazioni θ_t comprende solo la storia passata dei prezzi e dei rendimenti del mercato

- **Forma semi-forte dell'EMH**

In questo caso, l'insieme di informazioni θ_t è costituito da tutte le informazioni pubblicamente disponibili, che comprendono non solo la storia passata dei prezzi e dei rendimenti, ma anche relazioni finanziarie, articoli di giornale, dati meteorologici e così via.

- **Forma forte dell'EMH**

Questo caso è dato quando l'insieme di informazioni θ_t comprende tutte le informazioni disponibili dell'EMH che sono di vasta portata.



- **Efficienza in forma debole**

La forma debole della teoria dei mercati efficienti ipotizza che l'attuale prezzo di mercato rispecchi tutte le informazioni storiche relative al prezzo di un titolo. La tesi a sostegno dell'efficienza in forma debole afferma che tutti i nuovi movimenti di prezzo non hanno alcuna relazione con i dati storici. Pertanto, chi condivide questa teoria ritiene che tutti i movimenti futuri dei prezzi delle azioni non siano prevedibili in base ai movimenti dei prezzi precedenti. In sostanza, il mercato è totalmente imprevedibile come spiega la *teoria del cammino casuale o random walk*³.

- **Efficienza in forma semi-forte**

I sostenitori della forma semi-forte della teoria dei mercati efficienti ritengono che il prezzo di mercato tenga conto di tutte le informazioni pubblicamente disponibili. La teoria afferma che lo studio di queste informazioni, che vanno dagli estratti conto dell'azienda allo storico dei prezzi delle azioni, non può risultare in profitti troppo consistenti. Un mercato efficiente in forma semi-forte implica che né un'analisi fondamentale né un'analisi tecnica possano fornire informazioni utili, poiché tutte le nuove informazioni vengono immediatamente incorporate nei prezzi di mercato. La teoria dei mercati efficienti ritiene che solo chi detiene informazioni private possa godere di un vantaggio. In genere chi condivide la teoria EMH mette in discussione l'esigenza e l'esistenza di una consistente porzione di servizi finanziari, come analisti e ricercatori nel settore degli investimenti.

- **Efficienza in forma forte**

La forma forte della teoria dei mercati efficienti sostiene che tutte le informazioni disponibili, sia pubbliche che private, siano incorporate nel prezzo di un titolo. Ciò significa che nessun investitore potrebbe superare regolarmente il mercato nel suo complesso, ma che qualcuno potrebbe talvolta realizzare rendimenti fuori norma. La forma forte della teoria EMH presuppone che il mercato sia perfetto e che l'unico modo in cui sarebbe possibile registrare un rendimento superiore alla media si otterrebbe sfruttando informazioni interne.

Nel suo articolo pionieristico sull'EMH, Fama (1965) conclude come segue:

³La teoria del 'random walk' (in italiano 'teoria del percorso casuale') è un modello finanziario che presuppone che il mercato azionario si muova in modo completamente imprevedibile. L'ipotesi suggerisce che il prezzo futuro di ogni azione è indipendente dal proprio movimento storico e dal prezzo di altri titoli. Secondo la teoria del random walk, le forme di analisi delle azioni, sia tecnica che fondamentale, sono inaffidabili.

Per molti anni, economisti, statistici e docenti di finanza si sono interessati allo sviluppo e alla verifica di modelli di comportamento dei prezzi delle azioni. Un modello importante che si è evoluto da questa ricerca è la teoria dei percorsi casuali (theory of random walks). Questa teoria mette in serio dubbio molti altri metodi per descrivere e prevedere il comportamento dei prezzi azionari - metodi che hanno una notevole popolarità al di fuori del mondo accademico. Per esempio, se la teoria dei percorsi casuali (random walks) è una descrizione accurata della realtà, allora le varie procedure "tecniche" e "fondamentali" per prevedere i prezzi delle azioni sono completamente prive di valore.

In altre parole, se l'EMH è vera, qualsiasi tipo di ricerca o analisi dei dati per ottenere rendimenti superiori al mercato dovrebbe essere inutile nella pratica. D'altro canto, si è sviluppata un'industria della gestione patrimoniale da miliardi di dollari che promette rendimenti superiori al mercato grazie a una rigorosa ricerca e alla gestione attiva del capitale. In particolare, l'industria degli hedge fund si basa sulla promessa di fornire alfa, ossia rendimenti superiori al mercato e persino indipendenti, almeno in larga misura, dai rendimenti del mercato. Quanto sia difficile mantenere tale promessa è dimostrato dai dati di un recente studio di Preqin. Lo studio riporta un calo dell'indice Preqin All-Strategies Hedge Fund pari a -3,42% per l'anno 2018. Quasi il 40% di tutti gli hedge fund coperti dallo studio ha registrato perdite del 5% o superiori per quell'anno.

Se il prezzo di un'azione (o di un qualsiasi altro strumento finanziario) segue un percorso casuale (random-walk) standard, allora i rendimenti sono distribuiti normalmente con media zero. Il prezzo delle azioni sale con la probabilità del 50% e scende con il 50% di probabilità. In un contesto simile, il miglior predittore del prezzo delle azioni di domani è il prezzo delle azioni di oggi. Ciò è dovuto alla proprietà di Markov dei random-walks, vale a dire che la distribuzione dei prezzi azionari futuri è indipendente dalla storia dei prezzi, ma dipende solo dal livello attuale dei prezzi. Pertanto, nel contesto di un random-walk, l'analisi dei prezzi (o dei rendimenti) storici è inutile per prevedere i prezzi futuri.

4.1.1 Sostenitori della teoria dei mercati efficienti

Dopo la pubblicazione da parte di Fama negli anni '60, la teoria dei mercati efficienti (EMH) è rimasta estremamente popolare sia nell'ambito degli studi economici che di quelli commerciali e gran parte della ricerca sembrava supportare gli argomenti di questa ipotesi.

Tuttora esistono argomentazioni a favore della teoria dei mercati efficienti, tra cui:

- **Le sovraperformance dei fondi di investimento passivi**

La crescente popolarità degli investimenti passivi tramite fondi comuni di investimento ed ETF viene spesso citata per dimostrare che il pubblico sostiene la teoria dei mercati efficienti. Ipoteticamente, se la teoria dei mercati efficienti fosse errata e i mercati fossero inefficienti, i fondi attivi dovrebbero realizzare rendimenti maggiori rispetto ai fondi passivi. Questo, tuttavia, non succede spesso sul lungo periodo. Uno studio condotto da Morningstar ha rilevato che su un periodo di 10 anni conclusosi a giugno 2019, solo il 23% dei fondi attivi aveva superato i rendimenti medi della relativa controparte passiva. I sostenitori della teoria EMH citano questo studio e altri analoghi per dimostrare che i mercati sono efficienti e che la teoria è valida sul lungo termine.

Tuttavia, è emersa un'argomentazione contraria secondo cui se l'investimento passivo crescesse eccessivamente, potrebbe avere un impatto negativo sull'efficienza dei mercati. Gli investitori attivi sostengono il monitoraggio di ricerca, trading e mercato: aspetti essenziali per il buon funzionamento dei mercati. Ma perché un mercato sia realmente efficiente, occorrono sia partecipanti passivi che attivi. Laddove gli investitori attivi sono considerati "informati", poiché hanno raccolto tutte le informazioni disponibili allo scopo di sfruttare le inefficienze del mercato, dipendono comunque da altri trader "non informati" e dal loro occupare una posizione opposta alla loro sui mercati. Se gli investitori rinunciassero al rischio operando solo passivamente sui mercati finanziari, in teoria ci sarebbero molte meno opportunità di trading. Resta da vedere se gli enti normativi prenderanno provvedimenti rispetto al crescente squilibrio tra fondi attivi e passivi per conservare l'efficienza del mercato. In passato l'FCA (Financial Conduct Authority) ha dichiarato che potrebbe considerare la governance aziendale relativa al numero di azioni che può appartenere ai fondi passivi allo scopo di incoraggiare l'investimento attivo.

4.1.2 Detrattori della teoria dei mercati efficienti

Nel corso del tempo, la teoria dei mercati efficienti è stata oggetto di molte critiche. Abbiamo esaminato alcune delle argomentazioni più diffuse contro tale ipotesi, tra cui:

- **Bolle e crisi di mercato**

Le bolle speculative si hanno quando il prezzo di un asset aumenta ben oltre il suo valore equo (fair value) tanto che, quando avviene la correzione di mercato, i prezzi scendono rapidamente e si verifica una crisi

finanziaria. In base alla teoria dei mercati efficienti, le bolle del mercato e le crisi finanziarie non dovrebbero verificarsi. Anzi, la teoria sostiene che tali fenomeni non possono esistere perché il prezzo di un asset è sempre accurato. Fama, ad esempio, sosteneva che la crisi finanziaria del 2008 fosse il risultato di una recessione imminente anziché di una bolla creditizia. Era convinto che non si fosse trattato di una bolla speculativa poiché avrebbe potuto essere prevista e non identificata soltanto a posteriori. Tuttavia, molte delle critiche alla spiegazione di Fama sottolineano che la bolla creditizia era effettivamente prevedibile, come dimostrato da coloro che hanno scommesso contro il mercato delle opzioni di credito inadempienti e hanno fatto i milioni. Quando si verifica una bolla finanziaria, non significa che non esista consenso sul prezzo di un asset ma semplicemente che tale consenso è errato. Nel caso della bolla finanziaria del 2008, i partecipanti al mercato hanno ignorato informazioni di mercato essenziali per poter continuare a rilanciare il mercato delle opzioni di credito. Tale possibilità va contro tutto quanto sostiene la teoria dei mercati efficienti.

- **Economia comportamentale**

Anche l'introduzione dell'ambito dell'economia comportamentale ha consentito di rivolgere critiche alla teoria dei mercati efficienti. L'idea che i partecipanti al mercato siano, nel complesso, razionali è stata messa in discussione con intensità crescente man mano che si apprende di più sulla psicologia del trading. L'economia comportamentale consente anche di spiegare in buona parte le anomalie di mercato trattate sopra. Le pressioni sociali possono indurre i singoli a prendere decisioni irrazionali, che possono spingere i trader a fare errori e ad assumersi maggiori rischi di quelli che avrebbero normalmente assunto. In particolare, il fenomeno del comportamento imitativo, che descrive i singoli che “salgono sul carro della vittoria”, dimostra che non tutte le decisioni sono razionali e fondate sulle informazioni. Anche fattori quali i tratti caratteriali o le emozioni di un trader o di un investitore possono avere un impatto significativo sul comportamento e sul modo in cui interagiscono con il mercato.

- **Trader che hanno sovraperformato il mercato**

Esistono investitori che hanno sovraperformato costantemente la media di mercato. Il più noto è ovviamente Warren Buffett, la cui società Berkshire Hathaway ha registrato prestazioni migliori dell'indice S&P 500 il 73% delle volte tra il 2008 e il 2018. Nemmeno Buffett condivide la teoria

dei mercati efficienti e non ha mai nascosto il suo atteggiamento critico verso l'approccio passivo all'investimento. Buffett adotta invece un approccio basato sul valore degli investimenti, che punta a individuare le azioni sottovalutate tramite l'analisi fondamentale. Egli ammette che la teoria EMH è una teoria sufficientemente convincente da persuadere molti investitori a scegliere fondi indicizzati ed ETF. Personalmente, tuttavia, non ha mai investito in fondi indicizzati.

Un test semi-formale per i mercati efficienti può essere implementato come segue. Prendiamo una serie temporale finanziaria, ritardiamo i dati sui prezzi più volte e utilizziamo i dati sui prezzi ritardati come features per una regressione OLS che utilizza il livello dei prezzi corrente come data labels. Lo spirito è simile a quello delle tecniche grafiche che si basano sulle informazioni storiche dei prezzi per prevedere i prezzi futuri. Utilizzando Python implementiamo un'analisi basata sui dati dei prezzi ritardati per una serie di strumenti finanziari, sia commerciabili (tradable) che non commerciabili. Innanzitutto, importiamo i dati e mostriamo la relativa visualizzazione (vedi figura 4.1):



Figura 4.1: Dati di serie storiche normalizzati (fine giornata)

In secondo luogo, i dati sui prezzi di tutte le serie temporali finanziarie sono ritardati e memorizzati in un DataFrame:

	GLD	lag_1	lag_2	lag_3	lag_4	lag_5
Date						
2010-01-13	111.54	10.49	112.85	111.37	110.82	111.51
2010-01-14	112.03	111.54	110.49	112.85	111.37	110.82
2010-01-15	110.86	112.03	111.54	110.49	112.85	111.37

```

6 2010-01-19 111.52 110.86 112.03 111.54 110.49 112.85
7 2010-01-20 108.94 111.52 110.86 112.03 111.54 110.49
8 2010-01-21 107.37 108.94 111.52 110.86 112.03 111.54
9 2010-01-22 107.17 107.37 108.94 111.52 110.86 112.03
10
11 Date      lag_6    lag_7
12
13 2010-01-13 109.70 109.80
14 2010-01-14 111.51 109.70
15 2010-01-15 110.82 111.51
16 2010-01-19 111.37 110.82
17 2010-01-20 112.85 111.37
18 2010-01-21 110.49 112.85
19 2010-01-22 111.54 110.49

```

In terzo luogo, con i dati preparati, l'analisi di regressione OLS è semplice da condurre. La figura 4.2 mostra i risultati medi della regressione ottimale. Senza dubbio, i dati sui prezzi ritardati di un singolo giorno hanno il potere esplicativo più elevato. Il suo peso è prossimo a 1, a sostegno dell'idea che il miglior predittore del prezzo di domani di uno strumento finanziario sia il suo prezzo odierno. Questo vale anche per i risultati della regressione singola ottenuti per serie temporali finanziarie:

```

1
2          lag_1   lag_2   lag_3   lag_4   lag_5   lag_6
3 AAPL.O  1.0106 -0.0592  0.0258  0.0535 -0.0172  0.0060
4 MSFT.O  0.8928  0.0112  0.1175 -0.0832 -0.0258  0.0567
5 INTC.O  0.9519  0.0579  0.0490 -0.0772 -0.0373  0.0449
6 AMZN.O  0.9799 -0.0134  0.0206  0.0007  0.0525 -0.0452
7 GS.N   0.9806  0.0342 -0.0172  0.0042 -0.0387  0.0585
8 SPY    0.9692  0.0067  0.0228 -0.0244 -0.0237  0.0379
9 .SPX   0.9672  0.0106  0.0219 -0.0252 -0.0318  0.0515
10 .VIX   0.8823  0.0591 -0.0289  0.0284 -0.0256  0.0511
11 EUR=   0.9859  0.0239 -0.0484  0.0508 -0.0217  0.0149
12 XAU=   0.9864  0.0069  0.0166 -0.0215  0.0044  0.0198
13 GDX    0.9765  0.0096 -0.0039  0.0223 -0.0364  0.0379
14 GLD    0.9766  0.0246  0.0060 -0.0142 -0.0047  0.0223
15
16      lag_7
17 -0.0184
18  0.0323
19  0.0112
20  0.0056
21 -0.0215

```

```

22  0.0121
23  0.0063
24  0.0306
25 -0.0055
26 -0.0125
27 -0.0065
28 -0.0106

```

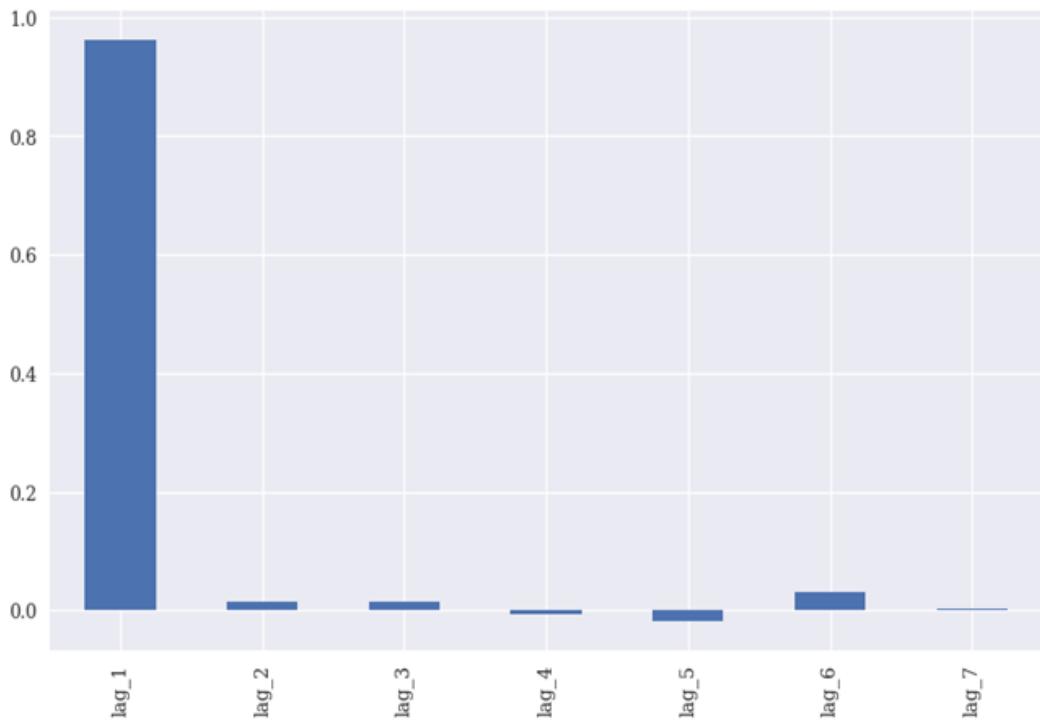


Figura 4.2: Parametri di regressione ottimali medi per i prezzi ritardati

Alla luce di questa analisi semi-formale, sembra esserci un forte evidenza a sostegno dell'EMH, almeno nella sua forma debole. È da notare che l'analisi di regressione OLS, così come è stata implementata, viola diverse ipotesi. Tra queste, si ipotizza che le features non siano correlate tra loro, mentre idealmente dovrebbero essere altamente correlate con i data labels. Tuttavia, i dati sui prezzi ritardati portano a features altamente correlate.

Il seguente codice Python presenta i dati di correlazione, che mostrano una correlazione quasi perfetta tra tutte le features. Ciò spiega perché una sola feature (“lag 1”) è sufficiente per ottenere l’approssimazione e la previsione basate sull’approccio di regressione OLS. L’aggiunta di features altamente correlate

non produce alcun miglioramento. Un'altra ipotesi fondamentale violata è la *stazionarietà*⁴ dei dati delle serie temporali.

In sintesi, se l'EMH è vera, la gestione attiva o algoritmica del portafoglio o il trading non avrebbe senso dal punto di vista economico. Il semplice investimento in un'azione o in un portafoglio efficiente nel senso della MVP, ad esempio, e la detenzione passiva dell'investimento per un lungo periodo produrrebbero senza alcuno sforzo almeno gli stessi rendimenti, se non superiori. Secondo il CAPM e la MVP, più alto è il rischio che l'investitore è disposto a sopportare, più alto dovrebbe essere il rendimento atteso. In effetti, come sottolineano Copeland et al. (2005, cap. 10), il CAPM e l'EMH formano un'ipotesi congiunta sui mercati finanziari: se l'EMH viene respinta, allora anche il CAPM deve essere respinto, poiché la sua derivazione presuppone che l'EMH sia vera.

4.2 Previsioni di mercato basate sui dati dei rendimenti

Negli ultimi anni gli algoritmi di ML e, in particolare, di DL, hanno fatto passi da gigante in campi che si sono dimostrati resistenti, per periodi di tempo piuttosto lunghi, ai metodi statistici o matematici standard. Che dire dei mercati finanziari? Gli algoritmi ML e DL potrebbero essere in grado di scoprire inefficienze laddove i metodi tradizionali di econometria finanziaria, come la regressione OLS, falliscono? Naturalmente, non esistono ancora risposte semplici e concise a queste domande. Tuttavia, alcuni esempi concreti potrebbero illuminare possibili risposte. A tal fine, si utilizzano i dati della sezione precedente per ricavare i rendimenti logaritmici dai dati dei prezzi. L'idea è quella di confrontare le prestazioni della regressione OLS con quelle delle reti neurali nel prevedere la direzione del movimento del giorno successivo per le diverse serie temporali.

L'obiettivo in questa fase è scoprire le inefficienze statistiche rispetto alle inefficienze economiche. Un'inefficienza statistica, ai fini di questa tesi, si riscontra quando un predittore (un modello o un algoritmo in generale o una rete neurale in particolare) prevede i mercati significativamente meglio di un predittore casuale che assegna uguale probabilità ai movimenti al rialzo e al ribasso. Le inefficienze statistiche si hanno quando un modello è in grado di prevedere la direzione del movimento futuro dei prezzi con un certo margine (ad

⁴Una serie storica è stazionaria quando oscilla intorno ad un determinato livello. Spesso questo livello è la media aritmetica dei valori della serie storica.

esempio, la previsione è corretta nel 55% o 60% dei casi). Le inefficienze economiche sono date solo se le inefficienze statistiche possono essere sfruttate con profitto attraverso una strategia di trading che tenga conto, ad esempio, dei costi di transazione.

La prima fase dell'analisi consiste nel creare insiemi di dati con ritardi nei rendimenti logaritmici. I dati normalizzati dei rendimenti logaritmici ritardati sono inoltre sottoposti a un test per la stazionarietà e le features sono sottoposte a un test di correlazione. Poiché le analisi che seguono si basano su dati relativi alle serie temporali, si tratta di *efficienza di mercato in forma debole*:

```

1 #mostra un esempio dei dati ritardati
2 dfs [ sym ] . head ()
3
4
5 Date
6 2010-01-14 0.0044 0.9570 -2.1692 1.3386 0.4959 -0.6434
7 2010-01-15 -0.0105 0.4379 0.9571 -2.1689 1.3388 0.4966
8 2010-01-19 0.0059 -1.0842 0.4385 0.9562 -2.1690 1.3395
9 2010-01-20 -0.0234 0.5967 -1.0823 0.4378 0.9564 -2.1686
10 2010-01-21 -0.0145 -2.4045 0.5971 -1.0825 0.4379 0.9571
11
12 lag_6 lag_7
13
14 1.6613 -0.1028
15 -0.6436 1.6614
16 0.4958 -0.6435
17 1.3383 0.4958
18 -2.1680 1.3384
19
20 #testa la stazionarieta' delle serie temporali
21 adfuller (dfs [ sym ] [ 'lag_1' ])
22 (-51.56825150582553,
23 0.0 ,
24 0 ,
25 2507 ,
26 { '1%' : -3.4329610922579095 ,
27 '5%' : -2.8626935681060375 ,
28 '10%' : -2.567384088736619 } ,
29 7017.165474260225 )
30
31 #mostra la corelazione tra i dati delle features
32 dfs [ sym ] . corr ()
33
```

```

34          GLD    lag_1    lag_2      lag_3    lag_4    lag_5
35 GLD     1.0000 -0.0297  0.0003  1.2635e-02 -0.0026 -5.9392e-03
36 lag_1   -0.0297  1.0000 -0.0305  8.1418e-04  0.0128 -2.8765e-03
37 lag_2   0.0003 -0.0305  1.0000 -3.1617e-02  0.0003  1.3234e-02
38 lag_3   0.0126  0.0008 -0.0316  1.0000e+00 -0.0313 -6.8542e-06
39 lag_4   -0.0026  0.0128  0.0003 -3.1329e-02  1.0000 -3.1761e-02
40 lag_5   -0.0059 -0.0029  0.0132 -6.8542e-06 -0.0318  1.0000e+00
41 lag_6   0.0099 -0.0053 -0.0043  1.4115e-02  0.0002 -3.2289e-02
42 lag_7   -0.0013  0.0098 -0.0052 -4.3869e-03  0.0141  2.1707e-04
43
44    lag_6    lag_7
45  0.0099 -0.0013
46 -0.0053  0.0098
47 -0.0043 -0.0052
48  0.0141 -0.0044
49  0.0002  0.0141
50 -0.0323  0.0002
51  1.0000 -0.0324
52 -0.0324  1.0000

```

In primo luogo, viene implementata la regressione OLS e vengono generate le previsioni risultanti dalla regressione. L'analisi viene eseguita sull'intero set di dati. Essa mostra la prestazione degli algoritmi nel campione. L'accuratezza con cui la regressione OLS predice la direzione del movimento del giorno successivo è leggermente, o addirittura di pochi punti percentuali, superiore al 50% con un'unica eccezione:

1	OLS	AAPL.O	acc=0.5056
2	OLS	MSFT.O	acc=0.5088
3	OLS	INTC.O	acc=0.5040
4	OLS	AMZN.O	acc=0.5048
5	OLS	GS.N	acc=0.5080
6	OLS	SPY	acc=0.5080
7	OLS	.SPX	acc=0.5167
8	OLS	.VIX	acc=0.5291
9	OLS	EUR=	acc=0.4984
10	OLS	XAU=	acc=0.5207
11	OLS	GDX	acc=0.5307
12	OLS	GLD	acc=0.5072

In secondo luogo, la stessa analisi è stata ripetuta, ma questa volta con una rete neurale di scikit-learn (MLPRegressor) come modello per l'apprendimento e la previsione. L'accuratezza della previsione nel campione è significativamente superiore al 50% in tutti casi e superiore al 60% in alcuni di essi:

1	MLP	AAPL.O	acc=0.6005
2	MLP	MSFT.O	acc=0.5853
3	MLP	INTC.O	acc=0.5766
4	MLP	AMZN.O	acc=0.5510
5	MLP	GS.N	acc=0.6527
6	MLP	SPY	acc=0.5419
7	MLP	.SPX	acc=0.5399
8	MLP	.VIX	acc=0.6579
9	MLP	EUR=	acc=0.5642
10	MLP	XAU=	acc=0.5522
11	MLP	GDX	acc=0.6029
12	MLP	GLD	acc=0.5259

In terzo luogo, la stessa analisi ma con una rete neurale del pacchetto Keras.

I risultati di accuratezza sono simili a quelli ottenuti con MLPRegressor, ma con un'accuratezza media più elevata:

1	DNN	AAPL.O	acc=0.6192
2	DNN	MSFT.O	acc=0.6216
3	DNN	INTC.O	acc=0.5634
4	DNN	AMZN.O	acc=0.5809
5	DNN	GS.N	acc=0.6240
6	DNN	SPY	acc=0.5734
7	DNN	.SPX	acc=0.5821
8	DNN	.VIX	acc=0.6033
9	DNN	EUR=	acc=0.5781
10	DNN	XAU=	acc=0.5726
11	DNN	GDX	acc=0.6288
12	DNN	GLD	acc=0.5781

Questo semplice esempio mostra che le reti neurali possono superare significativamente la regressione OLS, *nel campione*, nel prevedere la direzione dei movimenti di prezzo del giorno successivo. Come cambia il quadro della situazione quando il test delle prestazioni dei due tipi di modello è fatto *fuori dal campione*? A tal fine le analisi vengono ripetute, ma sul primo 80% dei dati viene implementata la fase di training (fitting) mentre sul restante 20% viene testata la performance. La regressione OLS viene implementata per prima. La regressione OLS fuori campione mostra livelli di accuratezza simili a quelli nel campione, circa il 50%:

1	OLS	AAPL.O	acc=0.5219
2	OLS	MSFT.O	acc=0.4960
3	OLS	INTC.O	acc=0.5418
4	OLS	AMZN.O	acc=0.4841
5	OLS	GS.N	acc=0.4980

6	OLS	SPY	acc=0.5020
7	OLS	.SPX	acc=0.5120
8	OLS	.VIX	acc=0.5458
9	OLS	EUR=	acc=0.4482
10	OLS	XAU=	acc=0.5299
11	OLS	GDX	acc=0.5159
12	OLS	GLD	acc=0.5100

Le prestazioni del modello MLPRegressor sono molto peggiori fuori dal campione rispetto ai numeri nel campione e simili ai risultati della regressione OLS:

1	MLP	AAPL.O	acc=0.4920
2	MLP	MSFT.O	acc=0.5279
3	MLP	INTC.O	acc=0.5279
4	MLP	AMZN.O	acc=0.4641
5	MLP	GS.N	acc=0.5040
6	MLP	SPY	acc=0.5259
7	MLP	.SPX	acc=0.5478
8	MLP	.VIX	acc=0.5279
9	MLP	EUR=	acc=0.4980
10	MLP	XAU=	acc=0.5239
11	MLP	GDX	acc=0.4880
12	MLP	GLD	acc=0.5000

Lo stesso vale per il modello sequenziale di Keras per il quale anche i numeri fuori campione mostrano valori di accuratezza compresi tra pochi punti percentuali al di sopra e al di sotto della soglia del 50%:

1	DNN	AAPL.O	acc=0.5518
2	DNN	MSFT.O	acc=0.5080
3	DNN	INTC.O	acc=0.4661
4	DNN	AMZN.O	acc=0.5239
5	DNN	GS.N	acc=0.5339
6	DNN	SPY	acc=0.5359
7	DNN	.SPX	acc=0.5478
8	DNN	.VIX	acc=0.5120
9	DNN	EUR=	acc=0.5159
10	DNN	XAU=	acc=0.5060
11	DNN	GDX	acc=0.4920
12	DNN	GLD	acc=0.4781

Efficienza di mercato in forma debole

Sebbene l'etichettatura di forma debole dell'efficienza di mercato possa far pensare il contrario, si tratta della forma più difficile, nel senso che solo i dati relativi alle serie temporali possono essere utilizzati per identificare le inefficienze statistiche. Con la forma semi-forte, è possibile aggiungere qualsiasi altra fonte di dati disponibili al pubblico per migliorare l'accuratezza della previsione.

Sulla base degli approcci scelti in questa sezione, i mercati sembrano essere efficienti almeno nella forma debole. La semplice analisi degli andamenti storici dei rendimenti basata sulla regressione OLS o sulle reti neurali potrebbe non essere sufficiente per scoprire le inefficienze statistiche. Ci sono due elementi principali dell'approccio scelto in questa sezione che possono essere modificati nella speranza di migliorare le previsioni:

Features

Oltre ai dati sui prezzi e rendimenti, è possibile aggiungere altre features ai dati, come gli indicatori tecnici (ad esempio, le medie mobili semplici o SMA). La speranza, secondo la tradizione dei grafici tecnici, è che tali indicatori migliorino l'accuratezza della previsione.

Lunghezza delle barre

Invece di lavorare con i dati di fine giornata, i dati intraday potrebbero consentire una maggiore precisione di previsione. In questo caso, la speranza è che sia più probabile scoprire le inefficienze statistiche durante il giorno rispetto alla fine della giornata, quando tutti i partecipanti al mercato in generale prestano la massima attenzione per effettuare le loro operazioni finali, tenendo conto di tutte le informazioni disponibili.

Le due sezioni seguenti affrontano questi elementi.

4.3 Previsione di mercato con più features

Nel trading esiste una lunga tradizione nell'utilizzo di indicatori tecnici per generare, sulla base di modelli osservati, segnali di acquisto o di vendita. Tali indicatori tecnici, praticamente di qualsiasi tipo, possono essere utilizzati anche come funzionalità per l'addestramento delle reti neurali. Useremo SMA (simple moving average), valori minimo e massimo di rolling, momentum e volatilità di rolling come features.

Indicatori tecnici come features

Come dimostrano gli esempi precedenti, qualsiasi indicatore tecnico tradizionale utilizzato per gli investimenti o il trading intraday può essere utilizzato come feature per addestrare gli algoritmi di ML. In tal senso, l'IA e il ML non rendono necessariamente obsoleti tali indicatori, ma possono anzi arricchire la derivazione di strategie di trading guidata dal ML.

Nel campione, le prestazioni del modello MLPClassifier sono ora molto migliori quando si tiene conto delle nuove caratteristiche e le si normalizza per l'addestramento. Il modello sequenziale di Keras raggiunge precisioni di circa il 70% per il numero di epoche di addestramento. Inoltre, tali prestazioni possono essere facilmente aumentate aumentando il numero di epoche e/o la capacità della rete neurale:

```
1
2 #MLPClassifier
3 IN-SAMPLE | AAPL.O | acc=0.5510
4 IN-SAMPLE | MSFT.O | acc=0.5376
5 IN-SAMPLE | INTC.O | acc=0.5607
6 IN-SAMPLE | AMZN.O | acc=0.5559
7 IN-SAMPLE | GS.N | acc=0.5794
8 IN-SAMPLE | SPY | acc=0.5729
9 IN-SAMPLE | .SPX | acc=0.5941
10 IN-SAMPLE | .VIX | acc=0.6940
11 IN-SAMPLE | EUR=
12 IN-SAMPLE | XAU=
13 IN-SAMPLE | GDX | acc=0.5847
14 IN-SAMPLE | GLD | acc=0.5567
15
16 #Keras
17 IN-SAMPLE | AAPL.O | acc=0.7017
18 IN-SAMPLE | MSFT.O | acc=0.6912
19 IN-SAMPLE | INTC.O | acc=0.7026
20 IN-SAMPLE | AMZN.O | acc=0.6786
21 IN-SAMPLE | GS.N | acc=0.6883
22 IN-SAMPLE | SPY | acc=0.6826
23 IN-SAMPLE | .SPX | acc=0.6940
24 IN-SAMPLE | .VIX | acc=0.7538
25 IN-SAMPLE | EUR=
26 IN-SAMPLE | XAU=
27 IN-SAMPLE | GDX | acc=0.6904
28 IN-SAMPLE | GLD | acc=0.7038
```

Questi miglioramenti devono essere trasferiti alla precisione di predizione fuori dal campione? A seguito ripetiamo l'analisi, questa volta con la suddivisione di training e test utilizzata in precedenza. Purtroppo, il quadro della situazione non è dei migliori. I numeri non rappresentano miglioramenti reali rispetto all'approccio che si basa solo sui dati dei rendimenti ritardati come caratteristiche. Per alcuni strumenti selezionati, sembra esserci un vantaggio di qualche punto percentuale nell'accuratezza della previsione rispetto al benchmark del 50%. Per altri, tuttavia, l'accuratezza è ancora inferiore al 50%, come illustrato per il MLPClassifier:

¹	#MLPClassifier		
2	OUT-OF-SAMPLE	AAPL.O	acc=0.4432
3	OUT-OF-SAMPLE	MSFT.O	acc=0.4595
4	OUT-OF-SAMPLE	INTC.O	acc=0.5000
5	OUT-OF-SAMPLE	AMZN.O	acc=0.5270
6	OUT-OF-SAMPLE	GS.N	acc=0.4838
7	OUT-OF-SAMPLE	SPY	acc=0.4811
8	OUT-OF-SAMPLE	.SPX	acc=0.5027
9	OUT-OF-SAMPLE	.VIX	acc=0.5676
10	OUT-OF-SAMPLE	EUR=	acc=0.4649
11	OUT-OF-SAMPLE	XAU=	acc=0.5514
12	OUT-OF-SAMPLE	GDX	acc=0.5162
13	OUT-OF-SAMPLE	GLD	acc=0.4946

Le buone prestazioni all'interno del campione e le non altrettanto buone prestazioni fuori dal campione suggeriscono che l'overfitting della rete neurale potrebbe giocare un ruolo cruciale. Un approccio per evitare l'overfitting è quello di utilizzare metodi di ensemble che combinano più modelli addestrati dello stesso tipo per ottenere un metamodello più robusto e migliori previsioni fuori dal campione. Uno di questi metodi è chiamato bagging. scikit-learn ha un'implementazione di questo approccio sotto forma della classe BaggingClassifier. L'uso di più stimatori consente di addestrare ciascuno di essi senza esporli all'intero set di dati di addestramento o a tutte le features. Questo dovrebbe aiutare a evitare l'overfitting.

Implementiamo un approccio bagging basato su una serie di stimatori di base dello stesso tipo (MLPClassifier). Le accuratezze di previsione sono ora costantemente superiori al 50%. Alcuni valori di accuratezza sono superiori al 55%, che può essere considerato piuttosto alto in questo contesto. Nel complesso, il bagging sembra evitare, almeno in una certa misura, l'overfitting e sembra migliorare sensibilmente le previsioni:

¹	#BaggingClassifier		
2	OUT-OF-SAMPLE	AAPL.O	acc=0.5000
3	OUT-OF-SAMPLE	MSFT.O	acc=0.5703

4	OUT-OF-SAMPLE		INTC.O		acc = 0.5054
5	OUT-OF-SAMPLE		AMZN.O		acc = 0.5270
6	OUT-OF-SAMPLE		GS.N		acc = 0.5135
7	OUT-OF-SAMPLE		SPY		acc = 0.5568
8	OUT-OF-SAMPLE		.SPX		acc = 0.5514
9	OUT-OF-SAMPLE		.VIX		acc = 0.5432
10	OUT-OF-SAMPLE		EUR=		acc = 0.5054
11	OUT-OF-SAMPLE		XAU=		acc = 0.5351
12	OUT-OF-SAMPLE		GDX		acc = 0.5054
13	OUT-OF-SAMPLE		GLD		acc = 0.5189

Efficienza del mercato di fine giornata

L'ipotesi del mercato efficiente risale agli anni '60 e '70, periodi in cui i dati di fine giornata erano sostanzialmente gli unici disponibili per le serie temporali. A quei tempi (e ancora oggi), si poteva ipotizzare che gli operatori di mercato prestassero particolare attenzione alle loro posizioni e ai loro scambi quanto più si avvicinava la fine della sessione di trading. Questo potrebbe essere più vero per le azioni, ad esempio, e un po' meno per le valute, che sono scambiate in linea di massima 24 ore su 24.

4.4 Previsione del mercato Intraday

Questo capitolo non ha prodotto prove conclusive, ma le analisi condotte fino ad ora indicano che i mercati sono debolmente efficienti a fine giornata. Che dire dei mercati intraday? Ci sono inefficienze statistiche più consistenti da individuare?

Per trovare una risposta a questa domanda, è necessario un altro set di dati. Utilizziamo un set di dati composto dagli stessi strumenti del set di dati di fine giornata, ma che ora contiene i prezzi di chiusura orari. Poiché gli orari di negoziazione (trading) possono variare da strumento a strumento, il set di dati è incompleto. Tuttavia, questo non è un problema, poiché le analisi sono implementate serie temporale per serie temporale. L'implementazione tecnica per i dati orari è essenzialmente la stessa di prima, basandosi sullo stesso codice dell'analisi di fine giornata.

Le accuratezze di previsione intraday sono ancora una volta distribuite intorno al 50% con uno spread relativamente ampio per la singola rete neurale. In positivo, alcuni valori di accuratezza sono superiori al 55%. Il metamodello bagging mostra invece una performance fuori dal campione più consistente,

con molti dei valori di accuratezza osservati che superano di qualche punto percentuale il benchmark del 50%:

```

1 train_test_model(model_mlp)
2 OUT-OF-SAMPLE | AAPL.O | acc=0.5420
3 OUT-OF-SAMPLE | MSFT.O | acc=0.4930
4 OUT-OF-SAMPLE | INTC.O | acc=0.5549
5 OUT-OF-SAMPLE | AMZN.O | acc=0.4709
6 OUT-OF-SAMPLE | GS.N | acc=0.5184
7 OUT-OF-SAMPLE | SPY | acc=0.4860
8 OUT-OF-SAMPLE | .SPX | acc=0.5019
9 OUT-OF-SAMPLE | .VIX | acc=0.4885
10 OUT-OF-SAMPLE | EUR=
11 OUT-OF-SAMPLE | XAU=
12 OUT-OF-SAMPLE | GDX | acc=0.4765
13 OUT-OF-SAMPLE | GLD | acc=0.5455
14
15
16 train_test_model(model_bag)
17 OUT-OF-SAMPLE | AAPL.O | acc=0.5660
18 OUT-OF-SAMPLE | MSFT.O | acc=0.5551
19 OUT-OF-SAMPLE | INTC.O | acc=0.5072
20 OUT-OF-SAMPLE | AMZN.O | acc=0.4830
21 OUT-OF-SAMPLE | GS.N | acc=0.5020
22 OUT-OF-SAMPLE | SPY | acc=0.4680
23 OUT-OF-SAMPLE | .SPX | acc=0.4677
24 OUT-OF-SAMPLE | .VIX | acc=0.5161
25 OUT-OF-SAMPLE | EUR=
26 OUT-OF-SAMPLE | XAU=
27 OUT-OF-SAMPLE | GDX | acc=0.5107
28 OUT-OF-SAMPLE | GLD | acc=0.5475

```

Efficienza del mercato intraday

Anche se i mercati sono debolmente efficienti a fine giornata, possono comunque essere debolmente inefficienti infragiornalieri. Tali inefficienze statistiche possono derivare da squilibri temporanei, pressioni di acquisto o di vendita, reazioni eccessive del mercato, ordini di acquisto o di vendita guidati da fattori tecnici e così via. **La questione centrale è se tali inefficienze statistiche, una volta scoperte, possano essere sfruttate con profitto attraverso specifiche strategie di trading.**

4.5 Conclusioni

Nel loro articolo ampiamente citato "L'irragionevole efficacia dei dati", Halevy et al. (2009) sottolineano che gli economisti soffrono di quella che chiamano invidia della fisica. Con ciò intendono l'incapacità di spiegare il comportamento umano nello stesso modo matematicamente elegante con cui i fisici sono in grado di descrivere anche i complessi fenomeni del mondo reale. Uno di questi esempi è probabilmente la formula più nota di Albert Einstein $E = mc^2$, che equipara l'energia alla massa di un oggetto moltiplicata per la velocità della luce al quadrato.

In economia e finanza, i ricercatori per decenni hanno cercato di emulare l'approccio fisico nel derivare e dimostrare equazioni semplici ed eleganti per spiegare i fenomeni economici e finanziari. Ma come mostrano insieme il Capitolo 2 e il Capitolo 3, molte delle più eleganti teorie finanziarie non hanno quasi alcuna prova a sostegno nel mondo finanziario reale in cui gli assunti semplificativi, come le distribuzioni normali e le relazioni lineari, non valgono.

Come Halevy et al. (2009) spiegano nel loro articolo, potrebbero esserci domini, come i linguaggi naturali e le regole che seguono, che sfidano la derivazione e la formulazione di teorie concise ed eleganti. I ricercatori potrebbero semplicemente dover fare affidamento su teorie e modelli complessi guidati dai dati. Per il linguaggio in particolare, il World Wide Web rappresenta uno tesoro di big data. E i big data sembrano essere necessari per addestrare algoritmi ML e DL su determinati compiti, come l'elaborazione del linguaggio naturale o la traduzione a livello umano. Dopotutto, la finanza potrebbe essere una disciplina che ha più in comune con il linguaggio naturale che con la fisica. Forse, dopo tutto, non esistono formule semplici ed eleganti che descrivano importanti fenomeni finanziari, come la variazione giornaliera del tasso di una valuta o il prezzo di un'azione. Forse la verità potrebbe essere trovata solo nei big data che oggi sono disponibili in modo programmatico sia per i ricercatori finanziari che per gli accademici.

Questo capitolo presenta solo l'inizio della ricerca della verità, per scoprire il Santo Graal della finanza: **dimostrare che i mercati non sono poi così efficienti**. Gli approcci relativamente semplici alle reti neurali di questo capitolo si basano solo su features legate alle serie temporali per l'addestramento. I labels sono semplici e diretti: se il mercato (il prezzo dello strumento finanziario) sale o scende. L'obiettivo è scoprire le inefficienze statistiche nel prevedere la direzione futura del mercato. Questo rappresenta a sua volta il primo passo per sfruttare economicamente tali inefficienze attraverso una strategia di trading implementabile. Nella ricerca delle inefficienze statistiche, questo capitolo si basa esclusivamente sui dati e sulle reti neurali. Non c'è teoria e non ci sono ipotesi sul comportamento dei partecipanti al mercato o

ragionamenti simili. Il maggior sforzo di modellazione viene fatto per quanto riguarda la preparazione delle caratteristiche, che ovviamente rappresentano ciò che il modellatore ritiene importante. Un'assunzione implicita nell'approccio adottato è che le inefficienze statistiche possano essere scoperte solo sulla base dei dati relativi alle serie temporali. Ciò significa che i mercati non sono nemmeno debolmente efficienti, la forma più difficile da confutare delle tre.

Basarsi solo sui dati finanziari e applicare ad essi algoritmi e modelli generali di ML e DL è ciò che si considera come finanza AI-first. Non sono necessarie teorie, né modellizzazioni del comportamento umano, né assunzioni sulle distribuzioni o sulla natura delle relazioni: solo dati e algoritmi. In questo senso, la finanza AI-first potrebbe anche essere etichettata come finanza senza teoria o senza modelli.

Parte III

Scoprire e Sfruttare le Inefficienze

L’obiettivo principale di questa parte è applicare le reti neurali per scoprire le inefficienze statistiche nei mercati finanziari. Ricordiamo che una inefficienza statistica si riscontra quando un predittore (un modello o un algoritmo in generale o una rete neurale in particolare) prevede i mercati significativamente meglio di un predittore casuale che assegna uguale probabilità ai movimenti al rialzo e al ribasso.

In un contesto di trading algoritmico, avere a disposizione un tale predittore è un prerequisito per la generazione di rendimenti alfa o superiori al mercato.

Questa parte è composta da due capitoli che forniscono un maggiore background, dettagli ed esempi relativi alle reti neurali profonde (Deep Neural Networks o DNNs), ed illustrano come sfruttare le inefficienze economiche attraverso il backtest vettorializzato delle strategie di trading.

Per lo scopo della presente tesi, l’approccio in questa parte, ma anche nella parte IV, è pratico, tralasciando molti dettagli importanti per quanto riguarda gli algoritmi e le tecniche applicate. Ciò è giustificato dal momento che l’obiettivo di questo elaborato non è illustrare il funzionamento degli algoritmi e tecniche di ML e DL, ma di mostrare come essi possono essere applicati nel campo finanziario. Per comprendere le tecniche di IA ci sono una serie di buone risorse in forma di libro disponibili che possono essere consultate per dettagli tecnici e informazioni di base. I codici completi e dettagliati dei seguenti capitoli sono messi a disposizione nei rispettivi file Jupyter Notebook allegati.

Capitolo 5

Reti neurali profonde (DNNs)

“Se stai cercando di prevedere i movimenti di un’azione sul mercato azionario data la recente cronologia dei prezzi, è improbabile che tu abbia successo, perché la cronologia dei prezzi non contiene molte informazioni predittive.”
—Francois Chollet (2017)

Sebbene la citazione introduttiva al capitolo possa dare poche ragioni di speranza, l’obiettivo principale di questo capitolo è scoprire le inefficienze statistiche nei mercati finanziari (serie temporali) applicando le reti neurali. I risultati numerici presentati in questo capitolo, come l’accuratezza delle previsioni del 60% e maggiore in alcuni casi, indicano che almeno qualche speranza è giustificata.

5.1 I Dati

Il capitolo 4 scopre indizi di inefficienze statistiche per, tra le altre serie temporali, la serie dei prezzi infragiornalieri della coppia di valute EUR/USD. Questo capitolo si concentra sul cambio estero (Foreign Exchange FX) come asset class e in particolare sulla coppia di valute EUR/USD. Tra le altre ragioni, lo sfruttamento economico delle inefficienze statistiche per il FX in generale non è così complicato come per gli altri asset classes. La disponibilità di dati gratuita e completa viene spesso fornita anche per il FX. Il seguente set di dati proviene da Refinitiv Eikon Data API. Il set di dati contiene valori di apertura, massimo, minimo e chiusura. La figura 5.1 mostra i prezzi di chiusura:

```
1 #legge i dati e li mette in un DataFrame  
2 url = 'http://hilpisch.com/aiif_eikon_id_eur_usd.csv'  
3 symbol = 'EUR.USD'  
4
```

		HIGH	LOW	OPEN	CLOSE
5	Date				
6	2019-10-01 00:00:00	1.0899	1.0897	1.0897	1.0899
7	2019-10-01 00:01:00	1.0899	1.0896	1.0899	1.0898
8	2019-10-01 00:02:00	1.0898	1.0896	1.0898	1.0896
9	2019-10-01 00:03:00	1.0898	1.0896	1.0897	1.0898
10	2019-10-01 00:04:00	1.0898	1.0896	1.0897	1.0898
11	2019-10-01 00:04:00	1.0898	1.0896	1.0897	1.0898

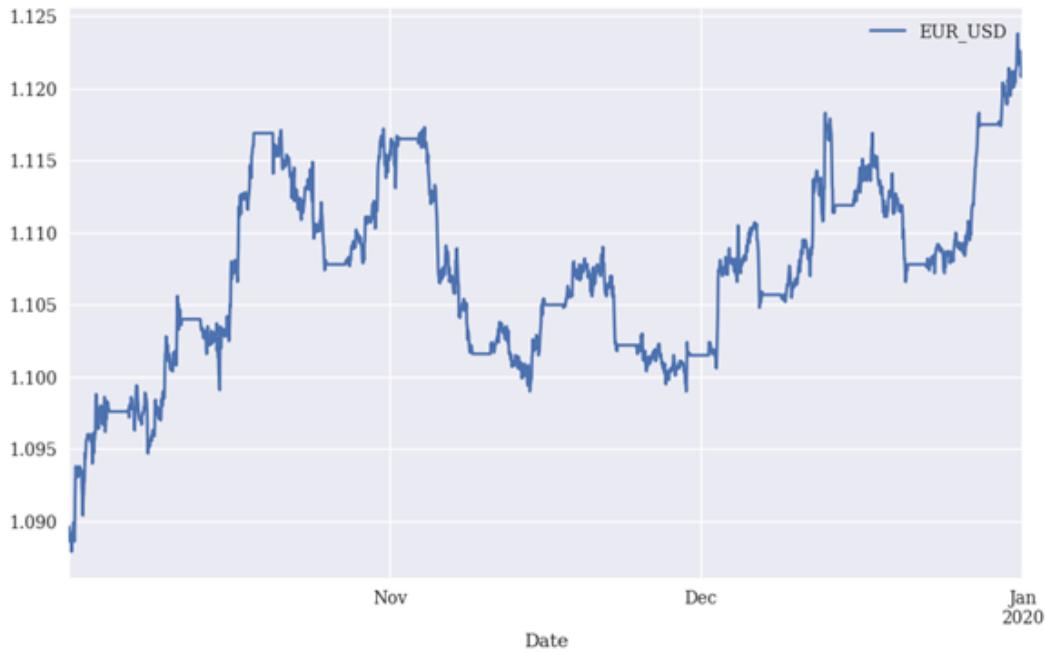


Figura 5.1: Prezzi di chiusura media per EUR/USD (Intraday)

5.2 Previsione di Base

Sulla base del nuovo set di dati, viene ripetuto l'approccio di previsione del capitolo 4. La prima fase è la creazione delle features ritardate. In secondo luogo, diamo uno sguardo ai dati delle etichette (labels).

Un grosso problema nella classificazione che può sorgere, a seconda del set di dati disponibile, è lo *squilibrio di classe*. Ciò significa, nel contesto delle etichette binarie, che la frequenza di una particolare classe rispetto all'altra classe potrebbe essere maggiore. Ciò potrebbe portare a situazioni in cui la rete neurale prevede semplicemente la classe con la frequenza più alta. Applicando pesi appropriati, si può fare in modo che entrambe le classi acquisiscano la stessa importanza durante la fase di addestramento.

La terza fase è la creazione del modello DNN con Keras e l’addestramento del modello sul set di dati completo. La prestazione di riferimento nel campione è di circa il 60%:

```

1 #valuta le prestazioni nel campione
2 model.evaluate(train[cols], train['d'])
3 loss: 0.6283 — accuracy: 0.5830
4 [0.6283310055732727, 0.5830469727516174]
```

Lo stesso vale per le prestazioni del modello fuori campione. È ancora ben al di sopra del 60%. Questo può essere considerato già abbastanza buono:

```

1 #valuta le prestazioni fuori dal campione
2 model.evaluate(test[cols], test['d'])
3 loss: 0.6115 — accuracy: 0.6087
4 [0.6114663481712341, 0.6086956262588501]
```

La figura 5.2 mostra come cambia l’accuratezza sui sottoinsiemi di dati di addestramento e di convalida durante le epoche di addestramento:



Figura 5.2: Valori di accuratezza nel training e convalida

L’analisi in questa sezione pone le basi per l’uso più elaborato delle DNN con Keras. Presenta un approccio base di previsione del mercato. Le sezioni seguenti aggiungono diversi elementi che dovrebbero principalmente migliorare le prestazioni del modello fuori campione ed evitare l’overfitting del modello nei dati di addestramento.

5.3 Normalizzazione

Nella sezione precedente si prendono le features così come sono. Nel capitolo 4, i dati delle features vengono normalizzati sottraendo la media dei dati di addestramento per ogni feature e dividendo per la deviazione standard dei dati di addestramento. Questa tecnica di normalizzazione è chiamata *normalizzazione gaussiana* e si rivela spesso, se non sempre, un aspetto importante durante l'addestramento di una rete neurale. Come osserviamo, le prestazioni nel campione aumentano in modo significativo quando si lavora con dati di features normalizzate. Anche le prestazioni fuori campione aumentano leggermente. Tuttavia, non vi è alcuna garanzia che le prestazioni fuori campione aumentino attraverso la normalizzazione delle features:

```

1 #calcola la media e la deviazione standard per tutte le
2 #features di addestramento
3 mu, std = train.mean(), train.std()
4
5 #normalizza il dataset di training in base alla
6 #normalizzazione gaussiana
7 train_ = (train - mu) / std
8
9 #valuta le prestazioni nel campione
10 model.evaluate(train_[cols], train['d'])
11 loss: 0.4254 — accuracy: 0.9244
12 [0.42540428042411804, 0.9243986010551453]
13 #normalizza il dataset di test in base alla
14 #normalizzazione gaussiana
15 test_ = (test - mu) / std
16
17 #valuta le prestazioni fuori campione
18 model.evaluate(test_[cols], test['d'])
19 loss: 1.5305 — accuracy: 0.6659
20 [1.5304620265960693, 0.6659038662910461]
```

Un grosso problema che si presenta spesso è l'overfitting. È visualizzato in modo impressionante nella figura 5.3, che mostra una precisione di addestramento in costante miglioramento mentre la precisione di convalida diminuisce lentamente:

Tre metodi candidati per evitare l'overfitting sono il *dropout*, la *regolarizzazione* e il *bagging*. Le sezioni seguenti illustrano questi metodi. L'impatto dell'ottimizzatore scelto è discusso più avanti in questo capitolo.



Figura 5.3: Valori di accuratezza dell’addestramento e della convalida (dati delle features normalizzate)

5.4 Dropout

L’idea del dropout è che le reti neurali non dovrebbero utilizzare tutte le unità nascoste (neuroni o nodi negli hidden layers) durante la fase di addestramento. L’analogia con il cervello umano è che un essere umano dimentica regolarmente informazioni apprese in precedenza. Questo, per così dire, mantiene il cervello umano ”di mentalità aperta”. Idealmente, una rete neurale dovrebbe comportarsi in modo simile: le connessioni nella DNN non dovrebbero diventare troppo forti per evitare l’overfitting nei dati di addestramento.

Tecnicamente, un modello Keras ha livelli aggiuntivi tra i livelli nascosti che gestiscono il dropout. Il parametro principale è la velocità con cui vengono rilasciate le unità nascoste di un livello. Questi drops in generale si verificano in modo casuale. Ciò può essere evitato fissando il parametro seed. Mentre le prestazioni nel campione diminuiscono, anche le prestazioni fuori campione diminuiscono leggermente. Tuttavia, la differenza tra le due misure di performance è minore, che è in generale una situazione auspicabile:

- ¹ `model.evaluate(train_[cols], train['d'])`
- ² loss: 0.4495 — accuracy: 0.7869
- ³ [0.44946548342704773, 0.7869415879249573]

```

4 model.evaluate(test_[cols], test['d'])
5 loss: 0.5743 — accuracy: 0.6499
6 [0.5743284821510315, 0.6498855948448181]

```

Come illustra la figura 5.4, l'accuratezza dell'addestramento e l'accuratezza della convalida ora non si allontanano così velocemente come prima:



Figura 5.4: Valori di accuratezza nel training e nella convalida (con Dropout)

Dimenticanza intenzionale

Il Dropout nel modello Sequential di Keras emula ciò che sperimentano tutti gli esseri umani: dimenticare le informazioni precedentemente memorizzate. Ciò si ottiene disattivando alcune unità nascoste (nodi o neuroni nascosti) di uno strato nascosto durante l'addestramento. In effetti, questo spesso evita l'overfitting di una rete neurale nei dati di addestramento.

5.5 Regolarizzazione

Un altro mezzo per evitare l'overfitting è la regolarizzazione. Con la regolarizzazione, i grandi pesi nella rete neurale vengono penalizzati nel calcolo della

loss function. Ciò evita la situazione in cui determinate connessioni nella DNN diventano troppo forti e dominanti. La regolarizzazione può essere introdotta in una DNN Keras attraverso un parametro nei livelli Dense. A seconda del parametro di regolarizzazione scelto, l'addestramento e l'accuratezza del test possono essere tenuti abbastanza vicini. In genere si utilizzano due regolarizzatori, uno basato sulla norma lineare, l_1 , e uno basato sulla norma euclidea, l_2 . Il seguente codice Python aggiunge la regolarizzazione alla funzione di creazione del modello:

```
model.evaluate(train_[cols], train['d'])
loss: 0.3973 — accuracy: 0.8832
[0.3972871005535126, 0.8831614851951599]
model.evaluate(test_[cols], test['d'])
loss: 1.0083 — accuracy: 0.6522
[1.0083409547805786, 0.6521739363670349]
```

La figura 5.5 mostra l'accuratezza dell'addestramento e della convalida durante la regolarizzazione. Le due misure di performance sono molto più vicine tra loro di quanto visto in precedenza:



Figura 5.5: Valori di accuratezza nel training e nella convalida (con Regolarizzazione)

Naturalmente, il dropout e la regolarizzazione possono essere usati insieme. L'idea è che le due misure combinate evitino l'overfitting ancora meglio e avvi-

cinino i valori di accuratezza in-sample e out-of-sample. E infatti la differenza tra le due misure è minima in questo caso:

```

1 model = create_model(hl=2, hu=128,
2                         #viene aggiunto il dropout al modello
3                         dropout=True, rate=0.3,
4                         #viene aggiunta la regolarizzazione
5                         #al modello
6                         regularize=True, reg=l2(0.001),
7                         )
8
9 model.evaluate(train_[cols], train['d'])
10 loss: 0.4399 — accuracy: 0.7944
11 [0.4399223327636719, 0.79438716173172]
12 model.evaluate(test_[cols], test['d'])
13 loss: 0.6080 — accuracy: 0.6430
14 [0.6080127358436584, 0.6430205702781677]
```

La figura 5.6 mostra l'accuratezza dell'addestramento e della convalida quando si combinano il dropout con la regolarizzazione. La differenza tra l'accuratezza dei dati di addestramento e di convalida nelle epoche di addestramento è in media di soli circa quattro punti percentuali:



Figura 5.6: Valori di accuratezza nel training e nella convalida (con Dropout e Regolarizzazione)

Penalizzazione dei pesi

La regolarizzazione evita l’overfitting penalizzando i grandi pesi in una rete neurale. I singoli pesi non possono diventare così grandi da dominare una rete neurale. Le sanzioni mantengono i pesi su un livello comparabile.

5.6 Bagging

Il metodo del bagging per evitare l’overfitting è già stato utilizzato nel Capitolo 4, anche se solo per il modello di scikit-learn MLPRegressor. Esiste anche un wrapper per un modello Keras di classificazione per le DNN, vale a dire la classe KerasClassifier. Di seguito combiniamo la modellazione Keras DNN basata sul wrapper con BaggingClassifier di scikit-learn. Le misure di performance all’interno e all’esterno del campione sono relativamente elevate, circa il 70%. Tuttavia, il risultato è guidato dallo squilibrio di classe, come affrontato in precedenza, e come si riflette qui nell’alta frequenza delle previsioni 0:

```
1 model_bag.score(train_[cols], train['d'])
2 0.7514318442153494
3 model_bag.score(test_[cols], test['d'])
4 0.6636155606407322
5 test['p'] = model_bag.predict(test_[cols])
6 test['p'].value_counts()
7 0 391
8 1 46
```

Distribuire l’apprendimento

Il bagging, in un certo senso, distribuisce l’apprendimento tra una serie di reti neurali (o altri modelli) in quanto ogni rete neurale, ad esempio, vede solo alcune parti del set di dati di addestramento e solo una selezione delle features. Ciò evita il rischio che una singola rete neurale si adatti eccessivamente al set di dati di addestramento completo. La previsione si basa su tutte le reti neurali addestrate selettivamente insieme.

5.7 Ottimizzatori

Il pacchetto Keras offre una selezione di ottimizzatori che possono essere utilizzati in combinazione con il modello Sequential (vedi <https://oreil.ly/atpu8>). Diversi ottimizzatori potrebbero mostrare prestazioni diverse, per quanto riguarda sia il tempo impiegato dall’addestramento che l’accuratezza della previsione. Il seguente codice Python utilizza diversi ottimizzatori e ne confronta le prestazioni. In tutti i casi, viene utilizzata la parametrizzazione predefinita di Keras. Le prestazioni fuori campione non variano molto. Tuttavia, le prestazioni nel campione, dati i diversi ottimizzatori, variano di un ampio margine:

1	sgd	time [s]: 8.8900	in-sample=0.6346
2	rmsprop	time [s]: 8.5427	in-sample=0.7703
3	adagrad	time [s]: 8.1901	in-sample=0.6254
4	adadelta	time [s]: 9.3164	in-sample=0.3396
5	adam	time [s]: 11.3129	in-sample=0.7766
6	adamax	time [s]: 11.1541	in-sample=0.6770
7	nadam	time [s]: 12.2695	in-sample=0.7852
8			
9		out-of-sample=0.6728	
10		out-of-sample=0.6545	
11		out-of-sample=0.6613	
12		out-of-sample=0.3501	
13		out-of-sample=0.6499	
14		out-of-sample=0.6293	
15		out-of-sample=0.6682	

5.8 Conclusioni

Questo capitolo approfondisce il mondo delle DNNs e utilizza Keras come pacchetto principale. Keras offre un alto grado di flessibilità nella composizione delle DNNs. I risultati in questo capitolo sono promettenti in quanto le prestazioni sia nel campione che fuori dal campione, per quanto riguarda l’accuratezza della previsione, sono costantemente del 60% e superiori. Tuttavia, l’accuratezza della previsione è solo una faccia della medaglia. Una strategia di trading appropriata deve essere disponibile e implementabile per trarre profitto economicamente dalle previsioni, o ”segnali”. Questo argomento di fondamentale importanza nel contesto del trading algoritmico viene discusso nel capitolo successivo.

Capitolo 6

Backtest Vettorializzato

“Successo significa realizzare profitti ed evitare le perdite.”

—Martin Zweig

Il capitolo precedente si occupa della scoperta delle inefficienze statistiche nei mercati finanziari mediante l'uso di tecniche di deep learning. Questo capitolo, invece, si occupa di identificare e sfruttare le inefficienze economiche per le quali le inefficienze statistiche sono in generale un prerequisito.

Lo strumento preferito per sfruttare le inefficienze economiche è il *trading algoritmico*, ovvero l'esecuzione automatizzata di strategie di trading basate su previsioni generate da un bot di trading.

Il trading algoritmico è un campo vasto e comprende diversi tipi di strategie di trading. Alcuni, ad esempio, cercano di minimizzare l'impatto sul mercato durante l'esecuzione di ordini di grandi dimensioni (algoritmi di liquidità). Altri cercano di replicare il più fedelmente possibile il payoff degli strumenti derivati (copertura dinamica/replica). Questi esempi illustrano che non tutte le strategie di trading algoritmico hanno l'obiettivo di sfruttare le inefficienze economiche. Ai fini di questa tesi, l'attenzione alle strategie di trading algoritmico risultanti dalle previsioni fatte da un bot di trading (ad esempio, sotto forma di un agente DNN) sembra appropriato e utile.

Questo capitolo riguarda il backtest vettorializzato delle strategie di trading algoritmico, come quelle basate sulle DNN per la previsione del mercato.

Il termine *backtesting¹ vettorializzato* si riferisce a un approccio tecnico al backtesting di strategie di trading algoritmico, come quelle basate su una rete neurale densa o profonda (DNN) per la previsione del mercato. *Vettorializzato*

¹Il backtesting è una modalità per analizzare le possibili prestazioni di una strategia di trading, applicandola a gruppi di dati storici del mondo reale. I risultati del test ti aiuteranno a scegliere la strategia più adatta per ottenere i risultati migliori.

in questo contesto si riferisce a un paradigma di programmazione che si basa pesantemente o addirittura esclusivamente su codice vettorializzato (ovvero codice senza alcun ciclo a livello di Python). La vettorizzazione del codice è una buona pratica con pacchetti come Numpy o pandas in generale ed è stato utilizzato intensamente anche nei capitoli precedenti. I vantaggi del codice vettorializzato sono un codice più conciso e di facile lettura, nonché un'esecuzione più rapida in molti scenari importanti.

Avere a disposizione un buon predittore basato sull'intelligenza artificiale che batte un semplice predittore di base è importante ma generalmente non è sufficiente per generare alfa (ovvero rendimenti superiori al mercato, possibilmente aggiustati per il rischio). Ad esempio, è anche importante per una strategia di trading basata sulla previsione prevedere correttamente i grandi movimenti di mercato e non solo la maggior parte dei movimenti di mercato (potenzialmente piuttosto piccoli). Il backtest vettorializzato è un modo facile e veloce per capire il potenziale economico di una strategia di trading.

6.1 Backtesting di una strategia basata sulle SMA

Media Mobile Semplice

Una media mobile semplice (SMA) è una media mobile aritmetica calcolata sommando i prezzi recenti e quindi dividendo tale cifra per il numero di periodi di tempo nella media di calcolo. Ad esempio, si potrebbe sommare il prezzo di chiusura di un titolo per un certo numero di periodi di tempo e poi dividere questo totale per lo stesso numero di periodi. Le medie a breve termine rispondono rapidamente alle variazioni del prezzo del titolo sottostante, mentre le medie a lungo termine reagiscono più lentamente.

Una media mobile semplice è personalizzabile perché può essere calcolata per un numero diverso di periodi di tempo. Questo viene fatto sommando il prezzo di chiusura del titolo per un certo numero di periodi di tempo e quindi dividendo questo totale per il numero di periodi di tempo, che fornisce il prezzo medio del titolo nel periodo di tempo.

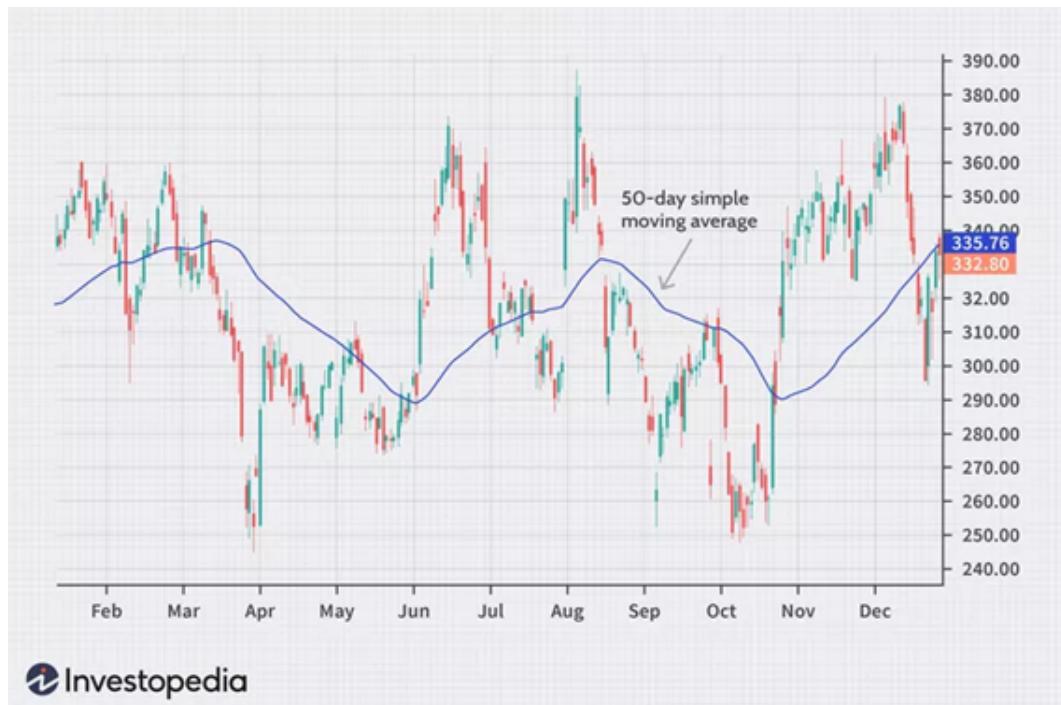


Figura 6.1: Media mobile semplice (SMA)

La formula della media mobile è semplice:

- $SMA = (p_1 + p_2 + p_3 + p_4 + \dots)/n$
- $SMA = p_1(0, 25) + p_2(0, 23) + p_3(0, 30) + p_4(0, 28) + p_5(0, 29)/n(5)$
- $SMA = 0, 27$

Dove:

- p – un prezzo di chiusura del periodo scelto
- n – numero di periodi di tempo

Una media mobile semplice attenua la volatilità e rende più facile visualizzare l'andamento del prezzo di un titolo. Se la media mobile semplice punta verso l'alto, significa che il prezzo del titolo sta aumentando. Se punta verso il basso, significa che il prezzo del titolo sta diminuendo. Più lungo è il periodo di tempo per la media mobile, più uniforme sarà la media mobile semplice. Una media mobile a breve termine è più volatile, ma la sua lettura è più vicina ai dati di origine.

Questa sezione introduce il backtest vettorializzato basato su una classica strategia di trading che utilizza le medie mobili semplici (SMA) come indicatori tecnici. Il codice seguente realizza le configurazioni necessarie e recupera i dati EOD per la coppia di valute EUR/USD:

```

1 #recupera i dati EOD per EUR/USD
2 url = 'http://hilpisch.com/aiif_eikon_eod_data.csv'
3 symbol = 'EUR='
4 data = pd.DataFrame(pd.read_csv(url, index_col=0,
5                               parse_dates=True).dropna()[symbol])
6
7 data.info()
8 DatetimeIndex: 2516 entries, 2010-01-04 to 2019-12-31
9 Data columns (total 1 columns):
10 #   Column Non-Null Count Dtype
11 --   --   --   --
12 0   EUR=    2516 non-null   float64

```

L'idea della strategia è la seguente. Calcola per un periodo più breve SMA1, diciamo per 42 giorni, e uno per un periodo più lungo SMA2, diciamo per 258 giorni. Ogni volta SMA1 che è superiore a SMA2, vai long (acquista) sullo strumento finanziario. Ogni volta SMA1 che è inferiore a SMA2, vai short (vendi) sullo strumento finanziario. Poiché l'esempio è basato su EUR/USD, andare long o short è facilmente realizzabile.

Il seguente codice Python calcola in modo vettoriale i valori SMA e visualizza le serie temporali risultanti insieme alle serie temporali originali (vedi figura 6.2):

```

1 #calcola la SMA piu' breve
2 data['SMA1'] = data[symbol].rolling(42).mean()
3
4 #calcola la SMA piu' lunga
5 data['SMA2'] = data[symbol].rolling(258).mean()
6 #mostra le tre serie temporali
7 data.plot(figsize=(10, 6));

```

Equipaggiati con i dati delle serie temporali SMA, le posizioni (long o short) risultanti possono, sempre in modo vettorializzato, essere derivate. Si noti lo spostamento di un giorno della posizione delle serie temporali risultanti per evitare errori di previsione nei dati. Questo spostamento è necessario poiché il calcolo degli SMA include i valori di chiusura dello stesso giorno. Pertanto, la posizione derivata dai valori SMA di un giorno deve essere applicata al giorno successivo per l'intera serie temporale.



Figura 6.2: Dati di serie temporali per EUR/USD e SMA

La figura 6.3 mostra le posizioni risultanti come sovrapposizione alle altre serie temporali:



Figura 6.3: Dati di serie temporali per EUR/USD, SMA e posizioni risultanti

Manca un passaggio cruciale: la combinazione delle posizioni con i rendimenti dello strumento finanziario. Poiché le posizioni sono opportunamente rappresentate da +1 per una posizione long e da -1 per una posizione short, questo passaggio si riduce alla moltiplicazione di due colonne dell'oggetto DataFrame, di nuovo in modo vettoriale. La strategia di trading basata sulla SMA supera di gran lunga l'investimento di riferimento passivo, come illustra la figura ??:



Figura 6.4: Performance lorda dell'investimento benchmark passivo e della strategia SMA

Finora, i dati sulla performance non tengono conto dei costi di transazione. Questi sono, ovviamente, un elemento cruciale quando si giudica il potenziale economico di una strategia di trading. Nella configurazione attuale, i costi di transazione proporzionali possono essere facilmente inclusi nei calcoli. L'idea è di determinare quando ha luogo uno scambio e di ridurre la performance della strategia di trading di un certo valore per tenere conto del relativo spread bid-ask (differenza tra domanda e offerta). Come è ovvio dalla figura 6.3, la strategia di trading non cambia posizione troppo spesso. Pertanto, al fine di avere alcuni effetti significativi dei costi di transazione, si presume che siano un po' più alti di quelli generalmente osservati per EUR/USD. L'effetto netto della sottrazione dei costi di transazione è di pochi punti percentuali al di sotto delle ipotesi fornite (vedi figura 6.5):



Figura 6.5: Performance linda della strategia SMA prima e dopo i costi di transazione

E il rischio derivante dalla strategia di trading? Per una strategia di trading basata su previsioni direzionali e che assume solo posizioni long o short, il rischio, espresso come volatilità (deviazione standard dei rendimenti logaritmici), è esattamente lo stesso dell’investimento passivo di riferimento:

```

1 #volatilita' giornaliera
2 data[['r', 's', 's_']].std()
3 r 0.0054
4 s 0.0054
5 s_- 0.0054
6
7 #volatilita' annualizzata
8 data[['r', 's', 's_']].std() * math.sqrt(252)
9 r 0.0853
10 s 0.0853
11 s_- 0.0855

```

6.2 Backtesting di una strategia giornaliera basata su DNN

La sezione precedente espone il progetto per il backtest vettorializzato sulla base di una strategia di trading semplice e facile da visualizzare. Lo stesso progetto può essere applicato, ad esempio, alle strategie di trading basate su DNN con aggiustamenti tecnici minimi. Di seguito addestriamo un modello Keras DNN, come discusso nel capitolo 5.

I dati utilizzati sono gli stessi dell'esempio precedente. Tuttavia, come nel capitolo 5, è necessario aggiungere al DataFrame diverse caratteristiche e relativi ritardi.

Sulla base di una suddivisione sequenziale del test di addestramento dei dati storici, addestriamo prima il modello DNN in base ai dati delle caratteristiche normalizzate:

¹ [#valuta le prestazioni del modello sui dati di addestramento](#)

² `model.evaluate(train_[cols], train['d'])`

³ `loss: 0.6748 — accuracy: 0.5951`

Finora, questo sostanzialmente ripete l'approccio centrale del capitolo 5. Ora è possibile applicare il backtest vettorializzato per giudicare la performance economica della strategia di trading basata su DNN nel campione in base alle previsioni del modello (vedi figura 6.6). In questo contesto, una previsione al rialzo viene naturalmente interpretata come una posizione long e una previsione al ribasso come una posizione short:



Figura 6.6: Rendimento lordo dell’investimento benchmark passivo e della strategia DNN giornaliera (nel campione)

La prossima è la stessa sequenza di calcoli per il set di dati di test. Mentre la sovrapreperformance nel campione è significativa, i numeri fuori campione non sono così impressionanti ma sono comunque convincenti (vedi figura 6.7):



Figura 6.7: Rendimento lordo dell'investimento benchmark passivo e della strategia DNN giornaliera (fuori campione)

La strategia di trading basata su DNN porta a un numero maggiore di operazioni rispetto alla strategia basata su SMA. Ciò rende l'inclusione dei costi di transazione un aspetto ancora più importante nel giudicare la performance economica.

Ora presupponiamo spread bid-ask (differenza tra domanda e offerta) realistici per EUR/USD al livello di 1,2 pips (ovvero 0,00012 in termini di unità di valuta). Per semplificare i calcoli, viene calcolato un valore medio per i costi di transazione proporzionali pc in base al prezzo di chiusura medio per EUR/USD (vedi figura 6.8):



Figura 6.8: Performance linda della strategia DNN giornaliera prima e dopo i costi di transazione (fuori campione)

La strategia di trading basata su DNN sembra promettente sia prima che dopo i tipici costi di transazione. Tuttavia, una strategia simile sarebbe economicamente praticabile anche intraday, quando si osservano ancora più scambi? La sezione successiva analizza una strategia intraday basata su DNN.

6.3 Backtesting di una strategia intraday basata su DNN

Per eseguire il training e il backtest di un modello DNN sui dati intraday, è necessario un altro set di dati:

```

1 #recupera i dati intraday per EUR/USD e
2 #seleziona i prezzi di chisura
3 url = 'http://hilpisch.com/aiif_eikon_id_eur_usd.csv'
4
5 symbol = 'EUR='
6 data = pd.DataFrame(pd.read_csv(url, index_col=0,
7                               parse_dates=True).dropna()[['CLOSE']])
8 data.columns = [symbol]
9
10 #ricampiona i dati in barre di cinque minuti

```

```

11 data = data.resample('5min', label='right').last().ffill()
12 data.info()
13 DatetimeIndex: 26486 entries, 2019-10-01 00:05:00 to
14 2019-12-31 23:10:00
15 Freq: 5T
16 Data columns (total 1 columns):
17 #   Column   Non-Null Count   Dtype  
18 --- 
19  0   EUR=     26486 non-null    float64

```

La procedura della sezione precedente può ora essere ripetuta con il nuovo set di dati.

```

model.evaluate(train_[cols], train['d'])
loss: 0.5254 — accuracy: 0.6697

```

Nel campione, le prestazioni sembrano promettenti, come illustrato nella figura 6.9:

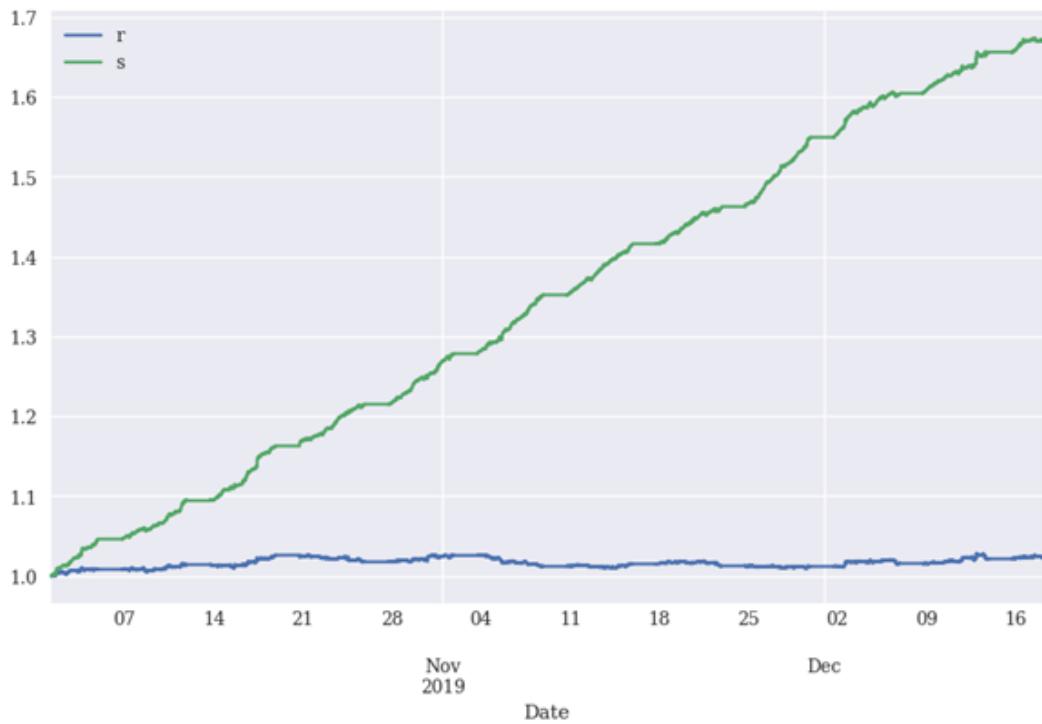


Figura 6.9: Rendimento lordo dell'investimento benchmark passivo e della strategia intraday DNN (nel campione)

Fuori dal campione, anche la performance sembra promettente prima dei costi di transazione. La strategia sembra sistematicamente sovrapreformare l'investimento di riferimento passivo (vedi figura 6.10):

```
1 model.evaluate(test_[cols], test['d'])
2 loss: 0.5336 — accuracy: 0.6572
```

La cartina di tornasole finale per quanto riguarda la pura performance economica avviene quando si aggiungono i costi di transazione. La strategia porta a centinaia di operazioni in un periodo di tempo relativamente breve. Come suggerisce la seguente analisi, sulla base degli spread bid-ask standard al dettaglio, la strategia basata su DNN non è praticabile.



Figura 6.10: Performance lorda dell'investimento benchmark passivo e della strategia intraday DNN (fuori campione)

Riducendo lo spread a un livello che potrebbero essere raggiunti da trader professionisti ad alto volume, la strategia continua a non andare in pareggio, ma piuttosto perde gran parte dei profitti a causa dei costi di transazione (vedi figura 6.11):

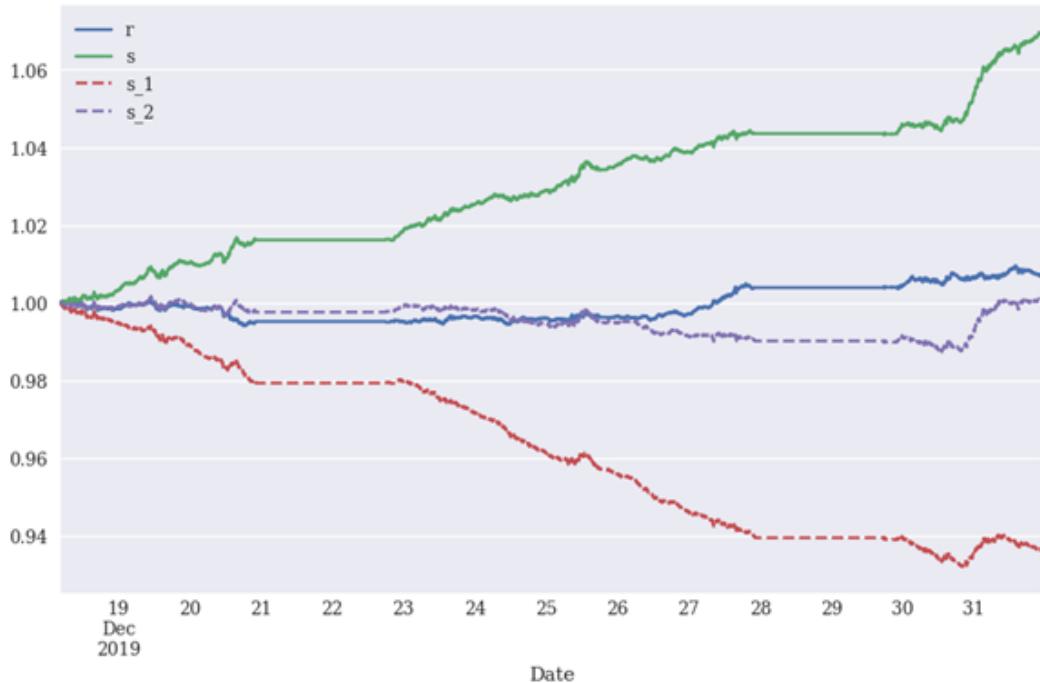


Figura 6.11: Performance lorda della strategia intraday DNN prima e dopo costi di transazione superiori/inferiori (fuori campione)

Trading Intraday

Il trading algoritmico infragiornaliero nella forma discussa in questo capitolo sembra spesso allettante da un punto di vista statistico. Sia nel campione che fuori dal campione, il modello DNN raggiunge un'elevata precisione nella previsione della direzione del mercato. Escludendo i costi di transazione, ciò si traduce anche sia in-sample che out-of-sample in una significativa sovrapreperformance della strategia basata su DNN rispetto all'investimento di riferimento passivo. Tuttavia, l'aggiunta dei costi di transazione al mix riduce considerevolmente le prestazioni della strategia basata su DNN, rendendola impraticabile per i tipici spread bid-ask al dettaglio e non molto attraente per spread bid-ask più bassi e ad alto volume.

6.4 Conclusioni

Il backtest vettorializzato si rivela un approccio efficiente e prezioso per il backtest delle prestazioni delle strategie di trading algoritmico basate sul-

l'intelligenza artificiale. Questo capitolo spiega innanzitutto l'idea alla base dell'approccio basato su un semplice esempio che utilizza due SMA per derivare i segnali. Ciò consente una semplice visualizzazione della strategia e delle posizioni risultanti. Procede quindi con il backtesting di una strategia di trading basata su DNN, come discusso in dettaglio nel capitolo 5, in combinazione con i dati EOD.

Sia prima che dopo i costi di transazione, le *inefficienze statistiche* scoperte nel capitolo 5 si traducono in *inefficienze economiche*, il che significa strategie di trading redditizie.

Quando si utilizzano gli stessi approcci di backtest vettorializzato con dati intraday, la strategia DNN mostra anche una significativa sovraperformance sia all'interno che all'esterno del campione rispetto all'investimento di riferimento passivo, almeno prima dei costi di transazione. L'aggiunta dei costi di transazione al backtest dimostra che questi devono essere piuttosto bassi, a un livello spesso non raggiunto nemmeno dai grandi trader professionisti, per rendere la strategia di trading economicamente sostenibile.

Parte IV

Metodi di IA & ML e applicazioni in Finanza

Capitolo 7

IA & ML nella finanza: Panoramica

“Il dataismo sostiene che l’universo è costituito da flussi di dati e che il valore di qualsiasi fenomeno o entità è determinato dal suo contributo all’elaborazione dei dati... Il dataismo fa quindi crollare la barriera tra animali (esseri umani) e macchine, e si aspetta che gli algoritmi elettronici finiscano per decifrare e superare gli algoritmi biochimici.”

— Yuval Noah Harari (2015)

C’è una nuova ondata di machine learning e data science nella finanza e le relative applicazioni trasformeranno il settore nei prossimi decenni. Attualmente, la maggior parte delle società finanziarie, compresi gli hedge fund, le banche di investimento e al dettaglio e le società fintech, stanno adottando e investendo molto nell’apprendimento automatico.

In futuro, le istituzioni finanziarie avranno bisogno di un numero crescente di esperti di machine learning e data science.

L’apprendimento automatico è diventato più importante di recente a causa della disponibilità di grandi quantità di dati e di una potenza di calcolo più accessibile. L’uso della scienza dei dati e dell’apprendimento automatico sta esplodendo in un modo esponenziale in tutte le aree della finanza. A seguito si fornisce una breve panoramica dei diversi tipi di machine learning e di deep learning.

7.1 Machine Learning, Deep Learning, Intelligenza Artificiale e Data Science

Per la maggior parte delle persone, i termini apprendimento automatico, apprendimento profondo, intelligenza artificiale e scienza dei dati creano confusione. In effetti, molte persone usano un termine in modo intercambiabile con gli altri.

La figura 7.1 mostra le relazioni tra IA, ML, DL e Data Science. L'apprendimento automatico è un sottoinsieme dell'intelligenza artificiale che consiste in tecniche che permettono ai computer di identificare patterns nei dati e di fornire applicazioni di intelligenza artificiale.

Il deep learning invece è un sottoinsieme dell'apprendimento automatico che consente ai computer di risolvere problemi più complessi. La scienza dei dati non è esattamente un sottoinsieme dell'apprendimento automatico, ma utilizza l'apprendimento automatico, il deep learning e l'intelligenza artificiale per analizzare i dati e raggiungere conclusioni fruibili. Combina machine learning, deep learning e intelligenza artificiale con altre discipline come l'analisi dei big data e il cloud computing.

Di seguito è riportato un riepilogo dei dettagli su intelligenza artificiale, machine learning, deep learning e data science:

- **Intelligenza Artificiale**

L'intelligenza artificiale è il campo di studio attraverso il quale un computer (e i suoi sistemi) sviluppa la capacità di svolgere con successo compiti complessi che di solito richiedono intelligenza umana. Queste attività includono, ma non sono limitate a, percezione visiva, riconoscimento vocale, processo decisionale e traduzione tra lingue. L'intelligenza artificiale è generalmente definita come la scienza che consente ai computer di fare cose che richiedono intelligenza quando vengono eseguite da esseri umani.

- **Apprendimento automatico**

L'apprendimento automatico è un'applicazione dell'intelligenza artificiale che fornisce al sistema di intelligenza artificiale la capacità di apprendere automaticamente dall'ambiente e applicare tali lezioni per prendere decisioni migliori. Esistono diversi algoritmi che il machine learning utilizza per apprendere, descrivere e migliorare i dati in modo iterativo, individuare modelli e quindi eseguire azioni su questi modelli.

- **Apprendimento profondo**

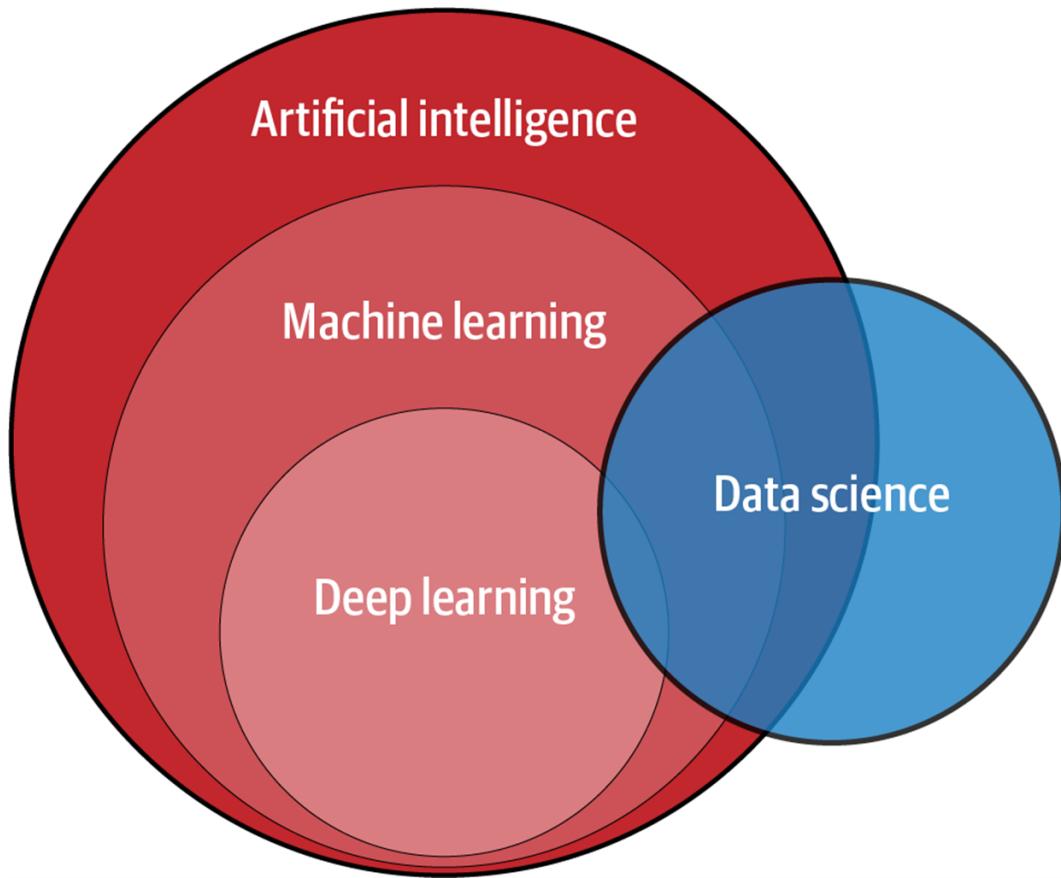


Figura 7.1: AI, ML, DL e Data Science

Il deep learning è un sottoinsieme del machine learning che prevede lo studio di algoritmi relativi a reti neurali artificiali che contengono molti blocchi (o layer) sovrapposti uno sull’altro. Il design dei modelli di deep learning si ispira alla rete neurale biologica del cervello umano. Si sforza di analizzare i dati con una struttura logica simile a come un essere umano trae conclusioni.

- **Scienza dei Dati**

La scienza dei dati è un campo interdisciplinare simile al data mining che utilizza metodi, processi e sistemi scientifici per estrarre conoscenza o intuizioni dai dati in varie forme, strutturate o non strutturate. La scienza dei dati è diversa da ML e AI perché il suo obiettivo è acquisire informazioni e comprendere i dati utilizzando diversi strumenti e tecniche scientifiche. Tuttavia, esistono diversi strumenti e tecniche comuni sia al machine learning che alla scienza dei dati.

7.2 Tipologie di apprendimento automatico

Questa sezione illustrerà tutti i tipi di apprendimento automatico che vengono utilizzati nelle varie applicazioni finanziarie. I tre tipi di machine learning, come mostrato nella figura 7.2, sono l'apprendimento supervisionato, l'apprendimento non supervisionato e l'apprendimento per rinforzo.

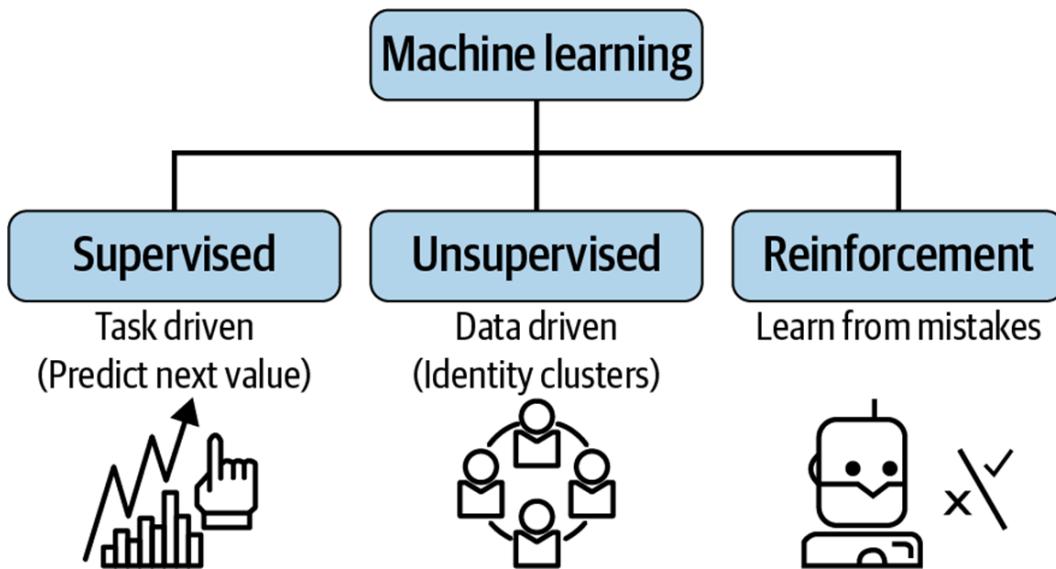


Figura 7.2: Tipologie di machine learning

7.2.1 Supervisionato

L'obiettivo principale nell'apprendimento supervisionato è addestrare un modello da dati etichettati (labels) che ci consenta di fare previsioni su dati non osservati o futuri. In questo caso, il termine supervisionato si riferisce a un insieme di campioni in cui sono già noti i segnali di output desiderati (etichette o labels). Esistono due tipi di algoritmi di apprendimento supervisionato: classificazione e regressione.

- **Classificazione**

La classificazione è una sottocategoria dell'apprendimento supervisionato in cui l'obiettivo è prevedere le etichette delle classi categoriali di nuove istanze basate su osservazioni passate.

- **Regressione**

La regressione è un'altra sottocategoria dell'apprendimento supervisionato utilizzata nella previsione di risultati continui. Nella regressione, ci viene fornito un numero di variabili predittive (esplicative) e una variabile di risposta continua (risultato o obiettivo) e proviamo a trovare una relazione tra queste variabili che ci permetta di prevedere un risultato.

Un esempio che mette a confronto la regressione con la classificazione è mostrato nella figura 7.3. Il grafico a sinistra mostra un esempio di regressione. La variabile di risposta continua è il rendimento e i valori osservati vengono tracciati rispetto ai risultati previsti. A destra, il risultato è un'etichetta di classe categoriale, indipendentemente dal fatto che il mercato sia rialzista o ribassista, ed è un esempio di classificazione.

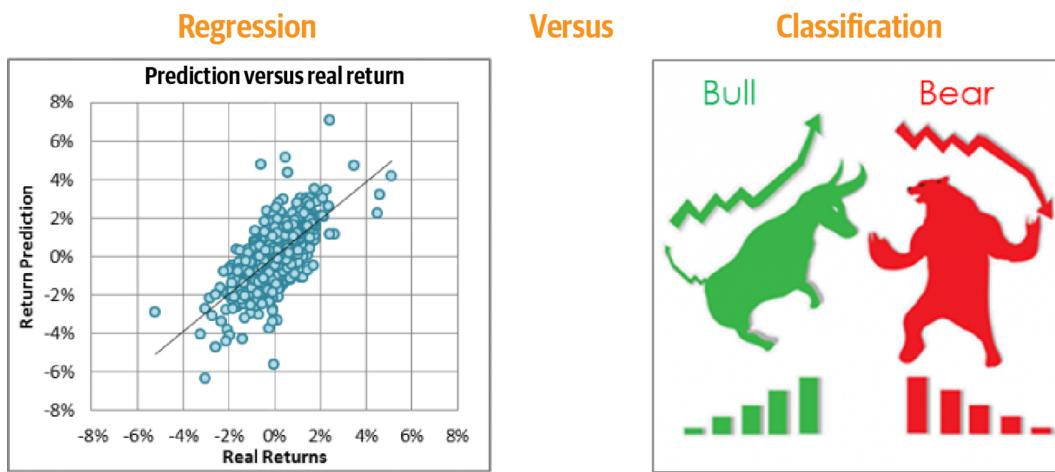


Figura 7.3: Regressione vs. Classificazione

7.2.2 Non Supervisionato

L'apprendimento non supervisionato è un tipo di apprendimento automatico utilizzato per trarre inferenze da set di dati costituiti da dati di input senza risposte etichettate. Esistono due tipi di apprendimento non supervisionato: la riduzione della dimensionalità e il clustering.

- **Riduzione della dimensionalità**

La riduzione della dimensionalità è il processo di riduzione del numero di caratteristiche, o variabili, in un set di dati preservando le informazioni e le prestazioni complessive del modello. È un modo comune e potente per gestire set di dati che hanno un gran numero di dimensioni. La figura

7.4 illustra questo concetto, in cui la dimensione dei dati viene convertita da due dimensioni (X_1 e X_2) a una dimensione (Z_1). Z_1 trasmette informazioni simili incorporate in X_1 e X_2 e riduce la dimensione dei dati.

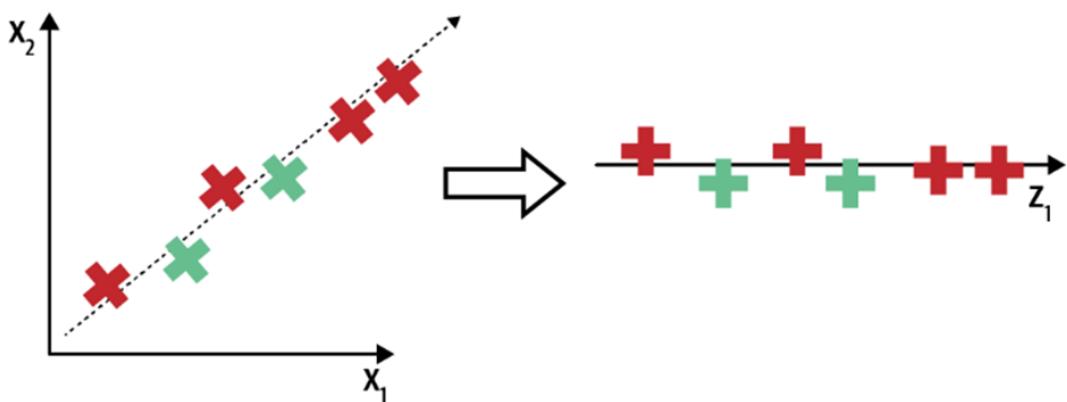


Figura 7.4: Riduzione della dimensionalità

- **Clustering**

Il clustering è una sottocategoria delle tecniche di apprendimento non supervisionato che ci consente di scoprire strutture nascoste nei dati. L'obiettivo del clustering è trovare un raggruppamento naturale nei dati in modo che gli elementi nello stesso cluster siano più simili tra loro rispetto a quelli di cluster diversi. Un esempio di clustering è mostrato nella figura 7.5, dove possiamo vedere tutti i dati raggruppati in due gruppi distinti dall'algoritmo di clustering.

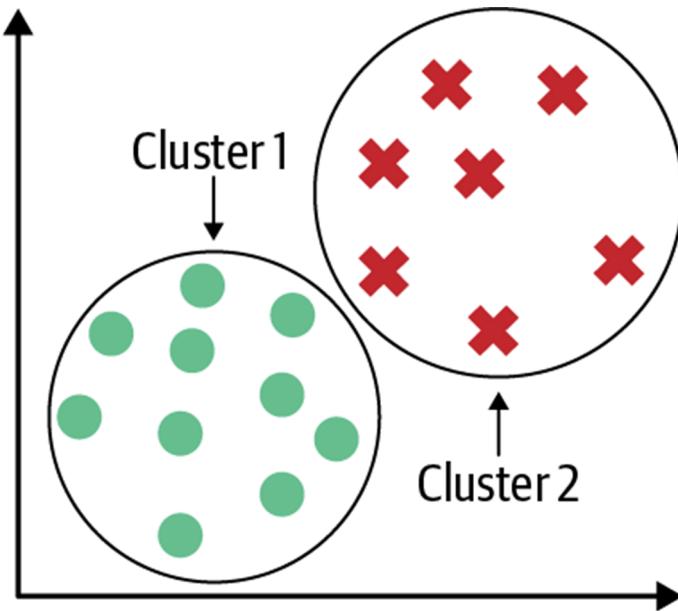


Figura 7.5: Clustering

- **Reinforcement Learning**

Imparare dalle esperienze e dalle relative ricompense o punizioni è il concetto centrale alla base dell'apprendimento per rinforzo (RL). Si tratta di intraprendere azioni adeguate a massimizzare la ricompensa in una situazione particolare. Il sistema di apprendimento, chiamato agente, può osservare l'ambiente, selezionare ed eseguire azioni e ricevere in cambio ricompense (o penalità sotto forma di ricompense negative), come mostrato nella figura 7.6. L'apprendimento per rinforzo differisce dall'apprendimento supervisionato in questo modo: nell'apprendimento supervisionato, i dati di addestramento hanno la chiave di risposta; quindi, il modello viene addestrato con le disponibili risposte corrette. Nell'apprendimento per rinforzo non esiste una risposta esplicita. Il sistema di apprendimento (agente) decide cosa fare per eseguire il compito assegnato e apprende se si trattava di un'azione corretta in base alla ricompensa. L'algoritmo determina la chiave di risposta attraverso la sua esperienza.

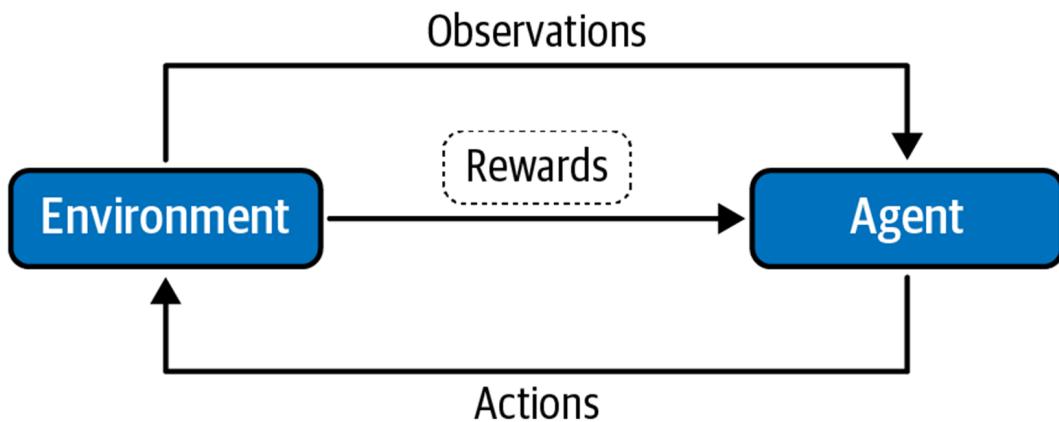


Figura 7.6: Reinforcement Learning

Le fasi dell'apprendimento per rinforzo sono le seguenti:

1. In primo luogo, l'agente interagisce con l'ambiente eseguendo un'azione.
2. Quindi l'agente riceve una ricompensa in base all'azione eseguita.
3. Sulla base della ricompensa, l'agente riceve un'osservazione e capisce se l'azione è stata buona o cattiva. Se l'azione è stata buona, ovvero l'agente ha ricevuto una ricompensa positiva, allora l'agente preferirà eseguire quell'azione. Se la ricompensa era meno favorevole, l'agente provverà a eseguire un'altra azione per ricevere una ricompensa positiva. È fondamentalmente un processo di apprendimento per tentativi ed errori.

7.2.3 Natural Language Processing

L'elaborazione del linguaggio naturale (NLP) è una branca dell'intelligenza artificiale che si occupa dei problemi di far comprendere a una macchina la struttura e il significato del linguaggio naturale utilizzato dagli esseri umani. Diverse tecniche di machine learning e deep learning sono utilizzate all'interno dell'NLP.

La NLP ha molte applicazioni nei settori finanziari in aree come l'analisi del sentimento, i chatbots e l'elaborazione dei documenti. Molte informazioni, come rapporti sulle vendite, chiamate sugli utili e titoli di giornali, vengono comunicate tramite SMS, rendendo l'NLP piuttosto utile nel dominio finanziario.

Data l'ampia applicazione degli algoritmi di NLP basati sull'apprendimento automatico in finanza, il seguente capitolo (Capitolo 8) è dedicato all'NLP e ai relativi casi di studio.

7.3 Reti Neurali Artificiali (ANNs)

Esistono molti tipi diversi di modelli utilizzati nell'apprendimento automatico. Tuttavia, una classe di modelli di machine learning che si distingue sono le reti neurali artificiali (ANN). Le ANN sono sistemi di calcolo basati su una raccolta di unità o nodi connessi chiamati neuroni artificiali, che modellano vagamente i neuroni in un cervello biologico. Ogni connessione, come le sinapsi in un cervello biologico, può trasmettere un segnale da un neurone artificiale a un altro. Un neurone artificiale che riceve un segnale può elaborarlo e quindi segnalare ulteriori neuroni artificiali ad esso collegati.

Il deep learning implica lo studio di algoritmi complessi relativi all'ANN. La complessità è attribuita a modelli elaborati di come le informazioni fluiscono attraverso il modello. Il deep learning ha la capacità di rappresentare il mondo come una gerarchia nidificata di concetti, con ogni concetto definito in relazione a un concetto più semplice. Le tecniche di deep learning sono ampiamente utilizzate nelle applicazioni di elaborazione del linguaggio naturale come vedremo dettagliatamente nel capitolo successivo.

Tra gli algoritmi più diffusi ed utilizzati nel deep learning ci sono le reti neurali convoluzionali o convolutive (CNNs). Le CNNs, note anche come ConvNet, sono costituite da più livelli e sono utilizzate principalmente per l'elaborazione delle immagini e il rilevamento di oggetti. Yann LeCun ha sviluppato la prima CNN nel 1988 che si chiamava LeNet. È stata utilizzata per riconoscere caratteri come codici postali e cifre.

Come vedremo più approfonditamente nel capitolo 9, le CNN sono ampiamente utilizzate per identificare immagini satellitari, prevedere serie temporali e rilevare anomalie e possono rivelarsi molto utili per la costruzione di una strategia di trading.

7.4 Applicazioni dell'Intelligenza Artificiale nella finanza

Nell'introduzione della tesi sono state esposte in maniera riassuntiva solo alcune delle applicazioni che l'apprendimento automatico trova nel settore finanziario. Ora verranno esposte in maniera più approfondita queste applicazioni. Come è stato già detto, ma è importante ribadirlo, l'apprendimento automatico nella finanza è diventato più importante di recente a causa della disponibilità di grandi quantità di dati e di una potenza di calcolo più accessibile. Le principali banche e società di servizi finanziari stanno implementando la tecnologia AI, incluso il machine learning, per semplificare i loro processi, ottimizzare i portafogli, ridurre il rischio e sottoscrivere prestiti, tra le altre cose.

In questa sezione esploreremo alcuni importanti modi in cui il machine learning sta trasformando il settore dei servizi finanziari ed esempi di applicazioni reali del machine learning nella finanza.

Per comprendere il ruolo del machine learning nella finanza, dobbiamo prima capire perché il machine learning è adatto alla finanza.

Perché l'apprendimento automatico è adatto alla finanza?

L'apprendimento automatico riguarda la digestione di grandi quantità di dati e l'apprendimento da tali dati su come svolgere un'attività specifica, ad esempio, come distinguere documenti legali fraudolenti da documenti autentici.

L'apprendimento automatico nella finanza è l'utilizzo di una varietà di tecniche per gestire in modo intelligente volumi di informazioni grandi e complessi.

Il machine learning eccelle nella gestione di questi grandi e complessi volumi di dati, qualcosa che il settore finanziario ha in eccesso. Con la natura quantitativa del settore finanziario e i grandi volumi di dati storici disponibili, l'apprendimento automatico nella finanza è pronto a migliorare diversi aspetti del settore.

Questo è il motivo per cui così tante istituzioni finanziarie stanno investendo molto nella ricerca e nello sviluppo del machine learning.

L'applicazione di algoritmi di apprendimento automatico per prevedere le performance finanziarie, rilevare le frodi e prevedere l'andamento delle azioni ha reso l'apprendimento automatico una competenza richiesta per la crescita della carriera per chiunque lavori nel settore bancario e finanziario.

Dunque, la tecnologia è arrivata a svolgere un ruolo fondamentale in molte fasi dell'ecosistema finanziario. Di seguito sono riportate alcune delle attuali applicazioni dell'apprendimento automatico nella finanza.

7.4.1 Trading Algoritmico

Il trading algoritmico (o semplicemente algo trade), come abbiamo già visto nella parte III, è l'uso di algoritmi per condurre operazioni in modo autonomo.

Questo è un altro esempio di come le aziende utilizzano l'apprendimento automatico nella finanza. Nel trading algoritmico, i computer eseguono programmi con una serie predeterminata di istruzioni (un algoritmo) per effettuare un'operazione per conto di un trader.

Queste istruzioni di solito coinvolgono parametri come tempistica, prezzo, quantità o altri vincoli. Il trading algoritmico consente l'esecuzione di un ordine di grandi dimensioni inviando al mercato a intervalli piccoli incrementi dell'ordine, chiamati "ordini figlio".

Quindi, sono principalmente i gestori di hedge fund che fanno uso di sistemi di trading automatizzati e quindi fanno uso dell'apprendimento automatico nella finanza.

L'apprendimento automatico è parte integrante dei vantaggi dei programmi algoritmici. Consente ai trader di automatizzare determinati processi garantendo un vantaggio competitivo. Il sistema consente inoltre di operare su più mercati, aumentando le opportunità di trading.

Inoltre, gli algoritmi sono in grado di apprendere e adattarsi ai cambiamenti in tempo reale, un altro vantaggio competitivo per quelle istituzioni che adottano il machine learning nella finanza. I sistemi algoritmici coinvolti qui sono un aiuto fenomenale per i trader. Ad esempio, gli algoritmi non sono sentimentali o emotivi, attributi che così spesso sabotano le aspirazioni umane quando si tratta di investimenti. Il trading algoritmico, quindi, semplifica il processo decisionale eludendo le emozioni umane.

Questo è un vantaggio cruciale dell'utilizzo dell'apprendimento automatico nella finanza.

Hedge Fund guidato dall'intelligenza artificiale

Un hedge fund guidato dall'intelligenza artificiale che effettua scambi di azioni senza intervento umano è l'ultima applicazione dell'apprendimento automatico nella finanza.

Aidyia, con sede a Hong Kong, utilizza algoritmi per condurre operazioni in modo autonomo. Ben Goertzel, esperto di intelligenza artificiale e fondatore e CEO di SingularityNet, un marketplace di intelligenza artificiale basato su blockchain, è il capo scienziato dell'azienda. Aidyia gestisce un hedge fund che utilizza l'intelligenza artificiale escludendo gli esseri umani per prendere tutte le decisioni di compravendita di azioni.

Gli esseri umani hanno costruito il sistema, ma il sistema funziona completamente da solo senza interferenze umane. Al lancio dell'hedge fund automatizzato Goertzel ha notoriamente osservato: "Se morissimo tutti, continuerebbe a fare trading". L'azienda utilizza una serie di capacità di intelligenza artificiale, tra cui una ispirata all'evoluzione genetica e un'altra alla logica probabilistica, per fare previsioni sul mercato e condurre operazioni in proprio.

Questo non è un nuovo sviluppo poiché l'intelligenza artificiale e l'apprendimento automatico fanno parte da molti anni di molte strategie di hedge fund, tuttavia, questo è il primo hedge fund completamente autonomo. Tech Revolution spiega che il sistema funziona fondamentalmente per trovare la popolazione di trader intelligenti definitiva, testando continuamente le prestazioni dei loro trader di azioni digitali. Il sistema conserva solo i "geni" dei miglio-

ri performer per creare una squadra di trader imbattibili. E questo processo continua all'infinito.

Sebbene l'algo trading semplifichi le cose per trader e gestori di fondi, scrivere un algoritmo di trading elettronico è un'impresa incredibilmente complicata.

Secondo Sigmoidal, una società di consulenza in machine learning di Varsavia sarà difficile per un individuo implementare con successo una strategia di investimento ML.

La ragione di ciò è che sarebbe necessario accedere a professionisti di grande talento con esperienza nel trading e nella scienza dei dati per sviluppare un algoritmo di trading. A questo proposito, le grandi banche di investimento e altri istituti finanziari trarranno maggiori benefici dall'apprendimento automatico nella finanza rispetto ai singoli individui.

Secondo un rapporto di ValueWalk, gli hedge fund di machine learning superano già in modo significativo gli hedge fund generalizzati, così come i fondi quantistici tradizionali.

Secondo l'edizione di luglio 2018 dell'Hedge Fund Sentiment Survey, più della metà dei gestori di hedge fund utilizza AI/ML per informare le decisioni di investimento; due terzi utilizzano AI/ML per generare idee di trading e ottimizzare i portafogli e più di un quarto utilizza l'automazione per eseguire operazioni.

7.4.2 Intercettazione delle frodi

Le frodi sono un enorme problema per le istituzioni finanziarie e uno dei motivi principali per sfruttare l'apprendimento automatico nella finanza. Le perdite per frode subite da banche e commercianti su tutte le carte di credito, di debito e prepagate per uso generico e privato emesse a livello globale ammontavano a 16,74 miliardi di sterline (21,84 miliardi di dollari) nel 2015, secondo un rapporto di Bloomberg. L'apprendimento automatico è ideale per combattere le transazioni finanziarie fraudolente. Questo perché i sistemi ML possono eseguire la scansione di vasti set di dati, rilevare attività insolite (anomalie) e contrassegnarle all'istante.

Falsi positivi

ML è anche il candidato perfetto per affrontare il problema dei falsi positivi, cosa che accade regolarmente in finanza. I falsi positivi, noti anche come "falsi rifiuti", si verificano quando i commercianti o gli istituti finanziari rifiutano erroneamente richieste di transazioni finanziarie legittime. I falsi positivi

rifiutati dalle carte sono un enorme punto dolente per gli istituti finanziari che rischiano di perdere la fedeltà dei clienti quando un'azienda rifiuta erroneamente le carte dei clienti. Nel 2015 Javelin Strategy and Research ha riferito che almeno il 15% di tutti i titolari di carte aveva almeno una transazione rifiutata in modo errato nell'anno precedente, il che rappresentava una perdita di entrate annuale per un totale di quasi 118 miliardi di dollari. Per le aziende, ciò significa perdita di entrate e diminuzione della fedeltà dei clienti. Identity-Mind Global fa parte di un numero crescente di società di intelligenza artificiale che aiutano commercianti, istituti finanziari e fornitori di servizi di pagamento a identificare i truffatori. Aiutare a combattere le frodi è un'eccellente applicazione dell'apprendimento automatico nella finanza. IdentityMind Global ha brevettato un software basato sull'apprendimento automatico chiamato DNA elettronico (eDNA) che utilizza più di 50 features di dati per stabilire l'identità di un individuo.

La società afferma che il suo servizio consente alle aziende di eseguire il controllo dell'identità, l'autenticazione basata sul rischio e l'identificazione normativa, prevenendo così le frodi sull'identità. Ancora più importante, il loro monitoraggio integrato delle transazioni consente di contrastare anche il riciclaggio di denaro e il finanziamento del terrorismo. Un'altra azienda che opera nel settore della verifica digitale è Socure.

L'azienda fornisce dati digitali in tempo reale e correla migliaia di data points (features e labels) online e offline per creare un'autentica identità del cliente. Socure ha sviluppato un bot chiamato Aida (Authentic Identity Agent) per aiutare a stabilire la fiducia nelle transazioni online. Aida utilizza l'intelligenza artificiale per elaborare miliardi di data points multidimensionali online e offline al secondo per convalidare l'autenticità delle identità digitali in tempo reale.

La società canadese Trulioo è una società globale di verifica dell'identità che fornisce un'identità elettronica istantanea e la verifica dell'indirizzo. L'azienda utilizza un software per confrontare le informazioni sull'identità di un individuo (nome completo, numero di telefono, indirizzo, ecc.) con i risultati del database come agenzie di credito, agenzie governative o altre fonti per verificare l'identità di un individuo. Questi tipi di servizi esemplificano i vantaggi dell'apprendimento automatico nella finanza.

7.4.3 Previsione del riciclaggio di denaro

Secondo un rapporto delle Nazioni Unite, si stima che la quantità di denaro riciclato a livello globale in un anno sia pari al 2-5% del PIL globale, ovvero dai 800 miliardi ai 2 trilioni di dollari. Se il riciclaggio di denaro fosse un paese, sarebbe la quinta economia più grande del mondo. Chiaramente, le

banche e gli altri istituti finanziari hanno molto da fare. Il gigante bancario HSBC prevede di incorporare la tecnologia di apprendimento automatico nella sua infrastruttura nel tentativo di combattere il riciclaggio di denaro. Utilizzando il software di Quantexa, HSBC valuterà miliardi di dati provenienti da fonti interne ed esterne. Il software AI raccoglierà dati interni, pubblicamente esistenti e transazionali dalla rete più ampia di un cliente nel tentativo di individuare segnali di riciclaggio di denaro. L'apprendimento automatico ha consentito alle istituzioni finanziarie di passare da un modello di business tradizionale a uno più dinamico e predittivo.

Commerzbank sta applicando la tecnologia di apprendimento automatico per automatizzare i controlli di pre-conformità per le transazioni di finanza commerciale tradizionalmente basate su carta. Intende automatizzare circa l'80% di tutti i controlli basati sulla conformità relativi ai processi di finanziamento commerciale della banca entro il 2020.

La tecnologia utilizza il riconoscimento ottico dei caratteri (OCR) e l'apprendimento automatico progressivo per estrarre i dati dai documenti fisici, riconoscere i modelli e contrassegnare le deviazioni. Enno-Burghard Weitzel, responsabile dei servizi commerciali di gestione prodotti presso Commerzbank, afferma:

“L’elaborazione delle transazioni di finanza commerciale sta diventando più complessa e soggetta a rischi più elevati, poiché i processi manuali faticano a tenere il passo con le crescenti tendenze normative e di mercato. Il nostro obiettivo è concentrare l’esperienza dei nostri specialisti di finanza commerciale nelle parti cruciali e complesse del business, utilizzando l’intelligenza artificiale per migliorare l’efficienza e ottimizzare ulteriormente i controlli del rischio”.

7.4.4 Analisi dei documenti

I recenti progressi nel deep learning hanno trasformato l'accuratezza del riconoscimento delle immagini oltre le capacità umane. L'analisi dei documenti è un perfetto esempio dei vantaggi dell'apprendimento automatico nella finanza. In realtà, la velocità e la precisione di questi sistemi ML sono fenomenali. Alla JP Morgan un programma chiamato COIN ha completato 360.000 ore di lavoro in pochi secondi. Ha comportato l'analisi di 12.000 contratti di credito commerciale. COIN, che utilizza l'apprendimento automatico per interpretare i documenti, sta per Contract Intelligence. JP Morgan è un precursore nell'applicazione dell'apprendimento automatico nella finanza. L'azienda sta investendo molto nella tecnologia per automatizzare i processi: il suo budget tecnologico è di 9,6 miliardi di dollari.

La capacità dei sistemi ML di scansionare e analizzare rapidamente documenti legali e di altro tipo aiuta le banche a far fronte ai problemi di conformità e a combattere le frodi. IPSoft e Onfido sono due società di intelligenza artificiale che operano in questo spazio. Amelia di IPSoft è stato riconosciuto come uno dei migliori sistemi di intelligenza artificiale al mondo. Più di 50 grandi aziende di tutti i settori utilizzano attualmente il sistema, che è programmato per automatizzare i processi IT e aziendali. Amelia è l’“agente cliente virtuale” o “collega digitale” di IPSoft, tra le sue innumerevoli capacità, Amelia scansiona anche testi legali e normativi per problemi di conformità. La piattaforma di Onfido si collega a vari database pubblicamente disponibili per offrire ai datori di lavoro una rapida verifica dell’identità e controlli dei precedenti per cose come la guida e i precedenti penali. Onfido si descrive come il nuovo standard di identità per Internet.

La loro tecnologia basata sull’intelligenza artificiale valuta se l’ID rilasciato dal governo di un utente è autentico o fraudolento; quindi, lo confronta con la biometria del riconoscimento facciale. Il motore di verifica di Onfido utilizza database pubblicamente disponibili per fornire ai datori di lavoro una verifica tempestiva dell’identità controllando che i documenti di identità siano autentici.

7.4.5 Gestione del rischio

Le grandi società e le istituzioni finanziarie dipendono da accurate previsioni di mercato per il successo delle loro attività. I mercati finanziari utilizzano sempre più i sistemi di AI e ML per sfruttare i dati attuali per individuare le tendenze e prevedere meglio i rischi incombenti. L’apprendimento automatico nella finanza sta migliorando la gestione del rischio nel settore finanziario.

Dataminr e Alphasense sono esempi di aziende che utilizzano queste tecnologie avanzate per aiutare le istituzioni finanziarie e di altro tipo a gestire il rischio. Affermano di scoprire eventi ad alto impatto e informazioni critiche molto prima che ci siano nelle notizie. Dataminr utilizza la sua innovativa tecnologia AI per raccogliere dati e avvisare i clienti all’istante, mettendoli in grado di rispondere alle sfide in tempo reale. L’azienda ottiene informazioni su possibili eventi ad alto impatto e ultime notizie critiche dai social media pubblici in tempo reale.

Alphasense svolge il lavoro in modo diverso. L’azienda fornisce un motore di ricerca per grandi società di investimento e consulenza, banche globali e società. Questo motore di ricerca AlphaSense restringe la ricerca ai data points (features e labels) critici e alle tendenze, facendo risparmiare tempo prezioso ai clienti. Utilizza l’elaborazione del linguaggio naturale (NLP) per trovare

e tenere traccia delle informazioni pertinenti, imparando dai successi e dagli errori con ogni ricerca.

7.4.6 Chatbot

L'apprendimento automatico nella finanza ha dato origine a migliori esperienze di chatbot e quindi a una migliore esperienza del cliente. Il machine learning ha dato nuova vita all'interazione uomo-macchina che può essere a volte molto frustrante per gli esseri umani. Grazie a robusti motori di elaborazione del linguaggio naturale e alla capacità di apprendere dalle interazioni precedenti, i chatbot basati su ML sono in grado di risolvere in modo rapido e accurato le richieste dei clienti. Questi chatbot sono in grado di adattarsi a ogni cliente e ai cambiamenti nel comportamento dei clienti. Pertanto, si presentano come umani, il che è più accettabile per i clienti. Questi sistemi ottengono il loro know-how finanziario dall'analisi di un sacco di domande finanziarie da parte dei clienti.

Per i clienti, i chatbot hanno il potenziale per automatizzare le operazioni e consentire un'esperienza bancaria più snella e senza attriti. Per le istituzioni finanziarie, la tecnologia farà risparmiare manodopera e fornirà sempre informazioni corrette e aggiornate. I chatbot più user-friendly sono un esempio di machine learning in finanza applicato a vantaggio di tutti gli utenti, ovvero istituti bancari e clienti. Una società che utilizza un assistente chatbot AI per monitorare le finanze personali è Kasisto. Gli utenti possono scaricare KAI, la piattaforma AI conversazionale di Kasisto sulle piattaforme mobile, di messaggistica e web della propria banca. KAI utilizza algoritmi di apprendimento automatico e altre strategie per mettere a punto e addestrare modelli statistici basati sui dati raccolti. Anche le principali banche commerciali considerano i chatbot un vantaggio tecnologico strategico.

Ad esempio, Wells Fargo ha iniziato a pilotare un chatbot basato sull'intelligenza artificiale nell'aprile 2017. Tale chatbot comunica tramite Facebook Messenger per fornire informazioni sull'account e reimpostare le password dei clienti. Bank of America ha sviluppato il proprio bot, Erica. Erica aiuta i clienti con le transazioni di base, dà suggerimenti di risparmio e fornisce informazioni sul saldo bancario e sui pagamenti con carta di credito. I clienti possono accedere a Erica tramite l'app di mobile banking Bank of America.

HSBC (Hong Kong) ha utilizzato tecnologie di intelligenza artificiale come l'elaborazione del linguaggio naturale per sviluppare Amy, un chatbot con assistente virtuale. Amy fornisce supporto immediato alle richieste dei clienti 24 ore su 24, 7 giorni su 7 sui loro desktop e telefoni cellulari in inglese, cinese tradizionale e semplificato. Un altro sviluppo sono le aziende che sviluppano

chatbot per le banche globali da integrare nei loro siti Web e app mobili, un'eccellente applicazione dell'apprendimento automatico nella finanza.

Ad esempio, Personetics Technologies ha costruito il suo chatbot Personetics Assist sull'elaborazione del linguaggio naturale, consentendogli di avere una conversazione intelligente con i clienti sulle loro finanze. Il chatbot utilizza l'analisi predittiva per fornire consigli approfonditi. La società ha la Royal Bank of Canada come uno dei suoi clienti bancari. Un'altra azienda che opera in questo settore è Finn AI, che ha integrato processi di apprendimento automatico nell'app bancaria. L'aspetto dell'apprendimento automatico consente al software, tramite un chatbot, di apprendere e migliorare continuamente attraverso le interazioni con i clienti. Le banche che hanno implementato il bot Finn AI per i propri clienti includono Bank of Montreal, Banpro e ATB Financial.

7.4.7 Sottoscrizione di prestiti e assicurazioni

Questa è un'altra applicazione ideale dell'apprendimento automatico nella finanza. Banche e compagnie assicuratrici hanno accesso a terabyte di dati dei consumatori su cui possono essere addestrati gli algoritmi ML. Gli algoritmi possono eseguire attività automatizzate come la corrispondenza dei record di dati, la ricerca di eccezioni e il calcolo se un richiedente è idoneo per un prestito o un'assicurazione.

Gli algoritmi ML possono eseguire le stesse attività di sottoscrizione e valutazione del credito che in passato richiedevano migliaia di ore umane. Gli ingegneri informatici addestrano gli algoritmi per individuare tutti i tipi di tendenze che potrebbero influenzare le decisioni di prestito o assicurazione.

Ci sono un certo numero di aziende che eccellono in questo caso d'uso dell'apprendimento automatico nella finanza. ZestFinance a Los Angeles aiuta altre società finanziarie a valutare i richiedenti di prestito che hanno poca o nessuna storia creditizia. La loro piattaforma Zest Automated Machine Learning (ZAML) utilizza migliaia di punti dati per valutare correttamente i candidati che le istituzioni avrebbero considerato troppo rischiosi in passato.

C'è una proliferazione di società di intelligenza artificiale che si sono fatte avanti per valutare l'affidabilità creditizia dei clienti per mutui, finanziamenti e rifinanziamenti di prestiti agli studenti, progetti di miglioramento della casa, prestiti per piccole imprese e altro ancora.

Assicurazioni basate sull'intelligenza artificiale

Lemonade è un chiaro esempio di una società assicurativa che adotta un approccio automatizzato all'assicurazione. A differenza delle compagnie assi-

curative tradizionali, Lemonade utilizza completamente l'apprendimento automatico ed i chatbot per fornire servizi dalla gestione dei sinistri assicurativi, ottenendo preventivi fino alla semplificazione dell'amministrazione del back-office.

I clienti possono utilizzare l'app Lemonade iOS o Android sul proprio smartphone per stipulare una polizza, pagare i premi, apportare modifiche alla propria polizza, segnalare un incidente o presentare un reclamo. Lemonade afferma che bastano 90 secondi sull'app per essere assicurati e 3 minuti per ottenere un rimborso. Cape Analytics utilizza la visione artificiale e l'apprendimento automatico per prendere le immagini geospatiali esistenti per creare un database di informazioni sulle proprietà adeguato.

L'azienda utilizza le immagini di una casa, ottenute da un partner come Nearmap, per stabilire il valore della casa e velocizzare così il processo di quotazione per le compagnie assicurative. Ciò significa anche che le compagnie assicurative non devono inviare qualcuno per ispezionare fisicamente una proprietà.

7.4.8 Applicazioni future dell'IA in finanza

Analisi del sentimento

Gli algoritmi ML e la loro attitudine all'analisi del sentimento influenzeranno sempre più il trading in futuro. L'analisi del sentimento è un ottimo esempio di apprendimento automatico nella finanza. Implica la lettura di enormi volumi di dati non strutturati come video e trascrizioni video, foto, file audio, post sui social media, presentazioni, pagine Web, articoli, blog e documenti aziendali per determinare il sentimento del mercato. L'analisi del sentimento consente alle aziende di capire cosa dicono le persone e, soprattutto, cosa intendono con ciò che dicono.

È fondamentale per tutti i leader aziendali nel posto di lavoro di oggi e un eccellente esempio di apprendimento automatico nella finanza. Molti credono che questa tecnologia possa trasformare i futuri mercati finanziari. Laddove gli esseri umani spesso commerciano sull'intuizione, gli algoritmi ML hanno così tante informazioni a loro disposizione che non hanno bisogno dell'intuizione. Le loro previsioni si baseranno su un'analisi accurata degli eventi in tempo reale.

Cosa ci riserva il futuro?

Un nuovo rapporto del World Economic Forum , *The New Physics of Financial Services – How artificial intelligence is transforming the financial Eco-*

system, avverte che l'adozione diffusa dell'IA potrebbe introdurre nuovi rischi sistematici e per la sicurezza nel sistema finanziario. Il rapporto rileva che i primi big mover stanno offrendo le loro applicazioni AI come "servizio" ai loro concorrenti; attirare gli utenti per accelerare l'apprendimento del loro sistema e trasformare i centri di costo in centri di profitto. Man mano che questa tendenza si allarga, il sistema finanziario potrebbe affrontare nuovi rischi.

Il comunicato stampa del WEF spiega che i clienti delle banche stanno sperimentando sempre più un mondo della finanza AI "a guida autonoma". Questo sviluppo può comportare rischi sistematici e per la sicurezza. Perché? Questo nuovo mondo finanziario sarà centralizzato con solo pochi attori in rete, tra cui, potenzialmente, le aziende big tech.

Ad esempio, negli Stati Uniti, la piattaforma di investimento Aladdin di BlacRock fornisce sofisticate analisi del rischio e strumenti completi di gestione del portafoglio che sfruttano il machine learning. L'amministratore delegato di BlackRock, Larry Fink, si aspetta che Aladdin realizzi il 30% dei ricavi dell'azienda.

Il rapporto prevede che l'intelligenza artificiale accelererà anche la "corsa al ribasso" per molti prodotti, poiché il prezzo diventa altamente comparabile tramite servizi di aggregazione e servizi di terze parti che mercificano l'eccellenza del back office.

Gli istituti finanziari sfrutteranno sempre più AI e ML per differenziarsi e fornire prodotti personalizzati secondo necessità. L'apprendimento automatico nella finanza sarà centrale per questi sviluppi. Il risultato netto per i clienti sarà una "finanza autonoma" - un'esperienza del cliente in cui le finanze di un individuo o di un'azienda si gestiscono effettivamente da sole, coinvolgendo il cliente ad agire come un consulente fidato sulle decisioni importanti, afferma il comunicato stampa.

7.5 Conclusioni

Il valore dell'apprendimento automatico nella finanza sta diventando sempre più evidente, ma il vero valore a lungo termine probabilmente diventerà evidente solo nei prossimi anni. Ci sono molti casi d'uso per l'apprendimento automatico nella finanza e le banche e altri istituti finanziari stanno investendo miliardi nella tecnologia.

I loro investimenti stanno portando alle loro aziende molti vantaggi, tra cui la riduzione dei costi operativi, l'aumento dei ricavi, l'aumento della fidelizzazione dei clienti grazie a una migliore esperienza del cliente e una migliore conformità e gestione del rischio. Nel frattempo, gli algoritmi ML forniscono

consulenza sugli investimenti, combattono le frodi finanziarie, autenticano documenti, negoziano in borsa e raccolgono informazioni cruciali che potrebbero influenzare mercati e investimenti.

E mentre gli algoritmi ML sono impegnati con tutte queste attività, stanno imparando e diventando più intelligenti, avvicinando il mondo a un sistema finanziario completamente automatizzato, che equivale al massimo risultato dell'apprendimento automatico nella finanza.

Questo capitolo ha trattato diverse applicazioni dell'apprendimento automatico nella finanza, dal trading algoritmico alle sottoscrizioni dei prestiti. Alcune di queste applicazioni ritenute tra le più importanti da applicare nella finanza — in particolare l'NLP per l'analisi del sentimento che, come abbiamo detto, è e sarà di rilevante importanza nella finanza e le CNNs per le immagini satellitari e serie temporali — saranno trattate nei capitoli successivi.

Capitolo 8

Natural Language Processing (NLP)

“L’analisi del sentimento è fondamentale per tutti i leader aziendali nel posto di lavoro di oggi e un eccellente esempio di apprendimento automatico nella finanza.”

— Algorithm-XLAB

L’elaborazione del linguaggio naturale (NLP) è un sottocampo dell’intelligenza artificiale utilizzato per aiutare i computer a comprendere il linguaggio umano. La maggior parte delle tecniche di NLP si basa sull’apprendimento automatico per derivare il significato dai linguaggi umani.

I dati di testo possono essere estremamente preziosi data la quantità di informazioni che gli esseri umani comunicano e memorizzano utilizzando il linguaggio naturale. La vasta gamma di fonti di dati rilevanti per gli investimenti spazia da documenti formali come dichiarazioni aziendali, contratti o brevetti a notizie, opinioni e ricerche o commenti di analisti a vari tipi di post o messaggi sui social media.

Quando il testo è stato fornito, il computer utilizza algoritmi per estrarre il significato associato a ogni frase e raccogliere dati essenziali da esso. L’NLP si manifesta in forme diverse in molte discipline sotto vari alias, tra cui (ma non limitato a) analisi testuale, text mining, linguistica computazionale e analisi del contenuto.

Nel panorama finanziario, una delle prime applicazioni dell’NLP è stata implementata dalla Securities and Exchange Commission (SEC) degli Stati Uniti. Il gruppo ha utilizzato il text mining e l’elaborazione del linguaggio naturale per rilevare frodi contabili. La capacità degli algoritmi di NLP nello scansionare e analizzare documenti legali e di altro tipo ad alta velocità fornisce

alle banche e ad altri istituti finanziari enormi vantaggi in termini di efficienza per aiutarli a soddisfare le normative di conformità e combattere le frodi.

Nel processo di investimento, scoprire informazioni sugli investimenti richiede non solo una conoscenza approfondita della finanza, ma anche una solida conoscenza dei principi della scienza dei dati e dell'apprendimento automatico. Gli strumenti di NLP possono aiutare a rilevare, misurare, prevedere e anticipare importanti caratteristiche e indicatori del mercato, come la volatilità del mercato, i rischi di liquidità, lo stress finanziario, i prezzi delle case e la disoccupazione.

Le notizie sono sempre state un fattore chiave nelle decisioni di investimento. È ben noto che le notizie specifiche dell'azienda, macroeconomiche e politiche influenzano fortemente i mercati finanziari. Man mano che la tecnologia avanza e i partecipanti al mercato diventano più connessi, il volume e la frequenza delle notizie continueranno a crescere rapidamente. Ancora oggi, il volume di dati di testo giornalieri prodotti rappresenta un compito insostenibile anche per un grande team di ricercatori fondamentali. L'analisi fondamentale assistita dalle tecniche di NLP è ora cruciale per sbloccare il quadro completo di come gli esperti e le masse si sentono riguardo al mercato.

Nelle banche e in altre organizzazioni, i team di analisti si dedicano a esaminare, analizzare e tentare di quantificare i dati qualitativi dalle notizie e dai report richiesti dalla SEC. L'automazione che utilizza l'NLP è adatta a questo contesto. L'NLP può fornire un supporto approfondito nell'analisi e nell'interpretazione di vari rapporti e documenti. Ciò riduce lo sforzo che le attività ripetitive e di basso valore esercitano sui dipendenti umani. Fornisce inoltre un livello di obiettività e coerenza a interpretazioni altrimenti soggettive; gli errori dovuti all'errore umano sono ridotti. L'NLP può anche consentire a un'azienda di raccogliere informazioni che possono essere utilizzate per valutare il rischio di un creditore o misurare il sentimento relativo al marchio dai contenuti sul Web.

Con l'aumento della popolarità del software di live chat nelle attività bancarie e finanziarie, i chatbot basati sull'NLP sono un'evoluzione naturale. La combinazione di robo-advisor con chatbot dovrebbe automatizzare l'intero processo di gestione patrimoniale e del portafoglio.

Principali sfide nel lavorare con i dati di testo

La conversione di testo non strutturato in un formato leggibile dalla macchina richiede un'attenta preelaborazione per preservare i preziosi aspetti semantici dei dati.

Il modo in cui gli esseri umani comprendono il contenuto del linguaggio non è completamente compreso e il miglioramento della capacità delle macchine di comprendere il linguaggio rimane un'area di ricerca molto attiva.

L'NLP è particolarmente impegnativo perché l'uso efficace dei dati di testo per il ML richiede una comprensione del funzionamento interno del linguaggio e la conoscenza del mondo a cui si riferisce.

Le sfide principali includono quanto segue:

- Ambiguità dovuta alla **polisemia**, cioè parola o frase con significati diversi a seconda del contesto.
- L'uso non standard e in **continua evoluzione** del linguaggio, in particolare sui social media.
- L'uso di **modi di dire** come “gettare la spugna”.
- **Nomi di entità** ingannevoli come “Dove stanno girando A Bug's Life?”
- Conoscenza del mondo: “Mary e Sue sono sorelle” vs. “Mary e Sue sono madri”.

In questo capitolo, presentiamo due casi di studio basati sull'NLP che coprono le applicazioni dell'NLP nel trading algoritmico e nell'interpretazione e automazione dei documenti. I passaggi chiave del modello per i problemi basati sull'NLP sono la **preelaborazione dei dati**, la **rappresentazione delle features** e **l'inferenza**. Pertanto, queste aree, insieme ai concetti correlati e agli esempi basati su Python, sono delineate in questo capitolo.

8.1 NLP: Teoria e concetti

Come abbiamo già detto, l'NLP è un sottocampo dell'intelligenza artificiale che si occupa di programmare computer per elaborare dati testuali al fine di ottenere utili intuizioni. Tutte le applicazioni di NLP passano attraverso passaggi sequenziali comuni, che includono una combinazione di preelaborazione di dati testuali e rappresentazione del testo come caratteristiche predittive prima di inserirli in un algoritmo di inferenza statistica. La figura 8.1, delinea i passaggi principali in un'applicazione basata sull'NLP.

Le seguenti sezioni esaminano questi passaggi.

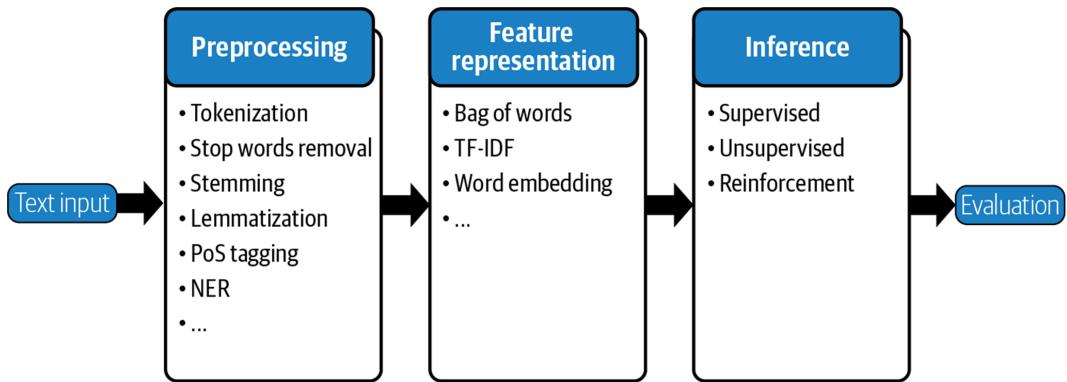


Figura 8.1: Passaggi principali nell'NLP

8.1.1 Preelaborazione

La figura 8.1, mostra alcune componenti chiave della fase di preelaborazione per l'NLP. Queste sono la tokenizzazione, la rimozione delle stop words, stemming, lemmatizzazione, tagging PoS (part-of-speech) e NER (Name Entry Recogniton).

Tokenizzazione

La tokenizzazione è il compito di suddividere un testo in segmenti significativi, chiamati token. Questi segmenti possono essere parole, punteggiatura, numeri o altri caratteri speciali che costituiscono gli elementi costitutivi di una frase. Un insieme di regole prestabilite ci permette di convertire efficacemente una frase in un elenco di token. I seguenti frammenti di codice mostrano un esempio di tokenizzazione delle parole usando i pacchetti Python NLTK e TextBlob:

```

1 nlp = en_core_web_sm.load()
2
3 #tokenizzazione
4
5 #text to tokenize
6 text = "This_is_a_tokenize_test"
7 from nltk.tokenize import word_tokenize
8 word_tokenize(text)
9 Output
10 ['This', 'is', 'a', 'tokenize', 'test']
11 #tokenizzazione con TextBlob
12 TextBlob(text).words
13 Output

```

¹⁴ WordList(['This', 'is', 'a', 'tokenize', 'test'])

Rimozione delle stop-words

A volte vengono escluse dal vocabolario parole estremamente comuni che offrono scarso valore nella modellazione. Queste parole sono chiamate stop words. Di seguito è riportato il codice per la rimozione delle stop words utilizzando la libreria NLTK:

```

1 text = "S&P_and_NASDAQ_are_the_two_most_popular_indices_in_US"
2
3 from nltk.corpus import stopwords
4 from nltk.tokenize import word_tokenize
5 nltk.download('stopwords')
6 stop_words = set(stopwords.words('english'))
7 text_tokens = word_tokenize(text)
8 tokens_without_sw= [word for word in text_tokens if
9                     not word in stop_words]
10
11 print(tokens_without_sw)
12 ['S', '&', 'P', 'NASDAQ', 'two', 'popular', 'indices', 'US']
```

Per prima cosa carichiamo il modello linguistico e lo memorizziamo nella variabile stop words. `stopwords.words('english')` è un insieme di stop words predefinite per il modello in lingua inglese in NLTK. Successivamente, iteriamo semplicemente attraverso ogni parola nel testo di input e, se la parola esiste nel set di parole non significative del modello di linguaggio NLTK, la parola viene rimossa. Come possiamo vedere, le stop words, come are e most, vengono rimosse dalla frase.

Stemming

Lo stemming è il processo di riduzione delle parole derivate alla loro forma base o radice (generalmente una forma di parola scritta). Ad esempio, se dovessimo derivare le parole *Stems*, *Stemming*, *Stemmed* e *Stemitzation*, il risultato sarebbe una singola parola: Stem. Il codice per lo stemming utilizzando la libreria NLTK è mostrato qui:

```

1 text = "It's_a_Stemming_testing"
2
3 parsed_text = word_tokenize(text)
4
5 # Initialize stemmer.
6 from nltk.stem.snowball import SnowballStemmer
```

```

7 stemmer = SnowballStemmer('english')
8
9 # Stem each word.
10 [(word, stemmer.stem(word)) for i, word in enumerate(parsed_text)
11   if word.lower() != stemmer.stem(parsed_text[i])]
12 Output
13 [('Stemming', 'stem'), ('testing', 'test')]

```

Lematizzazione

Una leggera variante dello stemming è la lematizzazione. La principale differenza tra i due processi è che lo stemming può spesso creare parole inesistenti, mentre i lemmi sono parole reali. Un esempio di lematizzazione è *run* come forma base per parole come *running* e *ran*, o che le parole *better* e *good* sono considerate lo stesso lemma. Il codice per la lematizzazione utilizzando la libreria TextBlob è mostrato di seguito:

```

1 text = "This_world_has_a_lot_of_faces"
2
3 from textblob import Word
4 nltk.download('wordnet')
5 parsed_data= TextBlob(text).words
6 [(word, word.lemmatize()) for i, word in enumerate(parsed_data)
7   if word != parsed_data[i].lemmatize()]
8 Output
9 ('has', 'ha'), ('faces', 'face')]

```

Tagging PoS (Part-of-Speech)

Il tagging PoS è il processo di assegnazione di un token alla sua categoria grammaticale (ad es. verbo, sostantivo, ecc.) per comprenderne il ruolo all'interno di una frase. I tag PoS sono stati utilizzati per una varietà di attività di NLP e sono estremamente utili poiché forniscono un segnale linguistico di come una parola viene utilizzata nell'ambito di una frase o documento. Dopo che una frase è stata suddivisa in token, viene utilizzato un tagger, o PoS tagger, per assegnare ciascun token a una categoria della parte del discorso. Storicamente, per creare tali tagger sono stati utilizzati gli Hidden Markov Models (HMM). Più recentemente, sono state sfruttate le reti neurali artificiali. Il codice per il tagging PoS utilizzando la libreria TextBlob è mostrato a seguito:

```

1 text = 'Google_is_looking_at_buying_U.K._startup_for_$1_billion'
2 nltk.download('averaged_perceptron_tagger')

```

```
Google ORG is looking at buying U.K. GPE startup for $1 billion MONEY
```

Figura 8.2: NER Output

```

3 TextBlob(text).tags
4 Output
5 [('Google', 'NNP'),
6 ('is', 'VBZ'),
7 ('looking', 'VBG'),
8 ('at', 'IN'),
9 ('buying', 'VBG'),
10 ('U.K.', 'NNP'),
11 ('startup', 'NN'),
12 ('for', 'IN'),
13 ('1', 'CD'),
14 ('billion', 'CD')]
```

Named entity recognition (NER)

Il riconoscimento di entità denominate (NER) è un passaggio successivo facoltativo nella preelaborazione dei dati che cerca di individuare e classificare le entità denominate nel testo in categorie predefinite. Queste categorie possono includere nomi di persone, organizzazioni, luoghi, espressioni di tempi, quantità, valori monetari o percentuali. Il NER eseguito utilizzando spaCy è mostrato di seguito:

```

1 text = 'Google_is_looking_at_buying_U.K._startup_for_$1_billion'
2
3 for entity in nlp(text).ents:
4     print("Entity:", entity.text)
5 Output
6 Entity: Google
7 Entity: U.K.
8 Entity: $1 billion
```

La visualizzazione di entità denominate nel testo, come mostrato nella figura 8.2, può anche essere incredibilmente utile per accelerare lo sviluppo e il debug del codice e il processo di addestramento.

```

1 from spacy import displacy
2 displacy.render(nlp(text), style="ent", jupyter = True)
```

spaCy: tutti i passaggi precenti in una sola volta

Tutti i passaggi di preelaborazione mostrati sopra possono essere eseguiti in un unico passaggio utilizzando spaCy. Quando chiamiamo la funzione `nlp` su un testo, spaCy prima tokenizza il testo per produrre un oggetto `Doc`. Il documento viene quindi elaborato in diversi passaggi. Questo, è anche indicato come la pipeline di elaborazione. La pipeline utilizzata dai modelli predefiniti è costituita da un tagger, un parser e un NER. Ogni componente della pipeline restituisce il `Doc` elaborato, che viene quindi passato al componente successivo, come mostrato nella figura 8.3.

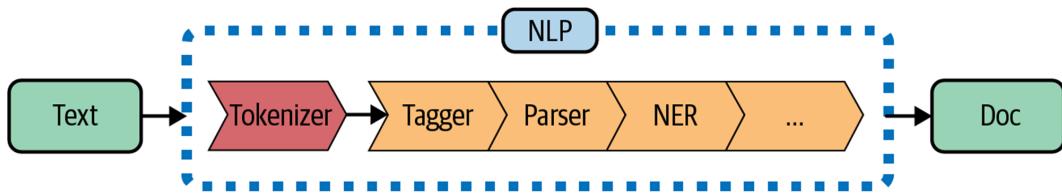


Figura 8.3: spaCy pipeline

```

1 text = 'Google_is_looking_at_buying_U.K._startup_for_$1_billion'
2 doc = nlp(text)
3 import pandas as pd
4 pd.DataFrame([ [ t.text , t.is_stop , t.lemma_ , t.pos_ ]
                for t in doc ],
5               columns=[ 'Token' , 'is_stop_word' , 'lemma' , 'POS' ])
6
7
8      Token  is_stop_word      lemma      POS
9  0   Google          False   Google  PROPN
10 1     is           True    be      AUX
11 2  looking          False   look     VERB
12 3     at            True    at      ADP
13 4  buying          False   buy      VERB
14 5    U.K.          False  U.K.  PROPN
15 6  startup          False startup    NOUN
16 7    for           True   for      ADP
17 8     $            False    $      SYM
18 9     1            False     1      NUM
19 10 billion          False billion    NUM
  
```

L'output per ciascuno dei passaggi di preelaborazione è illustrato nella tabella precedente.

Oltre alle fasi di preelaborazione di qui sopra, esistono altre fasi di preelaborazione utilizzate di frequente, come la rimozione di maiuscole e minuscole o

la rimozione di dati non alfanumerici, che possiamo eseguire a seconda del tipo di dati. Ad esempio, i dati estratti da un sito Web devono essere ulteriormente ripuliti, inclusa la rimozione dei tag HTML. I dati di un rapporto PDF devono essere convertiti in un formato di testo.

Altre fasi di preelaborazione opzionali includono l'analisi delle dipendenze, la risoluzione della coreferenza, l'estrazione di triplette e l'estrazione di relazioni:

- Analisi delle dipendenze: Assegna una struttura sintattica alle frasi per dare un senso al modo in cui le parole nella frase si relazionano tra loro.
- Risoluzione di co-referenza: Il processo di connessione dei token che rappresentano la stessa entità. È comune nelle lingue introdurre un soggetto con il proprio nome in una frase e poi riferirsi a lui/lei/esso nelle frasi successive.
- Estrazione di triplette: Il processo di registrazione delle terzine di soggetto, verbo e oggetto quando disponibili nella struttura della frase.
- Estrazione di relazioni: Una forma più ampia di estrazione di triplette in cui le entità possono avere più interazioni.

Questi passaggi aggiuntivi dovrebbero essere eseguiti solo se aiuteranno con l'attività a portata di mano.

8.2 Rappresentazione delle features

La stragrande maggioranza dei dati relativi all'NLP, come articoli di feed di notizie, report PDF, post sui social media e file audio, viene creata per il consumo umano. In quanto tale, è spesso archiviato in un formato non strutturato, che non può essere facilmente elaborato dai computer. Affinché le informazioni preelaborate vengano trasmesse all'algoritmo di inferenza statistica, i token devono essere tradotti in funzionalità predittive. Un modello viene utilizzato per incorporare testo non elaborato in uno spazio vettoriale.

La rappresentazione delle caratteristiche implica due cose:

1. Un vocabolario di parole conosciute.
2. Una misura della presenza di parole conosciute.

Alcuni dei metodi di rappresentazione delle features sono:

- Bag of Words

- TF-IDF
- Incorporamento di parole (Word embedding)
 - Modelli preaddestrati (ad es. word2vec, GloVe , modello di incorporamento di parole di spaCy)
 - Rappresentazione personalizzata delle features basata sul deep learning

Impariamo di più su ciascuno di questi metodi.

8.2.1 Bag of Words - word count

Nell'elaborazione del linguaggio naturale, una tecnica comune per estrarre caratteristiche dal testo consiste nel posizionare tutte le parole che ricorrono nel testo in un “secchio”. Questo approccio è chiamato modello bag of words. Viene definito “sacco di parole” perché ogni informazione sulla struttura della frase viene persa. In questa tecnica, costruiamo una singola matrice da una raccolta di testi, come mostrato nella figura 8.4, in cui ogni riga rappresenta un token e ogni colonna rappresenta un documento o una frase nel nostro corpus. I valori della matrice rappresentano il conteggio del numero di istanze del token che appare.

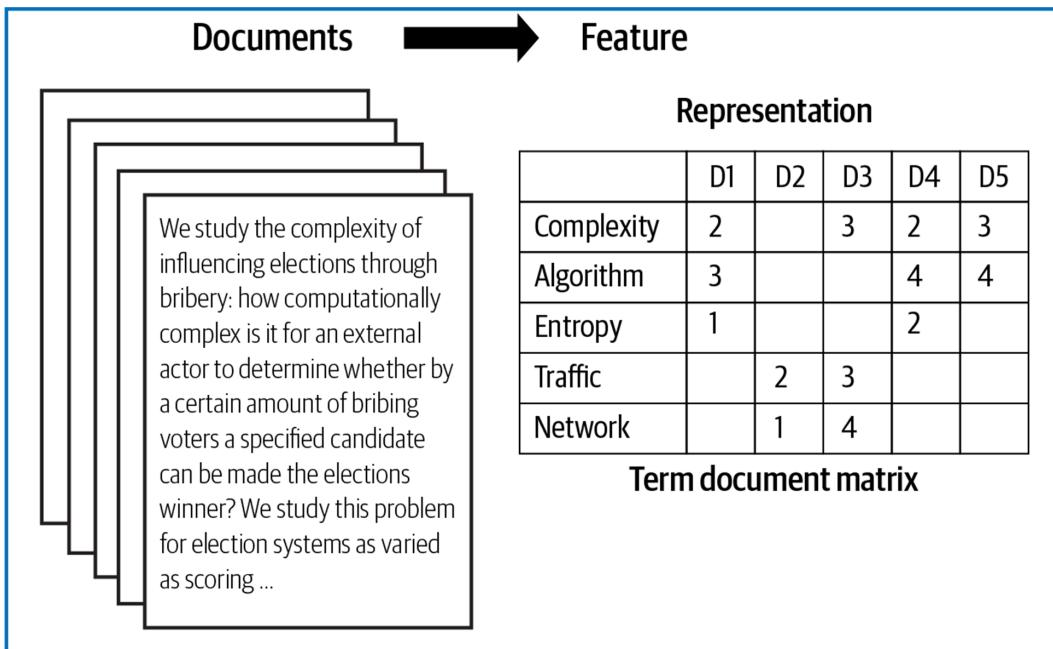


Figura 8.4: Bag of words

Il CountVectorizer di sklearn fornisce un modo semplice sia per tokenizzare una raccolta di documenti di testo sia per codificare nuovi documenti utilizzando quel vocabolario. La funzione fit_transform apprende il vocabolario da uno o più documenti e codifica ogni documento come vettore:

```
sentences = [  
    'The stock price of google jumps on the earning data today',  
    'Google plunge on China Data!',  
]  
from sklearn.feature_extraction.text import CountVectorizer  
vectorizer = CountVectorizer()  
print( vectorizer.fit_transform(sentences).todense() )  
print( vectorizer.vocabulary_ )  
Output  
[[0 1 1 1 1 1 0 1 1 2 1]  
 [1 1 0 1 0 0 1 1 0 0 0]]  
{'the': 10, 'stock': 9, 'price': 8, 'of': 5, 'google': 3,  
'jumps': 4, 'on': 6, 'earning': 2, 'data': 1, 'today': 11,  
'plunge': 7, 'china': 0}
```

Possiamo vedere una versione di matrice del vettore codificato che mostra un conteggio di un'occorrenza per ogni parola tranne *the* (indice 10), che ha un'occorrenza di due. I conteggi delle parole sono un buon punto di partenza, ma sono molto basilari. Un problema con i conteggi semplici è che alcune parole come *the* appariranno molte volte e i loro grandi conteggi non saranno molto significativi nei vettori codificati. Queste rappresentazioni di bag of words sono scarse perché i vocabolari sono vasti e una data parola o documento sarebbe rappresentata da un grande vettore composto principalmente da valori a zero.

8.2.2 TF-IDF

Un'alternativa è calcolare le frequenze delle parole, e il metodo di gran lunga più popolare è TF-IDF, che sta per Term Frequency–Inverse Document Frequency:

- Frequenza dei termini (TF): Questo riassume la frequenza con cui una data parola appare all'interno di un documento.
- Frequenza inversa del documento (IDF): Questo riduce le parole che appaiono molto nei documenti.

In parole povere, TF-IDF è un punteggio di frequenza delle parole che cerca di evidenziare le parole che sono più interessanti (cioè frequenti all'interno di un

documento, ma non tra i documenti). Il modulo di preelaborazione di scikit-learn offre due strumenti per creare una matrice dei termini del documento.

CountVectorizer utilizza conteggi binari o assoluti per misurare la frequenza del termine (TF) $tf(d, t)$ per ogni documento d e token t . TfidfVectorizer, al contrario, pesa la frequenza del termine (assoluta) per la frequenza inversa del documento (IDF). Di conseguenza, un termine che appare in più documenti riceverà un peso inferiore rispetto a un token con la stessa frequenza per un dato documento ma con una frequenza inferiore in tutti i documenti. Più specificamente, utilizzando le impostazioni predefinite, le voci $tf-idf(d, t)$ per la matrice documento-termine vengono calcolate come $tf-idf(d, t) = tf(d, t) \times idf(t)$ con:

$$idf(t) = \log \frac{(1 + n_d)}{(1 + df(d, t))} + 1$$

dove n_d è il numero di documenti e $df(d, t)$ la frequenza dei documenti del termine t . I vettori TF-IDF risultanti per ogni documento sono normalizzati rispetto ai loro totali assoluti o elevati al quadrato (per maggiori dettagli vedere la documentazione di sklearn). TfidfVectorizer tokenizzerà i documenti, apprenderà il vocabolario e le ponderazioni inverse della frequenza del documento e consentirà di codificare nuovi documenti:

```

1 from sklearn.feature_extraction.text import TfidfVectorizer
2 vectorizer = TfidfVectorizer(max_features=1000,
3                             stop_words='english')
4 TFIDF = vectorizer.fit_transform(sentences)
5 print(vectorizer.get_feature_names_out()[-10:])
6 print(TFIDF.shape)
7 print(TFIDF.toarray())
8 Output
9 ['china' 'data' 'earning' 'google' 'jumps' 'plunge'
10 'price' 'stock' 'today']
11 (2, 9)
12 [[0. 0.29017021 0.4078241 0.29017021 0.4078241 0.
13   0.4078241 0.4078241 0.4078241 ]
14 [0.57615236 0.40993715 0. 0.40993715 0.
15  0.57615236 0. 0. 0. ]]

```

Nel frammento di codice fornito, dai documenti viene appreso un vocabolario di nove parole. A ciascuna parola viene assegnato un indice intero univoco nel vettore di output. Le frasi sono codificate come un array sparso di nove elementi e possiamo rivedere i punteggi finali di ogni parola con valori diversi dalle altre parole nel vocabolario.

8.2.3 Word embedding

Il modello bag-of-words rappresenta i documenti come vettori sparsi e ad alta dimensione che riflettono i token che contengono. Gli incorporamenti di parole (word embeddings) rappresentano i token come *vettori densi*¹ e di dimensioni inferiori in modo che la posizione relativa delle parole riflette il modo in cui vengono utilizzate nel contesto. Incarnano l’ipotesi distributiva della linguistica che afferma che le parole sono meglio definite dalla compagnia che mantengono.

I vettori di parole sono in grado di catturare numerosi aspetti semantici; non solo i sinonimi sono assegnati vicino dalle incorporazioni, ma le parole possono avere diversi gradi di somiglianza. Ad esempio, la parola "driver" (conducente) può essere simile a "motorist" (automobilista) o a "factor" (agente). Inoltre, gli embeddings codificano le relazioni tra coppie di parole come le analogie (Tokyo è per il Giappone ciò che Parigi è per la Francia).

Come abbiamo detto, in un incorporamento, le parole sono rappresentate da vettori densi in cui un vettore rappresenta la proiezione della parola in uno spazio vettoriale continuo (vedi figura 8.5). La posizione di una parola all’interno dello spazio vettoriale viene appresa dal testo e si basa sulle parole che circondano la parola quando viene utilizzata. La posizione di un termine nello spazio vettoriale è indicata come il suo *incorporamento*.

Alcuni dei modelli di apprendimento degli incorporamenti di parole dal testo includono word2Vec, il modello di incorporamento di parole preaddestrato di spaCy e GloVe. Oltre a questi metodi progettati con cura, è possibile apprendere l’incorporamento di parole come parte di un modello di deep learning. Questo può essere un approccio più lento, ma adatta il modello a un set di dati di addestramento specifico.

Modello preaddestrato via spaCy

spaCy viene fornito con la rappresentazione integrata del testo come vettori a diversi livelli di parola, frase e documento. Le rappresentazioni vettoriali sottostanti provengono da un modello di word embedding, che generalmente produce una rappresentazione semantica densa e multidimensionale delle parole (come mostrato nell’esempio seguente).

Il modello di word embedding include 20.000 vettori univoci con 300 dimensioni. Usando questa rappresentazione vettoriale, possiamo calcolare somiglianze e differenze tra token, entità denominate (Named Entity), sostantivi, frasi e documenti.

¹Un vettore sparso (sparse) è quello che contiene principalmente zeri e poche voci diverse da zero. Un vettore denso (dense) contiene principalmente non zeri.

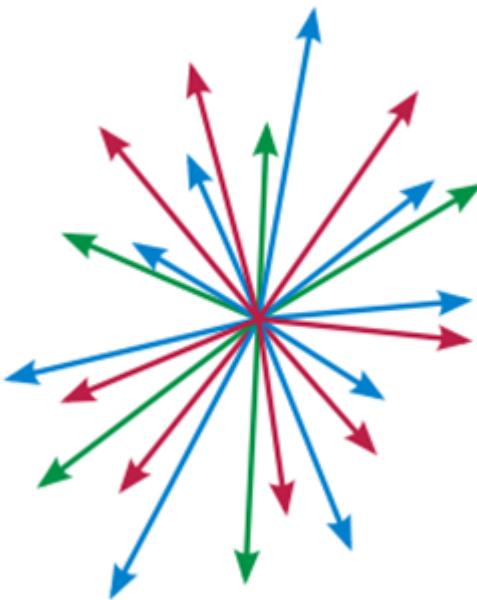


Figura 8.5: Spazio Vettoriale

Il word embedding in spaCy viene eseguito caricando prima il modello e quindi elaborando il testo. È possibile accedere direttamente ai vettori utilizzando l'attributo `.vector` di ciascun token elaborato (ovvero, parola). Anche il vettore medio per l'intera frase viene calcolato semplicemente utilizzando il vettore, fornendo un input molto conveniente per i modelli di apprendimento automatico basati su frasi:

```

1 doc = nlp("Apple..orange..cats..dogs")
2 print("Vector..representation..of..the..sentence")
3 for..first..10..features:..\n",\ doc.vector[0:10])
4
5 Output
6 Vector representation of the sentence for first 10 features:
7 [-0.43863457 -0.3626267 -0.20616335 1.0866606
8 0.44007337 -0.03703639 0.7782439
9 0.55959123 0.21888082 -0.40239647]
```

Nell'output viene mostrata la rappresentazione vettoriale della frase per le prime 10 features del modello preaddestrato.

Modello preaddestrato: Word2Vec utilizzando il pacchetto gensim

L'implementazione basata su Python del modello word2vec utilizzando il pacchetto gensim è dimostrata di seguito:

```
1
2 from gensim.models import Word2Vec
3
4 sentences = [
5 ['The', 'stock', 'price', 'of', 'Google', 'increases'],
6 ['Google', 'plunge', 'on', 'China', 'Data!']]
7
8 # train model
9 model = Word2Vec(sentences, min_count=1)
10
11 # summarize the loaded model
12 words = list(model.wv.key_to_index)
13 print(words)
14 print(model.wv.get_vector('Google')[1:5])
15 Output
16 ['Google', 'Data!', 'China', 'on', 'plunge',
17 'increases', 'of', 'price', 'stock', 'The']
18 [ 0.00023643 0.00510335 0.00900927 -0.00930295]
```

La rappresentazione vettoriale della frase per le prime cinque features del modello word2vec preaddestrato è mostrata sopra.

8.3 Inferenza

Come con altri compiti di intelligenza artificiale, un'inferenza generata da un'applicazione NLP di solito deve essere tradotta in una decisione per essere perseguitabile. L'inferenza rientra nelle tre categorie di apprendimento automatico trattate nel capitolo precedente (vale a dire, apprendimento supervisionato, non supervisionato e per rinforzo). Sebbene il tipo di inferenza richiesto dipenda dal problema aziendale e dal tipo di dati di addestramento, gli algoritmi più comunemente utilizzati sono supervisionati e non supervisionati.

Una delle metodologie supervisionate più frequentemente utilizzate nell'NLP è il modello Naive Bayes, in quanto può produrre una ragionevole accuratezza utilizzando semplici presupposti. Una metodologia supervisionata più complessa utilizza architetture di reti neurali artificiali. Negli anni passati, queste architetture, come le reti neurali ricorrenti (RNN), hanno dominato l'inferenza basata sull'NLP.

La maggior parte della letteratura esistente sull'NLP si concentra sull'apprendimento supervisionato. In quanto tali, le applicazioni di apprendimento non supervisionato costituiscono un sottodomain relativamente meno sviluppato in cui la misurazione della somiglianza dei documenti è tra le attività più comuni. Una tecnica popolare non supervisionata applicata nell'NLP è la

Latent Semantic Analysis (LSA). L'LSA esamina le relazioni tra un insieme di documenti e le parole che contengono producendo un insieme di concetti latenti relativi ai documenti e ai termini. L'LSA ha aperto la strada a un approccio più sofisticato chiamato Latent Dirichlet Allocation (LDA), in base al quale i documenti sono modellati come una miscela finita di argomenti. Questi argomenti a loro volta sono modellati come una miscela finita di parole nel vocabolario. L'LDA è stato ampiamente utilizzato per la modellazione di argomenti (topic modeling) — un'area di ricerca in crescita in cui i professionisti dell'NLP costruiscono modelli generativi probabilistici per rivelare probabili attribuzioni di argomenti per le parole.

8.3.1 Esempio di apprendimento supervisionato

Naive Bayes è una famiglia di algoritmi basati sull'applicazione del teorema di Bayes con una forte (ingenua) ipotesi che ogni feature utilizzata per prevedere la categoria di un dato campione sia indipendente dalle altre. Sono classificatori probabilistici e quindi calcoleranno la probabilità di ciascuna categoria utilizzando il teorema di Bayes. Verrà data in output la categoria con la probabilità più alta.

Nell'NLP, un approccio Naive Bayes presuppone che tutte le features delle parole siano indipendenti l'una dall'altra date le etichette o labels di classe.

A causa di questo presupposto semplificativo, Naive Bayes è molto compatibile con una rappresentazione di Bags of Words e si è dimostrato veloce, affidabile e accurato in numerose applicazioni di NLP. Inoltre, nonostante i suoi presupposti semplificativi, è competitivo con (e talvolta addirittura supera) i classificatori più complicati.

Diamo un'occhiata all'uso di Naive Bayes per l'inferenza in un problema di analisi del sentimento. Prendiamo un dataframe in cui ci sono due frasi con i sentimenti assegnati a ciascuna di esse. Nel passaggio successivo, convertiamo le frasi in una rappresentazione delle features utilizzando CountVectorizer. Le features e i sentimenti vengono utilizzati per addestrare e testare il modello utilizzando Naive Bayes:

```

1 sentences = [
2     'The_stock_price_of_google_jumps_on_the_earning_data_today',
3     'Google_plunge_on_China_Data!'
4 ]
5 sentiment = (1, 0)
6 data = pd.DataFrame({'Sentence':sentences,
7                     'sentiment':sentiment})
8 # feature extraction
9 from sklearn.feature_extraction.text import CountVectorizer

```

```

10 vect = CountVectorizer().fit(data['Sentence'])
11 X_train_vectorized = vect.transform(data['Sentence'])
12
13 # Running naive bayes model
14 from sklearn.naive_bayes import MultinomialNB
15 clfrNB = MultinomialNB(alpha=0.1)
16 clfrNB.fit(X_train_vectorized, data['sentiment'])
17
18 #Testing the model
19 preds = clfrNB.predict(vect.transform(['Apple_price_plunge',\
20 'Amazon_price_jumps']))
21 preds
22 Output
23 array([0, 1])

```

Come possiamo vedere, il Naive Bayes addestra abbastanza bene il modello dalle due frasi. Il modello fornisce un sentimento pari a zero e uno per le frasi di prova "Apple price plunge" e "Amazon price jumps", rispettivamente, dato che le frasi utilizzate per l'addestramento avevano anche le parole chiave "plunge" e "jumps", con le corrispondenti assegnazioni di sentimento.

8.3.2 Esempio di apprendimento non supervisionato

LDA è ampiamente utilizzato per la modellazione di argomenti perché tende a produrre argomenti significativi che gli esseri umani possono interpretare, assegna argomenti a nuovi documenti ed è estensibile. Funziona partendo da un presupposto chiave: i documenti vengono generati selezionando prima gli argomenti (i topic) e poi, per ogni argomento, un insieme di parole. L'algoritmo esegue quindi il reverse engineering di questo processo per trovare gli argomenti in un documento. Nel seguente frammento di codice viene mostrata un'implementazione di LDA per la modellazione degli argomenti. Prendiamo due frasi e le convertiamo in una rappresentazione di features usando CountVectorizer. Queste features e i sentimenti vengono utilizzati per addestrare il modello e produrre due matrici più piccole che rappresentano gli argomenti:

```

1 sentences =
2 'The_stock_price_of_google_jumps_on_the_earning_data_today',
3 'Google_plunge_on_China_Data!'
4 ]
5
6 #Getting the bag of words
7 from sklearn.decomposition import LatentDirichletAllocation
8 vect=CountVectorizer(ngram_range=(1, 1),stop_words='english')

```

```

9 sentences_vec=vect.fit_transform(sentences)
10
11 #Running LDA on the bag of words.
12 from sklearn.feature_extraction.text import CountVectorizer
13 lda=LatentDirichletAllocation(n_components=3)
14 lda.fit_transform(sentences_vec)
15
16 Output
17 array([[0.04311114, 0.91377772, 0.04311114],
18        [0.06869319, 0.86261362, 0.06869319]])

```

Utilizzeremo l'LDA per la modellazione degli argomenti nel secondo caso di studio di questo capitolo e discuteremo i concetti e l'interpretazione in dettaglio.

8.4 Caso di studio: NLP e strategie di trading basate sull'analisi del sentimento

L'elaborazione del linguaggio naturale offre la possibilità di quantificare il testo. Si può iniziare a porsi domande del tipo: Quanto è positiva o negativa questa notizia? Come possiamo quantificare le parole? Forse l'applicazione più notevole dell'NLP è il suo utilizzo nel trading algoritmico. L'NLP fornisce un mezzo efficiente per monitorare i sentimenti del mercato. Applicando tecniche di analisi del sentimento basate sull'NLP ad articoli di notizie, rapporti, social media o altri contenuti web, è possibile determinare efficacemente se tali fonti hanno un punteggio di sentimento positivo o negativo. I punteggi del sentimento possono essere utilizzati come segnale direzionale per acquistare titoli con punteggi positivi e vendere titoli con punteggi negativi. Le strategie di trading basate su dati di testo stanno diventando sempre più popolari con l'aumentare della quantità di dati non strutturati. In questo caso di studio vedremo come utilizzare i sentimenti basati sull'NLP per costruire una strategia di trading.

In questo caso di studio, ci concentreremo su:

- Produrre news sentiments utilizzando algoritmi supervisionati e non supervisionati.
- Migliorare l'analisi del sentimento utilizzando un modello di deep learning, come LSTM.
- Confrontare le diverse metodologie di generazione del sentimento allo scopo di costruire una strategia di trading.

- Usare efficacemente sentimenti e vettori di parole come features in una strategia di trading.
- Raccolta di dati da diverse fonti e preelaborazione per l'analisi del sentimento.
- Creazione di un framework per il backtesting dei risultati di una strategia di trading utilizzando i pacchetti Python disponibili.

Di seguito verranno illustrati i passaggi principali con le rispettive spiegazioni, il codice Python utilizzato nelle varie fasi verrà messo a disposizione in un file allegato per le consultazioni.

8.4.1 Costruire una strategia di trading basata sull'analisi del sentimento

Definizione del problema

Il nostro obiettivo è (1) utilizzare l'NLP per estrarre informazioni dai titoli delle notizie, (2) assegnare un sentimento a tali informazioni e (3) utilizzare l'analisi del sentimento per costruire una strategia di trading.

I dati utilizzati per questo caso di studio provengono dalle seguenti fonti:

- Dati dei titoli delle notizie raccolti dai *feed RSS*² di diversi siti Web di notizie

Ai fini di questo studio, esamineremo solo i titoli, non il testo completo delle storie. Il nostro dataset contiene circa 82.000 titoli da maggio 2011 a dicembre 2018.

- Sito web di Yahoo Finance per i dati di borsa

I dati sui rendimenti per le azioni utilizzati in questo caso di studio derivano dai dati sui prezzi di Yahoo Finance.

- Kaggle

Useremo i dati etichettati dei sentimenti delle notizie per un modello di analisi del sentimento basato sulla classificazione. Si noti che questi dati potrebbero non essere completamente applicabili al caso in questione e vengono utilizzati qui a scopo dimostrativo.

²Il Feed RSS (Really Simple Syndication) è un sistema per la distribuzione dei contenuti basato su XML.

- Lessico del mercato azionario

Il lessico si riferisce al componente di un sistema di NLP che contiene informazioni (semantiche, grammaticali) su singole parole o stringhe di parole. Questo viene creato sulla base di conversazioni in borsa nei servizi di microblogging.

I passaggi chiave di questo caso di studio sono delineati nella seguente figura 8.6.

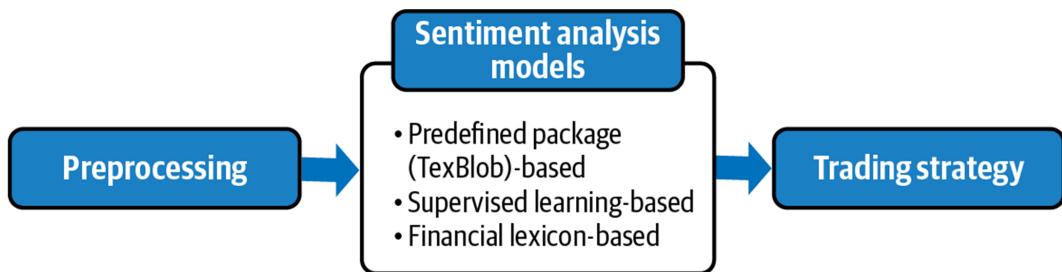


Figura 8.6: Passaggi della strategia di trading basata sull’analisi del sentimento

Una volta terminata la preelaborazione, esamineremo i diversi modelli di analisi del sentimento. I risultati della fase di analisi del sentimento vengono utilizzati per sviluppare le strategie di trading.

Caricamento e preparazione dei dati

In questa fase per prima cosa carichiamo i dati sui prezzi delle azioni da Yahoo Finance. Selezioniamo 10 azioni per questo caso di studio. Queste azioni sono i titoli di alcune dei più grandi titoli dello S&P 500 per quota di mercato come Apple (AAPL), Amazon (AMZN), Google (GOOG), Facebook (FB), Tesla (TSLA) e altri.

I dati caricati contengono i dati di prezzo e volume delle azioni insieme al loro ticker name o symbol.

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits	ticker
0	2010-01-04	26.40	26.53	26.27	26.47	123432400	0.0	0.0	AAPL
1	2010-01-05	26.54	26.66	26.37	26.51	150476200	0.0	0.0	AAPL
2	2010-01-06	26.51	26.62	26.06	26.09	138040000	0.0	0.0	AAPL
3	2010-01-07	26.19	26.22	25.85	26.04	119282800	0.0	0.0	AAPL
4	2010-01-08	26.01	26.22	25.85	26.22	111902700	0.0	0.0	AAPL

Nella fase di preparazione, caricchiamo e preelaboriamo i dati delle notizie, quindi combiniamo i dati delle notizie con i dati sui rendimenti delle azioni. Questo dataset combinato verrà utilizzato per lo sviluppo del modello.

Durante la valutazione dei modelli di analisi del sentimento, analizziamo anche la relazione tra i sentimenti e la successiva performance azionaria. Per capire la relazione, usiamo *event return*, che è il rendimento che corrisponde all'evento. Lo facciamo perché a volte le notizie vengono riportate in ritardo (ovvero, dopo che i partecipanti al mercato sono a conoscenza dell'annuncio) o dopo la chiusura del mercato. Avere una finestra leggermente più ampia ci assicura di catturare l'essenza dell'evento. L'*event return* è definito come:

$$R_{T-1} + R_T + R_{T+1}$$

Dove R_{T-1} e R_{T+1} sono i rendimenti prima e dopo i dati delle notizie, e R_T è il rendimento nel giorno della notizia.

Ora, abbiamo preparato un DataFrame, “`data_df`”, pulito (senza `NaN`) con ticker, titolo, data, rendimento dell'evento, rendimento per un determinato giorno e rendimento futuro per 10 titoli (ticker) di borsa univoci, per un totale di 2759 righe di dati.

```
1 print(data_df.shape, data_df.ticker.unique().shape)  
2 (2759, 5) (10,)
```

Valutazione dei modelli per l'analisi del sentimento

- Modello predefinito: pacchetto `TextBlob`
- Modello ottimizzato: algoritmi di classificazione e LSTM
- Modello basato sul lessico finanziario

Modello predefinito: pacchetto `TextBlob`

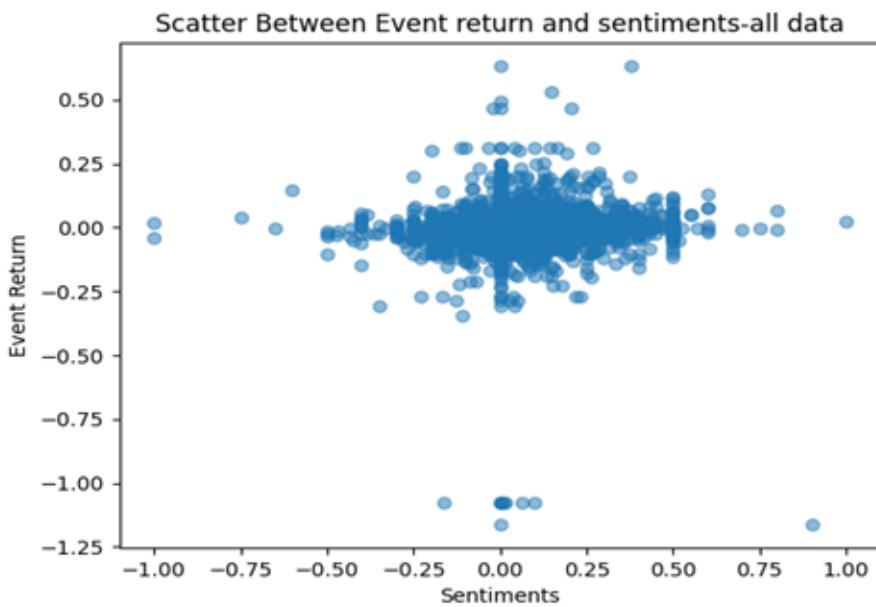
La funzione `TextBlob sentiment` è un modello preaddestrato basato sull'algoritmo di classificazione Naive Bayes. La funzione associa gli aggettivi che si trovano frequentemente nelle recensioni di film ai punteggi di polarità del sentimento che vanno da -1 a $+1$ (da negativo a positivo), convertendo una frase in un valore numerico. Lo applichiamo a tutti le headline (titoli) degli articoli. Di seguito è mostrato un esempio di come ottenere il sentimento per un testo di notizie:

```
text1 = "Bayer (OTCPK:BAYRY) started the week up  
3.5% to euro74/share in Frankfurt , touching their \  
highest level in 14 months , after the U.S. government said  
a $25M glyphosate decision against the \  
company should be reversed."
```

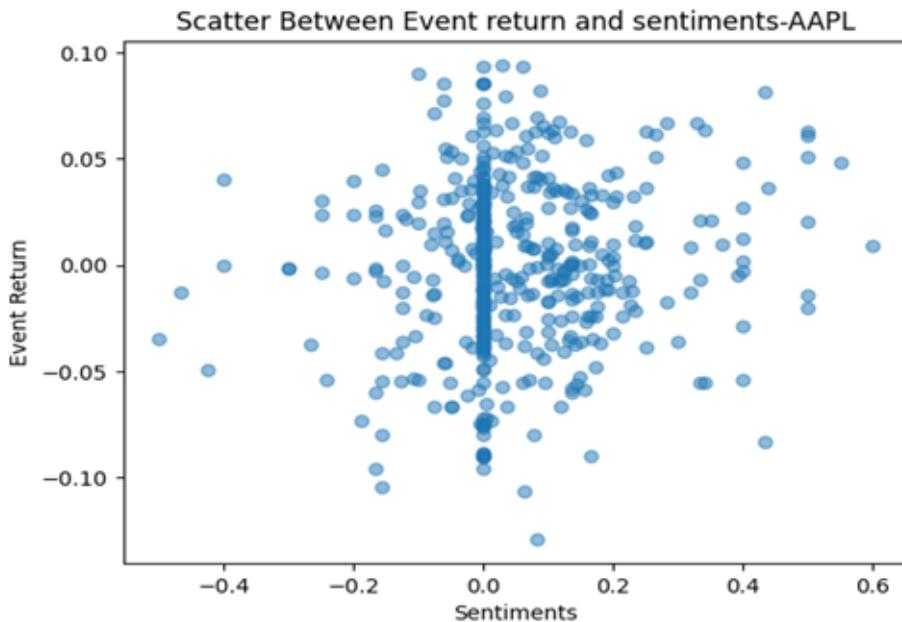
```
TextBlob(text1).sentiment.polarity  
Output  
0.5
```

Il sentimento per la dichiarazione è 0,5. Lo applichiamo a tutti i titoli che abbiamo nei dati.

Esaminiamo il grafico a dispersione dei sentimenti e dei rendimenti per esaminare la correlazione tra i due per tutti i 10 titoli.



Un grafico per un singolo titolo (APPL) è mostrato nel seguente grafico.



Dai grafici a dispersione possiamo vedere che non c'è una forte relazione tra le notizie e i sentimenti. La correlazione tra rendimento e sentimento è positiva (4,27%), il che significa che le notizie con sentimento positivi portano a rendimenti positivi. Tuttavia, la correlazione non è molto elevata. Anche guardando lo scatterplot generale, vediamo che la maggior parte dei sentimenti si concentra intorno allo zero. Ciò solleva la questione se un punteggio di sentimento addestrato sulle recensioni dei film sia appropriato per i prezzi delle azioni.

L'attributo `sentiment_assessments` elenca i valori sottostanti per ogni token e può aiutarci a capire il motivo del sentimento generale di una frase:

```
text = "Bayer (OTCPK:BAYRY) started the week up 3.5% to
euro74/share in Frankfurt, touching their highest l
evel in 14 months, after the U.S. government said
a $25M glyphosate decision against the
company should be reversed."
```

```
TextBlob(text).sentiment_assessments
```

Output

```
Sentiment(polarity=0.5, subjectivity=0.5,
assessments=[(['touching'], 0.5, 0.5, None)])
```

Vediamo che l'affermazione ha un sentimento positivo di 0,5, ma sembra che la parola "touching" abbia dato origine al sentimento positivo. Parole più intuitive, come "high", no. Questo esempio mostra che il contesto dei dati di addestramento è importante affinché il punteggio del sentimento sia significativo. Sono disponibili molti pacchetti e funzioni predefinite per l'analisi del sentimento, ma è importante prestare attenzione e avere una conoscenza approfondita del contesto del problema prima di utilizzare una funzione o un algoritmo per l'analisi del sentimento.

Detto ciò, per questo caso di studio potremmo aver bisogno di sentimenti formati sulle notizie finanziarie.

Apprendimento supervisionato: algoritmi di classificazione e LSTM

In questa fase, sviluppiamo un modello personalizzato per l'analisi del sentimento basato sui dati etichettati disponibili. I dati etichettati sono ottenuti dal sito web di Kaggle:

	datetime	headline	ticker	sentiment
0	1/16/2020 5:25	\$MMM fell on hard times but could be set to re...	MMM	0
1	1/11/2020 6:43	Wolfe Research Upgrades 3M \$MMM to j§Peer Perf...	MMM	1
2	1/9/2020 9:37	3M \$MMM Upgraded to j§Peer Performj" by Wolfe ...	MMM	1
3	1/8/2020 17:01	\$MMM #insideday follow up as it also opened up...	MMM	1
4	1/8/2020 7:44	\$MMM is best #dividend #stock out there and do...	MMM	0

I dati hanno i titoli delle notizie su 30 azioni diverse, per un totale di 9.470 righe, e hanno sentimenti etichettati zero e uno.

Per eseguire un modello di apprendimento supervisionato, dobbiamo prima convertire i titoli delle notizie in una rappresentazione delle caratteristiche. Per questo esercizio, le rappresentazioni vettoriali sottostanti provengono da un *modello di incorporamento di parole spaCy*, che generalmente produce una rappresentazione semantica densa e multidimensionale delle parole (come mostrato nell'esempio seguente).

Il modello di word embedding include 20.000 vettori univoci con 300 dimensioni. Lo applichiamo a tutti i titoli dei dati elaborati nel passaggio precedente.

Ora che abbiamo preparato la variabile indipendente, addestriamo il modello di classificazione. Abbiamo l'etichetta dei sentimenti zero o uno come

variabile dipendente. Per prima cosa dividiamo i dati in set di addestramento e di test ed eseguiamo i principali modelli di classificazione (ad esempio, regressione logistica, CART, SVM, foresta casuale e rete neurale artificiale).

Includeremo anche LSTM, che è un modello basato su RNN, nell'elenco dei modelli considerati. Un modello basato su RNN funziona bene per l'NLP, perché memorizza le informazioni per le caratteristiche correnti e quelle vicine per la previsione. Mantiene una memoria basata su informazioni passate, che consente al modello di prevedere l'output corrente condizionato da caratteristiche a lunga distanza e guarda le parole nel contesto dell'intera frase, piuttosto che guardare semplicemente le singole parole.

RNN

Le reti neurali ricorrenti (RNN) sono chiamate "ricorrenti" perché eseguono lo stesso compito per ogni elemento di una sequenza, con l'output che dipende dai calcoli precedenti. I modelli RNN hanno una memoria, che cattura informazioni su ciò che è stato calcolato finora. Come mostrato nella figura 8.7, una rete neurale ricorrente può essere pensata come copie multiple della stessa rete, ognuna delle quali passa un messaggio a un successore.

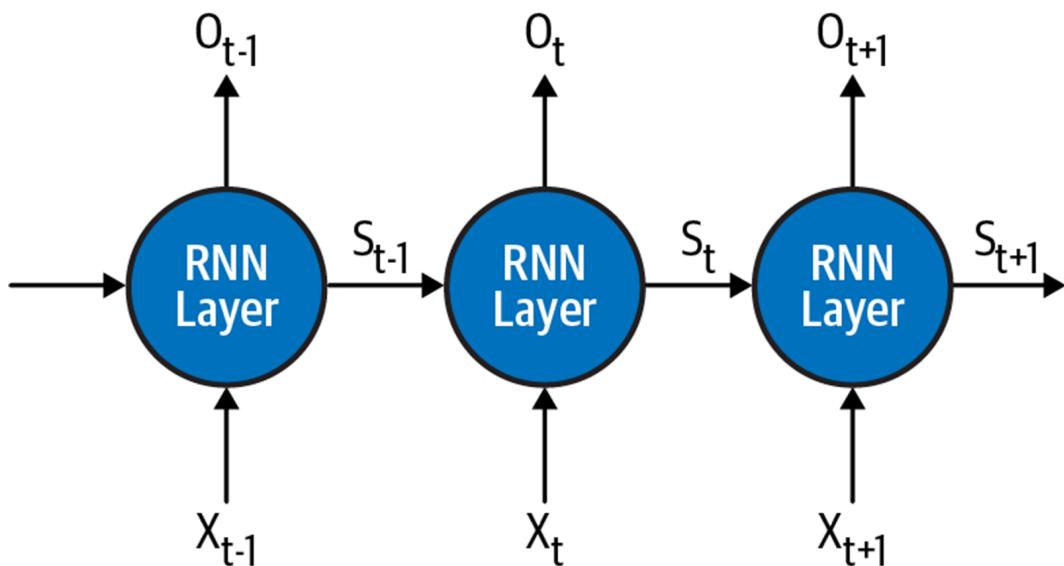


Figura 8.7: Rete neurale ricorrente

Nella figura 8.7:

- X_t è l'input al passo temporale t

- O_t è l'output al passo temporale t
- S_t è lo stato nascosto al passo temporale t. È la memoria della rete. Viene calcolato in base allo stato nascosto precedente e all'input nel passo corrente.

La caratteristica principale di un RNN è questo stato nascosto, che cattura alcune informazioni su una sequenza e le utilizza di conseguenza quando necessario.

Long short-term memory LSTM

La long short-term memory (LSTM) è un tipo speciale di RNN progettato esplicitamente per evitare il problema della dipendenza a lungo termine. Ricordare le informazioni per lunghi periodi di tempo è praticamente un comportamento predefinito per un modello LSTM. Questi modelli sono composti da un insieme di celle con caratteristiche per memorizzare la sequenza dei dati.

Queste celle catturano e memorizzano i flussi di dati. Inoltre, le celle interconnettono un modulo del passato a un altro modulo del presente per convegliare informazioni da diversi istanti temporali passati a quello presente. A causa dell'uso di porte in ogni cella, i dati in ogni cella possono essere eliminati, filtrati o aggiunti per le celle successive.

Le porte, basate su strati di rete neurale artificiale, consentono alle celle di far passare o eliminare i dati. Ogni livello produce numeri nell'intervallo da zero a uno, che rappresentano la quantità di ogni segmento di dati che dovrebbe essere lasciato passare in ogni cella. Più precisamente, una stima di valore zero implica "non far passare nulla". Una stima di uno indica "lascia passare tutto". In ogni LSTM sono coinvolti tre tipi di porte, con l'obiettivo di controllare lo stato di ogni cella:

- Porta dimenticata

Restituisce un numero compreso tra zero e uno, dove uno mostra "mantienilo completamente" e zero implica "ignoralo completamente". Questa porta decide condizionatamente se il passato debba essere dimenticato o preservato.

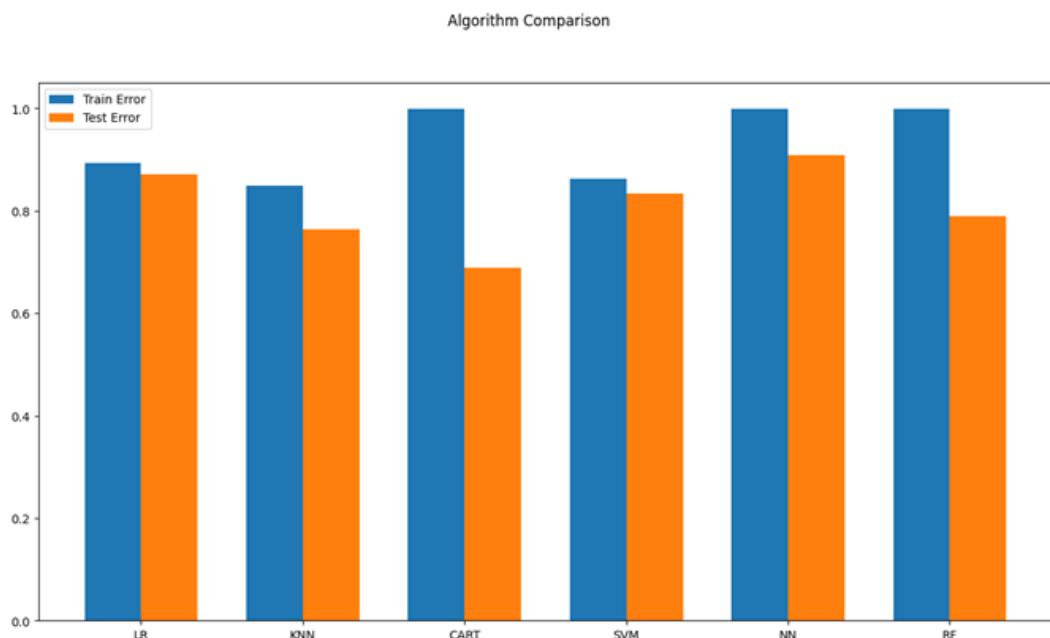
- Porta di ingresso

Scegli quali nuovi dati devono essere memorizzati nella cella.

- Porta d'uscita

Decide cosa produrrà da ogni cella. Il valore ottenuto sarà basato sullo stato della cella insieme ai dati filtrati e appena aggiunti.

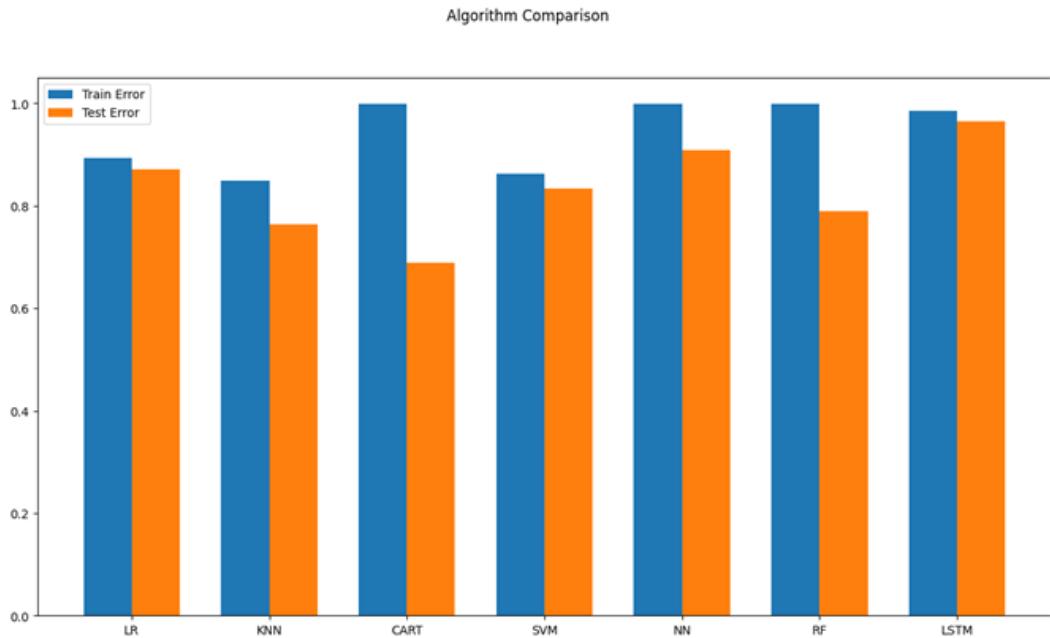
In primo luogo, addestriamo e testiamo i principali algoritmi di ML per la classificazione e poi confrontiamo le loro prestazioni.



Come possiamo vedere anche nel grafico, il modello NN è il migliore con un'accuratezza di training del 99% e un'accuratezza del test del 90%. Anche le prestazioni di Random Forest, SVM e Logistic Regression sono buone.

CART e KNN non funzionano bene come gli altri modelli. Ora, addestriamo e testiamo il modello basato sulle LSTM, e vediamo le sue prestazioni.

Il confronto di tutti i modelli di machine learning è il seguente:



Come previsto, il modello LSTM ha le migliori prestazioni nel set di test (precisione del 96,5%) rispetto a tutti gli altri modelli. Le prestazioni dell'ANN, con un'accuratezza del set di addestramento del 99% e un'accuratezza del set di test del 90,8%, sono paragonabili al modello basato su LSTM. Anche le prestazioni di Random Forest (RF), SVM e regressione logistica (LR) sono ragionevoli. CART e KNN non funzionano bene come altri modelli. Quindi, usiamo il modello LSTM per il calcolo dei sentimenti nei dati dei passaggi seguenti.

Apprendimento non supervisionato: modello basato su lessico finanziario

In questo caso di studio, aggiorniamo il lessico VADER con parole e sentimenti da un lessico adattato alle conversazioni di borsa nei servizi di micro-blogging:

- Lessici

Dizionari o vocabolari speciali che sono stati creati per analizzare i sentimenti. La maggior parte dei lessici ha un elenco di parole polari positive e negative con un punteggio ad esse associato. Utilizzando varie tecniche, come la posizione delle parole, le parole circostanti, il contesto, le parti del discorso e le frasi, i punteggi vengono assegnati ai documenti di testo per i quali vogliamo calcolare il sentimento. Dopo aver aggregato questi punteggi, otteniamo il sentimento finale.

- VADER(Valence Aware Dictionary for Sentiment Reasoning)

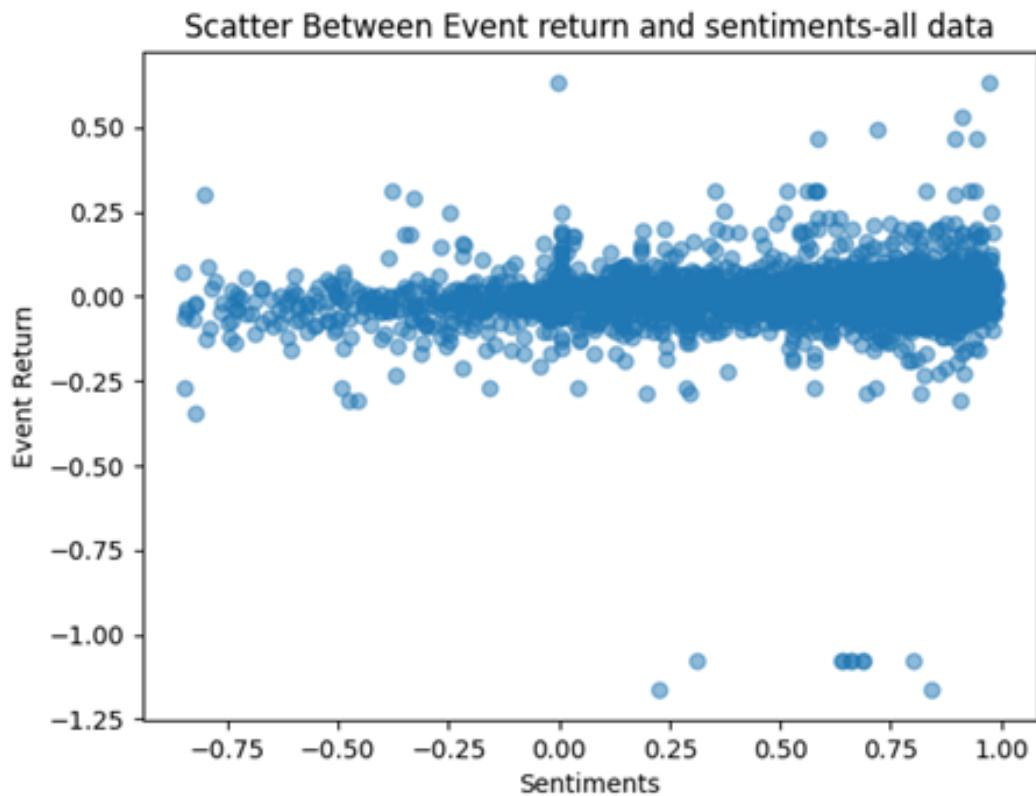
Un modello di analisi del sentimento predefinito incluso nel pacchetto NLTK. Può fornire punteggi di polarità sia positivi che negativi, nonché la forza dell'emozione di un campione di testo. È basato su regole e fa molto affidamento su testi classificati da persone. Si tratta di parole o di qualsiasi forma testuale di comunicazione etichettata in base al loro orientamento semantico come positiva o negativa.

Questa risorsa lessicale è stata creata automaticamente utilizzando diverse misure statistiche e un ampio set di messaggi etichettati da StockTwits, una piattaforma di social media progettata per condividere idee tra investitori, trader e imprenditori. I sentimenti sono compresi tra -1 e 1, simili ai sentimenti di TextBlob. A seguito addestriamo il modello in base ai sentimenti finanziari e osserviamo i risultati.

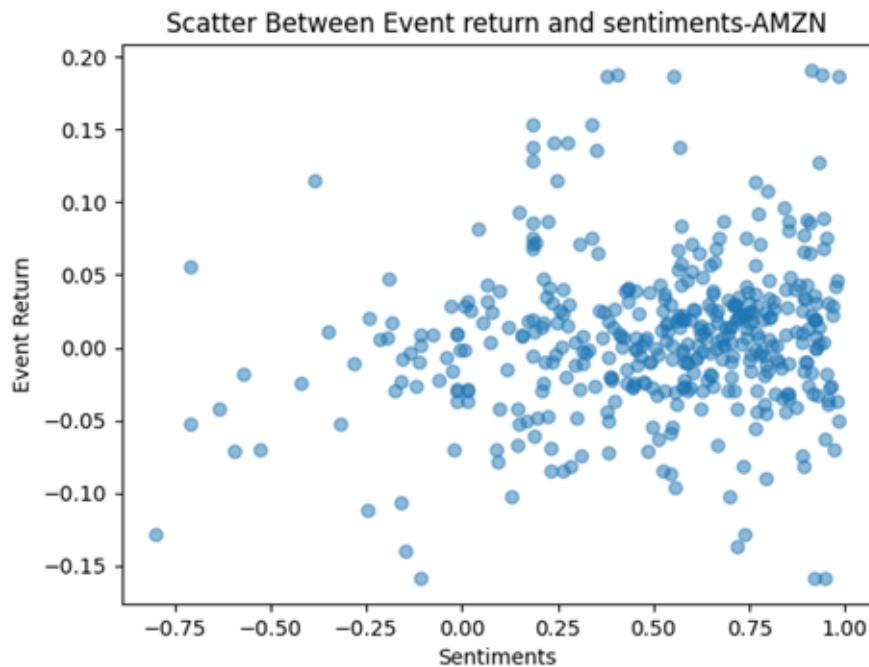
Controlliamo il sentimento per una notizia:

```
#estraiemo il punteggio del sentimento da una frase
text = "AAPL is trading higher after reporting
its October sales rose 12.6% M/M.
It has seen a 20%+ jump in orders"
sia.polarity_scores(text)[ 'compound' ]
Output
0.4535
```

Otteniamo i sentimenti per tutti i titoli delle notizie in base al nostro dataset. Esaminiamo la correlazione tra i rendimenti e i sentimenti, che viene calcolata utilizzando la metodologia basata sul lessico per l'intero dataset.



Non vediamo molti rendimenti elevati per sentimenti inferiori, ma i dati potrebbero non essere molto chiari. Diamo un'occhiata al risultato per uno dei titoli di borsa.



Vediamo una correlazione positiva tra il rendimento dell'evento e il sentimento. Approfondiremo il confronto tra diversi tipi di analisi del sentimento nella prossima sezione.

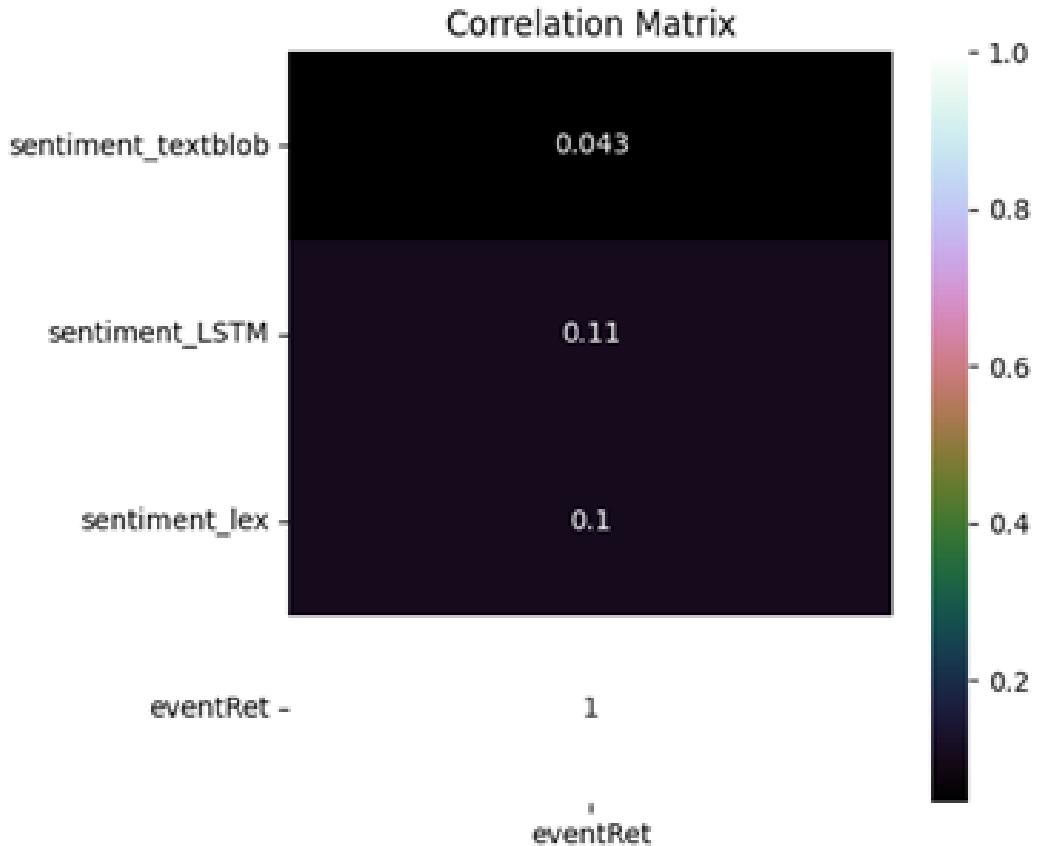
Analisi esplorativa dei dati e confronto

In questa sezione, confrontiamo i sentimenti calcolati utilizzando le diverse tecniche presentate sopra. Diamo un'occhiata ai titoli di esempio e ai sentimenti di tre diverse metodologie, seguiti da un'analisi visiva:

	ticker	headline	sentiment_textblob	sentiment_LSTM	sentiment_lex
1	NFLX	Netflix (NFLX +1.1%) shares post early gains after Citigroup ups its rating to Buy and lifts its price target to \$300 from \$245. U.S. revenue growth is sustainable, Citi says, "with a path to 50M subscribers by 2013," adding that NFLX has little competition in price, selection and convenience; mass market adoption of tablets will help, and the mass-market adoption phase is still to come.	-0.04375	1	0.8575

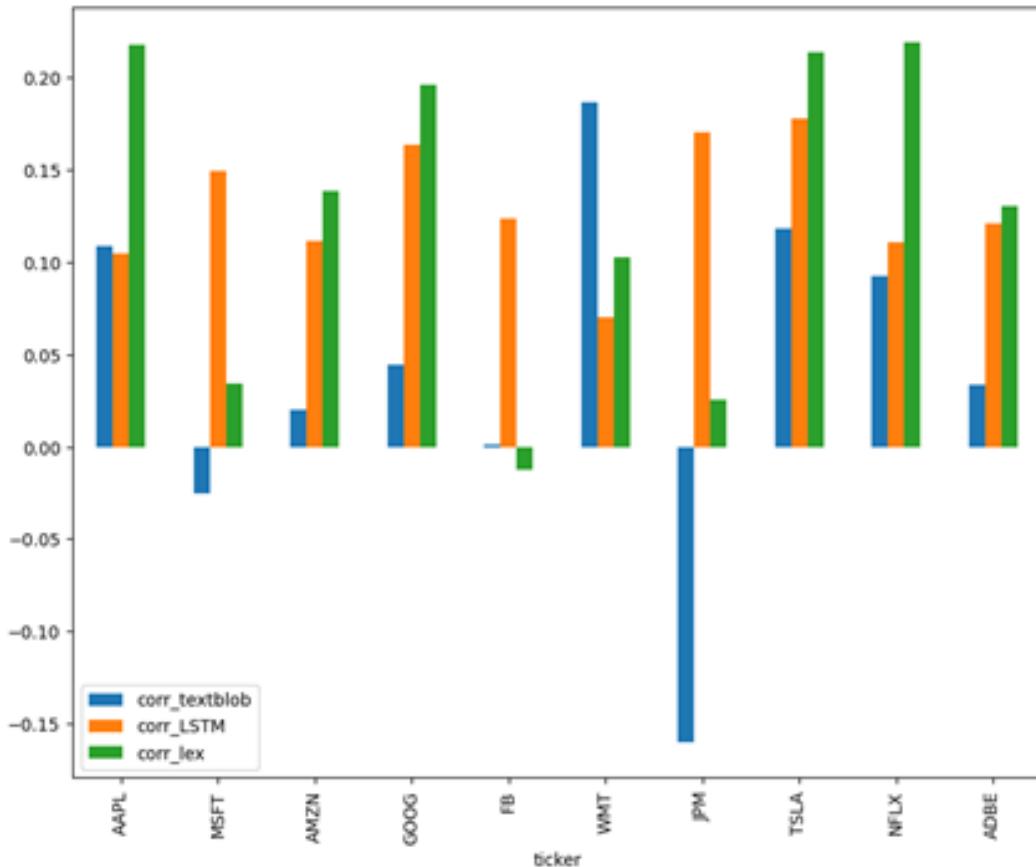
Guardando uno dei titoli, il sentimento di questa frase è positivo. Tuttavia, il risultato del sentiment TextBlob è di poco inferiore a zero, suggerendo che il sentimento è più neutro. Ciò rimanda all'ipotesi precedente secondo cui il modello addestrato sui sentimenti cinematografici probabilmente non sarà accurato per i sentimenti azionari. Il modello basato sulla classificazione (LSTM) suggerisce correttamente che il sentimento è positivo, ma è binario. Sentiment_lex ha un output più intuitivo con un sentimento significativamente positivo.

Rivediamo la correlazione di tutti i sentimenti di diverse metodologie rispetto ai rendimenti:

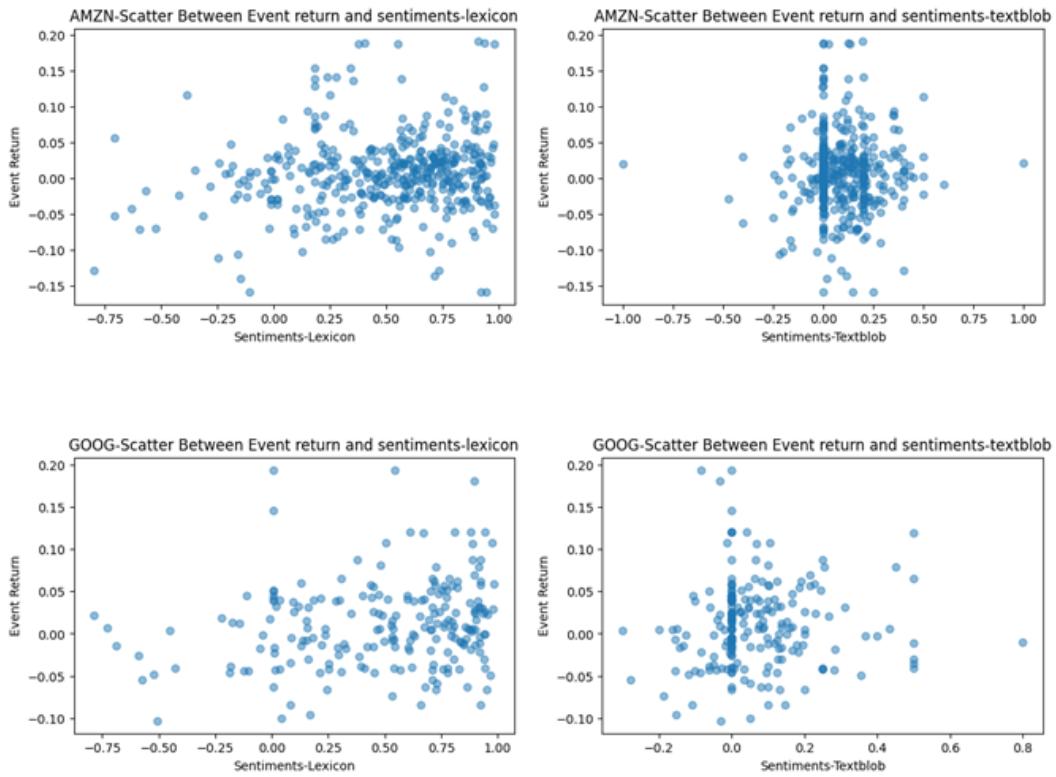


Tutti i sentimenti hanno relazioni positive con i rendimenti, il che è intuitivo e prevedibile. I sentimenti della metodologia del lessico sono i più alti, il che significa che il rendimento dell'evento del titolo (azione) può essere previsto al meglio utilizzando la metodologia del lessico. Ricordiamo che questa metodologia sfrutta i termini finanziari nel modello. Anche il metodo basato su LSTM offre prestazioni migliori rispetto all'approccio TextBlob, ma le prestazioni sono leggermente peggiori rispetto alla metodologia basata sul lessico.

Diamo un'occhiata alle prestazioni della metodologia a livello di ticker (azioni). Abbiamo scelto alcuni ticker con la capitalizzazione di mercato più alta per l'analisi:



Guardando il grafico, la correlazione dalla metodologia del lessico è più alta in tutti i titoli azionari, il che conferma la conclusione dell'analisi precedente. Significa che i rendimenti possono essere previsti al meglio utilizzando la metodologia del lessico. I sentimenti basati su TextBlob mostrano risultati non intuitivi in alcuni casi, ad esempio con JPM. Esaminiamo il grafico a dispersione per il lessico rispetto alle metodologie TextBlob per AMZN e GOOG. Metteremo da parte il metodo basato su LSTM poiché i sentimenti binari non saranno significativi nel grafico a dispersione:



I sentimenti basati sul lessico a sinistra mostrano una relazione positiva tra sentimenti e rendimenti. Alcuni dei punti con i rendimenti più alti sono associati alle notizie più positive. Inoltre, il grafico a dispersione è distribuito in modo più uniforme nel caso del lessico rispetto a TextBlob. I sentimenti per TextBlob sono concentrati intorno allo zero, probabilmente perché il modello non è in grado di classificare bene i sentimenti finanziari. Per la strategia di trading, utilizzeremo i sentimenti basati sul lessico, in quanto questi sono i più appropriati in base all'analisi in questa sezione. Anche i sentimenti basati su LSTM sono buoni, ma sono etichettati come zero o uno, sono quindi preferiti i sentimenti basati sul lessico più granulari.

Valutazione dei modelli: costruzione di una strategia di trading

I dati sul sentimento possono essere utilizzati in diversi modi per costruire una strategia di trading. I sentimenti possono essere utilizzati come segnale autonomo per decidere azioni di acquisto, vendita o attesa. Il punteggio del sentimento o i vettori di parole possono essere utilizzati anche per prevedere il rendimento o il prezzo di un'azione. Tale previsione può essere utilizzata per costruire una strategia di trading.

In questa sezione, dimostriamo una strategia di trading in cui acquistiamo o vendiamo azioni in base al seguente approccio:

- Acquista un'azione quando la variazione del punteggio del sentimento (punteggio del sentimento attuale/punteggio del sentimento precedente) è maggiore di 0,5. Vendì un'azione quando la variazione del punteggio del sentimento è inferiore a -0,5. Il punteggio del sentimento utilizzato qui si basa sui sentimenti basati sul lessico calcolati nel passaggio precedente.
- Oltre ai sentimenti, utilizziamo la media mobile (basata sugli ultimi 15 giorni) mentre prendiamo una decisione di acquisto o vendita.
- Le negoziazioni (ad esempio, acquisto o vendita) sono in *lotti*³ da 100 azioni (titoli/shares). L'importo iniziale disponibile per il trading è fissato a \$ 100.000.

La soglia della strategia, la dimensione del lotto e il capitale iniziale possono essere modificati in base alla performance della strategia.

Impostazione di una strategia

Per impostare la strategia di trading, utilizziamo backtrader, che è un comodo framework basato su Python per l'implementazione e il backtest delle strategie di trading. Backtrader ci consente di scrivere strategie, indicatori e analizzatori di trading riutilizzabili invece di dover dedicare tempo alla costruzione di infrastrutture. Utilizziamo il codice Quickstart della documentazione di backtrader come base e lo adattiamo alla nostra strategia di trading basata sul sentimento.

Il seguente frammento di codice riassume la logica di acquisto e vendita per la strategia. Fare riferimento al file Jupyter Notebook NLPAndSentimentAnalysisBasedTradingStrategy_cap8.ipynb, sviluppato da Hariom Tatsat, nella sezione 5.2 Results for Individual Stocks di questo caso di studio per l'implementazione dettagliata:

```
1 # buy if current close more than simple moving average (sma)
2 # AND sentiment increased by >= 0.5
3 if self.dataclose[0] > self.sma[0] and
4 self.sentiment - prev_sentiment >= 0.5:
```

³I lotti nei titoli e nel trading rappresentano il numero di unità di uno strumento finanziario acquistate in borsa. In genere, il numero di unità è indicato dal nome del lotto. Ad esempio, nel mercato azionario, un lotto arrotondato è di 100 azioni. Tuttavia, gli investitori non devono acquistare lotti arrotondati. Un lotto può essere qualsiasi numero di azioni. Ad esempio, un lotto dispari è il termine utilizzato quando vengono acquistate meno di 100 azioni.

```
5     self.order = self.buy()  
6  
7 # sell if current close less than simple moving average(sma)  
8 # AND sentiment decreased by >= 0.5  
9 if self.dataclose[0] < self.sma[0] and  
10 self.sentiment - prev_sentiment <= -0.5:  
11     self.order = self.sell()
```

Analisi dei risultati per i singoli titoli

Innanzitutto, eseguiamo la nostra strategia su GOOG e osserviamo i risultati:

```
1 ticker = 'GOOG'  
2 run_strategy(ticker, start = '2012-01-01',  
3                 end = '2018-12-12')
```

L'output mostra il log di trading per alcuni giorni e il rendimento finale:

```

GOOG
Starting Portfolio Value: 1000000.00
2012-04-12, Previous Sentiment 0.24, New Sentiment 0.80 BUY CREATE, 324.29
2012-04-13, BUY EXECUTED, Price: 322.57, Cost: 32257.00, Comm 0.00
2012-10-18, Previous Sentiment 0.98, New Sentiment 0.08 SELL CREATE, 346.20
2012-10-19, SELL EXECUTED, Price: 351.47, Cost: 32257.00, Comm 0.00
2012-10-19, OPERATION PROFIT, GROSS 2890.00, NET 2890.00
2013-01-10, Previous Sentiment 0.08, New Sentiment 0.80 BUY CREATE, 369.36
2013-01-11, BUY EXECUTED, Price: 369.61, Cost: 36961.00, Comm 0.00
2014-07-17, Previous Sentiment 0.73, New Sentiment -0.22 SELL CREATE, 572.16
2014-07-18, SELL EXECUTED, Price: 591.38, Cost: 36961.00, Comm 0.00
2014-07-18, OPERATION PROFIT, GROSS 22177.00, NET 22177.00
2014-07-18, Previous Sentiment -0.22, New Sentiment 0.77 BUY CREATE, 593.45
2014-07-21, BUY EXECUTED, Price: 590.13, Cost: 59013.00, Comm 0.00
2014-09-12, Previous Sentiment 0.66, New Sentiment -0.05 SELL CREATE, 574.04
2014-09-15, SELL EXECUTED, Price: 571.37, Cost: 59013.00, Comm 0.00
2014-09-15, OPERATION PROFIT, GROSS -1876.00, NET -1876.00
2015-07-17, Previous Sentiment 0.01, New Sentiment 0.90 BUY CREATE, 672.93
2015-07-20, BUY EXECUTED, Price: 659.24, Cost: 65924.00, Comm 0.00
2015-08-24, Previous Sentiment 0.91, New Sentiment -0.51 SELL CREATE, 589.61
2015-08-25, SELL EXECUTED, Price: 614.91, Cost: 65924.00, Comm 0.00
2015-08-25, OPERATION PROFIT, GROSS -4433.00, NET -4433.00
2015-10-22, Previous Sentiment -0.79, New Sentiment 0.73 BUY CREATE, 651.79
2015-10-23, BUY EXECUTED, Price: 727.50, Cost: 72750.00, Comm 0.00
2017-09-25, Previous Sentiment 0.69, New Sentiment 0.14 SELL CREATE, 920.97
2017-09-26, SELL EXECUTED, Price: 923.72, Cost: 72750.00, Comm 0.00
2017-09-26, OPERATION PROFIT, GROSS 19622.00, NET 19622.00
2017-10-27, Previous Sentiment 0.01, New Sentiment 0.60 BUY CREATE, 1019.27
2017-10-30, BUY EXECUTED, Price: 1014.00, Cost: 101400.00, Comm 0.00
2018-04-09, Previous Sentiment 0.54, New Sentiment -0.45 SELL CREATE, 1015.45
2018-04-10, SELL EXECUTED, Price: 1026.44, Cost: 101400.00, Comm 0.00
2018-04-10, OPERATION PROFIT, GROSS 1244.00, NET 1244.00
2018-04-23, Previous Sentiment 0.03, New Sentiment 0.87 BUY CREATE, 1067.45
2018-04-24, BUY EXECUTED, Price: 1052.00, Cost: 105200.00, Comm 0.00
2018-09-04, Previous Sentiment 0.76, New Sentiment 0.01 SELL CREATE, 1197.00
2018-09-05, SELL EXECUTED, Price: 1193.80, Cost: 105200.00, Comm 0.00
2018-09-05, OPERATION PROFIT, GROSS 14180.00, NET 14180.00
2018-10-01, Previous Sentiment 0.01, New Sentiment 0.88 BUY CREATE, 1195.31
2018-10-02, BUY EXECUTED, Price: 1190.96, Cost: 119096.00, Comm 0.00
2018-10-05, Previous Sentiment 0.88, New Sentiment 0.31 SELL CREATE, 1157.35
2018-10-08, SELL EXECUTED, Price: 1150.11, Cost: 119096.00, Comm 0.00
2018-10-08, OPERATION PROFIT, GROSS -4085.00, NET -4085.00
2018-12-11, (MA Period 15) Ending Value 149719.00|
Start Portfolio value: 1000000.00
Final Portfolio Value: 149719.00
Profit: 49719.00

```

Analizziamo il risultato del backtesting nel seguente grafico prodotto dal pacchetto backtrader. Fare riferimento al Jupyter Notebook NLPAndSentimentAnalysisBasedTradingStrategy_cap8.ipynb, sviluppato da Hariom Tatsat, nella sezione 5.2 Results for Individual Stocks di questo caso di studio per la versione dettagliata di questo grafico.



I risultati mostrano un profitto complessivo di \$ 49.719. Il grafico è un tipico grafico prodotto dal pacchetto backtrader ed è suddiviso in quattro riquadri:

- Pannello superiore

Il pannello superiore è l'osservatore del valore in denaro. Tiene traccia della liquidità e del valore totale del portafoglio durante la durata dell'esecuzione dei backtest. In questa esecuzione, abbiamo iniziato con \$ 100.000 e terminato con \$ 149.719.

- Secondo pannello

Questo pannello è l'osservatore commerciale. Mostra il profitto/la perdita realizzati di ogni operazione. Un'operazione è definita come l'apertura di una posizione e il ripristino della posizione a zero (direttamente o passando da long a short o da short a long). Guardando questo pannello, cinque operazioni su otto sono redditizie per la strategia.

- Terzo pannello

Questo panel è l'osservatore di acquisti e vendite. Indica dove sono avvenute le operazioni di acquisto e vendita. In generale, vediamo che l'azione di acquisto ha luogo quando il prezzo dell'azione è in aumento e l'azione di vendita ha luogo quando il prezzo dell'azione ha iniziato a scendere.

- Pannello inferiore

Questo pannello mostra il punteggio del sentimento, che varia tra -1 e 1.

Ora scegliamo uno dei giorni (17-07-2015) in cui è stata attivata un'azione di acquisto e analizziamo le notizie per Google in quel giorno e nel giorno precedente:

```
GOOG_ticker= data_df[ data_df[ 'ticker '].isin ([ ticker ]) ]
New= list (GOOG_ticker [GOOG_ticker [ 'date ']==
                  '2015-07-17'][ 'headline '])
Old= list (GOOG_ticker [GOOG_ticker [ 'date ']==
                  '2015-07-16'][ 'headline '])
print (" Current News:" ,New," \n\n" , " Previous News:" , Old)
```

Output

Current News: [”Axiom Securities has upgraded Google (GOOG +13.4%, GOOGL +14.8%) to Buy following the company’s Q2 beat and investor-pleasing comments about spending discipline, potential capital returns, and YouTube/mobile growth. MKM has launched coverage at Buy, and plenty of other firms have hiked their targets. Google’s market cap is now above \$450B.”]

Previous News: [”While Google’s (GOOG, GOOGL) Q2 revenue slightly missed estimates when factoring traffic acquisitions costs (TAC), its ex-TAC revenue of \$14.35B was slightly above a \$14.3B consensus. The reason: TAC fell to 21% of ad revenue from Q1’s 22% and Q2 2014’s 23%. That also, of course, helped EPS beat estimates.”, ‘Google (NASDAQ:GOOG): QC2 EPS of \$6.99 beats by \$0.28.’]

Chiaramente la notizia del giorno prescelto cita l’upgrade di Google, una notizia positiva. Il giorno precedente menziona le stime mancanti delle entrate, che è una notizia negativa. Quindi, c’è stato un cambiamento significativo del sentimento delle notizie nel giorno selezionato, che ha portato a un’azione di acquisto attivata dall’algoritmo di trading.

Ora eseguiamo la strategia per FB (Facebook).

¹ ticker = ‘FB’

² run_strategy(ticker, start = ‘2012-01-01’, end = ‘2018-12-12’)

3

4 Output

```

FB
Starting Portfolio Value: 100000.00
2012-11-14, Previous Sentiment -0.82, New Sentiment 0.71 BUY CREATE, 22.36
2012-11-15, BUY EXECUTED, Price: 22.34, Cost: 2234.00, Comm 0.00
2013-03-13, Previous Sentiment 0.95, New Sentiment -0.25 SELL CREATE, 27.08
2013-03-14, SELL EXECUTED, Price: 27.10, Cost: 2234.00, Comm 0.00
2013-03-14, OPERATION PROFIT, GROSS 476.00, NET 476.00
2013-04-10, Previous Sentiment -0.25, New Sentiment 0.94 BUY CREATE, 27.57
2013-04-11, BUY EXECUTED, Price: 27.48, Cost: 2748.00, Comm 0.00
2013-11-18, Previous Sentiment 0.98, New Sentiment 0.40 SELL CREATE, 45.83
2013-11-19, SELL EXECUTED, Price: 46.26, Cost: 2748.00, Comm 0.00
2013-11-19, OPERATION PROFIT, GROSS 1878.00, NET 1878.00
2014-01-30, Previous Sentiment 0.01, New Sentiment 0.82 BUY CREATE, 61.08
2014-01-31, BUY EXECUTED, Price: 60.47, Cost: 6047.00, Comm 0.00
2014-12-16, Previous Sentiment 0.68, New Sentiment -0.39 SELL CREATE, 74.69
2014-12-17, SELL EXECUTED, Price: 75.01, Cost: 6047.00, Comm 0.00
2014-12-17, OPERATION PROFIT, GROSS 1454.00, NET 1454.00
2014-12-22, Previous Sentiment -0.39, New Sentiment 0.73 BUY CREATE, 81.45
2014-12-23, BUY EXECUTED, Price: 82.02, Cost: 8202.00, Comm 0.00
2016-04-27, Previous Sentiment 0.91, New Sentiment 0.01 SELL CREATE, 108.89
2016-04-28, SELL EXECUTED, Price: 119.58, Cost: 8202.00, Comm 0.00
2016-04-28, OPERATION PROFIT, GROSS 3756.00, NET 3756.00
2016-07-06, Previous Sentiment -0.11, New Sentiment 0.81 BUY CREATE, 116.70
2016-07-07, BUY EXECUTED, Price: 116.63, Cost: 11663.00, Comm 0.00
2016-11-18, Previous Sentiment 0.78, New Sentiment -0.13 SELL CREATE, 117.02
2016-11-21, SELL EXECUTED, Price: 118.20, Cost: 11663.00, Comm 0.00
2016-11-21, OPERATION PROFIT, GROSS 157.00, NET 157.00
2016-11-21, Previous Sentiment -0.13, New Sentiment 0.83 BUY CREATE, 121.77
2016-11-22, BUY EXECUTED, Price: 122.40, Cost: 12240.00, Comm 0.00
2016-12-01, Previous Sentiment 0.83, New Sentiment -0.01 SELL CREATE, 115.10
2016-12-02, SELL EXECUTED, Price: 115.11, Cost: 12240.00, Comm 0.00
2016-12-02, OPERATION PROFIT, GROSS -729.00, NET -729.00
2018-04-10, Previous Sentiment 0.03, New Sentiment 0.55 BUY CREATE, 165.04
2018-04-11, BUY EXECUTED, Price: 165.36, Cost: 16536.00, Comm 0.00
2018-08-16, Previous Sentiment 0.80, New Sentiment -0.23 SELL CREATE, 174.70
2018-08-17, SELL EXECUTED, Price: 174.50, Cost: 16536.00, Comm 0.00
2018-08-17, OPERATION PROFIT, GROSS 914.00, NET 914.00
2018-12-03, Previous Sentiment -0.63, New Sentiment 0.47 BUY CREATE, 141.09
2018-12-04, BUY EXECUTED, Price: 140.73, Cost: 14073.00, Comm 0.00
2018-12-11, (MA Period 15) Ending Value 108041.00

Start Portfolio value: 100000.00
Final Portfolio Value: 108041.00
Profit: 8041.00

```



I dettagli dei risultati del backtesting della strategia sono i seguenti:

- Pannello superiore

Il pannello del valore in contanti mostra un profitto complessivo di \$ 8.041.

- Secondo pannello

L'osservatore commerciale mostra che sei azioni su sette sono state redditizie.

- Terzo pannello

L'osservatore di acquisto/vendita mostra che in generale l'azione di acquisto (vendita) ha avuto luogo quando il prezzo delle azioni stava aumentando (diminuendo).

- Pannello inferiore

Mostra un numero elevato di sentimenti positivi per FB nel periodo 2013-2014.

Analisi dei risultati su più titoli

Nel passaggio precedente, abbiamo eseguito la strategia di trading sui singoli titoli. Qui, lo eseguiamo su tutti i 10 titoli per i quali abbiamo calcolato i sentimenti:

```

1 results_tickers = {}
2 for ticker in tickers:
3     results_tickers[ticker] = run_strategy(ticker,
4         start = '2012-01-01',
5         end = '2018-12-12')
6 pd.DataFrame.from_dict(results_tickers).set_index([
7     [pd.Index(["PerUnitStartPrice", "StrategyProfit"])]])
8
9 Output

```



La strategia funziona abbastanza bene e produce un profitto complessivo per tutti i titoli. Come accennato in precedenza, le azioni di acquisto e vendita vengono eseguite in lotti di 100 (round lot). Pertanto, l'importo in dollari utilizzato è proporzionale al prezzo dei titoli (stock). Vediamo il più alto profitto nominale da AMZN e GOOG, che è principalmente attribuito agli elevati importi in dollari investiti per questi titoli dato il loro alto prezzo delle azioni. Oltre al profitto complessivo, per analizzare le prestazioni è possibile utilizzare diverse altre metriche, come lo Sharpe ratio e il drawdown massimo.

Variare il periodo di tempo della strategia

Nell'analisi precedente, abbiamo utilizzato il periodo di tempo dal 2011 al 2018 per il nostro backtesting. In questa fase, per analizzare ulteriormente

l'efficacia della nostra strategia, variamo il periodo di tempo del backtesting e analizziamo i risultati. Innanzitutto, eseguiamo la strategia per tutti i titoli per il periodo di tempo tra il 2012 e il 2014:

```

1 results_tickers = {}
2 for ticker in tickers:
3     results_tickers[ticker] = run_strategy(ticker,
4         start = '2012-01-01',
5         end = '2014-12-31')
6
7 Output

```



La strategia produce un profitto complessivo per tutti i titoli tranne che per AMZN e WMT (Walmart). Ora eseguiamo la strategia per il periodo tra il 2016 e il 2018:

```

1 results_tickers = {}
2 for ticker in tickers:
3     results_tickers[ticker] = run_strategy(ticker,
4         start = '2016-01-01',
5         end = '2018-12-31')
6
7 Output

```



Osserviamo una buona performance della strategia basata sul sentimento su tutti i titoli ad eccezione di AAPL, e possiamo concludere che si comporta abbastanza bene anche in diversi periodi di tempo.

8.4.2 Conclusioni del caso di studio

In questo caso di studio, abbiamo esaminato vari modi in cui i dati non strutturati possono essere convertiti in dati strutturati e quindi utilizzati per l'analisi e la previsione utilizzando strumenti per l'NLP. Abbiamo dimostrato tre diversi approcci, inclusi modelli di deep learning per sviluppare un modello per il calcolo dei sentimenti. Abbiamo eseguito un confronto dei modelli e abbiamo concluso che uno dei passaggi più importanti nell'addestramento del modello per l'analisi del sentimento è l'utilizzo di un vocabolario specifico del dominio.

Abbiamo anche utilizzato un modello preaddestrato da spaCy per convertire una frase in sentimenti e abbiamo utilizzato i sentimenti come segnali per sviluppare una strategia di trading. I risultati iniziali hanno suggerito che il modello addestrato su un sentimento basato sul lessico finanziario potrebbe rivelarsi un modello praticabile per una strategia di trading. Ulteriori miglioramenti possono essere apportati utilizzando modelli di analisi del sentimento preaddestrati più complessi, come BERT di Google, o diversi modelli di NLP preaddestrati disponibili nelle piattaforme open source. Le librerie NLP esistenti sono state utilizzate per estrarre i sentimenti dai titoli di azione, che sono poi stati utilizzati per formare un dataset di training per un modello di classificazione. Il modello ha mostrato una buona capacità di predire i movimenti dei prezzi delle azioni, consentendo di realizzare guadagni significativi.

stenti completano alcune delle fasi di preelaborazione e codifica per consentirci di concentrarci sulla fase di inferenza.

8.5 Caso di studio: Document Summarization (Sintesi del documento)

Il Document Summarization si riferisce alla selezione dei punti e degli argomenti più importanti in un documento e alla loro organizzazione in modo completo. Come discusso in precedenza, gli analisti delle banche e di altre organizzazioni di servizi finanziari esaminano, analizzano e tentano di quantificare i dati qualitativi da notizie, rapporti e documenti.

Il document summarization utilizzando l’NLP può fornire un supporto approfondito in questa analisi e interpretazione. Se adattato a documenti finanziari, come report sugli utili e notizie finanziarie, può aiutare gli analisti a ricavare rapidamente argomenti chiave e segnali di mercato dai contenuti. Il document summarization può essere utilizzato anche per migliorare gli sforzi di reporting e può fornire aggiornamenti tempestivi su questioni chiave.

Nell’NLP, i topic models (come LDA, introdotto in precedenza nel capitolo) sono gli strumenti usati più di frequente per l’estrazione di caratteristiche testuali sofisticate e interpretabili. Questi modelli possono far emergere argomenti, temi o segnali chiave da grandi raccolte di documenti e possono essere utilizzati efficacemente per il document summarization.

In questo caso di studio, ci concentreremo su:

- Implementazione del modello LDA per la modellazione degli argomenti.
- Comprendere la necessaria preparazione dei dati (ad esempio, convertire un PDF per un problema relativo all’NLP).
- Visualizzazione dell’argomento.

8.5.1 Utilizzo dell’NLP per il Document Summarization

Definizione del problema

L’obiettivo di questo caso di studio è scoprire in modo efficace argomenti comuni dalle *earnings call transcripts*⁴ delle società quotate in borsa che

⁴Una earning call è una teleconferenza tra la direzione di una società pubblica , analisti, investitori e media per discutere i risultati finanziari della società durante un determinato periodo di riferimento, ad esempio un trimestre o un anno fiscale. Una earning call è solitamente preceduta da un rapporto sugli utili, che contiene informazioni di riepilogo sulla performance finanziaria per il periodo.

utilizzano LDA. Un vantaggio fondamentale di questa tecnica rispetto ad altri approcci è che non è necessaria alcuna conoscenza preliminare degli argomenti.

Preparazione dei dati

Per questo caso di studio, estrarremo il testo da un PDF. Quindi, la libreria Python pdf-miner viene utilizzata per elaborare i file PDF in un formato di testo. Vengono usate anche le librerie per l'estrazione delle caratteristiche e la modellazione degli argomenti. La funzione convert_pdf_to_txt estrae tutti i caratteri da un documento PDF tranne le immagini. La funzione accetta semplicemente il documento PDF, estrae tutti i caratteri dal documento e restituisce il testo estratto come un elenco di stringhe Python. Per i dettagli implementativi dell'intero caso di studio si rimanda al file Jupyter Notebook NLPCasodiStudio2_cap8.ipynb messo in allegato.

Diamo un'occhiata al documento grezzo:

```
[ , ,
, ,
'SECURITIES AND EXCHANGE COMMISSION' ,
, ,
, ,
'Washington , D.C. 20549' ,
, ,
, ,
'\xa0' ,
'FORM ' ,
'\xa0' ,
, ,
'QUARTERLY REPORT PURSUANT TO SECTION 13 OR 15(d) OF' ,
, ]
```

Osserviamo che il testo estratto dal documento PDF contiene caratteri non informativi che devono essere rimossi. Questi caratteri riducono l'efficacia dei nostri modelli poiché forniscono rapporti di conteggio non necessari.

```
[ 'SECURITIES AND EXCHANGE COMMISSION' ,
'Washington D C ' ,
, ,
'FORM ' ,
, ,
'QUARTERLY REPORT PURSUANT TO SECTION OR d OF' ,
'THE SECURITIES EXCHANGE ACT OF ' ,
'For the quarterly period ended September ' ,
```

```
'Commission file number ',  
,  
,  
,  
,  
'WELLS FARGO COMPANY ',  
' Exact name of registrant as specified in its charter ',  
'Delaware ']
```

Come possiamo vedere i caratteri non informativi vengono sostituiti da uno spazio vuoto.

Costruzione ed addestramento del modello

La funzione CountVectorizer del modulo sklearn viene utilizzata con una regolazione minima dei parametri per rappresentare il documento pulito come una matrice di termini del documento. Questo viene eseguito perché la nostra modellazione richiede che le stringhe siano rappresentate come numeri interi. CountVectorizer mostra il numero di volte in cui una parola compare nell'elenco dopo la rimozione delle parole non significative. La matrice dei termini del documento è stata formattata in un dataframe per ispezionare il set di dati. Questo dataframe mostra il conteggio delle occorrenze di parola di ciascun termine nel documento:

aa	aaa	abbot	ability	able	abs	absorb	absorbed	absorbing	abusive	...	years	yes	yield	yielding	yields	york	yrs	zealand	zero	zip
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1 rows x 3128 columns

Nella fase successiva, la matrice dei termini del documento verrà utilizzata come input per l'algoritmo LDA per la modellazione degli argomenti. L'algoritmo è stato adattato per isolare cinque contesti tematici distinti. Questo valore può essere regolato in funzione del livello di granularità che si intende ottenere dalla modellazione.

Il codice seguente utilizza la libreria mglearn per visualizzare le prime 10 parole all'interno di ogni specifico topic model:

```
1 import mglearn
2 mglearn.tools.print_topics(topics=range(5), feature_names=features,
3    sorting=sorting, topics_per_chunk=5, n_words=10)
4
5 Output
6 topic 1          topic 2          topic 3          topic 4          topic 5
7 _____          _____          _____          _____          _____
8 assets          quarter         loans           securities      value
9 balance         million        mortgage        rate            total
```

10	losses	risk	loan	investment	income
11	credit	capital	commercial	contracts	net
12	period	months	total	credit	fair
13	derivatives	financial	real	market	billion
14	liabilities	management	estate	federal	equity
15	derivative	billion	securities	stock	september
16	allowance	ended	consumer	debt	december
17	average	september	backed	sales	table

Ogni argomento nella tabella dovrebbe rappresentare un tema più ampio. Tuttavia, dato che abbiamo addestrato il modello su un solo documento, i temi tra gli argomenti potrebbero non essere molto distinti l'uno dall'altro.

Guardando all'insieme, l'argomento 2 discute i trimestri, i mesi e le unità monetarie relative alla valutazione degli asset. L'argomento 3 rivela informazioni sui redditi da immobili, mutui e strumenti correlati. L'argomento 5 ha anche termini relativi alla valutazione delle attività. Il primo argomento fa riferimento alle voci di bilancio e ai derivati. L'argomento 4 è leggermente simile all'argomento 1 e contiene parole relative a un processo di investimento.

In termini di tema generale, gli argomenti 2 e 5 sono ben distinti dagli altri. Potrebbe esserci anche qualche somiglianza tra gli argomenti 1 e 4, in base alle parole principali. Nella prossima sezione cercheremo di comprendere la separazione tra questi argomenti utilizzando la libreria Python pyLDAvis.

Visualizzazione degli argomenti

In questa sezione, visualizziamo gli argomenti utilizzando diverse tecniche.

Visualizzazione dell'argomento

La visualizzazione dell'argomento facilita la valutazione della qualità dell'argomento utilizzando il giudizio umano. pyLDAvis è una libreria che mostra le relazioni globali tra argomenti facilitando al contempo la loro valutazione semantica ispezionando i termini più strettamente associati a ciascun argomento e, inversamente, gli argomenti associati a ciascun termine. Affronta anche la sfida in cui i termini usati di frequente in un documento tendono a dominare la distribuzione rispetto alle parole che definiscono un argomento. Di seguito, la libreria pyLDAvis viene utilizzata per visualizzare i topic models:

```

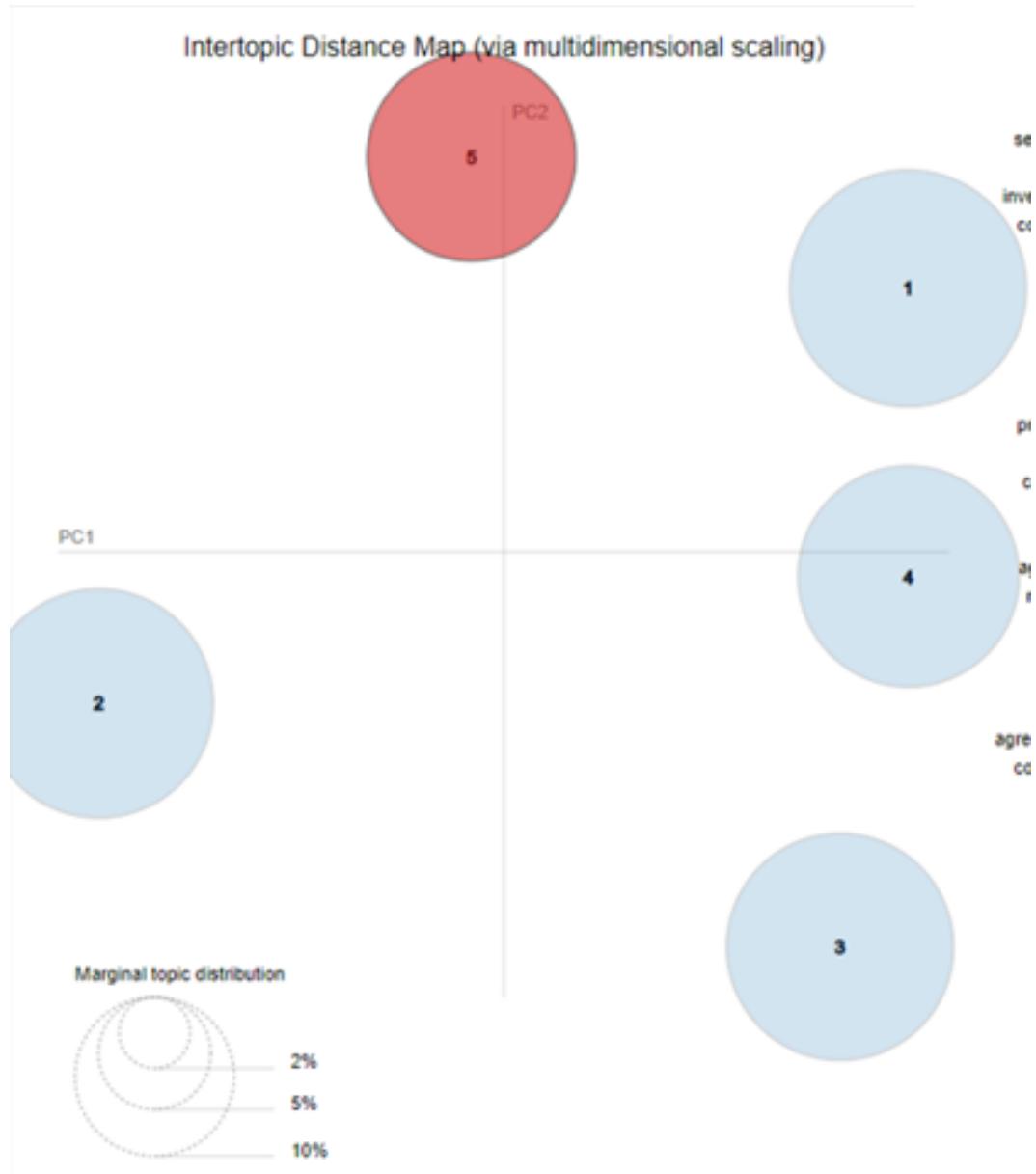
1 from __future__ import print_function
2 import pyLDAvis
3 import pyLDAvis.sklearn
4
5 zit=pyLDAvis.sklearn.prepare(lda,fin,vec)

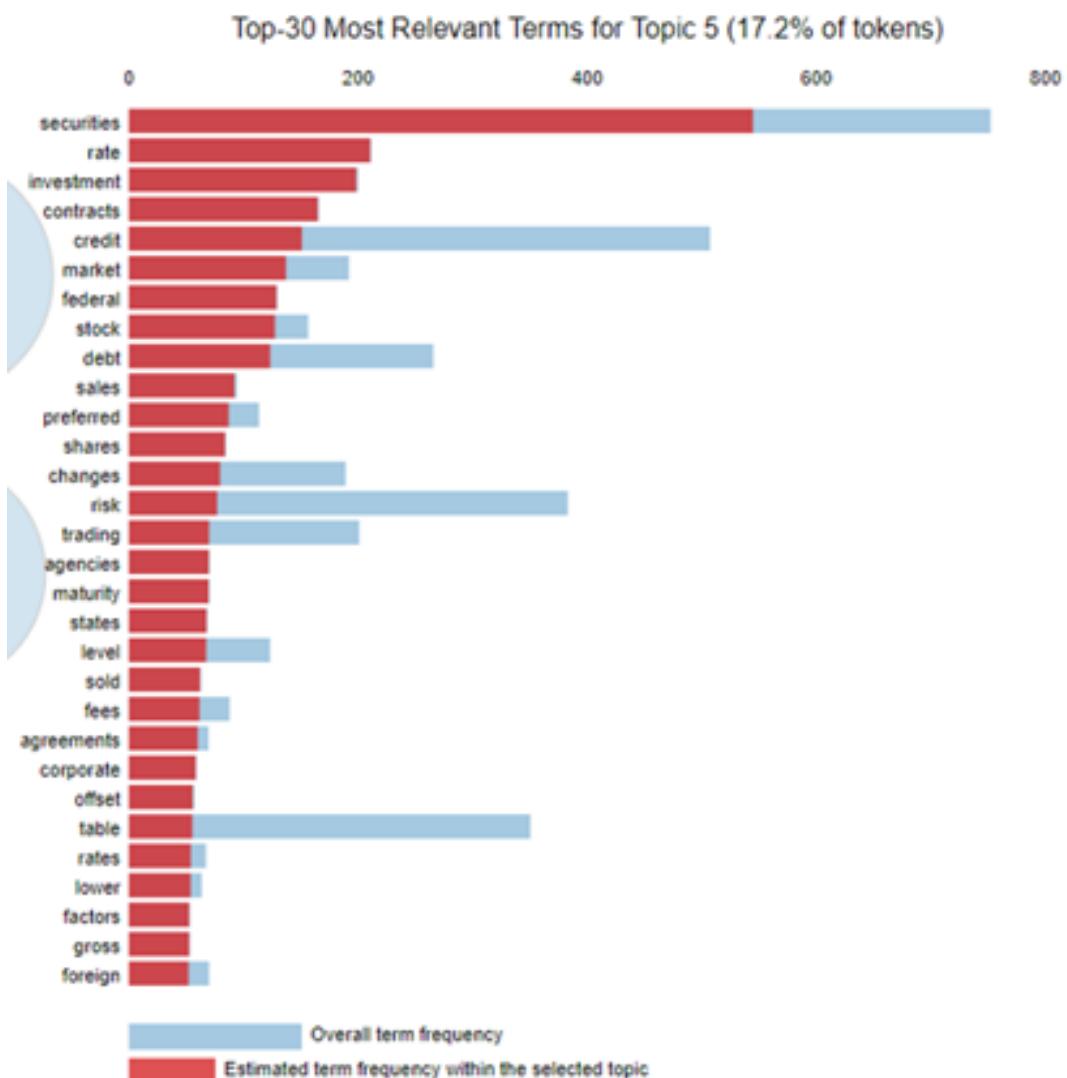
```

⁶ pyLDAvis.show(zit)

⁷

⁸ Output



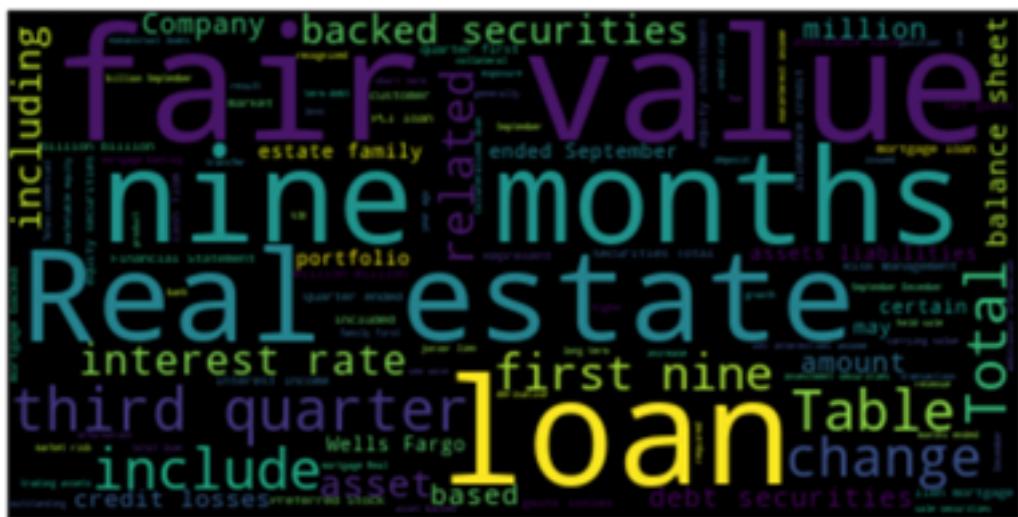


Notiamo che gli argomenti 2 e 5 sono piuttosto distanti tra loro. Questo è ciò che abbiamo osservato nella sezione precedente dal tema generale e dall'elenco di parole sotto questi argomenti. Gli argomenti 1 e 4 sono abbastanza vicini, il che convalida la nostra osservazione di sopra. Argomenti così vicini dovrebbero essere analizzati in modo più complesso e potrebbero essere combinati se necessario. La pertinenza dei termini sotto ciascun argomento, come mostrato nel secondo pannello del grafico (seconda immagine), può essere utilizzata anche per comprendere le differenze. Anche gli argomenti 3 e 4 sono relativamente vicini, sebbene l'argomento 3 sia piuttosto distante dagli altri.

Word Cloud

In questo passaggio, viene generato un word cloud per l'intero documento per annotare i termini più ricorrenti nel documento:

```
1 #Loading the additional packages for word cloud
2 from os import path
3 from PIL import Image
4 import numpy as np
5 import matplotlib.pyplot as plt
6 from wordcloud import WordCloud,STOPWORDS
7
8 #Loading the document and generating the word cloud
9 d = path.dirname(__name__)
10 text = open(path.join(d, 'Finance10k.txt')).read()
11
12 stopwords = set(STOPWORDS)
13 wc = WordCloud(background_color="black", max_words=2000, stopwords=stopwords)
14 wc.generate(text)
15
16 plt.figure(figsize=(16,13))
17 plt.imshow(wc, interpolation='bilinear')
18 plt.axis("off")
19 plt.show()
20
21 Output
```



La nuvola di parole generalmente concorda con i risultati della modellazione dell'argomento, poiché parole ricorrenti, come *loan*, *real estate*, *third quarter* e *fair value*, sono più grandi.

Integrando le informazioni dei passaggi precedenti, potremmo arrivare all'elenco degli argomenti rappresentati dal documento. Per il documento nel nostro caso di studio, vediamo che parole come *third quarter*, *first nine*, *nine* e *months* ricorrono abbastanza frequentemente. Nell'elenco delle parole ci sono diversi argomenti relativi alle voci di bilancio. Quindi il documento potrebbe essere un bilancio finanziario del terzo trimestre con tutti i valori di credito e attività in quel trimestre.

Conclusioni del caso di studio

In questo caso di studio, abbiamo esplorato l'uso della modellazione degli argomenti per ottenere informazioni sul contenuto di un documento. Abbiamo dimostrato l'uso del modello LDA, che estrae argomenti plausibili e ci consente di ottenere una comprensione di alto livello di grandi quantità di testo in modo automatizzato.

Abbiamo eseguito l'estrazione del testo da un documento in formato PDF ed eseguito un'ulteriore preelaborazione dei dati. I risultati, insieme alle visualizzazioni, hanno dimostrato che gli argomenti sono intuitivi e significativi.

Nel complesso, il caso di studio mostra come l'apprendimento automatico e l'NLP possono essere applicati in molti domini, come l'analisi degli investimenti, per riassumere documenti, notizie e report al fine di ridurre significativamente l'elaborazione manuale. Data questa capacità di accedere rapidamente e verificare le informazioni rilevanti e filtrate, gli analisti possono essere in grado di fornire report più completi e informativi su cui la direzione può basare le proprie decisioni.

8.6 Conclusione

Il campo dell'NLP ha compiuto progressi significativi, con il risultato di tecnologie che hanno e continueranno a rivoluzionare il modo in cui operano le istituzioni finanziarie. Nel breve termine, è probabile che assisteremo a un aumento delle tecnologie basate sull'NLP in diversi domini della finanza, tra cui la gestione patrimoniale, la gestione del rischio e l'automazione dei processi. L'adozione e la comprensione delle metodologie di NLP e delle relative infrastrutture sono e saranno ancora di più molto importanti per le istituzioni finanziarie.

Capitolo 9

CNNs per serie temporali finanziarie e immagini satellitari

”Dove c’è il fumo dei dati, c’è il fuoco degli affari.”

—Thomas Redmann

Le reti neurali convoluzionali (CNNs) consentono prestazioni sovraumane in varie attività di visione artificiale come la classificazione di immagini e il riconoscimento di oggetti nelle immagini. Le CNNs possono anche estrarre segnali dai dati di serie temporali che condividono determinate caratteristiche con i dati delle immagini, sono state applicate con successo anche nel riconoscimento vocale (Abdel-Hamid et al. 2014). Le CNN prendono il nome da un’operazione di algebra lineare chiamata convoluzione che sostituisce la moltiplicazione matriciale tipica delle reti feedforward in almeno uno dei loro strati.

In questo capitolo mostreremo come funzionano le convoluzioni e perché sono particolarmente adatte per i dati con una certa struttura regolare, tipicamente immagini, ma anche nelle serie temporali.

La ricerca sulle architetture CNN è proseguita molto rapidamente e continuano a emergere nuove architetture che migliorano le prestazioni dei benchmark. Descriveremo una serie di elementi costitutivi costantemente utilizzati da applicazioni di successo. Dimostreremo inoltre come il transfer learning può accelerare l’apprendimento utilizzando i pesi preaddestrati per gli strati della CNN più vicini all’ingresso, mentre si mettono a punto gli strati finali per un compito specifico. Le CNN possono aiutare a costruire una strategia di trading generando segnali da immagini o da serie temporali:

- **I dati satellitari** possono segnalare le tendenze future delle materie prime, tra cui l’offerta di determinate materie prime attraverso imma-

gini di aree agricole, miniere o reti di trasporto come le petroliere. I filmati delle telecamere di sorveglianza, per esempio, dei centri commerciali potrebbero essere utilizzati per tracciare e prevedere l'attività dei consumatori.

- **I dati delle serie temporali** comprendono una gamma molto ampia di fonti di dati e le CNN hanno dimostrato di poter fornire risultati di classificazione di alta qualità sfruttando la loro somiglianza strutturale con le immagini.

Creeremo una strategia di trading basata sulle previsioni di una CNN che utilizza dati di serie temporale che sono stati deliberatamente formattati come immagini e dimostreremo come costruire una CNN per classificare le immagini satellitari.

9.1 Come le CNN imparano dai dati di tipo reticolare

Le CNN sono concettualmente simili alle reti neurali feedforward (NN): sono costituite da unità con parametri chiamati pesi e polarizzazioni, e il processo di addestramento regola questi parametri per ottimizzare l'output della rete per un dato input in base a una funzione di perdita (loss function). Sono comunemente utilizzate per la classificazione. Ogni unità utilizza i suoi parametri per applicare un'operazione lineare ai dati di ingresso o alle attivazioni ricevute da altre unità, tipicamente seguito da una trasformazione non lineare. La rete nel suo complesso modella una funzione differenziabile che mappa i dati grezzi, come i pixel di un'immagine, alle probabilità di classe utilizzando una funzione di attivazione in uscita come softmax. Le NN feedforward con strati completamente connessi non si adattano bene ai dati di immagini ad alta dimensione con un gran numero di valori di pixel. Anche le immagini a bassa risoluzione incluse nel dataset CIFAR-10, che utilizzeremo nella prossima sezione, contengono 32×32 pixel con fino a 256 valori di colore diversi rappresentati da 8 bit ciascuno. Con tre canali, ad esempio, per i canali canali rosso, verde e blu del modello di colore RGB, una singola unità in uno strato di ingresso completamente connesso di ingresso implica $32 \times 32 \times 3 = 3.072$ pesi. Una risoluzione più standard di 640×480 pixel produce già quasi 1 milione di pesi per una singola unità di input. Le architetture profonde con diversi strati di ampiezza significativa portano rapidamente a un numero esplosivo di parametri che rendono quasi certo l'overfitting durante l'addestramento. Una NN feedforward completamente connessa non fa ipotesi sulla struttura locale

dei dati in ingresso, per cui il riordino arbitrario delle caratteristiche non ha alcun impatto sul risultato dell’addestramento.

Al contrario, le CNN, invece, assumono che i dati abbiano una topologia reticolare e che la struttura locale sia importante. In altre parole, codificano il presupposto che l’input abbia una struttura tipicamente riscontrabile nei dati delle immagini: i pixel formano una griglia bidimensionale, possibilmente con diversi canali per rappresentare le componenti del segnale a colori. Inoltre, i valori dei pixel vicini sono probabilmente più rilevanti per rilevare caratteristiche chiave come i bordi e gli angoli rispetto a punti di dati lontani. Naturalmente, le applicazioni iniziali delle CNN, come il riconoscimento della scrittura si sono concentrate solo sui dati delle immagini.

Col tempo, tuttavia, i ricercatori hanno riconosciuto caratteristiche simili nei dati delle serie temporali, ampliando l’ambito di utilizzo produttivo delle CNN. I dati delle serie temporali sono costituiti da misure a intervalli regolari che creano una griglia unidimensionale lungo l’asse temporale, come i rendimenti ritardati di un dato titolo. Può essere presente anche una seconda dimensione con caratteristiche aggiuntive per questo titolo e gli stessi periodi di tempo. Infine, potremmo rappresentare altri titoli utilizzando la terza dimensione.

Le CNN giocano un ruolo chiave anche in AlphaGo, il primo algoritmo a vincere una partita di Go contro gli esseri umani, in cui hanno valutato diverse posizioni sulla scacchiera a griglia.

L’elemento più importante per codificare **l’assunzione di una topologia a griglia** è **l’operazione di convoluzione** che dà il nome alle CNN, combinata con il **pooling**. Vedremo che le assunzioni specifiche sulla relazione funzionale tra i dati in ingresso e in uscita implicano che le CNN necessitano di un numero molto inferiore di parametri e calcolano in modo più efficiente.

In questa sezione spiegheremo come i livelli di convoluzione e di pooling apprendono i filtri che estraggono le caratteristiche locali e perché queste operazioni sono particolarmente adatte a dati con una struttura appena descritta. Le CNN di ultima generazione combinano molti di questi elementi di base per ottenere l’apprendimento della rappresentazione a strati. Concludiamo descrivendo le principali innovazioni architettoniche dell’ultimo decennio che hanno permesso di ottenere enormi miglioramenti nelle prestazioni.

9.1.1 Dalla codifica manuale all’apprendimento dei filtri dai dati

Per i dati delle immagini, questa struttura locale ha tradizionalmente motivato lo sviluppo di filtri codificati a mano che estraggono tali modelli per utilizzarli come caratteristiche nei modelli di apprendimento automatico (ML).

La Figura 9.1 mostra l'effetto di semplici filtri progettati per rilevare determinati bordi. Il file Jupyter Notebook filtro_esempio.ipynb, messo in allegato, illustra l'uso di filtri codificati a mano in una rete convolutiva e visualizza la trasformazione dell'immagine risultante. I filtri sono semplici $[-1, 1]$ disposti in una matrice 2×2 . Sotto ogni filtro, sono mostrati i suoi effetti.

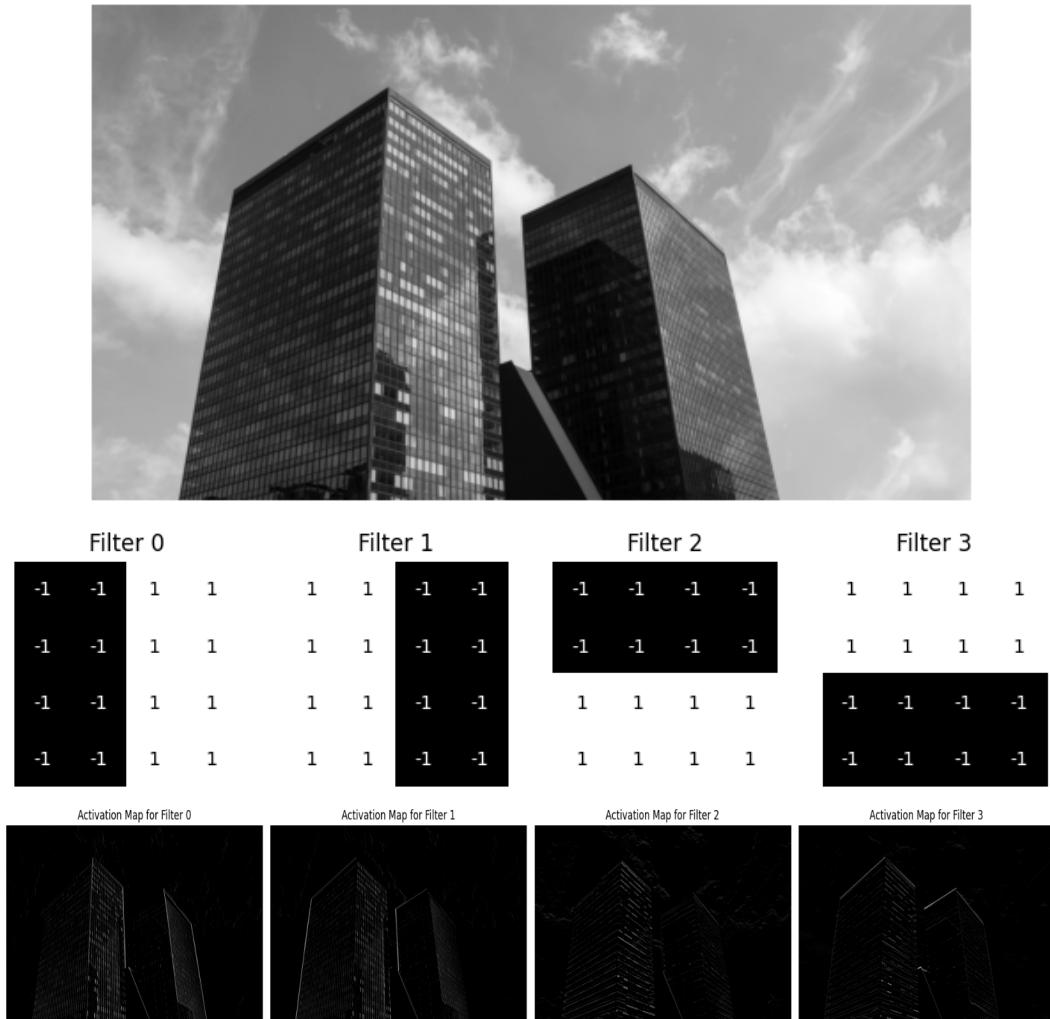


Figura 9.1: Risultato dei bordi in base ai filtri applicati all'immagine

Gli strati convolutivi, invece, sono progettati per apprendere tali rappresentazioni di caratteristiche locali dai dati. Un'intuizione chiave è quella di limitare il loro input, chiamato campo ricettivo, a una piccola area dell'input, in modo da catturare le costellazioni di pixel di base che riflettono modelli comuni come i bordi o gli angoli.

Tuttavia, tali schemi possono essere presenti ovunque in un'immagine, quindi le CNN devono anche riconoscere schemi simili in posizioni diverse e possibilmente con piccole variazioni.

I livelli successivi imparano a sintetizzare queste caratteristiche locali per rilevare caratteristiche di ordine superiore.

9.1.2 Come operano gli elementi di uno strato convoluto

Gli strati convolutivi integrano tre idee architettoniche che permettono l'apprendimento di rappresentazioni di caratteristiche che sono in qualche modo invarianti a spostamenti, cambiamenti di scala e distorsioni:

- Connettività sparsa piuttosto che densa
- Condivisione dei pesi
- Declassamento spaziale o temporale

Inoltre, gli strati convolutivi consentono ingressi di dimensioni variabili. Passeremo in rassegna un tipico strato convoluto e descriveremo ciascuna di queste idee.

La Figura 9.2 illustra l'insieme delle operazioni che avvengono tipicamente in uno strato convoluto tridimensionale, assumendo che i dati dell'immagine vengano immessi con le tre dimensioni di altezza, larghezza e profondità, ovvero il numero di canali. L'intervallo dei valori dei pixel dipende dalla rappresentazione in bit, ad esempio [0, 255] per 8 bit. In alternativa, l'asse della larghezza potrebbe rappresentare il tempo, l'altezza diverse caratteristiche e i canali potrebbero catturare osservazioni su oggetti distinti, come ad esempio i ticker.

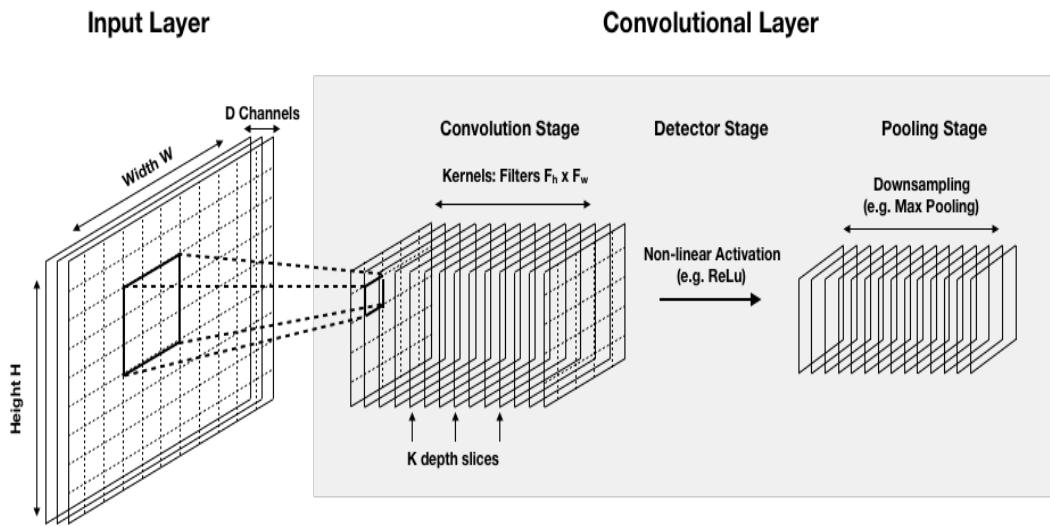


Figura 9.2: Operazioni tipiche in uno strato convolutivo bidimensionale

Nell'esempio illustrato nella Figura 9.2, lo strato convolutivo riceve un input tridimensionale e produce un output della stessa dimensionalità. Le CNN di ultima generazione sono composte da diversi strati di dimensioni variabili che vengono impilati l'uno sull'altro o che operano in parallelo su rami diversi. Con ogni strato, la rete è in grado di rilevare caratteristiche più astratte e di livello superiore.

Lo stadio convolutivo - estrazione delle caratteristiche

Il primo stadio applica un filtro, detto anche kernel, a toppe sovrapposte dell'immagine di ingresso. Il filtro è una matrice di dimensioni molto più piccole rispetto all'input, in modo che il suo campo ricettivo sia limitato a pochi valori contigui, come pixel o serie temporali. Di conseguenza, si concentra sugli schemi locali e riduce drasticamente il numero di parametri e di calcoli rispetto a uno strato completamente connesso. Uno strato convolutivo completo ha diverse mappe di caratteristiche organizzate come fette di profondità (figura 9.2), in modo che ogni livello possa estrarre più caratteristiche. Durante la scansione dell'input, il kernel (filtro) viene convoluto con ogni segmento di input coperto dal suo campo recettivo. L'operazione di convoluzione è semplicemente il prodotto scalare tra i pesi del filtro e i valori dell'area dell'input corrispondente dopo che entrambi sono stati rimodellati in vettori. Ogni convoluzione produce quindi un singolo numero e l'intera scansione produce una mappa di caratteristiche. Poiché il prodotto scalare è massimizzato per vettori identici, la mappa delle caratteristiche indica il grado di attivazione di ciascuna

regione di input.

La Figura 9.3 illustra il risultato della scansione di un input 5×5 utilizzando un filtro 3×3 con i valori dati, e come l'attivazione nell'angolo in alto a sinistra della mappa delle caratteristiche risulti dal prodotto scalare della regione di input appiattita e del kernel:

Input Data	Filter Matrix (Kernel)	Feature Map	
$\begin{array}{ c c c } \hline 1 & 1 & 1 \\ \hline 0 & 1 & 1 \\ \hline 0 & 0 & 1 \\ \hline 0 & 0 & 1 \\ \hline 0 & 1 & 1 \\ \hline \end{array}$	$\begin{array}{ c c c } \hline 0 & 0 \\ \hline 1 & 0 \\ \hline 1 & 1 \\ \hline 1 & 0 \\ \hline 0 & 0 \\ \hline \end{array} * \begin{array}{ c c c } \hline 1 & 0 & 1 \\ \hline 0 & 1 & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array}$	$\begin{array}{ c c c } \hline 4 & 3 & 4 \\ \hline 2 & 4 & 3 \\ \hline 2 & 3 & 4 \\ \hline \end{array} =$	$\begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} = 4$

Figura 9.3: Dalle convoluzioni alla mappa delle caratteristiche

L'aspetto più importante è che i valori dei filtri sono i parametri degli strati convolutivi, appresi dai dati durante l'addestramento per minimizzare la loss function scelta.

Come scansionare l'input - strides e padding

Lo stride definisce la dimensione del passo utilizzato per la scansione dell'input, ovvero il numero di pixel da spostare orizzontalmente e verticalmente. Gli strides più piccoli scansionano un maggior numero di aree (sovraposte), ma sono più costosi dal punto di vista computazionale. Lo zero padding è una tecnica che consiste nell'aggiungere all'immagine un "bordo" di zeri (vedi figura 9.4), allo scopo di preservare la dimensione dell'immagine in output dal layer, per non perdere informazioni.

Le quattro opzioni che sono comunemente utilizzate quando il filtro non si adatta perfettamente all'input e attraversa parzialmente il confine dell'immagine durante la scansione sono:

- Convoluzione valida: Scarta le scansioni in cui l'immagine e il filtro non combaciano perfettamente.
- Stessa convoluzione: Azzera l'input per produrre una mappa di caratteristiche di uguali dimensioni.

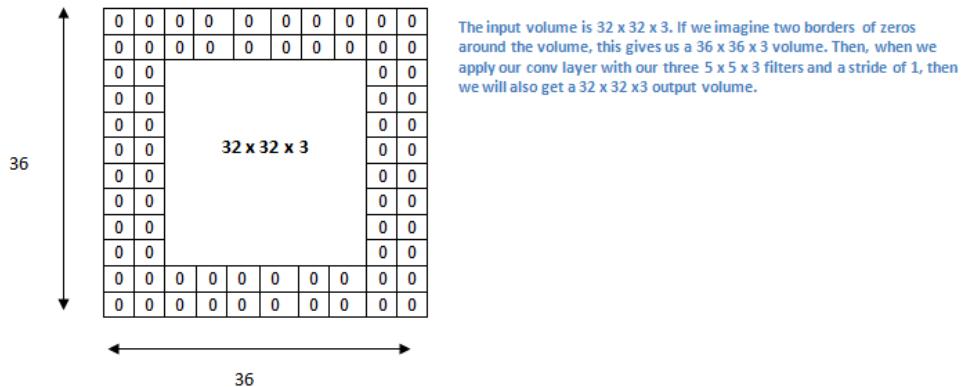


Figura 9.4: Padding

- Convoluzione completa: Azzera l'input in modo che ogni pixel venga scansionato un numero uguale di volte, compresi i pixel sul bordo (per evitare il sovraccampionamento dei pixel più vicini al centro).
- Causal: Azzeramento dell'input solo a sinistra, in modo che l'output non dipenda da un input di un periodo successivo; mantiene l'ordine temporale per i dati delle serie temporali.

Le scelte dipendono dalla natura dei dati e da dove è più probabile che si trovino le caratteristiche utili. In combinazione con il numero di fette di profondità, determinano la dimensione dell'output dello stadio di convoluzione.

Lo stadio del rivelatore - aggiunta di non linearità

Le mappe delle caratteristiche vengono solitamente sottoposte a una trasformazione non lineare. L'unità lineare rettificata (ReLU) è una funzione comune per questo scopo. Le ReLU sostituiscono le attivazioni negative in modo elementare con lo zero e mitigano il rischio di gradienti che svaniscono in altre funzioni di attivazione come la tanh. Un'alternativa popolare è la funzione softplus:

$$f(x) = \ln(1 + e^x)$$

A differenza della ReLU, questa funzione ha una derivata ovunque, ovvero la funzione sigmoide che si utilizza per la regressione logistica.

Lo stadio di pooling: sottocampionamento delle mappe di caratteristiche

L'ultimo stadio dello strato convolutivo può ricampionare la rappresentazione in input della mappa di caratteristiche per:

- ridurre la sua dimensionalità e prevenire l'overfitting
- ridurre il costo computazionale
- consentire l'invarianza di base della traduzione

Questo presuppone che la posizione precisa delle caratteristiche non solo è meno importante per l'identificazione di un modello, ma può addirittura essere dannosa, perché probabilmente varierà per diverse istanze dell'obiettivo. Il pooling riduce la risoluzione spaziale della mappa delle caratteristiche come un modo semplice per rendere le informazioni sulla posizione meno precise. Tuttavia, questo passaggio è facoltativo e molte architetture utilizzano il pooling solo per alcuni livelli o non lo utilizzano affatto. Un'operazione di pooling comune è il max pooling, che utilizza solo il valore massimo di attivazione da sottoregioni (tipicamente) non sovrapposte. Per una piccola mappa di caratteristiche 4×4 , ad esempio, il max pooling 2×2 produce il massimo per ciascuna delle quattro aree 2×2 non sovrapposte. Gli operatori di pooling meno comuni utilizzano la media o la mediana. Il pooling non aggiunge o apprende nuovi parametri, ma la dimensione della finestra di input e possibilmente lo stride sono iperparametri aggiuntivi.

9.1.3 L'evoluzione delle architetture CNN - innovazioni chiave

Negli ultimi due decenni diverse architetture CNN hanno spinto i limiti delle prestazioni introducendo importanti innovazioni. La crescita delle prestazioni predittive si è accelerata con l'arrivo dei big data sotto forma di ImageNet (Fei-Fei 2015), con 14 milioni di immagini assegnate a 20.000 classi da esseri umani tramite Amazon Mechanical Turk. Il ImageNet Large Scale Visual Recognition Challenge (ILSVRC) è diventato il punto focale dei progressi delle CNN. I progressi della CNN si sono concentrati su un insieme leggermente più piccolo di 1,2 milioni di immagini appartenenti a 1.000 classi. È utile avere familiarità con le architetture di riferimento che dominano queste competizioni per ragioni pratiche. Come vedremo nella prossima sezione sul lavoro con le CNN per i dati di immagine, esse offrono un buon punto di partenza per compiti standard. Inoltre, l'apprendimento per trasferimento ci permette di

affrontare molti compiti di visione computerizzata basandosi su un’architettura di successo con pesi preaddestrati. L’apprendimento per trasferimento non solo velocizza la selezione dell’architettura e l’addestramento ma consente anche applicazioni di successo su insiemi di dati molto più piccoli. Inoltre, molte pubblicazioni fanno riferimento a queste architetture, che spesso servono come base per le reti adattate a compiti di segmentazione o localizzazione.

9.2 CNNs per immagini satellitari

In questa sezione, dimostriamo come risolvere uno dei compiti chiave di computer vision ovvero la classificazione delle immagini. Come accennato nel capitolo 3, nella sezione dei Dati alternativi, i dati di immagine possono informare una strategia di trading fornendo indizi su tendenze future, cambiamenti dei fondamentali o eventi specifici relativi a una classe di attività o a un universo di investimento. Esempi popolari sono lo sfruttamento delle immagini satellitari per ottenere indizi sull’offerta di materie prime agricole, sull’attività economica e dei consumatori o sullo stato delle catene di approvvigionamento di materie prime o di produzione. I compiti specifici possono includere, ad esempio, i seguenti:

- Classificazione delle immagini: Identificare i terreni coltivati per determinate colture sono in espansione o prevedere la qualità e la quantità del raccolto.
- Rilevamento di oggetti: Contare il numero di petroliere su un determinato percorso di trasporto o il numero di auto in un parcheggio, oppure identificare il numero di autovetture in un parcheggio o identificare la posizione degli acquirenti in un centro commerciale.

In questa sezione, dimostreremo come progettare le CNN per automatizzare l’estrazione di tali informazioni, sia partendo da zero utilizzando le architetture più diffuse, sia attraverso l’apprendimento per trasferimento. Introdurremo le principali architetture CNN per questi compiti, spiegheremo perché funzionano bene, e mostreremo come addestrarle utilizzando TensorFlow 2. Sfortunatamente, le immagini satellitari con informazioni direttamente rilevanti per una strategia di trading sono molto costose da ottenere e non sono prontamente disponibili. Dimostreremo tuttavia come lavorare con il set di dati EuroSat per costruire un classificatore che identifica i diversi usi del suolo. Questa breve introduzione alle CNN per la computer vision ha lo scopo di dimostrare come affrontare i compiti più comuni che probabilmente bisogna affrontare quando si vuole progettare una strategia di trading basata su immagini rilevanti per

l'universo di investimento di nostra scelta. Per prima cosa illustreremo l'architettura LeNet5 utilizzando il dataset MNIST di cifre scritte a mano. Successivamente, dimostreremo l'uso dell'aumento dei dati con AlexNet su CIFAR-10, una versione semplificata dell'ImageNet originale. Continueremo poi con l'apprendimento per trasferimento basato su architetture all'avanguardia prima di applicare quanto appreso alle immagini satellitari reali.

9.2.1 LeNet5 - La prima CNN con applicazioni industriali

Yann LeCun, oggi direttore della ricerca sull'intelligenza artificiale di Facebook, è stato uno dei principali pionieri nello sviluppo delle CNN. Nel 1998, dopo diverse iterazioni iniziate negli anni '80, LeNet5 è diventata la prima CNN moderna utilizzata in applicazioni reali ed introdusse elementi architettonici ancora oggi rilevanti. LeNet5 è stata pubblicata in un documento molto istruttivo, Gradient-Based Learning Applied to Document Recognition (LeCun et al. 1989), che ne esponeva molti dei concetti centrali. Soprattutto, promuoveva l'intuizione che le convoluzioni con filtri apprendibili sono efficaci per estrarre caratteristiche correlate in più punti con pochi parametri. Date le limitate risorse computazionali dell'epoca, l'efficienza era di fondamentale importanza. LeNet5 è stata progettata per riconoscere la scrittura a mano sugli assegni ed è stata utilizzata da diverse banche. Ha stabilito un nuovo punto di riferimento per l'accuratezza della classificazione, con un risultato del 99,2% sul dataset MNIST, un dataset di cifre scritte a mano. È costituito da tre strati convolutivi, ciascuno dei quali contiene una trasformazione tanh non lineare, un'operazione di pooling e uno strato di uscita completamente connesso. In tutti gli strati convolutivi, il numero di mappe di caratteristiche aumenta mentre le loro dimensioni diminuiscono. Ha un totale di 60.850 parametri addestrabili (Lecun et al. 1998).

"Hello World" per le CNN - classificazione delle cifre scritte a mano

Il dataset MNIST originale contiene 60.000 immagini in scala di grigi con risoluzione di 28×28 pixel, ciascuna contenente una singola cifra scritta a mano da 0 a 9. Una buona alternativa è il dataset Fashion, più impegnativo ma strutturalmente simile. Implementeremo una versione leggermente semplificata di LeNet5 per dimostrare come costruire una CNN utilizzando un'implementazione di TensorFlow. Vedere il Notebook `digit_classification_with_lenet5` per i dettagli.

Carichiamo il dataset MNIST:

```
from tensorflow.keras.datasets import mnist
(X_train, y_train), (X_test, y_test) = mnist.load_data()
X_train.shape, X_test.shape
((60000, 28, 28), (10000, 28, 28))
```

La Figura 9.5 mostra le prime dieci immagini del set di dati. Inoltre, mostra come i valori dei pixel per una singola immagine vanno da 0 a 255:

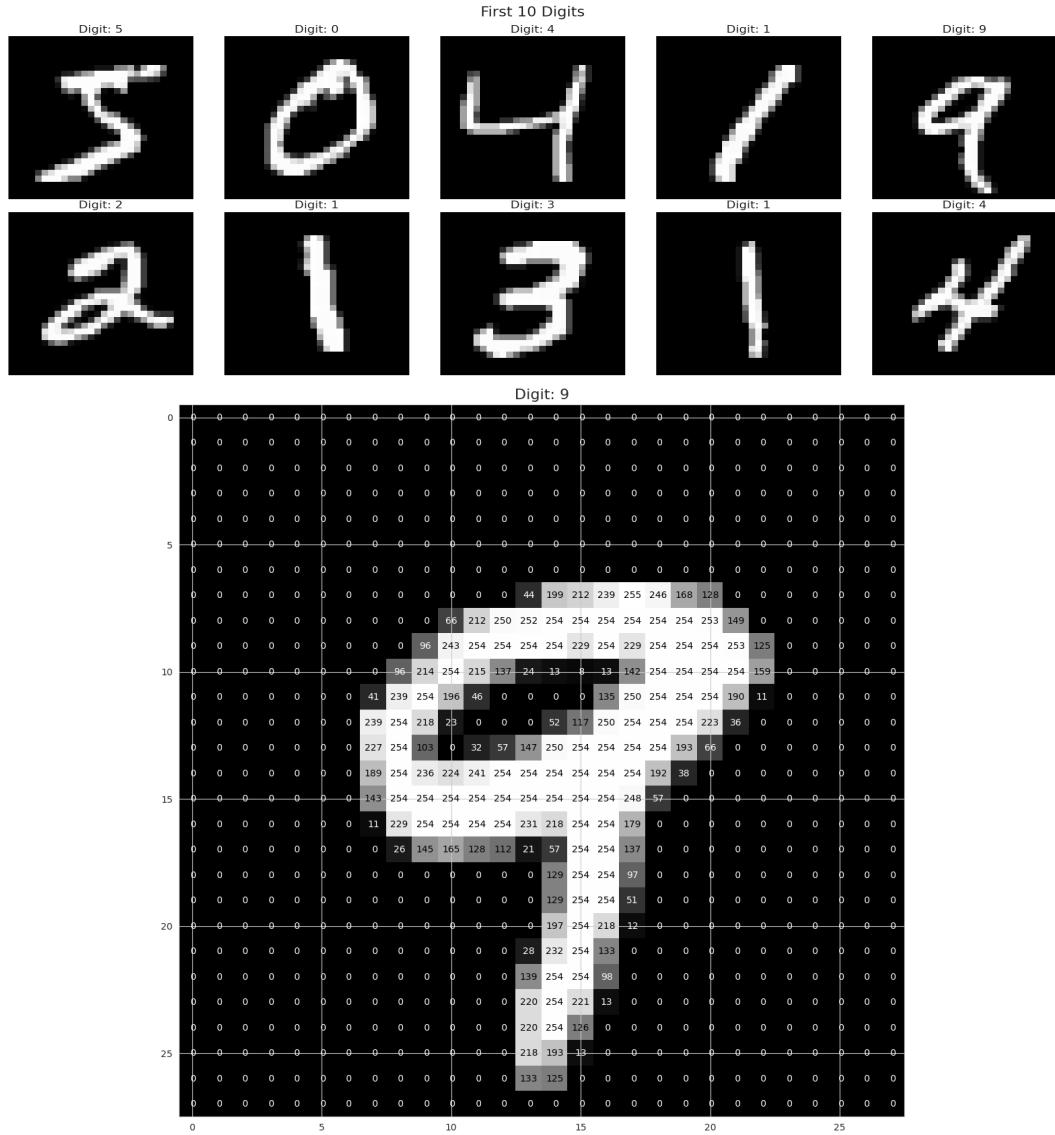


Figura 9.5: Esempi del dataset MNIST

Ridimensioniamo i valori dei pixel nell'intervallo [0, 1] per normalizzare i

dati di addestramento e facilitare il processo di backpropagation, e convertiamo i dati in float a 32 bit, che riducono i requisiti di memoria e il costo computazionale e forniscono una precisione sufficiente per il nostro caso d'uso:

```
X_train = X_train.astype('float32')/255
X_test = X_test.astype('float32')/255
```

Definizione dell'architettura di LeNet5

Possiamo definire una versione semplificata di LeNet5 che omette lo strato finale originale contenente funzioni a base radiale come segue, utilizzando il padding "valido" di default e gli strides a passo singolo, a meno che non sia stato definito diversamente:

```
lenet5 = Sequential([
    Conv2D(filters=6,
           kernel_size=5,
           activation='relu',
           input_shape=(28, 28, 1),
           name='CONV1'),
    AveragePooling2D(pool_size=(2, 2),
                     strides=(1, 1),
                     padding='valid',
                     name='POOL1'),
    Conv2D(filters=16, kernel_size=(5, 5), activation='tanh',
           name='CONV2'),
    AveragePooling2D(pool_size=(2, 2), strides=(2, 2), name='POOL2'),
    Conv2D(filters=120, kernel_size=(5, 5), activation='tanh',
           name='CONV3'),
    Flatten(name='FLAT'),
    Dense(units=84, activation='tanh', name='FC6'),
    Dense(units=10, activation='softmax', name='FC7')
])
```

La sintesi indica che il modello così definito ha oltre 300.000 parametri:

Layer (type)	Output Shape	Param #
CONV1 (Conv2D)	(None, 24, 24, 6)	156
POOL1 (AveragePooling2D)	(None, 23, 23, 6)	0

CONV2 (Conv2D)	(None, 19, 19, 16)	2416
POOL2 (AveragePooling2D)	(None, 9, 9, 16)	0
CONV3 (Conv2D)	(None, 5, 5, 120)	48120
FLAT (Flatten)	(None, 3000)	0
FC6 (Dense)	(None, 84)	252084
FC7 (Dense)	(None, 10)	850

Total params: 303,626
 Trainable params: 303,626
 Non-trainable params: 0

Addestramento e valutazione del modello

Ora siamo pronti ad addestrare il modello. Il modello si aspetta un input quadridimensionale, quindi lo rimodelliamo di conseguenza. Utilizziamo la dimensione standard di 32 batch e una divisione 80:20 tra addestramento e convalida. Inoltre, sfruttiamo il checkpoint per memorizzare i pesi del modello se l'errore di convalida migliora e ci assicuriamo che il set di dati sia rimescolato in modo casuale. Definiamo anche una callback early_stopping per interrompere l'addestramento una volta che l'accuratezza della validazione non migliora più per 20 iterazioni:

```
lenet_history = lenet5.fit(
    X_train.reshape(-1, 28, 28, 1),
    y_train,
    batch_size=batch_size,
    epochs=epochs,
    validation_split=0.2, # use 0 to train on all data
    callbacks=[checkpointer, early_stopping],
    verbose=1,
    shuffle=True)
```

La cronologia dell'allenamento registra l'ultimo miglioramento dopo 81 epoch che richiedono circa 4 minuti su una singola GPU. L'accuratezza del test di questa esecuzione campione è del 99,09%, quasi esattamente lo stesso risultato di LeNet5 originale:

```
lenet_accuracy = lenet5.evaluate(X_test.reshape(-1, 28, 28, 1),  
y_test, verbose=0)[1]  
  
print('Test accuracy: {:.2%}'.format(lenet_accuracy))  
Test accuracy: 99.04%
```

A titolo di confronto, una semplice rete feedforward a due strati raggiunge "solo" il 97,97% di accuratezza del test (si veda il file Jupyter Notebook). Il miglioramento di LeNet5 su MNIST è in effetti modesto. Anche i metodi non neurali hanno ottenuto un'accuratezza di classificazione maggiore o uguale al 99%, tra cui K-nearest neighbors e support vector machines. Le CNN brillano davvero con i dataset più impegnativi, come vedremo in seguito.

9.2.2 AlexNet - riaccendere la ricerca sull'apprendimento profondo

AlexNet, sviluppato da Alex Krizhevsky, Ilya Sutskever e Geoff Hinton presso l'Università di Toronto, ha ridotto drasticamente il tasso di errore e ha superato in modo significativo il secondo classificato dell'ILSVRC 2012, ottenendo un errore nella top-5 del 16% contro il 26% (Krizhevsky, Sutskever e Hinton 2012). Questa scoperta ha dato il via a una rinascita della ricerca sull'ML e ha portato l'apprendimento profondo per la visione computerizzata sulla mappa tecnologica globale.

L'architettura di AlexNet è simile a LeNet, ma molto più profonda e ampia. È spesso accreditata per aver scoperto l'importanza della profondità con circa 60 milioni di parametri, superando LeNet5 di un fattore pari a 1.000, a testimonianza dell'aumento della potenza di calcolo, in particolare dell'uso delle GPU, e di dataset molto più grandi.

Ha incluso convoluzioni sovrapposte piuttosto che combinare ogni convoluzione con uno stadio di pooling e ha utilizzato con successo il dropout per la regolarizzazione e ReLU per trasformazioni non lineari efficienti. Inoltre, ha impiegato l'incremento dei dati per aumentare il numero di campioni di addestramento, ha aggiunto il decadimento dei pesi e ha utilizzato un'implementazione più efficiente delle convoluzioni. Ha inoltre accelerato l'addestramento distribuendo la rete su due GPU.

Il Notebook `image_classification_with_alexnet.ipynb` contiene una versione leggermente semplificata di AlexNet adattato al set di dati CIFAR-10 che contiene 60.000 immagini di 10 delle 1.000 classi originali. È stato compresso a una risoluzione di 32×32 pixel dai 224×224 originali, ma ha ancora tre canali di colore. Per i dettagli sull'implementazione si veda il notebook

image_classification_with_alexnet; in questa sezione salteremo alcuni passaggi ripetitivi.

Preelaborazione dei dati CIFAR-10 con l'aumento dell'immagine

CIFAR-10 può essere scaricato anche tramite l'interfaccia Keras di TensorFlow e noi ridimensioniamo i valori dei pixel e codifichiamo le dieci etichette di classe come abbiamo fatto con MNIST nella sezione precedente.

Per prima cosa addestriamo una rete feedforward a due strati su 50.000 campioni di addestramento per 57 epochs per ottenere una test accuracy del 55,43%. Sperimentiamo anche una rete convoluzionale a tre strati con oltre 528.000 parametri che raggiunge un'accuratezza del 74,51% (vedi notebook).

Un trucco comune per migliorare le prestazioni è quello di aumentare artificialmente la dimensione del training set creando dati sintetici. Ciò comporta lo spostamento casuale o il capovolgimento orizzontale dell'immagine o l'introduzione di rumore nell'immagine. TensorFlow comprende una classe ImageDataGenerator per questo scopo. Possiamo configurarla e adattare i dati di addestramento come segue:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(
    width_shift_range=0.1, # randomly horizontal shift
    height_shift_range=0.1, # randomly vertical shift
    horizontal_flip=True) # randomly horizontal flip
datagen.fit(X_train)
```

Il risultato (Figura 9.6) mostra come le immagini aumentate (a bassa risoluzione 32×32) siano state alterate in vari modi, come previsto.

L'accuratezza del test per la CNN a tre strati migliora in modo modesto, raggiungendo il 76,41% dopo l'addestramento su dati più ampi e aumentati.

Definizione dell'architettura del modello

È necessario adattare l'architettura di AlexNet alla minore dimensionalità delle immagini di CIFAR-10 rispetto ai campioni di ImageNet. A tal fine, utilizziamo il numero originale numero di filtri, ma riducendoli (si veda il notebook).

Il riepilogo (si veda il notebook) mostra i cinque strati convoluzionali seguiti da due livelli completamente connessi con un uso frequente della normalizzazione batch, per un totale di 21,5 milioni di parametri.



Figura 9.6: Campioni originali e aumentati

Confronto delle prestazioni di AlexNet

Oltre ad AlexNet, abbiamo addestrato una NN feedforward a 2 strati e una CNN a 3 strati, quest'ultima con e senza incremento dell'immagine. Dopo 100 epoche (con interruzione anticipata se l'accuratezza di convalida non migliora per 20 round), otteniamo le traiettorie di convalida incrociata e l'accuratezza del test per i quattro modelli come mostrato nella Figura 9.7:

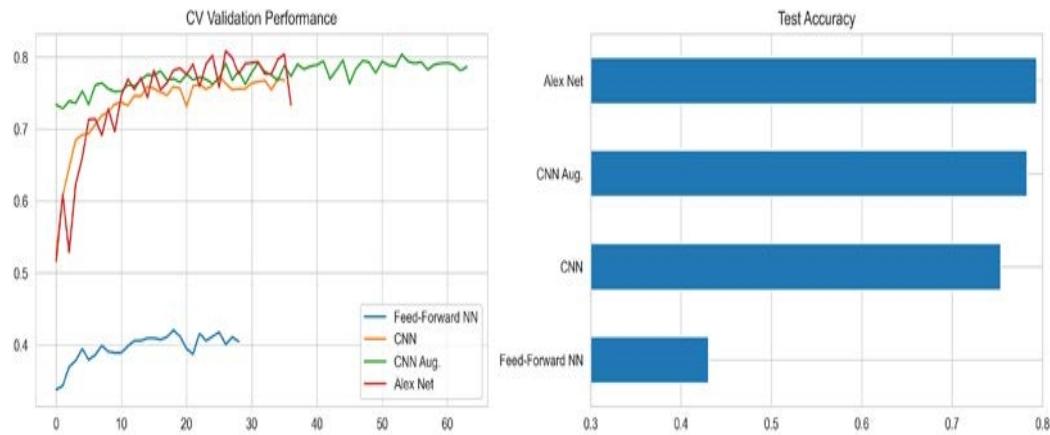


Figura 9.7: Prestazioni di convalida e test accuracy su CIFAR-10

AlexNet raggiunge la massima accuratezza nel test con il 79,33% dopo circa 35 epoche, seguita dalla CNN con immagini aumentate, con il 78,29%, che si allena più a lungo grazie al dataset più ampio. La NN feedforward ha ottenuto prestazioni molto peggiori rispetto a quelle di MNIST su questo dataset più complesso, con un'accuratezza del 43,05%.

9.2.3 Transfer Learning - Apprendimento più rapido con meno dati

In pratica, a volte non abbiamo abbastanza dati per addestrare una CNN da zero con un'inizializzazione casuale. L'apprendimento per trasferimento è una tecnica di ML che riutilizza un modello addestrato su un insieme di dati per un altro compito. Naturalmente, funziona se l'apprendimento del primo compito si trasferisce al compito di interesse. Se ha successo, può portare a prestazioni migliori ed ad un addestramento più rapido che richiede meno dati etichettati rispetto all'addestramento di una rete neurale da zero sull'attività di destinazione.

L'approccio alla CNN basato sull'apprendimento per trasferimento si basa su un preallenamento su un set di dati molto ampio, come ad esempio ImageNet. L'obiettivo è far sì che i filtri convoluzionali estraggano una rappresentazione delle caratteristiche che si generalizza a nuove immagini. In un secondo momento, si sfrutta il risultato per inizializzare e riqualificare (ri-addestrare) una nuova CNN oppure per utilizzarla come input per una nuova rete che affronti il compito di interesse.

Come abbiamo già discusso, le architetture CNN utilizzano tipicamente una sequenza di strati convoluzionali per rilevare schemi (patterns) gerarchici, aggiungendo uno o più strati completamente connessi per mappare le attivazioni convoluzionali alle classi o ai valori dei risultati. L'uscita dell'ultimo strato convoluzionale che alimenta la parte completamente connessa è chiamata caratteristica del collo di bottiglia. Possiamo utilizzare le caratteristiche del collo di bottiglia di una rete preaddestrata come input di una nuova rete completamente connessa, di solito dopo aver applicato una funzione di attivazione ReLU.

In altre parole, congeliamo gli strati convoluzionali e sostituiamo la parte densa della rete. Un ulteriore vantaggio è che possiamo utilizzare input di dimensioni diverse, perché sono gli strati densi a vincolare le dimensioni degli input.

In alternativa, è possibile utilizzare le caratteristiche del collo di bottiglia come input per un altro algoritmo di machine learning. Nell'architettura AlexNet, ad esempio, il livello di collo di bottiglia calcola un vettore con 4.096 voci per ogni immagine in ingresso 224 x 224. Utilizziamo questo vettore come caratteristiche per un nuovo modello.

Possiamo anche fare un ulteriore passo avanti e non solo sostituire e riqualificare gli strati finali utilizzando nuovi dati, ma anche perfezionare i pesi della CNN pre-addestrata. Per ottenere questo risultato, continuiamo l'addestramento, o solo per gli strati successivi, congelando i pesi di alcuni strati precedenti, o per tutti gli strati. La motivazione è presumibilmente quella di

preservare gli schemi (patterns) più generici appresi dai livelli inferiori, come i bordi o i colori, consentendo agli strati successivi della CNN di adattarsi ai dettagli di un nuovo compito. ImageNet, ad esempio, contiene un'ampia varietà di razze di cani, che può portare a rappresentazioni di caratteristiche specificamente utili per differenziare tra queste classi.

Basarsi su architetture all'avanguardia

L'apprendimento per trasferimento ci permette di sfruttare le architetture più performanti senza dover sostenere l'addestramento potenzialmente molto impegnativo per le GPU e i dati. Descriviamo brevemente le caratteristiche di alcune architetture molto diffuse, che rappresentano dei punti di partenza molto apprezzati.

VGGNet

Il secondo classificato dell'ILSVRC 2014 è stato sviluppato dal Visual Geometry Group (VGG) dell'Università di Oxford (VGG, Simonyan 2015). Ha dimostrato l'efficacia di filtri convoluzionali 3×3 molto più piccoli e combinati in sequenza, e ha rafforzato l'importanza della profondità per prestazioni elevate. VGG16 contiene 16 strati convoluzionali e completamente connessi che eseguono solo convoluzioni 3×3 ed il pooling 2×2 (vedi Figura 9.8).

VGG16 ha 140 milioni di parametri che aumentano i costi computazionali dell'addestramento e dei requisiti di memoria. Tuttavia, la maggior parte dei parametri si trova negli strati connessi, che si è scoperto non essere essenziali, per cui la loro rimozione riduce notevolmente il numero di parametri senza impattare negativamente sulle prestazioni.

GoogLeNet

Christian Szegedy di Google ha ridotto i costi di calcolo utilizzando implementazioni di CNN più efficienti per facilitare le applicazioni pratiche. La GoogLeNet risultante (Szegedy et al. 2015) ha vinto l'ILSVRC 2014 con soli 4 milioni di parametri grazie al modulo Inception, rispetto ai 60 milioni di AlexNet e ai 140 milioni di VGG16.

Il modulo Inception si basa sul concetto di network-in-network che utilizza le convoluzioni 1×1 per comprimere uno stack profondo di filtri convoluzionali e ridurre così i costi computazionali. Il modulo utilizza filtri paralleli 1×1 , 3×3 e 5×5 , combinando questi ultimi due con convoluzioni 1×1 per ridurre la dimensionalità dei filtri passati dal livello precedente.

Inoltre, utilizza il pooling medio invece di strati completamente connessi in cima agli strati convoluzionali per eliminare molti parametri di minore impatto.

Ci sono state diverse versioni migliorate, la più recente delle quali è Inception-v4.

ResNet

L'architettura della rete residua (ResNet) è stata sviluppata da Microsoft e ha vinto l'ILSVRC 2015. Ha ridotto l'errore della top-5 al 3,7%, al di sotto del livello delle prestazioni umane su questo compito, pari a circa il 5% (He et al. 2015).

Introduce connessioni di scorciatoia per l'identità che saltano diversi livelli e superano alcune sfide dell'addestramento di reti profonde, consentendo l'uso di centinaia o addirittura più di migliaia di strati. Utilizza inoltre in modo massiccio la normalizzazione dei lotti (batch), che si è dimostrata in grado di consentire tassi di apprendimento più elevati ed è più indulgente sull'inizializzazione dei pesi. L'architettura omette anche gli strati finali completamente connessi.

L'addestramento delle reti profonde si trova ad affrontare la famigerata sfida del gradiente che svanisce: quando il gradiente si propaga ai livelli precedenti, la moltiplicazione ripetuta di piccoli pesi rischia di ridurre il gradiente a zero. Quindi, l'aumento della profondità può limitare l'apprendimento.

La connessione a scorciatoia che salta due o più livelli è diventata uno degli sviluppi più popolari nelle architetture CNN e ha dato il via a numerosi sforzi di ricerca per perfezionare e spiegare le sue prestazioni.

9.2.4 Transfer Learning con VGG16

Le moderne CNN possono richiedere settimane di addestramento su più GPU su ImageNet, ma fortunatamente, molti ricercatori condividono i pesi finali. TensorFlow 2, ad esempio, contiene modelli preaddestrati per diverse architetture di riferimento discusse in precedenza, in particolare VGG16 e la sua versione più ampia, VGG19, ResNet50, InceptionV3 e InceptionResNetV2, oltre a MobileNet, DenseNet, NASNet e MobileNetV2.

Come estrarre le caratteristiche del collo di bottiglia

Il notebook bottleneck_features.ipynb illustra come scaricare il modello VGG16 preaddestrato, sia con gli strati finali per generare previsioni, sia senza gli strati finali, come illustrato nella Figura 9.8, per estrarre gli output prodotti dalle caratteristiche del collo di bottiglia:

TensorFlow 2 rende molto semplice il download e l'utilizzo di modelli preaddestrati:

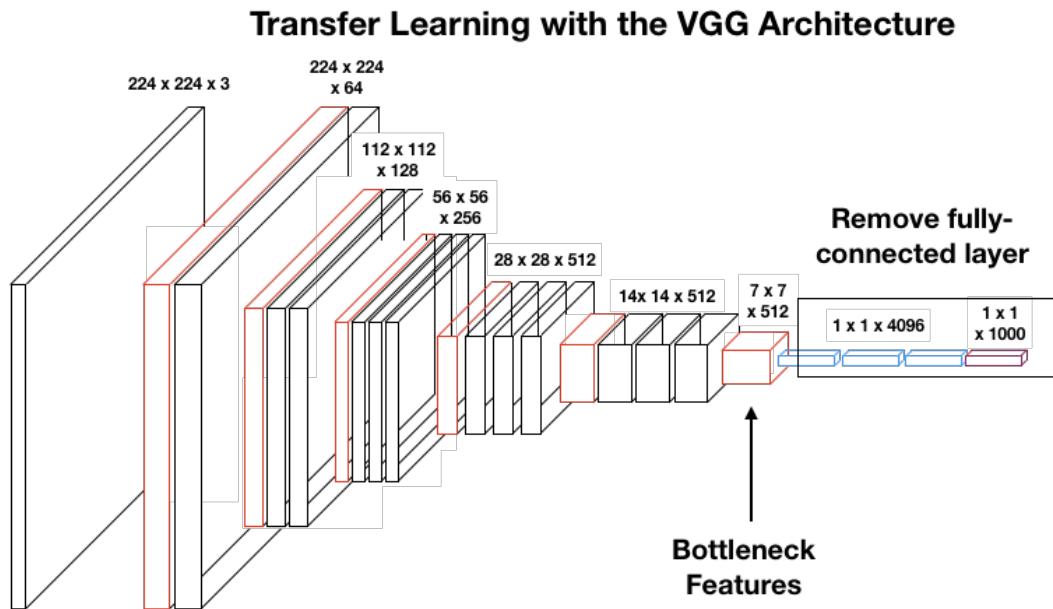


Figura 9.8: L'architettura VGG16

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168

```

24    block3_conv2 (Conv2D)           (None, 56, 56, 256)      590080
25
26    block3_conv3 (Conv2D)           (None, 56, 56, 256)      590080
27
28    block3_pool (MaxPooling2D)     (None, 28, 28, 256)      0
29
30    block4_conv1 (Conv2D)           (None, 28, 28, 512)     1180160
31
32    block4_conv2 (Conv2D)           (None, 28, 28, 512)     2359808
33
34    block4_conv3 (Conv2D)           (None, 28, 28, 512)     2359808
35
36    block4_pool (MaxPooling2D)     (None, 14, 14, 512)      0
37
38    block5_conv1 (Conv2D)           (None, 14, 14, 512)     2359808
39
40    block5_conv2 (Conv2D)           (None, 14, 14, 512)     2359808
41
42    block5_conv3 (Conv2D)           (None, 14, 14, 512)     2359808
43
44    block5_pool (MaxPooling2D)     (None, 7, 7, 512)       0
45
46    flatten (Flatten)             (None, 25088)            0
47
48    fc1 (Dense)                  (None, 4096)            102764544
49
50    fc2 (Dense)                  (None, 4096)            16781312
51
52    predictions (Dense)          (None, 1000)            4097000
53
54 =====
55 Total params: 138,357,544
56 Trainable params: 138,357,544
57 Non-trainable params: 0
58 -----

```

È possibile utilizzare questo modello per le predizioni come qualsiasi altro modello Keras: passiamo sette immagini campione e otteniamo le probabilità di classe per ciascuna delle 1.000 categorie di ImageNet:

```

1 y_pred = vgg16.predict(img_input)
2 y_pred.shape
3 (7, 1000)

```

Per escludere gli strati completamente connessi, basta aggiungere la parola chiave `include_top=False`. Le previsioni sono ora emesse dallo strato convoluzionale finale `block5_pool` e corrispondono alla forma di questo strato:

Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[None, None, None, 3]	0
block1_conv1 (Conv2D)	(None, None, None, 64)	1792
block1_conv2 (Conv2D)	(None, None, None, 64)	36928
block1_pool (MaxPooling2D)	(None, None, None, 64)	0
block2_conv1 (Conv2D)	(None, None, None, 128)	73856
block2_conv2 (Conv2D)	(None, None, None, 128)	147584
block2_pool (MaxPooling2D)	(None, None, None, 128)	0
block3_conv1 (Conv2D)	(None, None, None, 256)	295168
block3_conv2 (Conv2D)	(None, None, None, 256)	590080
block3_conv3 (Conv2D)	(None, None, None, 256)	590080
block3_pool (MaxPooling2D)	(None, None, None, 256)	0
block4_conv1 (Conv2D)	(None, None, None, 512)	1180160
block4_conv2 (Conv2D)	(None, None, None, 512)	2359808
block4_conv3 (Conv2D)	(None, None, None, 512)	2359808
block4_pool (MaxPooling2D)	(None, None, None, 512)	0
block5_conv1 (Conv2D)	(None, None, None, 512)	2359808
block5_conv2 (Conv2D)	(None, None, None, 512)	2359808

```

41  block5_conv3 (Conv2D)           (None, None, None, 512)    2359808
42
43  block5_pool (MaxPooling2D)     (None, None, None, 512)    0
44
45 -----
46 Total params: 14,714,688
47 Trainable params: 14,714,688
48 Non-trainable params: 0
49 -----
50
51 vgg16.predict(img_input).shape
52 (7, 7, 7, 512)

```

Omettendo gli strati completamente connessi e mantenendo solo i moduli convoluzionali, non siamo più costretti a utilizzare una dimensione di input fissa per il modello, come il formato originale 224×224 di ImageNet. Possiamo invece adattare il modello a dimensioni di input arbitrarie.

Come perfezionare un modello preaddestrato

Dimostreremo come congelare alcuni o tutti gli strati di un modello preaddestrato e come continuare l'addestramento utilizzando un nuovo insieme di strati completamente connessi e dati con un formato diverso (si veda il notebook transfer_learning.ipynb per esempi di codice, adattati da un tutorial di TensorFlow 2).

Utilizziamo i pesi di VGG16, preaddestrati su ImageNet con le immagini di gatti e cani incorporate in TensorFlow (si veda il notebook su come ottenere il dataset). La pre-elaborazione ridimensiona tutte le immagini a 160×160 pixel. Indichiamo la nuova dimensione dell'input mentre istanziamo l'istanza VGG16 preaddestrata e poi congeliamo tutti i pesi:

```

1 vgg16 = VGG16(input_shape=IMG_SHAPE, include_top=False,
2                  weights='imagenet')
3
4 vgg16.trainable = False
5 vgg16.summary()
6 -----
7 Layer (type)          Output Shape      Param #
8 -----
9 input_1 (InputLayer)   [(None, 160, 160, 3)]  0
10
11 block1_conv1 (Conv2D)  (None, 160, 160, 64)   1792
12
13 block1_conv2 (Conv2D)  (None, 160, 160, 64)   36928

```

```

14
15   block1_pool (MaxPooling2D)    (None, 80, 80, 64)          0
16
17   block2_conv1 (Conv2D)        (None, 80, 80, 128)        73856
18
19   block2_conv2 (Conv2D)        (None, 80, 80, 128)        147584
20
21   block2_pool (MaxPooling2D)   (None, 40, 40, 128)          0
22
23   block3_conv1 (Conv2D)        (None, 40, 40, 256)        295168
24
25   block3_conv2 (Conv2D)        (None, 40, 40, 256)        590080
26
27   block3_conv3 (Conv2D)        (None, 40, 40, 256)        590080
28
29   block3_pool (MaxPooling2D)   (None, 20, 20, 256)          0
30
31   block4_conv1 (Conv2D)        (None, 20, 20, 512)        1180160
32
33   block4_conv2 (Conv2D)        (None, 20, 20, 512)        2359808
34
35   block4_conv3 (Conv2D)        (None, 20, 20, 512)        2359808
36
37   block4_pool (MaxPooling2D)   (None, 10, 10, 512)          0
38
39   block5_conv1 (Conv2D)        (None, 10, 10, 512)        2359808
40
41   block5_conv2 (Conv2D)        (None, 10, 10, 512)        2359808
42
43   block5_conv3 (Conv2D)        (None, 10, 10, 512)        2359808
44
45   block5_pool (MaxPooling2D)   (None, 5, 5, 512)           0
46
47 =====
48 Total params: 14,714,688
49 Trainable params: 0
50 Non-trainable params: 14,714,688
51 -----

```

La forma dell'output del modello per 32 immagini campione ora corrisponde a quella dell'ultimo strato convoluzionale del modello senza testa:

```

1 feature_batch = vgg16(image_batch)
2 Feature_batch.shape

```

³ TensorShape([32, 5, 5, 512])

Possiamo aggiungere nuovi livelli al modello headless usando l'API sequenziale o quella funzionale. Per quanto riguarda l'API sequenziale, l'aggiunta dei livelli GlobalAveragePooling2D, Dense e Dropout funziona come segue:

```

1 global_average_layer = GlobalAveragePooling2D()
2 dense_layer = Dense(64, activation='relu')
3 dropout = Dropout(0.5)
4 prediction_layer = Dense(1, activation='sigmoid')
5 seq_model = tf.keras.Sequential([
6     global_average_layer,
7     dense_layer,
8     dropout,
9     prediction_layer])
10 seq_model.compile(loss = tf.keras.losses.BinaryCrossentropy(from_logits=True),
11                     optimizer = 'Adam',
12                     metrics=['accuracy'])

```

Abbiamo impostato from_logits=True per BinaryCrossentropy loss perché il modello fornisce un risultato lineare. Il riepilogo mostra come il nuovo modello combina gli strati convoluzionali VGG16 pre-addestrati e i nuovi strati finali:

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 5, 5, 512)	14714688
global_average_pooling2d (GlobalAveragePooling2D)	(None, 512)	0
dense (Dense)	(None, 64)	32832
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65
Total params: 14,747,585		
Trainable params: 32,897		
Non-trainable params: 14,714,688		

Prima di addestrare il nuovo strato finale, il VGG16 preaddestrato fornisce un'accuratezza di convalida del 48,75%. Ora si procede all'addestramento del modello per 10 epoche, regolando solo i pesi dello strato finale:

```

1 history = transfer_model.fit(train_batches,
2                               epochs=initial_epochs,
3                               validation_data=validation_batches)

```

10 epoche aumentano l'accuratezza della validazione oltre il 94%. Per perfezionare il modello, possiamo scongelare i modelli VGG16 e continuare l'addestramento. Si noti che questa operazione va eseguita solo dopo l'addestramento dei nuovi strati finali: gli strati di classificazione inizializzati in modo casuale produrranno probabilmente aggiornamenti del gradiente di grandi dimensioni che possono eliminare i risultati del pre-addestramento.

Per scongelare le parti del modello, si seleziona uno strato, dopo di che si impostano i pesi su addestrabili; in questo caso, lo strato 12 dei 19 strati totali dell'architettura VGG16:

```

1 vgg16.trainable = True
2 len(vgg16.layers)
3 19
4 # Fine-tune from this layer onward
5 start_fine_tuning_at = 12
6 # Freeze all the layers before the 'fine_tune_at' layer
7 for layer in vgg16.layers[:start_fine_tuning_at]:
8     layer.trainable = False

```

Ora è sufficiente ricompilare il modello e continuare l'addestramento per un massimo di 50 epoche utilizzando l'arresto anticipato, a partire dall'epoca 10, come segue:

```

1 fine_tune_epochs = 50
2 total_epochs = initial_epochs + fine_tune_epochs
3
4 history_fine_tune = transfer_model.fit(train_batches,
5                                         epochs=total_epochs,
6                                         initial_epoch=history.epoch[-1],
7                                         validation_data=validation_batches,
8                                         callbacks=[early_stopping])

```

La Figura 9.9 mostra come l'accuratezza della validazione aumenti sostanzialmente, raggiungendo il 97,89% dopo altre 22 epoche:

Il Transfer Learning è una tecnica importante quando i dati di addestramento sono limitati, come spesso accade nella pratica. Anche se è improbabile che cani e gatti producano segnali negoziabili (tradeable), il Transfer Learning

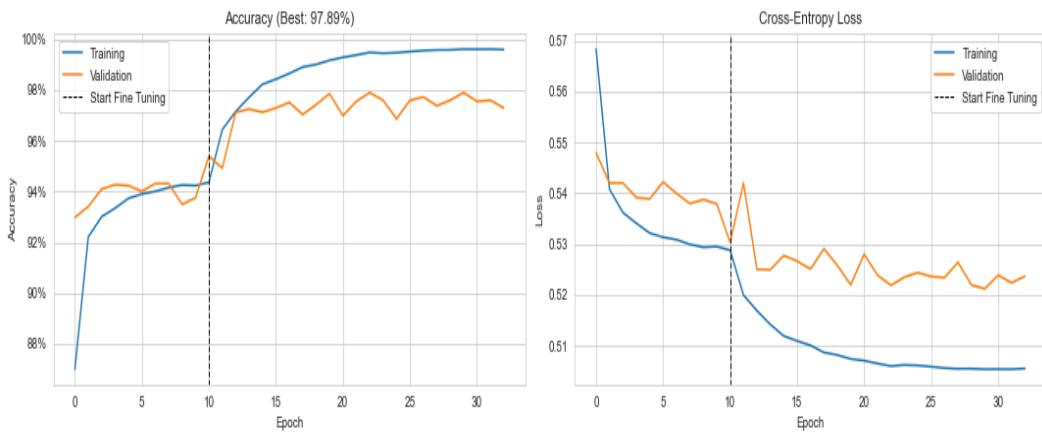


Figura 9.9: Cross-validation performance: accuracy and cross-entropy loss

potrebbe certamente aiutare a migliorare l'accuratezza delle previsioni su un set di dati alternativo come le immagini satellitari che affronteremo in seguito.

9.2.5 Classificare le immagini satellitari con il Transfer Learning

Le immagini satellitari occupano un posto di rilievo tra i dati alternativi (si veda il Capitolo 3). Per esempio, i commercianti di materie prime possono fare affidamento sulle immagini satellitari per prevedere l'offerta di determinate colture o risorse, monitorando l'attività delle aziende agricole, dei siti minerari o del traffico di petroliere.

Il dataset EuroSat

Per illustrare il lavoro con questo tipo di dati, carichiamo il set di dati EuroSat incluso nei set di dati TensorFlow 2 (Helber al 2019). Il datset EuroSat comprende circa 27.000 immagini in formato 64×64 che rappresentano 10 diversi tipi di uso del suolo. La Figura 9.10 mostra un esempio per ogni etichetta:

Una serie temporale di dati simili potrebbe essere utilizzata per tracciare le dimensioni relative delle aree coltivate, industriali e residenziali o lo stato di coltivazioni specifiche per prevedere la quantità o la qualità del raccolto, ad esempio per il vino.



Figura 9.10: Dieci tipi di uso del suolo contenuti nel dataset

Messa a punto di una CNN molto profonda - DenseNet201

Huang et al. (2018) hanno sviluppato una nuova architettura, denominata densamente connessa, basata sull'idea che le CNN possono essere più profonde, più precise e più efficienti da addestrare se contengono connessioni più corte tra gli strati vicini all'ingresso e quelli vicini all'uscita.

Un'architettura, denominata DenseNet201, collega ogni strato a tutti gli altri strati in maniera feedforward. Utilizza le mappe di caratteristiche di tutti i livelli precedenti come input, mentre le mappe di caratteristiche di ogni strato diventano input per tutti gli strati successivi.

Scarichiamo l'architettura DenseNet201 da tensorflow.keras.applications e sostituiamo i suoi strati finali con i seguenti strati densi intervallati da normalizzazione batch per mitigare l'esplosione o lo svanire dei gradienti in questa rete molto profonda con oltre 700 strati:

Layer (type)	Output Shape	Param #
densenet201 (Functional)	(None, 1920)	18321984
batch_normalization (BatchN ormalization)	(None, 1920)	7680
dense (Dense)	(None, 2048)	3934208
batch_normalization_1 (BatchNormalization)	(None, 2048)	8192

```

13
14  dense_1 (Dense)           (None, 2048)      4196352
15
16  batch_normalization_2 (Batch Normalization) (None, 2048)      8192
17
18  dense_2 (Dense)           (None, 2048)      4196352
19
20  batch_normalization_3 (Batch Normalization) (None, 2048)      8192
21
22  dense_3 (Dense)           (None, 2048)      4196352
23
24  batch_normalization_4 (Batch Normalization) (None, 2048)      8192
25
26  dense_4 (Dense)           (None, 10)        20490
27
28
29
30
31 -----
32 Total params: 34,906,186
33 Trainable params: 34,656,906
34 Non-trainable params: 249,280
35 -----

```

Addestramento del modello e valutazione dei risultati

Utilizziamo il 10% delle immagini di addestramento per la convalida e otteniamo la migliore accuratezza di classificazione fuori campione del 97,96% dopo 10 epoche. Questo risultato supera le prestazioni citate nell'articolo originale per l'architettura ResNet-50 dalle migliori prestazioni con una divisione 90-10.

Probabilmente si otterrebbe un ulteriore guadagno di prestazioni aumentando il relativamente piccolo set di addestramento.

9.3 Le CNN per i dati delle serie temporali

Le CNN sono state originariamente sviluppate per l'elaborazione dei dati delle immagini e hanno raggiunto prestazioni sovrumane in vari compiti di computer vision. Come discusso nella prima sezione, i dati delle serie temporali hanno una struttura a griglia simile a quella delle immagini e le CNN sono

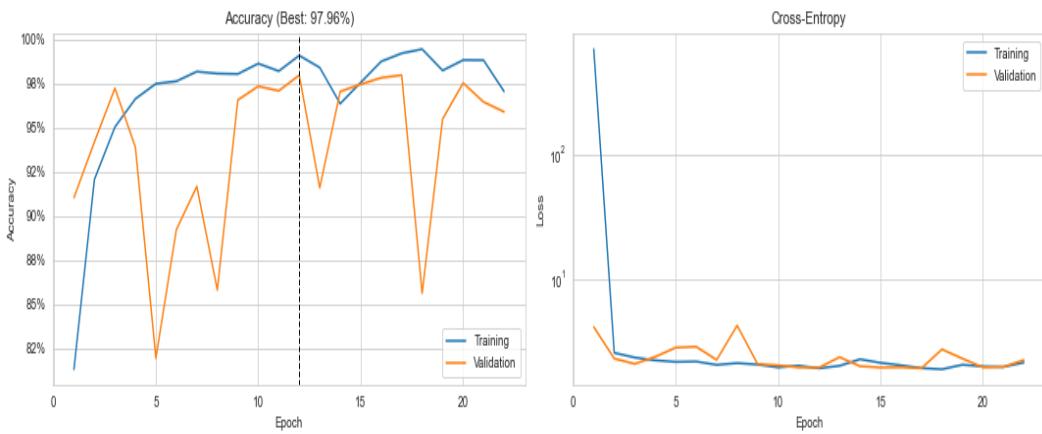


Figura 9.11: Cross-validation performance

state applicate con successo a rappresentazioni mono, bi e tridimensionali di dati temporali.

L'applicazione delle CNN alle serie temporali darà probabilmente i suoi frutti se i dati soddisfano l'assunto chiave del modello, ossia che i modelli o le relazioni locali aiutano a prevedere i risultati. Nel contesto delle serie temporali, gli schemi locali potrebbero essere l'autocorrelazione o simili relazioni non lineari a intervalli rilevanti. Lungo la seconda e la terza dimensione, gli schemi locali implicano relazioni sistematiche tra le diverse componenti di una serie multivariata o tra queste serie per diversi ticker. Poiché la localizzazione è importante, è altrettanto importante che i dati siano organizzati di conseguenza, a differenza delle reti feed-forward, in cui il rimescolamento degli elementi di ogni dimensione non influisce negativamente sul processo di apprendimento.

In questa sezione, replichiamo un recente lavoro di ricerca che ha ottenuto buoni risultati formattando dati multivariati di serie temporali come immagini per prevedere i rendimenti. Svilupperemo e testeremo anche una strategia di trading basata sui segnali contenuti nelle previsioni.

9.3.1 CNN-TA - clustering di serie temporali in formato 2D

Per sfruttare la struttura reticolare dei dati delle serie temporali, possiamo utilizzare architetture CNN per serie temporali univariate e multivariate. In questo caso, consideriamo le diverse serie temporali come canali, simili ai diversi segnali a colori.

Un approccio alternativo converte una serie temporale di fattori alfa in un formato bidimensionale per sfruttare la capacità delle CNN di rilevare i pattern locali. Sezer e Ozbayoglu (2018) propongono **CNN-TA**, che calcola 15 indicatori tecnici per intervalli diversi e utilizza il clustering gerarchico per individuare gli indicatori che si comportano in modo simile l'uno all'altro in una griglia bidimensionale.

Gli autori addestrano una CNN per prevedere se comprare, tenere o vendere un'attività in un determinato giorno. Confrontano le prestazioni della CNN con il modello "buy-and-hold" e con altri modelli e scoprono che supera tutte le alternative utilizzando le serie di prezzi giornalieri dei titoli del Dow 30 e dei nove ETF più scambiati nel periodo 2007-2017.

In questa sezione, sperimentiamo questo approccio utilizzando i dati giornalieri dei prezzi azionari statunitensi e dimostriamo come calcolare e convertire un insieme simile di indicatori in formato immagine. Poi addestriamo una CNN per prevedere i rendimenti giornalieri e valutiamo una semplice strategia long-short basata sui segnali risultanti.

Creazione di indicatori tecnici a intervalli diversi

Selezioniamo innanzitutto un universo dei 500 titoli statunitensi più scambiati dal set di dati Quandl Wiki per per volume di dollari per periodi di cinque anni nel periodo 2007-2017. Per via di problemi nel caricare i file .csv usati in questa sezione su Google Colab, alcuni non erano più reperibili gratuitamente e altri sono troppo grandi, la scrittura del codice non è stata possibile anche se sono stati provati svariati tentativi per risolvere suddetti problemi, perciò è stato scelto di rimandare ai file .ipynb presenti su GitHub e scaricati contenenti il codice e le spiegazioni dettagliate di ciò che viene svolto nella presente sezione. Tali file sono stati scritti da Stefan Jansen autore del libro Machine Learning for Algorithmic Trading da cui sono state prese le informazioni per scrivere questo capitolo sulle CNN nella finanza. Detto ciò, si veda il notebook engineer_cnn_features.ipynb per gli esempi di codice di questa sezione e per alcuni ulteriori dettagli di implementazione.

Le nostre caratteristiche sono costituite da 15 indicatori tecnici e fattori di rischio che calcoliamo per 15 intervalli diversi e poi li disponiamo in una griglia 15×15 . La tabella seguente elenca alcuni degli indicatori tecnici; inoltre, seguiamo l'autore nell'utilizzo delle seguenti metriche:

- Medie mobili ponderate ed esponenziali (WMA e EMA) del prezzo finale
- Tasso di variazione (ROC) del prezzo di chiusura
- Oscillatore Chande Momentum (CMO)

- Oscillatori Chaikin A/D (ADOSC)
- Indice di movimento direzionale medio (ADX)

Per ogni indicatore, il periodo di tempo varia da 6 a 20 per ottenere 15 misurazioni distinte. Ad esempio, il seguente esempio di codice calcola l'**indice di forza relativa (RSI)**:

```

1 T = list(range(6, 21))
2 for t in T:
3     universe[f'{t:02}_RSI'] = universe.groupby(level='symbol').close.
4     apply(RSI, timeperiod=t)

```

Per il **Normalized Average True Range (NATR)**, che richiede diversi input, il calcolo funziona come segue:

```

1 for t in T:
2     universe[f'{t:02}_NATR'] = universe.groupby(
3         level='symbol', group_keys=False).apply(
4             lambda x: NATR(x.high, x.low, x.close, timeperiod=t))

```

Calcolo dei rolling factor betas per diversi orizzonti

Utilizziamo anche cinque fattori di rischio di Fama-French. Essi riflettono la sensibilità dei rendimenti di un titolo ai fattori che hanno dimostrato di avere un impatto sui rendimenti azionari. Catturiamo questi fattori calcolando i coefficienti di una regressione rolling OLS dei rendimenti giornalieri di un titolo sui rendimenti di portafogli progettati per riflettere i fattori sottostanti:

- **Premio di rischio azionario:** Rendimenti ponderati per il valore delle azioni statunitensi meno il tasso di interesse del Tesoro USA a 1 mese.
- **Size (SMB):** rendimenti dei titoli classificati come piccoli (in base alla capitalizzazione di mercato) meno quelli dei titoli grandi.
- **Value (HML):** I rendimenti dei titoli con un alto valore di mercato meno quelli dei titoli con un basso valore di mercato.
- **Investimento (CMA):** Differenze di rendimento per le società con spese di investimento conservativo rispetto a quelle con spese aggressive.
- **Redditività (RMW):** Allo stesso modo, le differenze di rendimento per i titoli con una redditività robusta rispetto a quelli con una metrica debole.

I dati provengono dalla libreria di dati di Kenneth French utilizzando pandas_datareader:

```

1 import pandas_datareader.data as web
2 factor_data = (web.DataReader(
3     'F-F_Research_Data_5_Factors_2x3_daily',
4     'famafrench', start=START)[0])

```

Successivamente, applichiamo RollingOLS() di statsmodels per eseguire regressioni su periodi di diverse lunghezze, da 15 a 90 giorni. Abbiamo impostato il parametro params_only sul metodo .fit() per accelerare il calcolo e catturare i coefficienti utilizzando l'attributo .params del fitted factor_model:

```

1 factors = [Mkt-RF, 'SMB', 'HML', 'RMW', 'CMA']
2 windows = list(range(15, 90, 5))
3 for window in windows:
4     betas = []
5     for symbol, data in universe.groupby(level='symbol'):
6         model_data = data[[ret]].merge(factor_data, on='date').dropna()
7         model_data[ret] == model_data.RF
8         rolling_ols = RollingOLS(endog=model_data[ret],
9                               exog=sm.add_constant(model_data[factors]),
10                              window=window)
11        factor_model= rolling_ols.fit(params_only=True)
12        .params.drop('const', axis=1)
13        result = factor_model.assign(symbol=symbol).set_index('symbol',
14                                         append=True)
15        betas.append(result)
16        betas = pd.concat(betas).rename(columns=lambda x: f'{window:02}-{x}')
17        universe = universe.join(betas)

```

Selezione delle caratteristiche in base alle informazioni reciproche

Il passo successivo consiste nel selezionare le 15 caratteristiche più rilevanti tra le 20 candidate per riempire la griglia di input 15×15 . Gli esempi di codice per i passaggi successivi si trovano nel notebook convert_cnn_features_to_image_format.ipynb.

A tal fine, stimiamo l'informazione reciproca per ogni indicatore e gli intervalli di 15 rispetto al nostro obiettivo, i rendimenti a termine di un giorno. scikit-learn mette a disposizione la funzione mutual_info_regression() che rende questa operazione semplice, anche se richiede tempo e memoria. Per accelerare il processo, campioniamo a caso 100.000 osservazioni:

```

1 df = features.join(targets[target]).dropna().sample(n=100000)
2 X = df.drop(target, axis=1)
3 y = df[target]

```

⁴ `mi[t] = pd.Series(mutual_info_regression(X=X, y=y), index=X.columns)`

Il pannello di sinistra della Figura 9.12 mostra l'informazione reciproca, mediata sui 15 intervalli per ciascun indicatore. NATR, PPO e Bande di Bollinger sono i più importanti dal punto di vista di questa metrica:

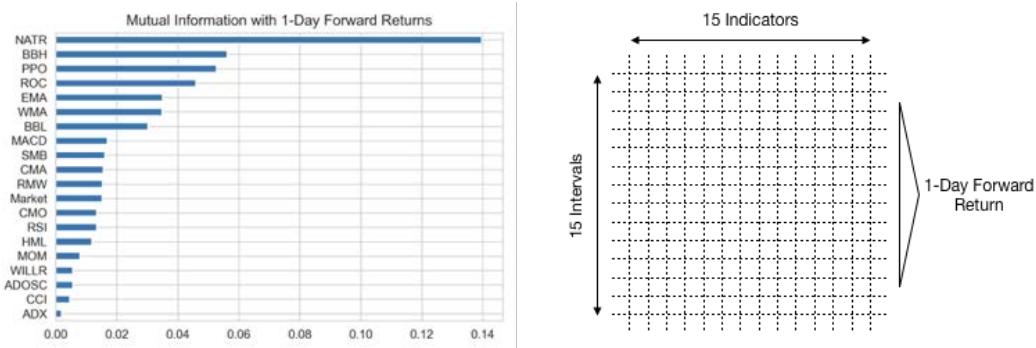


Figura 9.12: Informazioni reciproche e layout di griglia bidimensionale per le serie temporali

Clustering gerarchico delle caratteristiche

Il pannello di destra della Figura 9.12 illustra la griglia bidimensionale di 15 X 15 caratteristiche che alimenteremo nella nostra CNN. Come discusso nella prima sezione di questo capitolo, le CNN fanno affidamento sulla località dei patterns rilevanti che si trovano tipicamente nelle immagini in cui i pixel vicini sono strettamente correlati e cambiano da un pixel all'altro. I cambiamenti da un pixel all'altro sono spesso graduali.

Per organizzare i nostri indicatori in maniera simile, seguiremo l'approccio di Sezer e Ozbayoglu per applicare il clustering gerarchico. L'obiettivo è identificare le caratteristiche che si comportano similmente e ordinare le colonne e le righe della griglia di conseguenza.

Possiamo basarci sulle funzioni `pairwise_distance()`, `linkage()` e `dendrogram()` di SciPy. Creiamo una funzione ausiliaria che standardizza l'input in senso colonna per evitare di distorcere le distanze tra le caratteristiche a causa delle differenze di scala, e utilizziamo il criterio di Ward che unisce i cluster per minimizzare la varianza. La funzione restituisce l'ordine dei nodi foglia nel dendrogramma che, a sua volta, visualizza la successiva formazione di cluster più grandi:

```
1 def cluster_features(data, labels, ax, title):
2     data = StandardScaler().fit_transform(data)
3     pairwise_distance = pdist(data)
```

```

4 Z = linkage(data, 'ward')
5 dend = dendrogram(Z,
6     labels=labels,
7     orientation='top',
8     leaf_rotation=0.,
9     leaf_font_size=8.,
10    ax=ax)
11 return dend['ivl']

```

Per ottenere l'ordine ottimizzato degli indicatori tecnici nelle colonne e dei diversi intervalli nelle righe, utilizziamo il metodo `.reshape()` di NumPy per assicurarci che la dimensione che desideriamo raggruppare appaia nelle colonne dell'array bidimensionale che passiamo a `cluster_features()`:

```

1 labels = sorted(best_features)
2 col_order = cluster_features(features.dropna().values.reshape(-1, 15).T, labels)
3 labels = list(range(1, 16))
4 row_order = cluster_features(
5     features.dropna().values.reshape(-1, 15, 15).transpose((0, 2,
6         1)).reshape(-1, 15).T, labels)

```

La Figura 9.13 mostra i dendrogrammi per le caratteristiche di riga e colonna:

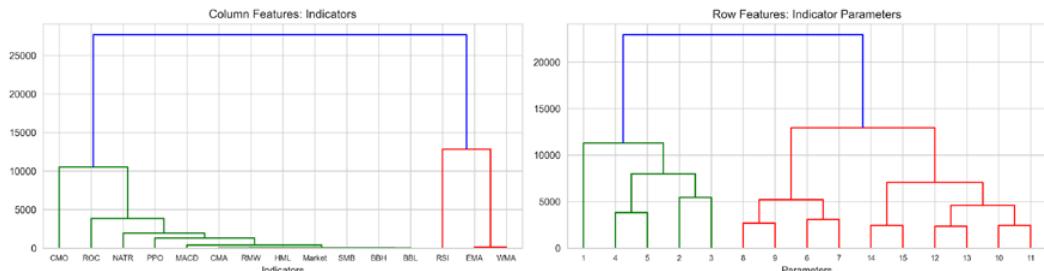


Figura 9.13: Dendrogrammi per caratteristiche di riga e colonna

Riordiniamo le caratteristiche di conseguenza e memorizziamo il risultato come input per la CNN che creeremo nella fase successiva.

Creare e addestrare una rete neurale convolutiva

Ora siamo pronti a progettare, addestrare e valutare una CNN seguendo i passi descritti nella sezione precedente. Il notebook `cnn_for_trading.ipynb` contiene gli esempi di codice pertinenti.

Anche in questo caso seguiamo fedelmente l'autore creando una CNN con 2 livelli convoluzionali con kernel di dimensione 3 e 16 e 32 filtri, rispettivamente,

seguiti da uno strato di max pooling di dimensione 2. Appiattiamo l'uscita dell'ultima pila di filtri e colleghiamo le 1.568 uscite risultanti a uno strato denso di dimensione 32, applicando il 25 e il 50 per cento di probabilità di abbandono alle connessioni in entrata e in uscita per mitigare l'overfitting. La tabella seguente riassume la struttura della CNN che contiene 55.041 parametri addestrabili:

Layer (type)	Output Shape	Param #
CONV1 (Conv2D)	(None, 15, 15, 16)	160
CONV2 (Conv2D)	(None, 15, 15, 32)	4640
POOL2 (MaxPooling2D)	(None, 7, 7, 32)	0
DROP1 (Dropout)	(None, 7, 7, 32)	0
FLAT1 (Flatten)	(None, 1568)	0
FC1 (Dense)	(None, 32)	50208
DROP2 (Dropout)	(None, 32)	0
FC2 (Dense)	(None, 1)	33
Total params: 55,041		
Trainable params: 55,041		
Non-trainable params: 0		

Validazione incrociata (Cross-validation) del modello con il generatore di indici MutipleTimeSeriesCV train e validation set. Forniamo 5 anni di giorni di trading durante il periodo di training in lotti (batches) di 64 campioni casuali e convalidiamo usando i 3 mesi successivi, coprendo gli anni 2014-2017.

Scaliamo le caratteristiche nell'intervallo [-1, 1] e utilizziamo nuovamente il metodo .reshape() di NumPy per creare il formato richiesto N x 15 x 15 x 1:

```

1 def get_train_valid_data(X, y, train_idx, test_idx):
2     x_train, y_train = X.iloc[train_idx, :], y.iloc[train_idx]
3     x_val, y_val = X.iloc[test_idx, :], y.iloc[test_idx]
4     scaler = MinMaxScaler(feature_range=(-1, 1))
5     x_train = scaler.fit_transform(x_train)
6     x_val = scaler.transform(x_val)

```

```

7  return (x_train.reshape(-1, size, size, 1), y_train,
8  x_val.reshape(-1, size, size, 1), y_val)

```

L’addestramento e la validazione seguono il processo di memorizzare i pesi dopo ogni epoca e di generare previsioni per le iterazioni con le migliori prestazioni, senza dover ricorrere a un costoso retraining.

Per valutare l’accuratezza predittiva del modello, calcoliamo il coefficiente di **informazione giornaliero (IC)** per il validation set come segue:

```

1 checkpoint_path = Path('models', 'cnn_ts')
2 for fold, (train_idx, test_idx) in enumerate(cv.split(features)):
3     X_train, y_train, X_val, y_val = get_train_valid_data(features, target, train_idx, test_idx)
4
5     preds = y_val.to_frame('actual')
6     r = pd.DataFrame(index=y_val.index.unique(level='date'))
7     .sort_index()
8     model = make_model(filter1=16, act1='relu', filter2=32,
9                         act2='relu', do1=.25, do2=.5, dense=32)
10    for epoch in range(n_epochs):
11        model.fit(X_train, y_train,
12                    batch_size=batch_size,
13                    validation_data=(X_val, y_val),
14                    epochs=1, verbose=0, shuffle=True)
15        model.save_weights(
16            (checkpoint_path / f'ckpt_{fold}_{epoch}').as_posix())
17        preds[epoch] = model.predict(X_val).squeeze()
18        r[epoch] = preds.groupby(level='date').apply(
19            lambda x: spearmanr(x.actual, x[epoch])[0]).to_frame(epoch)

```

Addestriamo il modello per un massimo di 10 epoche utilizzando la discesa stocastica del gradiente con Nesterov e scopriamo che le epoche più performanti, 8 e 9, raggiungono un IC medio giornaliero (basso) di circa 0,009.

Assemblare i migliori modelli per generare segnali negoziabili (tradeable signals)

Per ridurre la varianza delle previsioni per il periodo di prova, generiamo e facciamo la media delle previsioni dei 3 modelli che hanno ottenuto i migliori risultati durante la convalida incrociata, che in questo caso corrispondono all’addestramento per 4, 8 e 9 epoche. Il periodo di addestramento relativamente breve sottolinea che la quantità di segnali nelle serie temporali finanziarie è bassa rispetto all’informazione sistematica contenuta nei dati delle immagini.

La funzione generate_predictions() ricarica i pesi del modello e restituisce le previsioni per il periodo di riferimento:

```

1 def generate_predictions(epoch):
2     predictions = []
3     for fold, (train_idx, test_idx) in enumerate(cv.split(features)):
4         X_train, y_train, X_val, y_val = get_train_valid_data(
5             features, target, train_idx, test_idx)
6         preds = y_val.to_frame('actual')
7         model = make_model(filter1=16, act1='relu', filter2=32,
8             act2='relu', do1=.25, do2=.5, dense=32)
9         status = model.load_weights(
10            (checkpoint_path / f'ckpt_{fold}_{epoch}').as_posix())
11         status.expect_partial()
12         predictions.append(pd.Series(model.predict(X_val).squeeze(),
13             index=y_val.index))
14     return pd.concat(predictions)
15
16 preds = []
17 for i, epoch in enumerate(ic.drop('fold', axis=1)
18 .mean().nlargest(3).index):
19     preds[i] = generate_predictions(epoch)

```

Memorizziamo le previsioni e procediamo al backtest di una strategia di trading basata su queste previsioni di rendimento giornaliero.

Backtesting di una strategia di trading long-short

Per avere un'idea della qualità del segnale, calcoliamo lo spread tra portafogli equamente ponderati investiti in titoli selezionati in base ai quantili del segnale utilizzando Alphalens (per i dettagli implementativi di Alphalens vedi notebook `performance_eval_alphalens.ipynb`).

La Figura 9.14 mostra che, per un orizzonte di investimento di un giorno, questa strategia ingenua avrebbe guadagnato poco più di quattro punti base al giorno nel periodo 2013-2017.

Traduciamo questo risultato leggermente incoraggiante in una semplice strategia che inserisce posizioni long (short) per i 25 titoli con le previsioni di rendimento più alte (più basse), operando su base giornaliera. La Figura 9.15 mostra che questa strategia è competitiva con il benchmark S&P 500 per gran parte del periodo di backtesting (pannello di sinistra), con un rendimento cumulativo del 35,6% e uno Sharpe ratio di 0,53 (prima dei costi di transazione; pannello di destra).

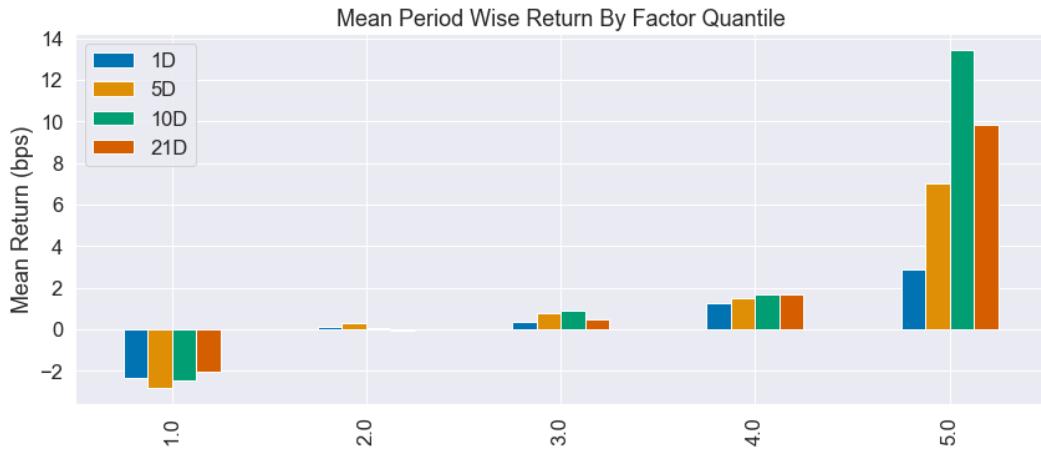


Figura 9.14: Alphalens valutazione della qualità del segnale

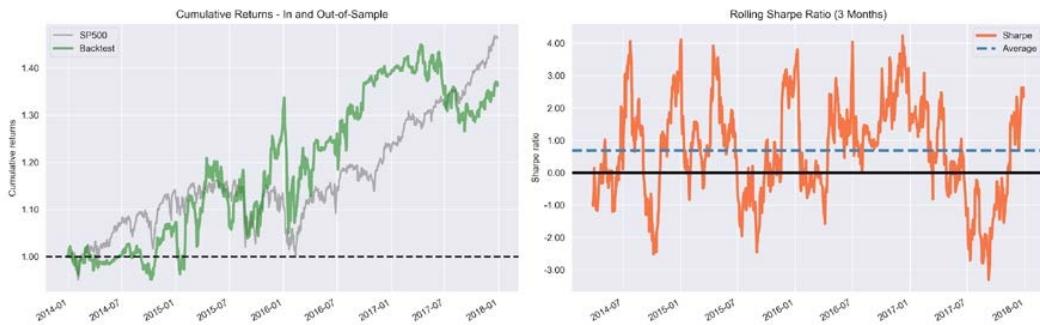


Figura 9.15: Performance del backtest dentro e fuori dal campione

9.4 Conclusioni

In questo capitolo abbiamo introdotto le CNN, un’architettura NN specializzata che ha preso spunto dalla nostra (limitata) comprensione della visione umana e che funziona particolarmente bene su dati di tipo reticolare. Abbiamo trattato l’operazione centrale della convoluzione o della correlazione incrociata che guida la scoperta di filtri che a loro volta individuano caratteristiche utili per risolvere il compito da svolgere. Abbiamo esaminato diverse architetture allo stato dell’arte che rappresentano dei buoni punti di partenza, soprattutto perché perché l’apprendimento per trasferimento ci permette di riutilizzare i pesi preaddestrati e di ridurre il lavoro di training, altrimenti piuttosto impegnativo dal punto di vista computazionale e dei dati.

Nella prossima parte, ci occuperemo delle regolamentazioni dell’IA nel campo finanziario e delle prospettive future che l’impiego dell’IA nella finanza potrà

avere.

Parte V

Regolamentazione e Prospettive future

Capitolo 10

Regolamentazione dell'IA in Finanza

“Un ambiente ad alto rischio ed estremamente competitivo in cui operano oggi i sistemi di intelligenza artificiale è il mercato finanziario globale.”
—Nick Bostrom (2014)

In questo capitolo vogliamo esaminare alcuni argomenti non matematici circa l'utilizzo dell'intelligenza artificiale nelle applicazioni finanziarie.

10.1 Regtech

Con il termine **”Regtech”** ci si riferisce all'uso dell'intelligenza artificiale e del machine learning nella gestione dei processi normativi per il settore finanziario all'interno del settore finanziario. Questo è il caso in cui le banche, le compagnie di assicurazione ed i fornitori di servizi finanziari utilizzano l'IA ed il ML per rendere più veloci e più efficienti i loro processi interni per conformarsi alle leggi ed ai regolamenti.

Dunque le principali funzioni sono il monitoraggio normativo, il reporting normativo e la conformità alle normative. Ogni volta che una banca, compagnia assicurativa o istituto finanziario utilizza l'intelligenza artificiale per migliorare i processi normativi interni è ciò che chiamiamo **Regtech**. Oggi-giorno, sulla scia della crisi finanziaria, abbiamo molti più regolamenti, molte più supervisioni delle istituzioni finanziarie, quindi le autorità di vigilanza e di regolamentazione richiedono sempre più informazioni e ci sono sempre più regole a cui le banche e le compagnie assicurative devono attenersi.

Per molte banche ed istituti finanziari soddisfare tutti questi requisiti normativi è diventato un lavoro a tempo pieno; hanno bisogno di assumere avvo-

cati negli uffici di conformità. Di norma, per la maggior parte di queste cose, le agenzie di vigilanza non sono molto sofisticate quando si tratta della tecnologia che viene utilizzata, quindi in molti casi i professionisti devono compilare fogli Excel enormi che poi devono essere inviati alle autorità di regolamentazione e questo richiede molto tempo, per accelerare questo processo e renderlo più efficiente ma anche sicuro si utilizza l'intelligenza artificiale. Ad esempio troviamo due aziende che offrono questa tipologia di servizio:

- **IdentityMind Global:** offre servizi di anti-frode e di gestione del rischio per le transazioni digitali.
- **Trunomi:** offre servizi per la gestione del consenso per i dati del servizio clienti.

Ci sono molte altre società e società di consulenza che usano l'intelligenza artificiale ed il machine learning per fornire i servizi di consulenza su questi argomenti, ad esempio **Deloitte** è una azienda punto di riferimento di questo tipo.

Per avere una migliore idea su **Regtech**, illustriamo tutte le sezioni di mercato che essa comprende:

- **Profilazione e Due Diligence:** Si raccolgono e si integrano dati da molteplici fonti interne ed esterne. Ha come obiettivo profilare un'entità per confermare l'identità di una persona o di un'azienda, oppure per classificarli in base ai requisiti normativi.
- **Reporting and Dashboards:** Si raccolgono e si integrano dati da molteplici fonti interne. Ha come obiettivo quello di creare report standardizzati per scopi gestionali o di conformità normativa.
- **Risk Analytics:** Anche qui si raccolgono e si integrano dati da molteplici fonti interne ed esterne. L'obiettivo è valutare il rischio di frodi, abusi di mercato, o cattiva condotta a livello di transazione, ad esempio nell'investment banking o nel trading si potrebbe voler utilizzare strumenti di IA e di ML per analizzare le transazioni e osservare se la società ha un'esposizione al rischio troppo alta o se potrebbe esserci un rischio di frode.
- **Conformità Dinamica:** Si utilizzano metodi di machine learning per facilitare il monitoraggio delle modifiche normative per garantire un adattamento flessibile delle politiche e, più importante, dei processi in atto, altrimenti potrebbe accadere che ogni volta che qualcosa cambia nella regolamentazione rende necessaria l'assunzione di consulenti esterni per modificare i processi che sono stati messi in atto qualche tempo prima.

- **Monitoraggio del mercato:** Si raccolgono e integrano dati da molteplici fonti esterne. L'obiettivo è abbinare i risultati negativi a livello di mercato alle regole normative o aziendali. Per esempio, si vuole identificare la scarsa performance di un prodotto, la manipolazione del mercato ecc.

10.1.1 Regtech in numeri

Oggigiorno il settore Regtech è ancora dominato dalle start-up, quasi il 70% delle aziende ha meno di cinque anni.

Secondo il rapporto *The Global RegTech Industry Report by Cambridge Center for Alternative Finance supportato da EY Japan*, il settore Regtech impiega solo 44.000 persone a livello globale, tuttavia ci sono i presupposti per far sì che questa parte del settore finanziario cresca enormemente nei prossimi anni, dovuto anche all'aumento dei requisiti normativi e di vigilanza. Il settore Regtech, sempre secondo il suddetto report, raccoglie circa 5 miliardi di entrate annuali, e circa 9.7 miliardi sono stati raccolti dal settore da finanziamenti esterni.

Il mercato e l'ambiente normativo sono classificati come favorevoli alle società di tecnologia ed il ritmo dei cambiamenti normativi è aumentato sin dalla grande crisi finanziaria del 2008/2009 e aumenterà ancora ancora di più, specialmente ora che siamo in un periodo di crisi continua. Vi sono punizioni e sanzioni per la non conformità con le decisioni normative e questo ha portato ad un aumento della domanda non solo di responsabili della conformità ed esperti in conformità e supervisione bancaria e assicurativa, ma anche di metodologie automatizzate ed affidabili per accelerare le elaborazioni.

Il seguente grafico (Figura 10.1) mostra alcune informazioni sui top 10 mercati Regtech secondo il CCAF Report del 2019:

Osserviamo che i top 10 mercati Regtech sono nel Regno Unito, negli Stati Uniti ma anche nell'Unione Europea, in Lussemburgo, in Svizzera ed in Irlanda, questo perché, come ben sappiamo, sono paesi in cui sono presenti molte industrie finanziarie, società e fornitori di servizi finanziari.

Chi sono i clienti RegTech?

Il Cambridge Center for Alternative Finance in collaborazione con EY ha intervistato 658 società del RegTech (circa l'80% delle società del settore) ottenendo i seguenti risultati circa la clientela delle compagnie RegTech.

- l'89-94% delle compagnie RegTech prende di mira le banche come clienti.
- il 61% ha come target le assicurazioni.

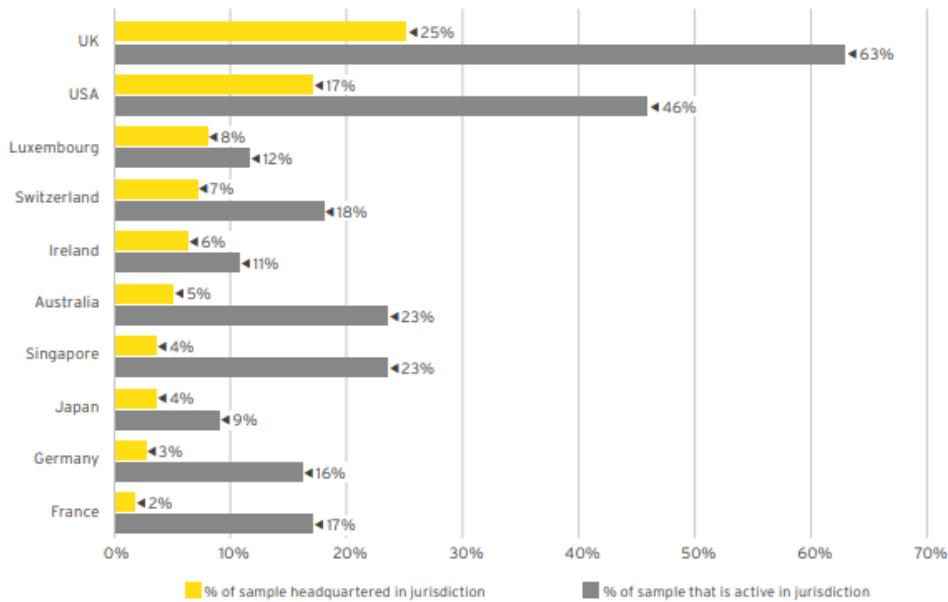


Figura 10.1: Top 10 RegTech markets

- il 58% serve le FinTechs.

Tuttavia, il 58% delle compagnie RegTech afferma di avere anche clienti al di fuori del settore dei servizi finanziari, il che ha senso poiché le grandi società industriali di solito hanno meno regolamentazione e di solito nessuna regolamentazione, ma la maggior parte di queste grandi società ha anche problemi simili a quelli del settore finanziario, ad esempio il rischio di transazioni fraudolente o errate, e quindi ogni grande azienda industriale è alla ricerca di processi automatizzati all'interno della sua funzione finanziaria.

Quali sono le principali tecnologie e strumenti utilizzati dalle compagnie Regtech?

Come osserviamo nel grafico (Figura 10.2).

Il Cloud Computing, Machine Learning, Analisi predittiva dei dati, NLP ed il Deep Learning sono le più importanti tecnologie utilizzate dalle società Regtech per fornire i loro servizi di regolamentazione alle aziende, banche ed assicurazioni.

I Regolatori

I Regolatori devono sempre adattarsi a nuovi servizi finanziari abilitati dalla tecnologia, questi possono rappresentare una sfida soprattutto per la fusione

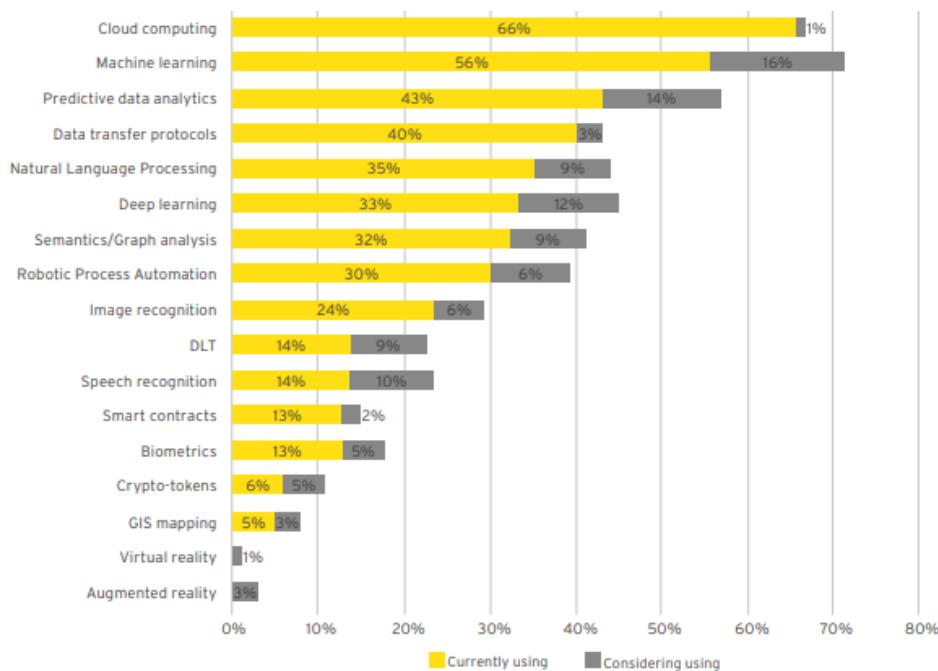


Figura 10.2: Tecnologie e strumenti usati dalle compagnie Regtech, Fonte: CCFA Report

e lo sviluppo economico delle nazioni. Alcuni regolatori come **BaFin** hanno escogitato qualcosa che si chiama **Sandbox Normativo** che è un programma formale che consente fornire determinati servizi finanziari e alcuni modelli di business che non sono ancora pienamente conformi alle leggi esistenti e l'obiettivo è conoscere le opportunità ed i rischi che una particolare innovazione nella tecnologia finanziaria comporta, ad esempio, sviluppare politiche basate su prove per vedere cosa dovrebbero fare le autorità di regolamentazione e cosa non dovrebbero fare, per esempio, l'introduzione delle criptovalute ,come i bitcoin, che almeno all'inizio non sono regolamentate e quindi i regolatori ed i supervisori devono pensare a come regolamentarle e se regolamentarle del tutto, questo può essere fatto in una Sandbox Normativa per consentire un tipo di innovazione in modo limitato e per monitorare attentamente per vedere cosa dovrebbe essere fatto e cosa non dovrebbe essere fatto.

Alcuni regolatori come l'FCA (Financial Conduct Authority) del Regno Unito hanno centri o uffici per l'innovazione, questi sono luoghi in cui innovatori e regolatori si incontrano per discutere soluzioni alle sfide per il settore finanziario.

Ciò che è stato fatto , ad esempio nel Regno Unito, si trattava di set-

te **TechSprints** e di eventi di due giorni che includevano rappresentanti del settore ed innovatori, questi riguardavano i servizi finanziari di segnalazione normativa, salute mentale e antiriciclaggio, in questo modo le autorità di regolamentazione e l'industria cercano di discutere nuove idee riguardanti le nuove tecnologie e le nuove sfide.

10.2 Suptech

La Suptech è una sottodisciplina della Regtech, ma in realtà può essere vista come un'estensione di questa. La Suptech si focalizza su tecnologie innovative come i Big Data e IA utilizzate dalle autorità finanziarie come parte delle azioni di vigilanza, in questo modo abbiamo autorità finanziarie, come BaFin in Germania e FCA in UK, che utilizzano i Big Data ed il ML per supportare la supervisione delle istituzioni finanziarie, questo è ciò che chiamiamo SupTech.

SubTech è strettamente correlato a RegTech, ovviamente le aziende hanno un incentivo a rispettare le normative e rendere tale processo più efficiente ed economico possibile, SupTech non deve solo rendere questo processo più efficiente dal lato delle autorità di vigilanza ma anche identificare frodi e potenziali minacce alla stabilità finanziaria. Per questo motivo l'attenzione è solitamente rivolta all'analisi della cattiva condotta, alle segnalazioni fatte dagli istituti finanziari e alla gestione di tutti i dati che arrivano alle autorità di vigilanza finanziaria.

Nell'ambito delle autorità di vigilanza finanziaria c'è stato un sondaggio tra 39 autorità finanziarie di 31 paesi sull'implementazione di strategie Suptech e ciò che è stato scoperto è che hanno identificato due ampi approcci:

- **Specific Suptech Roadmaps** basata su particolari esigenze di un dipartimento, questo approccio tende ad essere più sperimentale, ad esempio, alcuni dipartimenti all'interno di un supervisore finanziario potrebbero aver bisogno di un modello per identificare le transazioni fraudolente in borsa dopodiché tale modello viene sperimentato.
- **Programmi di Trasformazione Digitale a livello di istituzione e di Innovazione basata sui dati (DT&DI)** è un approccio molto più ampio che comprende l'intera agenzia di vigilanza, ad esempio, la gestione e la governance dell'agenzia hanno deciso che hanno bisogno di una trasformazione del loro IT complessivo e di conseguenza ha senso concentrarsi sulle tecnologie IA e ML come parte di questa trasformazione IT all'interno dell'intera agenzia.

Come si osserva nella seguente Figura 10.3, in realtà il 50% delle 39 autorità di vigilanza ha affermato che non alcuna strategia, ciò indica come tutto questo sia ancora molto sperimentale per la maggior parte dei supervisori finanziari.

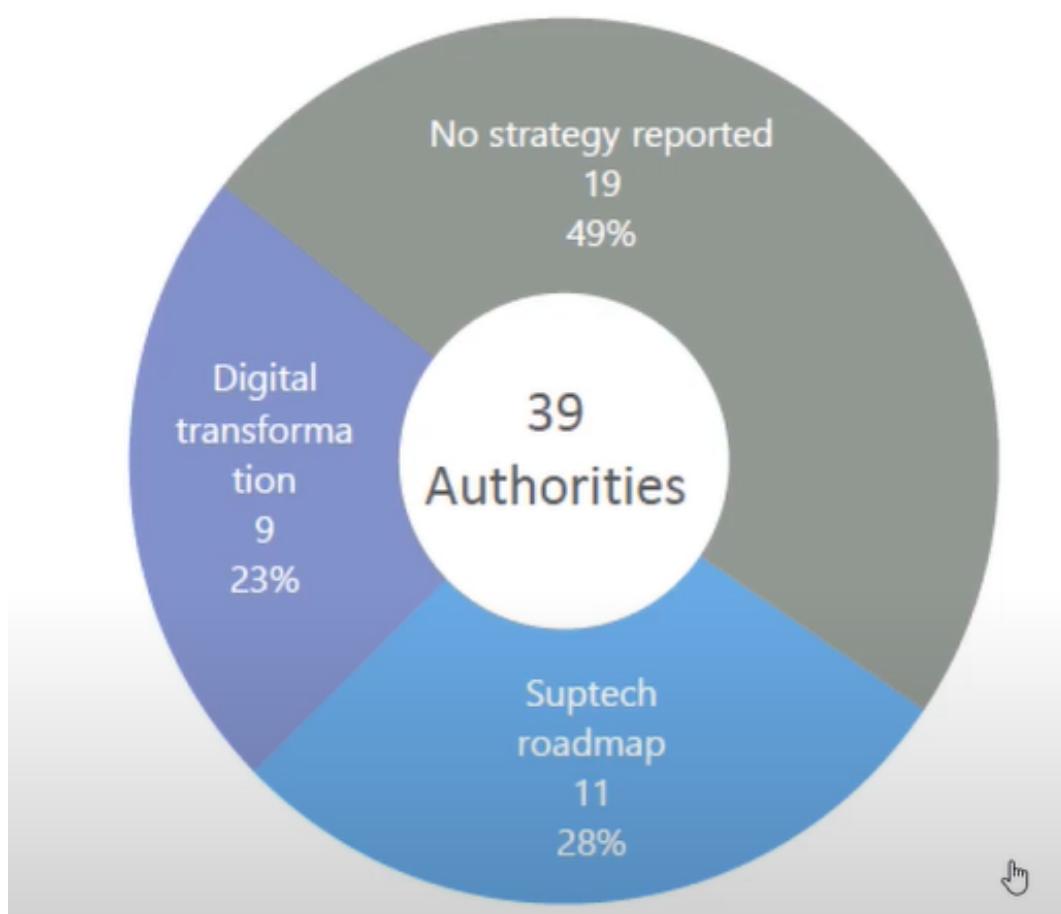


Figura 10.3: SupTech-Focus Areas, Fonte: FSI Report

In conclusione, SupTech è ancora agli albori ma sta guadagnando slancio poiché, come abbiamo visto, le istituzioni che sono supervisionate e regolamentate stanno investendo in RegTech, dunque ha senso che le agenzie di supervisione tengano traccia di questo sviluppo e seguano le orme delle istituzioni che dovrebbero supervisionare, anche se questo è ancora sperimentale vediamo molti supervisori finanziari investire sempre di più nelle tecnologie di IA e di ML come parte della supervisione finanziaria. Ovviamente, queste agenzie hanno a disposizione molti meno fondi rispetto alle società controllate, ma se prendiamo come esempio la Germania ed il recente scandalo con Wirecard, è

probabile che il BaFin verrà riformato in una certa misura e probabilmente esplorerà nuovi modi per identificare le minacce alla stabilità finanziaria e la cattiva condotta finanziaria.

10.3 Rischio Sistematico dovuto all'IA in finanza

Nella supervisione e regolamentazione finanziaria la stabilità finanziaria è uno degli obiettivi principali. I supervisori mirano a raggiungere la stabilità finanziaria per prevenire le crisi finanziarie.

Come l'utilizzo dell'IA in finanza è in relazione con il rischio sistematico?

Ci sono due facce della stessa medaglia, la prima è che l'IA ed il ML possono essere utilizzati dalle istituzioni finanziarie, ad esempio, per tenere traccia dell'esposizione al rischio, e gestirlo in modo più efficiente. Dunque l'IA può essere utilizzata dalle banche e dalle assicurazioni per migliorare la stabilità finanziaria semplicemente facendo un miglior lavoro nella gestione del rischio e nel trading. L'altra faccia della medaglia è che le tecnologie di IA e di ML ed il loro diffuso utilizzo nel settore finanziario possono esse stesse causare un rischio sistematico. Come ben sappiamo un aspetto molto importante dei metodi di ML è la loro capacità di generalizzare su nuovi dati, gli esseri umani possono attingere a un'ampia gamma di esperienze precedenti ed utilizzare la loro immaginazione, tuttavia quando si utilizza l'IA nelle funzioni di regolamentazione essa non è in grado di ragionare su eventi a cui non è ancora stata esposta, può solo estrapolare da ciò che ha visto in precedenza. Possiamo sperare che un modello sia stato adeguatamente addestrato e che non esegua l'overfitting e generalizzi bene a nuovi dati. Questo è il problema che abbiamo visto in molte crisi finanziarie, ovvero che a volte gli eventi si ripetono ma di solito le crisi finanziarie si verificano perché accade qualcosa di piuttosto nuovo o una nuova combinazione di eventi che porta ad una nuova crisi.

Oltre a questo non sappiamo come l'IA prende le decisioni, è troppo complesso per noi da seguire, di solito è una black-box e sicuramente è una black-box per le parti interessate esterne in modo tale che solo le aziende sanno come la loro IA ed i loro modelli di ML funzionano. Questa opacità non è mai una buona cosa nelle istituzioni finanziarie, ciò porta all'incertezza, quando gli investitori ed i detentori del debito non sanno cosa succede all'interno di una banca o di una compagnia di assicurazioni, ciò potrebbe portare a reazioni

di panico nelle notizie e quindi causare crisi finanziarie. L'IA è per sé una black-box e ciò potrebbe causare problemi lungo la strada.

Un altro fattore è che l'IA ha maggiori probabilità di amplificare cicli attuali, è incline alla prociclicità rispetto ai regolatori umani, il problema è che l'automazione favorisce metodologie e standardizzazione omogenee che portano al problema della *prociclicità*.¹ La regolamentazione anti-ciclica è di solito quello che oggi gli economisti finanziari ritengono essere un approccio più intelligente, il che significa che se l'economia prende una brutta piega, si perda la regolamentazione in modo che l'economia si riprenda più rapidamente e che nella fase di crescita economica si inasprisca la regolamentazione al fine di prevenire comportamenti eccessivi nel dare prestiti da parte delle banche che causerebbero la prossima crisi.

L'ultimo fattore del rischio sistematico dovuto all'IA in finanza è dovuto alla elevata prevedibilità e alla trasparenza dell'IA nel modo in cui funziona, ciò consentirà alle persone di scommettere contro di essa. Immaginiamo, ad esempio, che tutti gli investitori nel mercato azionario fossero dei metodi di IA e dei robot automatizzati che sono stati addestrati sui dati basati su algoritmi si ML, allora bisognerebbe soltanto capire come tutti questi robo-traders funzionano dopodiché un essere umano sarà in grado di battere questi sistemi di IA abbastanza facilmente poiché tutti questi modelli sono prevedibili e altamente trasparenti quando si tratta del modo in cui essi funzionano. Ecco perché alcuni problemi potrebbero essere causati semplicemente dall'effetto che l'IA è molto prevedibile in tal senso.

Rischio Esogeno e rischio Endogeno

Il rischio Esogeno è causato da eventi esterni, è molto simile ad un asteroide che cade a Londra o a Roma, è facile da misurare e non si può far nulla quando si tratta di investimenti finanziari. L'IA ed il ML sono molto adatti per la valutazione sulla gestione del rischio esogeno. Ad esempio, possiamo avere molti dati sugli avvistamenti degli asteroidi passati e molti altri dati astrofisici per addestrare i nostri modelli con essi per poter prevedere l'arrivo di futuri asteroidi. L'IA è molto adatta per la microregolazione e la gestione del rischio interno di eventi esogeni.

Il rischio Endogeno è causato da eventi all'interno del sistema quindi inizia quando gli individui e le singole entità all'interno del sistema smettono

¹essa consiste nel timore che il nuovo sistema di requisiti patrimoniali relativi al rischio di credito, fondati sui sistemi di rating interni delle banche, possa accentuare le fluttuazioni del ciclo economico aggravando in particolare le fasi recessive. Cioè quando il sistema economico attraversa una fase recessiva, le condizioni economico-finanziarie delle imprese tendono a deteriorarsi.

di agire in modo indipendente ma sincronizzano il loro comportamento ed è molto difficile da misurare. I metodi di IA vengono addestrati con vari giochi per situazioni comprese le interazioni tra entità quindi con informazioni complete, ad esempio, negli scacchi tutte le mosse possono essere conosciute, le reti neurali profonde hanno successo in questi giochi perché presumono che il loro avversario sia il loro clone e non si preparano per situazioni di rischio endogeno. Quando abbiamo giochi di informazione incompleti, per esempio nel poker, le carte degli avversari sono sconosciute e quindi l'IA gioca peggio degli umani perché non può teorizzare sulle intenzioni dell'avversario può solo imparare dalle mosse passate. Con i giochi cooperativi è ancora più difficile per l'IA, quando la cooperazione porta a multipli local optima, come nel caso della diplomazia e della teoria dei giochi, in questi casi diventa molto difficile per l'IA emulare l'essere umano e questo limiterà l'uso dell'IA in queste situazioni.

10.4 Considerazioni Etiche

Ora tratteremo un enorme argomento, un argomento che diventerà ancora più grande nei prossimi anni e decenni, ovvero il lato etico dell'utilizzo dell'intelligenza artificiale e del machine learning.

Iniziamo con esaminare alcune considerazioni prese dallo studio di Thomas et al. (2021), tale studio mostra che abbiamo rischi etici su molti livelli quando si usa l'IA ed il ML, ad esempio:

- **Data**

- dati di addestramento devono essere privi di **pregiudizi** (bias) e dovrebbero includere tutti i tipi di stimoli pertinenti, ad esempio i pregiudizi sessisti e razziali dovrebbero essere esclusi. Questa è una parte molto importante che bisogna considerare quando si addestrano i modelli di ML.
- inoltre, l'IA dovrebbe essere implementata solo in ambienti in cui è stata addestrata, ovvero non dovremmo avere alcun utilizzo incrociato per uno scopo diverso, ciò potrebbe essere problematico soprattutto nel settore finanziario, se si pensa a modelli che sono stati addestrati con un certo tipo di dati e poi vengono utilizzati in un contesto diverso dalla provenienza dei dati usati per l'addestramento si dice che i modelli hanno preso in "prestito" tali dati, ebbene in tali casi si potrebbero avere problemi con la **privacy dei dati**, dato che la privacy dei dati deve essere rispettata durante la raccolta e l'elaborazione dei dati.

- **Algoritmi**

- una codifica non etica potrebbe essere introdotta dai programmati e gli sviluppatori stessi, la codifica invece deve essere priva di pregiudizi. Ciò non significa necessariamente che un algoritmo diventi non etico a causa della selezione dei dati di addestramento, ma che gli sviluppatori stessi impostino un modello inserendo nella codifica alcuni pregiudizi.
- l'IA deve essere controllata dai pregiudizi emergenti durante l'intero ciclo di vita.

- **Uso Aziendale**

- lo scopo dell'utilizzo dell'IA deve essere etico, ad esempio non deve discriminare alcune parti della popolazione.
- l'impatto non intenzionale dell'IA può essere anti-etico.

10.4.1 Razzismo e sessismo nell'IA

Le macchine ed i modelli non hanno di per sé pregiudizi, ma i set di dati di apprendimento e gli algoritmi sono molto probabilmente distorti come la nostra società attuale. La maggior parte degli sviluppatori di intelligenza artificiale sono solitamente maschi bianchi a cui mancano le prospettive delle minoranze e di conseguenza c'è la possibilità che vengano introdotti pregiudizi nelle macchine e negli algoritmi e possono essere amplificati se non lavoriamo attivamente contro di loro.

Negli Stati Uniti ci sono organizzazioni che lavorano in questo senso, ad esempio, la ACLU (American Civil Liberties Union), ha combattuto in un'ampia varietà di questioni e hanno esposto il programma *Rekognition* di Amazon come razzista. Anche la AJI (Algorithmic Justice League), fondata nel 2016 con la missione di sensibilizzare l'opinione pubblica sui pregiudizi (bias) nell'intelligenza artificiale.

È una di queste organizzazioni che ha evidenziato i pericoli dell'utilizzo di algoritmi di intelligenza artificiale in situazioni in cui ci sono interazioni umane.

10.4.2 Raccomandazioni pratiche

Lo studio di Thomas et al. (2021) ha anche fornito alcune raccomandazioni pratiche, ad esempio:

- si dovrebbe implementare una dichiarazione generale sulle aziende e sull'intenzione dell'istituzione per l'IA etica.
- dovrebbe esserci un'estensione della missione o della dichiarazione di scopo già esistente nell'azienda.
- si dovrebbe implementare un piano di progettazione specifico dell'applicazione interna.
- si dovrebbe controllare regolarmente i processi in modo che se si osservano rischi etici e alcune preoccupazioni che questi programmi possano produrre decisioni non etiche, questi devono essere contrassegnati con il fine di "ripararli".
- si devono tenere dei registri delle decisioni riguardanti i compromessi etici, a volte potrebbe essere necessario fare un compromesso, ciò non significa che bisogna agire in modo non etico ma in alcune situazioni potrebbe essere necessario soppesare alcune decisioni riguardo alla privacy dei dati di addestramento rispetto agli scopi commerciali. Queste decisioni di compromessi potrebbero essere registrate per questioni di trasparenza.

10.4.3 Regolamentazione dell'IA e del ML in finanza

La regolamentazione dell'IA da una prospettiva etica è ancora nella sua infanzia.

La Commissione Europea ha dichiarato la sua strategia per la finanza digitale che intende riunirsi con le autorità di vigilanza europee entro il 2024, si tratterà se e come la regolamentazione dei mercati finanziari esistente debba essere applicata all'uso dei Big Data e all'intelligenza artificiale.

La definizione dell'intelligenza artificiale secondo **BaFin** è la seguente:

L'intelligenza artificiale è una combinazione di grandi quantità di dati (Big Data), risorse computazionali e apprendimento automatico (ML). Nell'apprendimento automatico, viene data ai computer la capacità di apprendere dai dati e dall'esperienza sulla base di speciali algoritmi. Rispetto ai metodi basati su regole, l'apprendimento avviene senza che il programmatore specifichi quali risultati devono essere derivati da determinate costellazioni di dati ed in che modo.

Quindi i Big Data più apprendimento automatico e l'elevata potenza computazionale sono l'intelligenza artificiale nella definizione di BaFin.

Sempre secondo il BaFin Tedesco gli **Algoritmi** sono regole di azione che sono solitamente integrati in una programma per computer e risolvono un problema di ottimizzazione o una classe di problemi. In aggiunta alla distinzione in base al tipo di algoritmi (come si risolve tecnicamente il problema) le applicazioni di machine learning possono essere differenziate anche in base ai tipi di risultato (la distinzione base tra classificazione, regressione, e clustering) ed in base ai tipi di dati (e.g. testo, linguaggio, e image data).

Principi Prioritari

Nel BaFin hanno anche definito alcuni principi prioritari:

- **Responsabilità del top management**

- Il management è il responsabile delle strategie e delle linee guida o politiche a livello aziendale per l'utilizzo di processi decisionali basati su algoritmi.
- Le potenzialità di tali processi così come i loro limiti e rischi dovrebbero essere presi in considerazione e chiaramente dichiarati.
- Una strategia a livello aziendale per l'uso di processi decisionali basati su algoritmi dovrebbe anche essere riflessa nella strategia IT.

- **Adeguata gestione del rischio e dell'outsourcing**

- Gli istituti finanziari dovrebbero stabilire la gestione del rischio del sistema adattato all'uso dei processi decisionali basati su algoritmi.
- Se le applicazioni provengono da un fornitore di servizi la gestione deve anche avere una gestione efficace dell'outsourcing. Se si utilizzano l'IA ed il ML ed il loro utilizzo viene esternalizzato il loro utilizzo, allora bisogna anche mantenere traccia del nostro fornitore di servizi.
- Responsabilità, strutture di controllo e di segnalazione devono essere chiaramente definite.
- Quando si stabilisce un'adeguata gestione del rischio è necessario considerare i rischi di un processo decisionale algoritmico. Si tratta di misure di mitigazione del rischio ed i processi dovrebbero iniziare esattamente dove il rischio ha origine secondo il principio del polluter pays.

- **Prevenzione dei pregiudizi (Bias)**

- Evitare un bias, ovvero una distorsione sistematica dei risultati nei processi decisionali basati su algoritmi.
- I bias devono essere evitati per poter prendere decisioni aziendali basate su risultati che non sono sistematicamente distorti ed escludono la possibilità sistematica basata su pregiudizi nei confronti di determinati gruppi di clienti e quindi evitare anche rischi reputazionali delle società che usano la tecnologia IA.
- In conformità con il polluter pays principle, il rischio deve essere identificato dove può sorgere e deve essere analizzato ed eliminato o almeno mitigato.

- **Escludere la differenziazione vietata dalla legge**

- Per alcuni servizi finanziari è effettivamente stabilito per legge che alcune caratteristiche non possono essere utilizzate per la differenziazione, cioè per il calcolo del rischio e dei prezzi.
- Il pericolo di discriminazione esiste se queste caratteristiche sono sostituite da un'approssimazione, cioè se invece di usare il termine genere o etnia utilizzassimo gruppi di età e di reddito della città natale, alla fine potrebbe portare all'effetto che gli algoritmi sostituiscano semplicemente una caratteristica con altre tre caratteristiche che sono correlate.
- Sarebbe associato a maggiori rischi reputazionali e anche a rischi legali, quindi è nella migliore delle ipotesi interesse di un istituto finanziario impedire che ciò accada effettivamente.
- Le compagnie dovrebbero stabilire processi di verifica statistica che escludano la discriminazione e tale sostituzione delle caratteristiche all'interno dei processi IA e ML.

10.4.4 Etica dell'intelligenza artificiale in finanza

Possiamo trovare qualche informazione in più circa le questioni etiche dell'IA nella finanza da EIOPA l'autorità di vigilanza sulle assicurazioni e sui fondi pensione dell'Unione Europea. La EIOPA ha un gruppo sull'etica digitale (GDE) e hanno anche formulato i principi di governance dell'IA suddivisi in:

- Supervisione umana
- Solidità e performance

- Governance dei dati
- Trasparenza e spiegabilità
- Equità e non discriminazione
- Principio di proporzionalità

Supervisione umana

Le compagnie di assicurazione e le istituzioni finanziarie europee dovrebbero stabilire livelli adeguati di supervisione umana tenendo conto dell'impatto degli specifici casi d'uso dell'IA e altre misure di governance e controllo in atto. Dovrebbero selezionare il livello di supervisione umana e la selezione dovrebbe essere proporzionata alla natura, all'entità e alla complessità del rischio inherente allo specifico caso d'uso dell'IA in una certa compagnia di assicurazioni. Diversi ruoli e responsabilità per il personale coinvolto nei processi di intelligenza artificiale dovrebbero essere chiaramente definite nei documenti di policy.

Solidità e performance

L'azienda dovrebbe valutare e monitorare le prestazioni dei sistemi di intelligenza artificiale su base continuativa e tenere in considerazione i loro limiti e delle potenziali carenze.

Le metriche delle prestazioni dovrebbero essere adattate al perseguimento dell'obiettivo e alla natura dei dati utilizzati, bisognerebbe verificare se si stanno effettivamente raggiungendo gli obiettivi che si sono prefissati.

Una buona gestione dei dati è la chiave per garantire le prestazioni dei sistemi di IA. I sistemi di IA dovrebbero produrre risultati stabili nel tempo, altrimenti non ha molto senso dal punto di vista aziendale. Le compagnie assicurative dovrebbero sviluppare sistemi e infrastrutture IT resilienti che non possono essere manomessi.

Data governance

Le compagnie di assicurazione dovrebbero adattarsi alle misure di data governance per l'impatto di casi d'uso specifici dell'IA. I dati utilizzati nei modelli di IA dovrebbero essere accurati, completi e adeguati. Una certa data governance dovrebbe essere applicata durante l'intero ciclo di vita del modello di IA, i dati utilizzati nei modelli di IA dovrebbero essere gestiti e archiviati in modo sicuro, poiché di solito, specialmente nelle assicurazioni, si tratta di dati molto sensibili e riservati dei clienti, devono essere conservate registrazioni

appropriate dei dati e dei metodi di modellazione per garantire la riproduzione e la tracciabilità.

Trasparenza e spiegabilità

L'azienda dovrebbe adattare i tipi di spiegazioni a casi d'uso specifici dell'IA e alle parti destinararie interessate. Le aziende quindi devono adattare le loro spiegazioni ai diversi tipi di stakeholders e devono sforzarsi di utilizzare modelli di IA spiegabili, in particolare nei casi d'uso dell'IA ad alto impatto.

I dati utilizzati devono essere comunicati in modo trasparente e di conseguenza abbiamo bisogno della sicurezza dei data governance e gestione dei dati sensibili.

Equità e non discriminazione

Processi di governance solidi e trasparenti sono fondamentali per garantire equità e non discriminazione, specialmente quando si tratta del calcolo del premio assicurativo, altrimenti ciò potrebbe portare a rischi reputazionali. Le compagnie assicurative dovrebbero condurre la propria attività in modo equo quando utilizzano l'IA, e compiere sforzi ragionevoli per tenere conto dei risultati dei sistemi di intelligenza artificiale. I consumatori che non sono disposti a condividere dati personali e sensibili non strettamente necessari per le valutazioni del rischio dovrebbero avere accesso a una copertura assicurativa a prezzi accessibili. Le imprese dovrebbero rispettare il principio dell'autonomia umana sviluppando sistemi di IA che supportino i consumatori nel loro processo decisionale ed evitino pratiche sleali introdotte dall'uso di metodi di IA.

Principio di proporzionalità

Le compagnie di assicurazione dovrebbero stabilire la governance necessaria, le misure proporzionate alla natura, alla portata, e alla complessità delle loro operazioni.

La valutazione dell'impatto dei casi d'uso dell'IA e le misure di governance dovrebbero essere proporzionali al potenziale impatto di uno specifico caso d'uso di IA sui consumatori o sulle imprese.

Le compagnie di assicurazione dovrebbero quindi valutare la combinazione di misure messe in atto per garantire un uso etico e affidabile dell'IA.

Questi sono i principi stabiliti dal gruppo GDE di EIOPA sull'etica digitale e l'etica IA nelle assicurazioni ma che possono essere applicate anche ad altre istituzioni finanziarie. Questo sarà un argomento enorme negli anni a venire

con più applicazioni, con più modelli e un'utilità ancora maggiore derivante dall'uso dell'IA e del ML in finanza.

Capitolo 11

Competizione basata sull'intelligenza artificiale

"Le società di servizi finanziari si stanno appassionando all'intelligenza artificiale, utilizzandola per automatizzare attività umili, analizzare dati, migliorare il servizio clienti e rispettare le normative."

- Nick Huber (2020)

Questo capitolo affronta argomenti relativi alla concorrenza nel settore finanziario sulla base dell'applicazione sistematica e strategica dell'IA.

11.1 Istruzione e formazione

L'ingresso nel campo della finanza e dell'industria finanziaria avviene abbastanza spesso attraverso un'istruzione formale sul campo. Le lauree tipiche hanno nomi come i seguenti:

- Master of Finance
- Master of Quantitative Finance
- Master of Computational Finance
- Master of Financial Engineering
- Master of Quantitative Enterprise Risk Management

In sostanza, tutte queste lauree oggi richiedono agli studenti di padroneggiare almeno un linguaggio di programmazione, spesso Python, per soddisfare

i requisiti di elaborazione dei dati della finanza basata sui dati. A questo proposito, le università rispondono alla domanda di queste competenze da parte dell'industria. Murray (2019) sottolinea:

La forza lavoro dovrà adattarsi poiché le aziende utilizzano l'intelligenza artificiale per più attività.

Non sono solo le università ad adeguare i loro programmi di studio in corsi di laurea in finanza per includere programmazione, scienza dei dati e intelligenza artificiale. Le aziende stesse investono anche molto in programmi di formazione per il personale nuovo ed esistente per essere pronto per la finanza basata sui dati e basata sull'intelligenza artificiale. Noonan (2018) descrive gli sforzi di formazione su larga scala di JPMorgan Chase, una delle più grandi banche del mondo, come segue:

JPMorgan Chase sta sottoponendo centinaia di nuovi banchieri d'investimento e gestori patrimoniali a lezioni obbligatorie di codifica, in segno dell'accresciuta necessità di competenze tecnologiche di Wall Street. La formazione sulla codifica per i ragazzi di quest'anno si è basata sulla programmazione Python, che li aiuterà ad analizzare set di dati molto grandi e interpretare dati non strutturati come il testo in linguaggio libero. Il prossimo anno, la divisione di asset management amplierà la formazione tecnica obbligatoria per includere concetti di data science, machine learning e cloud computing.

In sintesi, sempre più ruoli nel settore finanziario richiederanno personale esperto in programmazione, concetti di base e avanzati di scienza dei dati, apprendimento automatico e altri aspetti tecnici, come il cloud computing. Le università e gli istituti finanziari, sia sul lato dell'acquisto che su quello della vendita, reagiscono a questa tendenza rispettivamente adeguando i loro programmi di studio e investendo pesantemente nella formazione della propria forza lavoro. In entrambi i casi, si tratta di competere in modo efficace, o anche di rimanere rilevanti e riuscire a sopravvivere, in un panorama finanziario cambiato per sempre dalla crescente importanza dell'intelligenza artificiale.

11.2 Lotta per le risorse

Nella ricerca di utilizzare l'intelligenza artificiale in modo scalabile e significativo nella finanza, gli attori dei mercati finanziari competono per le risorse migliori. Quattro risorse principali sono di fondamentale importanza: risorse umane, algoritmi, dati e hardware.

Probabilmente la risorsa più importante e, allo stesso tempo, più scarsa sono gli esperti di IA in generale e di IA per la finanza in particolare. A questo proposito, le istituzioni finanziarie competono con società tecnologiche, startup

di tecnologia finanziaria (fintech) e altri gruppi per i migliori talenti. Sebbene le banche siano generalmente disposte a pagare stipendi relativamente alti a tali esperti, gli aspetti culturali delle società tecnologiche e, ad esempio, la promessa di stock option nelle startup potrebbero rendere loro difficile attrarre i migliori talenti. Spesso le istituzioni finanziarie ricorrono a coltivare i talenti internamente.

Molti algoritmi e modelli in machine e deep learning possono essere considerati algoritmi standard ben studiati, testati e documentati. In molti casi, tuttavia, non è chiaro fin dall'inizio come applicarli al meglio in un contesto finanziario. È qui che le istituzioni finanziarie investono molto negli sforzi di ricerca. Per molte delle più grandi istituzioni buy-side, come gli hedge fund sistematici, la ricerca sulle strategie di investimento e trading è al centro dei loro modelli di business. Tuttavia, la distribuzione e la produzione sono di pari importanza. Sia la ricerca che l'implementazione della strategia sono, ovviamente, discipline altamente tecniche in questo contesto.

Gli algoritmi senza dati sono spesso inutili. Allo stesso modo, gli algoritmi con dati "standard" provenienti da fonti di dati tipiche, come scambi o fornitori di servizi dati come Refinitiv o Bloomberg, potrebbero avere solo un valore limitato. Ciò è dovuto al fatto che tali dati vengono analizzati intensamente da molti, se non da tutti, i principali attori del mercato, rendendo difficile o addirittura impossibile identificare opportunità di generazione di alpha o vantaggi competitivi simili. Di conseguenza, le grandi istituzioni buy-side investono molto pesantemente per ottenere l'accesso a dati alternativi.

L'importanza dei dati alternativi al giorno d'oggi si riflette negli investimenti che gli attori buy-side e altri investitori fanno in società attive nel settore. Ad esempio, nel 2018 un gruppo di società di investimento ha investito 95 milioni di dollari nel gruppo di dati Enigma. Fortado (2018) descrive l'accordo e la sua logica come segue:

Hedge fund, banche e società di capitali di rischio si stanno accumulando in investimenti in società di dati nella speranza di incassare un'attività che stanno utilizzando molto di più. Negli ultimi anni, c'è stata una proliferazione di start-up che setacciano risme di dati e li vendono a gruppi di investimento alla ricerca di un vantaggio. L'ultima ad attirare l'interesse degli investitori è Enigma, una start-up con sede a New York che ha ricevuto finanziamenti da fonti tra cui il gigante quantistico Two Sigma, l'hedge fund attivista Third Point e le società di venture capital NEA e Glynn Capital in una raccolta di capitale da 95 milioni di dollari annunciata martedì.

La quarta risorsa per cui le istituzioni finanziarie competono sono le migliori opzioni hardware per elaborare grandi quantità di dati finanziari, implementa-

re gli algoritmi basati su set di dati tradizionali e alternativi e quindi applicare l'intelligenza artificiale in modo efficiente alla finanza. Gli ultimi anni hanno visto un'enorme innovazione nell'hardware dedicato a rendere gli sforzi di machine e deep learning più veloci, più efficienti dal punto di vista energetico e più convenienti. Mentre i processori tradizionali, come le CPU, svolgono un ruolo minore nel campo, hardware specializzato come GPU di Nvidia o opzioni più recenti come TPU di Google e IPU della startup Graphcore hanno preso il sopravvento in IA. L'interesse delle istituzioni finanziarie per il nuovo hardware specializzato si riflette, ad esempio, negli sforzi di ricerca di Citadel, uno dei più grandi hedge fund e market maker, nelle IPU. I suoi sforzi sono documentati nel rapporto di ricerca completo Jia et al. (2019), che illustra i potenziali vantaggi dell'hardware specializzato rispetto alle opzioni alternative.

Nella corsa al dominio nella finanza AI-first, le istituzioni finanziarie investono miliardi all'anno in talento, ricerca, dati e hardware. Mentre le grandi istituzioni sembrano ben posizionate per stare al passo con il ritmo nel settore, i player di piccole o medie dimensioni troveranno difficile passare in modo completo a un approccio basato sull'intelligenza artificiale per la loro attività.

11.3 Impatto sul mercato

L'uso crescente e ormai diffuso di algoritmi di data science, machine learning e deep learning nel settore finanziario ha senza dubbio un impatto sui mercati finanziari, sugli investimenti e sulle opportunità di trading. I metodi ML e DL sono in grado di scoprire inefficienze statistiche e persino inefficienze economiche che non sono rilevabili dai metodi econometrici tradizionali, come la regressione OLS multivariata. È quindi da presumere che nuovi e migliori metodi di analisi rendano più difficile scoprire opportunità e strategie che generano alpha.

Confrontando la situazione attuale dei mercati finanziari con quella dell'estrazione dell'oro, Lopéz de Prado (2018) descrive la situazione come segue:

Se un decennio fa era relativamente comune per un individuo scoprire l'alfa macroscopico (vale a dire, utilizzando semplici strumenti matematici come l'econometria), attualmente le possibilità che ciò accada stanno rapidamente convergendo a zero. Gli individui che oggi cercano l'alfa macroscopico, indipendentemente dalla loro esperienza o conoscenza, stanno combattendo contro probabilità schiaccianti. L'unico vero alfa rimasto è microscopico e trovarlo richiede metodi industriali ad alta intensità di capitale. Proprio come con l'oro, l'alfa microscopico non significa minori profitti complessivi. L'alfa microscopico oggi è molto più abbondante di quanto l'alfa macroscopico sia mai stato

nella storia. Ci sono un sacco di soldi da guadagnare, ma dovrai usare pesanti strumenti ML.

In questo contesto, le istituzioni finanziarie sembrano quasi obbligate ad abbracciare la finanza AI-first per non essere lasciate indietro e alla fine forse anche fallire. Ciò vale non solo per gli investimenti e il trading, ma anche per altri settori. Mentre le banche storicamente hanno coltivato relazioni a lungo termine con debitori commerciali e al dettaglio e hanno costruito organicamente la loro capacità di prendere decisioni affidabili in materia di credito, l'IA oggi livella il campo di gioco e rende le relazioni a lungo termine quasi prive di valore. Pertanto, i nuovi entranti nel settore, come le startup fintech, che si affidano all'intelligenza artificiale possono spesso acquisire rapidamente quote di mercato dagli operatori storici in modo controllato e fattibile. D'altra parte, questi sviluppi incentivano gli operatori storici ad acquisire e fondere startup fintech più giovani e innovative per rimanere competitivi.

11.4 Scenari competitivi

Guardando avanti, diciamo, da tre a cinque anni, come potrebbe apparire il panorama competitivo guidato dalla finanza AI-first? Sono possibili tre scenari:

- **Monopolio**

Un istituto finanziario raggiunge una posizione dominante grazie a scoperte importanti e senza pari nell'applicazione dell'intelligenza artificiale, ad esempio, al trading algoritmico. Questa è, ad esempio, la situazione delle ricerche su Internet, dove Google detiene una quota di mercato globale di circa il 90

- **Oligopolio**

Un numero minore di istituzioni finanziarie è in grado di sfruttare la finanza AI-first per raggiungere posizioni di leadership. Un oligopolio è, ad esempio, presente anche nel settore degli hedge fund, in cui un piccolo numero di grandi attori domina il campo in termini di asset under management.

- **Competizione perfetta**

Tutti gli attori dei mercati finanziari beneficiano in modo simile dei progressi della finanza AI-first. Nessun singolo giocatore o gruppo di giocatori gode di vantaggi competitivi rispetto agli altri. Tecnologicamente

parlando, questo è paragonabile alla situazione degli scacchi al computer al giorno d'oggi. Un certo numero di programmi di scacchi, eseguiti su hardware standard come gli smartphone, sono significativamente più bravi a giocare a scacchi rispetto all'attuale campione del mondo.

È difficile prevedere quale scenario sia più probabile. Si possono trovare argomenti e descrivere possibili percorsi per tutti e tre. Ad esempio, un argomento a favore di un monopolio potrebbe essere che un importante passo avanti nel trading algoritmico, ad esempio, potrebbe portare a una sovraperformance rapida e significativa che aiuta ad accumulare più capitale attraverso reinvestimenti, nonché attraverso nuovi afflussi. Ciò a sua volta aumenta la tecnologia disponibile e il budget per la ricerca per proteggere il vantaggio competitivo e attrae talenti che altrimenti sarebbero difficili da conquistare. L'intero ciclo si autoalimenta e l'esempio di Google nella ricerca, in connessione con il core business della pubblicità online, è un buon esempio in questo contesto.

Allo stesso modo, ci sono buone ragioni per anticipare un oligopolio. Attualmente, è lecito ritenere che qualsiasi grande attore nel settore commerciale investa pesantemente in ricerca e tecnologia, con iniziative relative all'intelligenza artificiale che costituiscono una parte significativa del budget. Come in altri campi, ad esempio, i motori di raccomandazione, ad esempio Amazon per i libri, Netflix per i film e Spotify per la musica, più aziende potrebbero essere in grado di raggiungere progressi simili contemporaneamente. È concepibile che gli attuali principali trader sistematici saranno in grado di utilizzare la finanza AI-first per consolidare le loro posizioni di leadership.

Infine, molte tecnologie sono diventate onnipresenti nel corso degli anni. I forti programmi di scacchi sono solo un esempio. Altri potrebbero essere mappe e sistemi di navigazione o assistenti personali basati sulla voce. In uno scenario di concorrenza perfetta, un numero piuttosto elevato di attori finanziari competerebbe per minuscole opportunità di creazione di alfa o potrebbe addirittura non essere in grado di generare rendimenti distinguibili dai normali rendimenti di mercato.

Allo stesso tempo, ci sono argomenti contro i tre scenari. Il panorama attuale ha molti attori con mezzi e incentivi uguali per sfruttare l'IA nella finanza. Ciò rende improbabile che un solo giocatore si distingua e conquisti quote di mercato nella gestione degli investimenti paragonabili a Google nella ricerca. Allo stesso tempo, il numero di operatori di piccole, medie e grandi dimensioni che effettuano ricerche sul campo e le basse barriere all'ingresso nel trading algoritmico rendono improbabile che pochi eletti possano assicurarsi vantaggi competitivi difendibili. Un argomento contro la concorrenza perfetta è che, nel prossimo futuro, il trading algoritmico su larga scala

richiede un'enorme quantità di capitale e altre risorse. Per quanto riguarda gli scacchi, DeepMind ha dimostrato con AlphaZero che c'è sempre spazio per l'innovazione e miglioramenti significativi.

È difficile prevedere uno scenario finale competitivo per il settore finanziario in un momento in cui l'IA avrà preso il sopravvento. Gli scenari che vanno dal monopolio all'oligopolio alla concorrenza perfetta sembrano ancora ragionevoli. AI-first finance mette a confronto ricercatori, professionisti e autorità di regolamentazione con nuovi rischi e nuove sfide per affrontare questi rischi in modo appropriato. Uno di questi rischi, che gioca un ruolo di primo piano in molte discussioni, è la scatola nera (black-box) caratteristica di molti algoritmi di intelligenza artificiale. Un tale rischio di solito può essere mitigato solo in una certa misura con l'intelligenza artificiale spiegabile all'avanguardia di oggi.

Capitolo 12

Singolarità finanziaria

"Ci troviamo in un cespuglio di complessità strategica, circondati da una fitta nebbia di incertezza."

-Nick Bostrom (2014)

La concorrenza basata sull'intelligenza artificiale nel settore finanziario può portare a una singolarità finanziaria? Questa è la questione principale discussa in questo capitolo finale.

12.1 Nozioni e definizioni

L'espressione singolarità finanziaria risale almeno al post sul blog del 2015 di Shiller. In questo post, Shiller scrive: *L'alfa alla fine andrà a zero per ogni strategia di investimento immaginabile? Più fondamentalmente, si avvicina il giorno in cui, grazie a così tante persone intelligenti e computer più intelligenti, i mercati finanziari diventano davvero perfetti e possiamo semplicemente sederci, rilassarci e presumere che tutti gli asset abbiano un prezzo corretto? Questo stato di cose immaginato potrebbe essere chiamato la singolarità finanziaria, analoga all'ipotetica futura singolarità tecnologica, quando i computer sostituiranno l'intelligenza umana. La singolarità finanziaria implica che tutte le decisioni di investimento sarebbero meglio lasciate a un programma per computer, perché gli esperti con i loro algoritmi hanno capito cosa guida i risultati del mercato e lo hanno ridotto a un sistema senza soluzione di continuità.*

Un po' più in generale, si potrebbe definire la singolarità finanziaria come il momento in cui i computer e gli algoritmi cominciano a prendere il controllo della finanza e dell'intero settore finanziario, comprese le banche, i gestori patrimoniali, le borse e così via, con gli esseri umani che prendono un posto in secondo piano come manager, supervisori e controllori.

Per capire meglio di cosa si andrà a discutere a breve, facciamo un breve excursus sulle forme di IA.

12.1.1 Forme di Intelligenza

È difficile dirlo senza una definizione specifica di intelligenza . Il ricercatore di intelligenza artificiale Max Tegmark (2017) definisce sinteticamente l'intelligenza come "la capacità di raggiungere obiettivi complessi".

Questa definizione è abbastanza generale da comprendere definizioni più specifiche. AlphaZero è intelligente data questa definizione poiché è in grado di raggiungere un obiettivo complesso, vale a dire vincere partite di Go o scacchi contro giocatori umani o altri agenti di intelligenza artificiale. Naturalmente anche gli esseri umani e gli animali in generale sono considerati intelligenti.

Le seguenti definizioni più specifiche sembrano appropriate e abbastanza precise.

- **Intelligenza Artificiale Ristretta (ANI)**

Questa specifica un agente di intelligenza artificiale che supera le capacità e le abilità a livello di esperto umano in un campo ristretto. AlphaZero può essere considerato una ANI nei campi del Go, degli scacchi e dello shogi. Un agente AI algoritmico di trading azionario che realizza un rendimento netto costantemente del 100% all'anno (per anno) sul capitale investito potrebbe essere considerato una ANI.

- **Intelligenza artificiale generale (AGI)**

Questa specifica un agente di intelligenza artificiale che raggiunge l'intelligenza a livello umano in qualsiasi campo, come scacchi, matematica, composizione del testo o finanza, e potrebbe superare l'intelligenza a livello umano in alcuni altri domini.

- **Superintelligenza (SI)**

Questa specifica un intelletto o un agente di intelligenza artificiale che supera l'intelligenza a livello umano sotto ogni aspetto.

Una ANI ha la capacità di raggiungere un obiettivo complesso in un campo ristretto a un livello superiore a qualsiasi essere umano. Una AGI è altrettanto valido come qualsiasi essere umano nel raggiungere obiettivi complessi in un'ampia varietà di campi. Infine, una superintelligenza è significativamente migliore di qualsiasi essere umano, o anche di un collettivo di esseri umani, nel raggiungere obiettivi complessi in quasi tutti i campi immaginabili.

La precedente definizione di superintelligenza è in linea con quella fornita da Nick Bostrom nel suo libro intitolato *Superintelligence* (2014):

Possiamo provvisoriamente definire una superintelligenza come qualsiasi intelletto che superi di gran lunga le prestazioni cognitive degli esseri umani praticamente in tutti i domini di interesse.

Ora che conosciamo le varie forme di Intelligenza Artificiale possiamo definire la singolarità finanziaria come il momento in cui esiste un bot di trading che mostra una capacità costante di prevedere i movimenti nei mercati finanziari a livelli sovrumani e superistituzionali, così come con una precisione senza precedenti. In tal senso, un tale trading bot sarebbe caratterizzato come un'intelligenza artificiale ristretta (ANI) invece di un'intelligenza artificiale generale (AGI) o di una superintelligenza.

Si può presumere che sia molto più facile costruire un tale AFI sotto forma di un trading bot piuttosto che un AGI o addirittura una superintelligenza. Ciò vale allo stesso modo per AlphaZero, poiché è più facile costruire un agente di intelligenza artificiale che sia superiore a qualsiasi essere umano o qualsiasi altro agente nel gioco del Go. Pertanto, anche se non è ancora chiaro se esisterà mai un agente AI qualificabile come AGI o superintelligenza, è comunque molto più probabile che emerga un trading bot qualificabile come ANI o AFI.

12.2 Qual'è il rischio?

La ricerca di una ANI potrebbe essere stimolante ed eccitante di per sé. Tuttavia, come di consueto in finanza, non molte iniziative sono guidate da motivazioni altruistiche; piuttosto, la maggior parte è guidata dagli incentivi finanziari (cioè denaro contante). Ma qual è esattamente la posta in gioco nella corsa alla costruzione di una ANI? Non si può rispondere con certezza o generalità, ma alcuni semplici calcoli possono far luce sulla questione.

Per capire quanto sia prezioso avere un ANI rispetto a strategie di trading inferiori, considera i seguenti benchmark:

- Strategia del toro

Una strategia di trading che va long solo su uno strumento finanziario in attesa di un aumento dei prezzi.

- Strategia casuale

Una strategia di trading che sceglie casualmente una posizione long o short per un dato strumento finanziario.

- Strategia dell'orso

Una strategia di trading che va short solo su uno strumento finanziario in previsione di un calo dei prezzi.

Queste strategie di riferimento devono essere confrontate con una ANI con le seguenti caratteristiche di successo:

- X% top

L'ANI azzecca i movimenti al rialzo e al ribasso del X%, mentre i restanti movimenti di mercato sono previsti in modo casuale.

- X% ANI

L'ANI azzecca il X% di tutti i movimenti di mercato scelti a caso, mentre i restanti movimenti di mercato vengono predetti in modo casuale.

Il seguente codice Python importa il set di dati di serie temporali noto con i dati EOD per una serie di strumenti finanziari. Gli esempi da seguire si basano su cinque anni di dati EOD per un singolo strumento finanziario:

```

1 url = 'https://hilpisch.com/aiif_eikon_eod_data.csv'
2
3 raw = pd.read_csv(url, index_col=0, parse_dates=True)
4
5 symbol = 'EUR='
6
7 #rendimenti del benchmark rialzista (solo long)
8 raw['bull'] = np.log(raw[symbol] / raw[symbol].shift(1))
9
10 data = pd.DataFrame(raw['bull']).loc['2015-01-01':]
11
12 data.dropna(inplace=True)
13
14 data.info()
15
16 DatetimeIndex: 1305 entries, 2015-01-01 to 2020-01-01
17 Data columns (total 1 columns):
18 # Column Non-Null Count Dtype
19 --- 
20 0 bull 1305 non-null float64

```

Con la strategia rialzista già definita dai rendimenti logaritmici dello strumento finanziario di base, il seguente codice Python specifica le altre due strategie di riferimento e ricava le performance per le strategie AFI. In questo contesto, vengono prese in considerazione una serie di strategie AFI per illustrare l'impatto dei miglioramenti nell'accuratezza delle previsioni dell'AFI:

```

1 np.random.seed(100)
2
3 #i rendimenti del benchmark casuale
4 data['random'] = np.random.choice([-1, 1], len(data)) * data['bull']
5
6 #i rendimenti del benchmark ribassista (solo short)
7 data['bear'] = -data['bull']
8
9 def top(t):
10    top = pd.DataFrame(data['bull'])
11    top.columns = ['top']
12    top = top.sort_values('top')
13    n = int(len(data) * t)
14    top['top'].iloc[:n] = abs(top['top'].iloc[:n])
15    top['top'].iloc[n:] = abs(top['top'].iloc[n:])
16    top['top'].iloc[n:-n] = np.random.choice([-1, 1],
17        len(top['top'].iloc[n:-n])) * top['top'].iloc[n:-n]
18    data[f'int(t_*_100)'].top] = top.sort_index()
19
20 #i rendimenti della strategia X% top
21 for t in [0.1, 0.15]:
22    top(t)
23
24 def afi(ratio):
25    correct = np.random.binomial(1, ratio, len(data))
26    random = np.random.choice([-1, 1], len(data))
27    strat = np.where(correct, abs(data['bull']), random * data['bull'])
28    data[f'int(ratio_*_100)'].afi] = strat
29
30 #Il rendimento della strategia X% ANI
31 for ratio in [0.51, 0.6, 0.75, 0.9]:
32    afi(ratio)

```

Utilizzando l'approccio standardizzato di backtesting vettoriale, come introdotto nel Capitolo 6 (trascurando i costi di transazione), diventa chiaro quali aumenti significativi nell'accuratezza della previsione implicano in termini finanziari. Consideriamo il "90% ANI", che non è perfetto nelle sue previsioni, ma manca piuttosto di qualsiasi vantaggio nel 10% di tutti i casi. L'accuratezza presunta del 90% porta a una performance linda che in cinque anni restituisce quasi 100 volte il capitale investito (al lordo dei costi di transazione). Con una precisione del 75%, l'ANI restituirebbe comunque quasi 50 volte il capitale investito (vedi Figura 12.1). Ciò esclude la leva finanziaria, che può essere facilmente aggiunta in modo quasi privo di rischi in presenza di

tali accuratezze di previsione:

```
1 data.head()
```

Date	bull	random	bear	10_top	15_top	51_afi	60_afi	75_afi	90_afi
2015-01-01	0.000413	-0.000413	-0.000413	0.000413	-0.000413	0.000413	0.000413	0.000413	0.000413
2015-01-02	-0.008464	0.008464	0.008464	0.008464	0.008464	0.008464	0.008464	0.008464	0.008464
2015-01-05	-0.005767	-0.005767	0.005767	-0.005767	0.005767	-0.005767	0.005767	-0.005767	0.005767
2015-01-06	-0.003611	-0.003611	0.003611	-0.003611	0.003611	0.003611	0.003611	0.003611	0.003611
2015-01-07	-0.004299	-0.004299	0.004299	0.004299	0.004299	0.004299	0.004299	0.004299	0.004299

```
1 data.sum().apply(np.exp)
2 bull          0.926676
3 random        1.097137
4 bear          1.079126
5 10_top        9.815383
6 15_top        21.275448
7 51_afi        12.272497
8 60_afi        22.103642
9 75_afi        49.227314
10 90_afi       98.176658
```

Le analisi mostrano che la posta in gioco è piuttosto alta, anche se ovviamente vengono fatte diverse ipotesi semplificative. Il tempo gioca un ruolo importante in questo contesto. Reimplementare le stesse analisi su un periodo di 10 anni rende i numeri ancora più impressionanti, quasi inimmaginabili in un contesto commerciale. Come illustra il seguente output per "90% ANI", il rendimento lordo sarebbe più di 16.000 volte il capitale investito (prima dei costi di transazione). L'effetto della capitalizzazione e del reinvestimento è enorme:

```
1 bull          0.782657
2 random        0.800253
3 bear          1.277698
4 10_top        165.066583
5 15_top        1026.275100
6 51_afi        206.639897
7 60_afi        691.751006
8 75_afi        2947.811043
9 90_afi       16581.526533
```

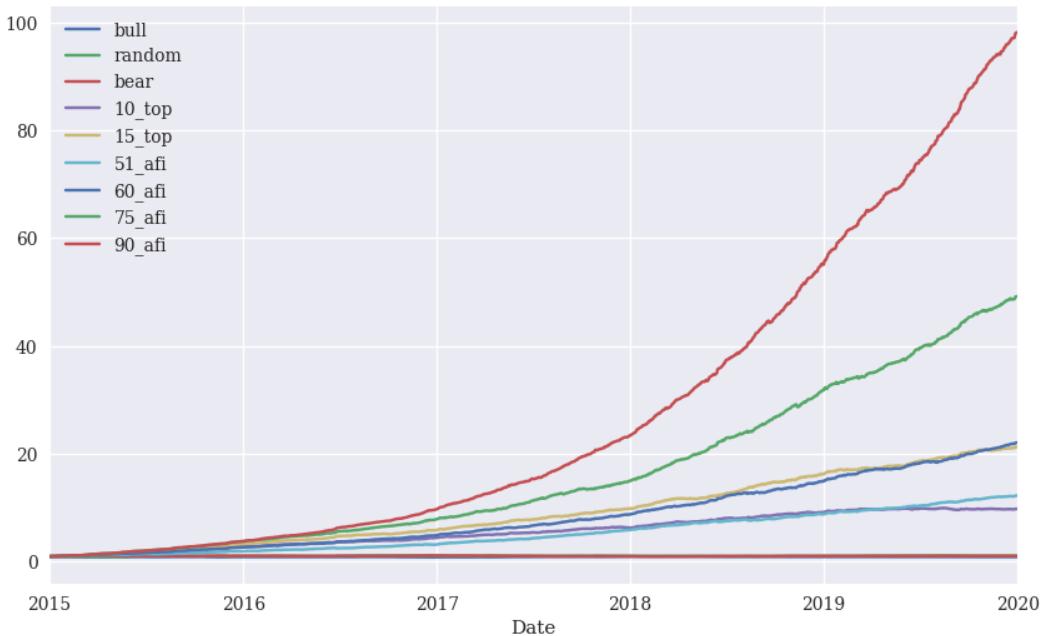


Figura 12.1: Performance linda nel tempo delle strategie benchmark e teoriche ANI

12.3 Percorsi verso la singolarità finanziaria

L’emergere di una ANI sarebbe un evento piuttosto specifico in un ambiente piuttosto specifico. Ad esempio, non è necessario emulare il cervello di un essere umano poiché l’AGI o la superintelligenza non è l’obiettivo principale. Dato che non esiste un essere umano che sembra essere costantemente superiore nel trading nei mercati finanziari rispetto a tutti gli altri, potrebbe anche essere una strada senza uscita, cercando di emulare un cervello umano per arrivare ad una ANI. Inoltre, non è necessario preoccuparsi dell’incarnazione. Un ANI può vivere come software solo su un’infrastruttura appropriata che si connette ai dati richiesti e alle API di trading.

D’altra parte, l’intelligenza artificiale sembra essere un percorso promettente per una ANI a causa della natura stessa del problema: prendere come input grandi quantità di dati finanziari e di altro tipo e generare previsioni sulla direzione futura di un movimento di prezzo.

Un’altra opzione potrebbe essere un ibrido di intelligenza umana e macchina. Mentre le macchine hanno supportato i commercianti umani per decenni, in molti casi i ruoli sono cambiati. Gli esseri umani supportano le macchine nel trading fornendo l’ambiente ideale e dati aggiornati, intervenendo solo in

situazioni estreme e così via. In molti casi, le macchine sono già completamente autonome nelle loro decisioni di trading algoritmico. O come afferma Jim Simons, fondatore di Renaissance Technologies, uno degli hedge fund di trading sistematico di maggior successo e segreti: "L'unica regola è che non abbiamo mai l'override del computer".

Sebbene non sia abbastanza chiaro quali percorsi potrebbero portare alla superintelligenza, dal punto di vista odierno, sembra molto probabile che l'IA possa aprire la strada alla singolarità finanziaria e ad una ANI.

12.4 Scenari prima e dopo

È lecito ritenere che tutti i principali istituti finanziari di tutto il mondo e molte altre entità non finanziarie attualmente svolgano ricerche e abbiano esperienza pratica con l'IA applicata alla finanza. Tuttavia, non tutti gli attori del settore finanziario sono posizionati ugualmente bene per arrivare prima ad una ANI di trading. Alcuni, come le banche, sono piuttosto limitati dai requisiti normativi. Altri seguono semplicemente diversi modelli di business, come gli scambi. Altri, come alcuni gestori patrimoniali, si concentrano sulla fornitura di prodotti di investimento a basso costo e mercificati, come gli ETF, che imitano la performance di indici di mercato più ampi. In altre parole, generare alfa non è l'obiettivo principale di ogni istituto finanziario.

Da una prospettiva esterna, gli hedge fund più grandi sembrano quindi nella posizione migliore per ottenere il massimo dalla finanza basata sull'intelligenza artificiale e dal trading algoritmico basato sull'intelligenza artificiale. In generale, hanno già molte delle risorse richieste importanti in questo campo: persone di talento e ben istruite, esperienza con algoritmi di trading e accesso quasi illimitato a fonti di dati tradizionali e alternative, nonché un'infrastruttura di trading scalabile e professionale. Se manca qualcosa, i grandi budget tecnologici assicurano investimenti rapidi e mirati.

Non è chiaro se prima ci sarà una ANI con altre che verranno dopo o se potrebbero emergere più ANI contemporaneamente. Se sono presenti più ANI, si potrebbe parlare di uno scenario multipolare od oligopolistico. Le ANI o AFI probabilmente competerebbero per lo più l'una contro l'altra, con i giocatori "non AFI" messi da parte. Gli sponsor dei singoli progetti si sforzerebbero di ottenere vantaggi, anche piccoli, perché ciò potrebbe consentire a un'AFI di subentrare completamente e diventare finalmente un singolo o un monopolista.

È anche ipotizzabile che prevalga fin dall'inizio uno scenario del tipo "chi vince prende tutto". In uno scenario del genere emerge un'unica AFI in grado di raggiungere rapidamente un livello di dominio nel trading finanziario che non può essere eguagliato da nessun altro concorrente. Ciò potrebbe essere

dovuto a diversi motivi. Uno dei motivi potrebbe essere che la prima AFI genera rendimenti così impressionanti che gli asset in gestione si gonfiano a un ritmo enorme, portando a budget sempre più elevati che a loro volta gli consentono di acquisire risorse sempre più rilevanti. Un altro motivo potrebbe essere che la prima AFI raggiunge rapidamente una dimensione in cui le sue azioni possono avere un impatto sul mercato - con la capacità di manipolare i prezzi di mercato, ad esempio - in modo tale da diventare la principale, o addirittura l'unica, forza trainante nei mercati finanziari.

La regolamentazione potrebbe in teoria impedire ad una AFI di diventare troppo grande o di guadagnare troppo potere di mercato. Le domande principali sarebbero se tali leggi siano applicabili nella pratica e come esattamente dovrebbero essere progettate per avere gli effetti desiderati.

12.5 Star Trek o Star Wars

L'industria finanziaria, per molte persone, rappresenta la forma più pura di capitalismo: l'industria in cui l'avida guida tutto. È stato ed è sicuramente un settore altamente competitivo, su questo non si discute. Il trading e la gestione degli investimenti in particolare sono spesso simboleggiati da manager e proprietari miliardari che sono disposti a scommettere in grande e ad andare testa a testa con i loro rivali per ottenere il prossimo mega affare o scambio. L'avvento dell'IA fornisce ai manager ambiziosi un ricco set di strumenti per spingere la concorrenza al livello successivo.

Tuttavia, la domanda è se la finanza AI-first, che potrebbe culminare in una AFI, porterà all'utopia finanziaria o alla distopia. L'accumulazione sistematica e infallibile della ricchezza potrebbe teoricamente servire solo poche persone, o potrebbe potenzialmente servire l'umanità. Sfortunatamente, si presume che solo gli sponsor del progetto che porta ad una AFI beneficeranno direttamente del tipo di AFI immaginato in questo capitolo. Questo perché una tale AFI genererebbe profitti solo commerciando nei mercati finanziari e non inventando nuovi prodotti, risolvendo problemi importanti o facendo crescere imprese e industrie. In altre parole, una AFI che commercia nei mercati finanziari solo per generare profitti sta partecipando a un gioco a somma zero e non aumenta direttamente la ricchezza distribuibile.

Si potrebbe sostenere che, ad esempio, i fondi pensione che investono in un fondo gestito dall'AFI beneficierebbero anche dei suoi rendimenti eccezionali. Ma questo gioverebbe ancora una volta solo a un certo gruppo e non all'umanità nel suo insieme. Ci sarebbe anche da chiedersi se gli sponsor di un progetto AFI di successo sarebbero disposti ad aprirsi a investitori esterni. Un buon esempio in tal senso è il fondo Medallion, gestito da Renaissance Technologies

e uno dei veicoli di investimento più performanti della storia. Renaissance ha chiuso Medallion, che è essenzialmente gestito esclusivamente da macchine, a investitori esterni nel 1993. Le sue prestazioni stellari avrebbero sicuramente attirato grandi quantità di risorse aggiuntive. Tuttavia, considerazioni specifiche, come la capacità di determinate strategie, svolgono un ruolo in questo contesto e considerazioni simili potrebbero valere anche per una AFI.

Pertanto, mentre ci si potrebbe aspettare che una superintelligenza aiuti a superare i problemi fondamentali affrontati dall'umanità nel suo insieme - malattie gravi, problemi ambientali, minacce sconosciute provenienti dallo spazio e così via - un AFI molto probabilmente porta a una maggiore disuguaglianza e a una concorrenza più agguerrita nei mercati . Invece di un mondo simile a Star Trek , caratterizzato da uguaglianza e risorse inesauribili, non si può escludere che un AFI possa piuttosto portare a un mondo simile a Star Wars , caratterizzato da intense guerre commerciali e lotte per le risorse disponibili. Al momento, le guerre commerciali globali, come quella tra Stati Uniti e Cina, sembrano più intense che mai e la tecnologia e l'intelligenza artificiale sono importanti campi di battaglia.

12.6 Conclusioni

Anche se potrebbe rivelarsi impossibile creare un AFI come delineato in questo capitolo, l'introduzione sistematica dell'IA nella finanza stimolerà sicuramente l'innovazione e in molti casi intensificherà la concorrenza nel settore. Piuttosto che essere una moda passeggera, l'intelligenza artificiale è una tendenza che alla fine porterà a un cambio di paradigma nel settore.

Capitolo 13

Conclusioni

Abbiamo iniziato facendo un breve excursus sui concetti fondamentali e di base legati al campo finanziario per poi trasferirci a trattare le principali teorie normative finanziarie evidenziando come nella pratica queste teorie non diano risultati soddisfacenti. Quindi siamo entrati nel mondo della finanza guidata dai dati in cui grazie all'esplosione dei dati digitali e l'emergere dell'apprendimento automatico (ML) si da origine ad un approccio model-free ed incentrato sull'intelligenza artificiale come capacità strategica per le strategie di investimento e di trading.

Transitando verso la terza parte del presente elaborato, abbiamo compreso come scoprire e sfruttare le inefficienze del mercato attraverso l'utilizzo delle reti neurali.

Successivamente siamo passati a presentare una panoramica sulle tipologie di apprendimento automatico utili nel campo della finanza per poi passare a mostrare quali siano le principali applicazioni dell'Intelligenza Artificiale nel contesto finanziario esplicando due tra le più importanti fra queste applicazioni, ovvero servirsi del Natural Language Processing per l'analisi del sentimento e delle reti neurali convolutive per le immagini satellitari e le serie temporali finanziarie.

In via di conclusione, nell'ultima parte, sono state trattate tematiche di rilevante importanza come la regolamentazione delle nuove tecnologie IA-driven e le questioni etiche conseguenti alla diffusione dei Big Data e all'utilizzo sempre maggiore dell'IA da parte delle istituzioni finanziarie.

Abbiamo visto quali siano le tendenze globali del business e della tecnologia al di là della finanza, ed è molto più probabile che tali tendenze continuino anziché arrestarsi o invertire la rotta.

Molte società di investimento stanno appena iniziando a sfruttare la gamma di strumenti di intelligenza artificiale, così come i singoli individui stanno

acquisendo le competenze necessarie e i processi aziendali si stanno adattando a queste nuove opportunità per la creazione di valore.

All'orizzonte si profilano anche numerosi ed entusiasmanti sviluppi per l'applicazione dell'IA nella finanza che probabilmente alimenteranno l'attuale slancio. È probabile che diventino nei prossimi anni, tra cui l'automazione del processo di ML, la generazione di dati di formazione sintetici e l'emergere dell'informatica quantistica, le tecnologie più pervasive del nuovo secolo.

13.1 Ottimismo consapevole

Per via del rapido e pervasivo ingresso dell'IA nel mondo della finanza e non solo, sorgono naturalmente alcuni importanti problemi riguardanti alla sicurezza e all'utilizzo etico dell'IA. Tuttavia se saremo capaci di utilizzare l'IA in modo consapevole, etico e sicuro, essa potrebbe esserci grande aiuto per risolvere molti tra i più importanti problemi che l'umanità ed il nostro pianeta Terra presentano. Pertanto la seguente è una lista dei principi, tratti dal libro Vita 3.0 di Max Tegmark, sui quali la prosecuzione dello sviluppo dell'IA dovrebbe basarsi per far sì che essa possa offrire grandi opportunità per aiutare e stimolare le persone nei decenni e nei secoli a venire.

Problemi di ricerca

1. Scopo della ricerca: obiettivo della ricerca sull'IA deve essere creare non intelligenza senza orientamento, bensì intelligenza benefica.
2. Finanziamento della ricerca: gli investitori nell'IA devono essere accompagnati da finanziamenti per la ricerca volta a garantire il suo uso benefico, che affronti, tra le altre, domande spinose nell'ambito dell'informatica, dell'economia, del diritto, dell'etica e degli studi sociali; per esempio:
 - Come possiamo rendere molto robusti i futuri sistemi di IA, in modo che facciano quello che vogliamo senza malfunzionamenti o attacchi informatici?
 - Come possiamo aggiornare i nostri sistemi giuridici in modo che siano più equi ed efficaci, stiano al passo con l'IA e gestiscano i rischi associati all'IA?
 - Come possiamo far crescere la nostra prosperità per mezzo dell'automazione, mantenendo però le risorse e gli obiettivi delle persone?

- Con quale insieme di valori deve stare in linea l'IA e quale status legale ed etico deve avere?
3. Collegamento tra scienza e politica: deve esistere uno scambio costruttivo e sano fra ricercatori dell'IA e politici.
 4. Cultura della ricerca: fra i ricercatori e gli sviluppatori di IA deve essere promossa una cultura di cooperazione, fiducia e trasparenza.
 5. Evitare la competizione: le équipe che sviluppano sistemi di IA devono collaborare attivamente affinché non si corra il rischio di scorciatoie in merito agli standard di sicurezza.

Etica e valori

1. Sicurezza: i sistemi di IA devono essere sicuri, protetti per tutta la loro vita operativa e devono esserlo in modo verificabile, laddove siano applicabili e fattibili.
2. Trasparenza dei guasti: se un sistema di IA provoca danni, e deve essere possibile stabilirne il motivo.
3. Trasparenza giudiziaria: qualsiasi coinvolgimento di un sistema autonomo nelle decisioni giuridiche deve fornire una spiegazione soddisfacente, verificabile da un'autorità umana competente.
4. Responsabilità: progettisti e costruttori di sistemi di IA avanzati sono parti interessate per le conseguenze morali del loro uso, del loro abuso e delle loro azioni, con la responsabilità e l'opportunità di plasmare quelle conseguenze.
5. Allineamento dei valori: i sistemi di IA altamente autonomi devono essere progettati in modo che i loro fini e i loro comportamenti siano sicuramente in linea con i valori umani per tutto l'arco del loro esercizio.
6. Valori umani: i sistemi di IA devono essere progettati e gestiti in modo da essere compatibili con gli ideali di dignità umana, diritti, libertà e diversità culturale.
7. Privacy personale: le persone devono avere il diritto di accedere ai dati che generano, di gestirli e controllarli, dato il potere che hanno i sistemi di IA di analizzare e utilizzare quei dati.
8. Libertà e privacy: l'applicazione dell'IA a dati personali non deve limitare in modo irragionevole la libertà, reale o percepita delle persone.

9. Benefici condivisi: le tecnologie dell'IA devono andare a vantaggio del maggior numero possibile di persone e dare loro più potere.
10. Prosperità condivisa: la prosperità economica creata dall'IA deve essere ampiamente condivisa, a beneficio di tutta l'umanità.
11. Controllo umano: gli esseri umani devono scegliere se e come delegare decisioni a sistemi di IA, per raggiungere obiettivi scelti da esseri umani.
12. Non sovversione: il potere conferito dal controllo, di sistemi di IA altamente avanzati deve rispettare e migliorare, non sovertire i processi sociali e civici da cui dipende la salute della società.
13. Corsa agli armamenti con IA: deve essere evitata una corsa alle armi letali autonome.

Problemi a lungo termine

1. Attenzione alle capacità: non esistendo consenso, dobbiamo evitare ipotesi forti sui limiti massimi delle capacità di future IA.
2. Importanza: l'IA avanzata può rappresentare un cambiamento profondo nella storia della vita sulla Terra, che va pianificato e gestito con attenzione e risorse adeguate.
3. Rischi: i rischi associati ai sistemi di IA, in particolare quelli catastrofici o esistenziali, devono essere oggetto di pianificazione e sforzi di mitigazione commisurati al loro presunto impatto.
4. Automiglioramento ricorsivo: i sistemi di IA progettati per automigliorarsi ricorsivamente o autoreplicarsi in modo che potrebbe condurre a un rapido aumento di qualità o quantità devono andare soggetti a severe misure di sicurezza e controllo.
5. Bene comune: la superintelligenza deve essere sviluppata solo al servizio di ideali etici ampiamente condivisi e a vantaggio dell'intera umanità, non di uno Stato o di un'organizzazione.

Come abbiamo visto anche negli ultimi capitoli della presente tesi, probabilmente l'IA ci presenterà sia grandiose opportunità, sia sfide difficili. Una strategia che forse può aiutarci sostanzialmente in tutte le sfide dell'IA è agire insieme e migliorare la nostra società umana **prima** che l'IA decolla a pieno.

Faremo meglio a educare i nostri figli ed i giovani perché rendano la tecnologia solida e benefica prima di cederle un grande potere. Faremo meglio a modernizzare le nostre leggi prima che la tecnologia le renda obsolete. Faremo meglio a risolvere i conflitti internazionali prima che si trasformino in una corsa agli armamenti con le armi autonome. Faremo meglio a creare un'economia che garantisca la prosperità di tutti, prima che l'IA possa aumentare le disuguaglianze. Staremo meglio in una società in cui i risultati delle ricerche sulla sicurezza dell'IA vengono messi in pratica anziché ignorati. Guardando ancora più avanti, alle sfide legate a un'IAG superumana, faremo meglio ad accordarci almeno su alcuni standard etici di base, prima di iniziare a insegnare quegli standard a macchine potenti.

Giunti ormai alla fine dell'elaborato, desidero concludere con le parole di uno dei massimi esponenti dell'IA: Max Tegmark, che recitano come segue:

In un mondo polarizzato e caotico, chi avrà il potere di usare l'IA per fini malvagi avrà più motivazioni e migliori possibilità di farlo, e le squadre che concorrono per costruire l'IAG saranno sottoposte a una pressione maggiore affinchè non prendano precauzioni sulla sicurezza, piuttosto che cooperare. In breve, se possiamo creare una società umana più armoniosa, caratterizzata dalla collaborazione in vista di fini condivisi, questo aumenterà le possibilità che la rivoluzione dell'IA si concluda bene. In altre parole, uno dei modi migliori che avete per migliorare il futuro della vita è migliorare il domani. Avete il potere di farlo in molti modi. Ovviamente potete votare e dire ai vostri politici quello che pensate su istruzione, privacy, armi letali autonome, disoccupazione tecnologica e altri problemi. Ma potete anche votare ogni giorno con quello che scegliete di acquistare con le notizie che scegliete di consumare, con quello che scegliete di condividere e con il genere di modello che scegliete di seguire. Volete essere qualcuno che interrompe tutte le conversazioni per controllare lo smartphone, o qualcuno che si sente stimolato dall'uso della tecnologia in modo pianificato e deliberato? Desiderate essere i padroni della vostra tecnologia o che sia la tecnologia la vostra padrona? Che cosa volete che significhi esseri umani nell'era dell'IA? Discutete di tutte queste cose con quanti vi stanno vicino: non è solo una conversazione importante, è anche affascinante. Siamo custodi del futuro della vita, ora, mentre diamo forma all'era dell'IA. A Londra ho pianto, ma adesso sento che non c'è nulla di inevitabile in questo futuro, e so che fare una differenza è molto più facile di quanto pensassi. Il nostro futuro non è scritto nella roccia, in attesa solo di accedere: sta a noi crearlo. Creiamone insieme uno motivante!

13.2 Citazioni risorse web

- Canale YouTube della cattedra di finanza & finanza sostenibile della Leipzig University (Università di Lipsia) [web1].
- Investopedia [web2]
 - Abbreviation for a Company's Stock [web3]
 - What Are Fundamentals? [web4]
 - Simple Moving Average [web5]
 - Lot: What It Means in Stock and Bond Trading [web6]
 - Earnings Call [web7]
- Introduction of Artificial Intelligence in Stock Market [web8]
- Algotrading, la finanza senza umani - Raffaele Mauro [web9]
- Probability space [web10]
- Headless Browser [web11]
- Vendita allo scoperto [web12]
- Fatto stilizzato [web13]
- Stazionarietà e serie storiche finanziarie [web14]
- Che cos'è l'ipotesi del mercato efficiente o EMH? [web15]
- Che cosa è il backtesting e come si esegue su una strategia di trading? [web16]
- 10 Applications of Machine Learning in Finance [web17]
- Le Reti Neurali Convoluzionali, ovvero come insegnare alle macchine a riconoscere per astrazione [web18]
- Corso LaTeX 2021 [web19]

Bibliografia

- [1] JOSEPH GREEN. *ChatGPT for Stock Trading: Future of Stock Trading*. Independently published, 2023.
- [2] YVES HILPISCH. *Artificial Intelligence in Finance: A Python-Based Guide*. O'Reilly, 2020.
- [3] BRAD LOOKABAUGH HARIOM TATSAT, SAHIL PURI. *Machine Learning and Data Science Blueprints for Finance: From Building Trading Strategies to Robo-Advisors Using Python*. O'Reilly, 2020.
- [4] STEFAN JANSEN. *Machine Learning for Algorithmic Trading: Predictive models to extract signals from market and alternative data for systematic trading strategies with Python*. Packt Publishing, 2020.
- [5] MAX TEGMARK. *Vita 3.0, Essere umani nell'era dell'intelligenza artificiale*. Raffaello Cortina Editore, 2018.
- [6] Prof. Dr. GREGOR WEISS. *Artificial Intelligence & Machine Learning in Finance*. Video-corso della Leipzig University, 2021.

Sitografia

- [web1] Università di lispia, finanza & finanza sostenibile. <https://www.youtube.com/@ULFinance>.
- [web2] investopedia.com. [https://www.investopedia.com/.](https://www.investopedia.com/)
- [web3] Abbreviation for a company's stock. <https://www.investopedia.com/terms/s/stocksymbol.asp>.
- [web4] What are fundamentals? <https://www.investopedia.com/terms/f/fundamentals.asp#:~:text=in%20the%20market.-,For%20businesses%2C%20information%20such%20as%20profitability%2C%20revenue%2C%20assets%2C,the%20feasibility%20of%20the%20investment.7.>
- [web5] Simple moving average. [https://www.investopedia.com/terms/s/sma.asp#:~:text=A%20simple%20moving%20average%20\(SMA\)%20is%20an%20arithmetic%20moving%20average,periods%20in%20the%20calculation%20average.](https://www.investopedia.com/terms/s/sma.asp#:~:text=A%20simple%20moving%20average%20(SMA)%20is%20an%20arithmetic%20moving%20average,periods%20in%20the%20calculation%20average.)
- [web6] Lot: What it means in stock and bond trading. [https://www.investopedia.com/terms/l/lot.asp#:~:text=our%20editorial%20policies-,What%20Is%20a%20Lot%20\(Securities%20Trading\)%3F,round%20lot%20is%20100%20shares.](https://www.investopedia.com/terms/l/lot.asp#:~:text=our%20editorial%20policies-,What%20Is%20a%20Lot%20(Securities%20Trading)%3F,round%20lot%20is%20100%20shares.)
- [web7] Earnings call. <https://www.investopedia.com/terms/e/earnings-call.asp>.
- [web8] Introduction of artificial intelligence in stock market. <https://smartmoney.angelone.in/blog/introduction-of-artificial-intelligence-in-stock-market/>.
- [web9] Algotrading, la finanza senza umani - raffaele mauro. https://issuu.com/pietrodonnini/docs/algotrading__la_finanza_senza_umani_-_raffaele_mau.

- [web10] Probability space. https://en.wikipedia.org/wiki/Probability_space.
- [web11] Headless browser. https://en.wikipedia.org/wiki/Headless_browser.
- [web12] Vendita allo scoperto. https://it.wikipedia.org/wiki/Vendita_allo_scoperto.
- [web13] Fatto stilizzato. https://it.frwiki.wiki/wikif/Fait_stylis%C3%A9.
- [web14] Stazionarietà e serie storiche finanziarie. <https://www.cgsfinancial.it/post/stazionariet%C3%A0-e-serie-storiche-finanziarie#:~:text=Solitamente%20nella%20letteratura%20accademica%2C%20quando,%20appunto%2C%20mean%2Dreverting>.
- [web15] Che cos'è l'ipotesi del mercato efficiente o emh? <https://www.ig.com/it/strategie-di-trading/che-cos-e-l-ipotesi-del-mercato-efficiente-o-emh-200313>.
- [web16] Che cosa è il backtesting e come si esegue su una strategia di trading? <https://www.ig.com/it/strategie-di-trading/che-cosa-e-il-backtesting-e-come-si-esegue-su-una-strategia-di-t-221123>.
- [web17] 10 applications of machine learning in finance. <https://algorithmxlab.com/blog/applications-machine-learning-finance/>.
- [web18] Le reti neurali convoluzionali, ovvero come insegnare alle macchine a riconoscere per astrazione. <https://www.spindox.it/it/reti-neurali-convoluzionali-il-deep-learning-ispirato-all-a-corteccia-visiva#note2>.
- [web19] Corso latex 2021. <https://users.dimi.uniud.it/~gianluca.gorni/TeX/TeX.html>.