

Introduzione

Il dataismo sostiene che l'universo sia costituito da flussi di dati, e che il valore di qualsiasi fenomeno o entità sia determinato dal suo contributo all'elaborazione dei dati. [...] Il dataismo fa così crollare la barriera tra gli animali [umani] e le macchine, prefigura un tempo in cui gli algoritmi elettronici decifreranno e supereranno gli algoritmi biochimici.

Yuval Noah Harari, *Homo Deus: A Brief History of Tomorrow*, Harvill Secker, Londra 2015

Trovare l'algoritmo giusto per negoziare automaticamente e con successo nei mercati finanziari significa trovare il Santo Graal del trading finanziario. Non molto tempo fa, il trading algoritmico era disponibile e possibile solo per i grandi trader, dotati di ricchi portafogli e che potevano contare sulla gestione di molti asset. I recenti sviluppi nei campi dell'*open source*, dell'*open data*, del *cloud computing* e del *cloud storage*, nonché delle piattaforme di trading online hanno livellato il campo da gioco anche per gli operatori più piccoli e i singoli trader, e ora è possibile esplorare questa disciplina così affascinante anche contando solo su un tipico computer portatile o desktop e una connessione affidabile a Internet.

Al giorno d'oggi, Python, con il suo ecosistema di potenti package, rappresenta la piattaforma tecnologica privilegiata per il trading algoritmico. Tra l'altro, Python permette di svolgere in modo efficiente l'analisi dei dati (per esempio con *pandas*, <http://pandas.pydata.org>), di applicare il *machine learning* alla predizione del mercato azionario (per esempio con *scikit-learn*, <http://scikit-learn.org>) o addirittura di fare uso della tecnologia *deep learning* di Google (con *tensorflow*, <http://tensorflow.org>).

Questo libro si occupa di Python impiegato per il trading algoritmico, principalmente nel contesto di strategie di *alpha generating* (vedi il Capitolo 1). Un libro del genere, che si colloca all'intersezione di due campi così vasti ed entusiasmanti, difficilmente può riuscire a fondo a trattare tutti gli argomenti di rilievo. Tuttavia, può introdurre in modo piuttosto approfondito tutta una serie di importanti meta-argomenti.

- *Dati finanziari*: sono al centro di ogni progetto di trading algoritmico. Python e package come *NumPy* e *pandas* svolgono un ottimo lavoro nel gestire e sfruttare dati finanziari strutturati di qualsiasi tipo (*end-of-day*, *intraday*, ad alta frequenza).
- *Backtesting*: non dovrebbe esistere alcun trading automatizzato e algoritmico senza un rigoroso test delle strategie di trading implementate. Il libro tratta, tra le altre cose,

le strategie di trading basate su medie mobili semplici, *momentum*, *mean-reversion* e le previsioni basate sul *machine learning* e il *deep learning*.

- *Dati in tempo reale*: il trading algoritmico richiede la gestione di dati in tempo reale, con algoritmi online rivolti all'analisi e alla visualizzazione real-time. Il libro introduce anche alla programmazione socket, con ZeroMQ, e alla visualizzazione in streaming.
- *Piattaforme online*: nessuna negoziazione può avvenire senza contare su una piattaforma di trading. Il libro tratta due piattaforme di trading elettronico: Oanda (<http://oanda.com>) e FXCM (<http://fxcm.com>).
- *Automazione*: la bellezza, e alcune delle principali sfide, del trading algoritmico derivano dall'automazione delle operazioni di trading. Il libro mostra come impiegare Python nel cloud e come impostare un ambiente appropriato per il trading automatizzato e algoritmico.

Il libro offre un'esperienza unica di apprendimento, dovuta ai seguenti approcci.

- *Trattazione di argomenti rilevanti*: è l'unico libro che tratta con questa ampiezza e profondità tutto ciò che riguarda l'uso di Python per il trading algoritmico.
- *Ricca base di codice*: il libro è accompagnato da un repository Git con tutti i listati in un formato eseguibile. Il repository è disponibile sulla piattaforma Quant (<http://py4at.pqp.io>) e all'URL <https://github.com/yhilpisch/py4at>.
- *Il trading reale come obiettivo*: la copertura di due diverse piattaforme di trading online mette il lettore in condizioni di iniziare in modo efficiente a entrare nel mondo del trading virtuale e reale. A tale scopo, il libro fornisce al lettore conoscenze di base ma pertinenti, pratiche e utili.
- *Approccio fai-da-te di autoapprendimento*: poiché il materiale e i listati sono autonomi e si basano solo su package Python standard, il lettore ha sempre la piena conoscenza e il pieno controllo di ciò che sta facendo, di come utilizzare gli esempi di codice, di come modificarli e così via. Non si rimanda a piattaforme terze, per esempio, per eseguire il backtesting o per connettersi a piattaforme di trading. Il lettore può fare tutto da solo, semplicemente seguendo le istruzioni fornite e al ritmo che trova più congeniale.
- *Forum degli utenti*: il lettore dovrebbe essere in grado di seguire senza problemi il libro, ma l'autore e i *Python Quants* sono sempre a disposizione. Il lettore può inviare in qualsiasi momento domande e commenti nel forum degli utenti sulla piattaforma Quant (<http://py4at.pqp.io>). L'account è gratuito.
- *Formazione online/video* (con sottoscrizione a pagamento): i *Python Quants* offrono programmi di formazione online completi che fanno uso dei contenuti presenti nel libro e altri contenuti aggiuntivi, trattando argomenti importanti come la scienza dei dati finanziari, l'uso dell'intelligenza artificiale nella finanza, Python per Excel e i database, più altri strumenti e competenze per Python.

Contenuti e struttura

Ecco una rapida panoramica degli argomenti e dei contenuti presentati in ogni capitolo.

- Il Capitolo 1, *Python e il trading algoritmico*, è un'introduzione al tema del trading algoritmico, ovvero il trading automatizzato di strumenti finanziari basato su algoritmi. Tratta i concetti fondamentali relativi a questo contesto e presenta, tra le altre cose, i requisiti per l'utilizzo dei contenuti del libro.
- Il Capitolo 2, *L'infrastruttura Python*, getta le basi tecniche per tutti i capitoli successivi, in quanto spiega come impostare un ambiente Python adatto. Questo capitolo utilizza principalmente *conda* come *package manager* ed *environment manager*. Inoltre illustra l'implementazione di Python tramite container Docker nel cloud.
- Quanto al Capitolo 3, *Lavorare con i dati finanziari*, i dati delle serie storiche sono fondamentali per ogni progetto di trading algoritmico. Questo capitolo mostra come recuperare i dati finanziari da diverse fonti pubbliche e proprietarie. Mostra anche come archiviare i dati delle serie storiche finanziarie in modo efficiente con Python.
- Il Capitolo 4, *Padroneggiare il backtesting vettorizzato*, si occupa della vettorizzazione, un potente approccio per il calcolo numerico in generale e per l'analisi finanziaria in particolare. Questo capitolo introduce la vettorizzazione con NumPy e pandas, e tratta l'approccio al backtesting di strategie basate su SMA, *momentum* e *mean reversion*.
- Il Capitolo 5, *Predire i movimenti di mercato con il machine learning*, è dedicato alla generazione di predizioni di mercato mediante l'applicazione di approcci di *machine learning* e *deep learning*. Facendo affidamento principalmente sulle osservazioni sui rendimenti passati, presenta approcci impiegabili per predire la direzione del mercato di domani, utilizzando package Python come *scikit-learn* e *tensorflow*.
- Il Capitolo 6, *Creazione di classi per il backtesting basato su eventi*, supera il backtesting vettorizzato, che pur presentando vantaggi in termini di concisione di codice e prestazioni, è limitato per quanto riguarda la rappresentazione di alcune caratteristiche delle strategie di trading; il backtesting basato su eventi, implementato mediante l'uso della programmazione a oggetti, consente una modellazione piuttosto granulare e più realistica di tali caratteristiche. Questo capitolo presenta e spiega in dettaglio una classe base e due classi per il backtesting di strategie di trading *long-only* e *long-short*.
- Il Capitolo 7, *Lavorare con dati e socket in tempo reale*, si spinge oltre: la necessità di gestire i dati in tempo reale o in streaming è una realtà anche per i singoli trader algoritmici. Lo strumento privilegiato è la programmazione socket, per svolgere la quale questo capitolo introduce ZeroMQ (<http://zeromq.org>), una tecnologia leggera e scalabile. Il capitolo spiega anche come utilizzare Plotly (<http://plot.ly>) per creare grafici in streaming piacevoli e interattivi. Presenta inoltre una classe wrapper che semplifica la creazione di tali grafici nei casi in cui sia necessario visualizzare contemporaneamente più flussi di dati (come avviene in una dashboard).
- Il Capitolo 8 è intitolato *Trading di CFD con Oanda*. Oanda (<http://oanda.com>) è una piattaforma di trading forex e CFD (*Contracts for Difference*) che offre un'ampia gamma di strumenti negoziabili, per esempio analisi su coppie di valute, indici azionari, materie prime o strumenti di tasso (obbligazioni di riferimento). Questo capitolo fornisce una guida su come implementare strategie di trading automatizzate

e algoritmiche con Oanda, utilizzando il package wrapper per Python `tpqoa` (<http://github.com/yhilpisch/tpqoa>).

- Il Capitolo 9, *Trading di FX con FXCM*, si occupa di FXCM, un'altra piattaforma di trading forex e CFD (*Contracts for Difference*) che ha recentemente rilasciato una moderna API RESTful per il trading algoritmico. Gli strumenti disponibili riguardano più classi di attività, come forex, indici azionari e materie prime. È disponibile un package wrapper per Python che rende il trading algoritmico basato sul codice Python piuttosto comodo ed efficiente (<http://fxcmpy.tpq.io>).
- Il Capitolo 10, *Automazione delle operazioni di trading*, tratta la gestione del capitale, l'analisi e la gestione del rischio, nonché i compiti tipici dell'automazione tecnica delle operazioni di negoziazione algoritmica. Considera in dettaglio il criterio di Kelly per l'allocazione del capitale e la leva finanziaria.
- L'Appendice fornisce una breve introduzione agli argomenti più importanti di Python, NumPy e pandas nel contesto del materiale trattato nei capitoli. Rappresenta un punto di partenza per raffinare nel tempo la conoscenza di Python.

La Figura P.1 mostra i livelli di trading algoritmico trattati dai capitoli, dal basso verso l'alto. Si parte necessariamente dall'infrastruttura Python (Capitolo 2), cui si aggiungono i dati finanziari (Capitolo 3), le strategie e il backtesting vettorizzato (Capitolo 4 e Capitolo 5). Fino a quel momento, vengono utilizzati e manipolati solo dataset. Il backtesting basato su eventi introduce per la prima volta l'idea che i dati del mondo reale arrivino in modo incrementale (Capitolo 6). Questo concetto apre la strada alle attività di comunicazione via socket e alla gestione dei dati in tempo reale (Capitolo 7). Inoltre, le piattaforme di trading e le loro API devono essere in grado di effettuare ordini (Capitolo 8 e Capitolo 9). Infine, vengono trattati alcuni importanti aspetti dell'automazione e della distribuzione al pubblico (Capitolo 10). In questo senso, i capitoli del libro formano gli strati rappresentati nella Figura P.1, che mostra una sequenza naturale degli argomenti trattati.

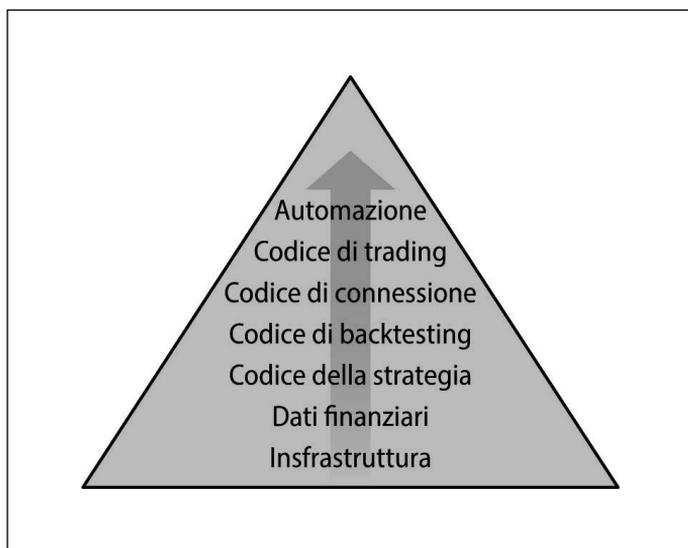


Figura I.1 I livelli di questo libro.

A chi è rivolto questo libro

Il libro è rivolto agli studenti, agli accademici e ai professionisti che intendono applicare Python all'affascinante campo del trading algoritmico. Il libro presuppone che il lettore sia dotato di alcune conoscenze di base sia della programmazione Python, sia del trading finanziario. Come riferimento e ripasso, l'Appendice presenta alcuni elementi importanti di Python, NumPy, matplotlib e pandas. Di seguito sono segnalati alcuni testi di riferimento utili per ottenere una buona comprensione degli argomenti di Python trattati nel libro. La maggior parte dei lettori trarrà vantaggio dall'utilizzo del testo di Hilpisch (2018). Per quanto riguarda gli approcci al *machine learning* e al *deep learning* applicati al trading algoritmico, Hilpisch (2020) offre molte informazioni di base e una ricca collezione di esempi specifici.

- Yves Hilpisch, *Python for Finance - Mastering Data-Driven Finance*, 2a ed., O'Reilly, Beijing 2018.
- Yves Hilpisch, *Artificial Intelligence in Finance - A Python-based Guide*, O'Reilly, Beijing 2020.
- Wes McKinney, *Python for Data Analysis - Data Wrangling con Pandas*, NumPy e IPython. 2a ed., O'Reilly, Beijing 2017.
- Luciano Ramalho, *Fluent Python - Clear, Concise, and Effective Programming*. 2a ed., O'Reilly, Beijing 2021.
- Jake VanderPlas, *Python Data Science Handbook - Essential Tools for Working with Data*, O'Reilly, Beijing 2016.

Per informazioni di base sul trading algoritmico consultate, per esempio, i seguenti testi.

- Ernest Chan, *Quantitative Trading - How to Build Your Own Algorithmic Trading Business*, John Wiley & Sons, Hoboken 2009.
- Ernest Chan, *Algorithmic Trading - Winning Strategies and Their Rationale*, John Wiley & Sons, Hoboken 2013.
- Robert Kissel, *The Science of Algorithmic Trading and Portfolio Management*. Elsevier/Academic Press, Amsterdam 2013.
- Rishi Narang, *Inside the Black Box - A Simple Guide to Quantitative and High Frequency Trading*, John Wiley & Sons, Hoboken 2013.

Mi auguro che apprezzerete questo vostro viaggio nel mondo del trading algoritmico con Python. Per domande o commenti potete contattarmi scrivendo a pyalgo@tpq.io.

Convenzioni usate in questo libro

In questo libro vengono utilizzate le seguenti convenzioni tipografiche.

- *Corsivo*: indica i nuovi termini e i termini tecnici e stranieri.
- Carattere monospaziato: utilizzato per i listati nonché, all'interno dei paragrafi, per far riferimento agli elementi del linguaggio, come nomi di variabili o funzioni, di database, tipi di dati, variabili di ambiente, istruzioni, indirizzi web e parole chiave.
- *Carattere monospaziato corsivo*: mostra nei listati il testo che dovrebbe essere sostituito con valori forniti dall'utente o valori determinati dal contesto.

SUGGERIMENTO

Questo tipo di nota indica un trucco o un suggerimento utile.

NOTA

Questo tipo di nota indica un approfondimento generale.

ATTENZIONE

Questo tipo di nota indica un'avvertenza o cautela.

Utilizzo degli esempi di codice

Potete trovare ed eseguire il codice che accompagna questo libro sulla piattaforma Quant, all'indirizzo <https://py4at.pqp.io>, registrandovi gratuitamente. Il codice degli esempi è inoltre accessibile al seguente repository GitHub: <https://github.com/yhilpisch/py4at>.

Lo scopo di questo manuale è quello di aiutarvi a portare a termine il vostro lavoro. In generale, il codice di esempio presentato nel volume può essere liberamente utilizzato nei programmi e nella documentazione e non avete bisogno di ottenere il permesso di utilizzarlo, sempre che non intendiate riprodurre una porzione significativa del codice. Per esempio, la scrittura di un programma che utilizzi vari blocchi di codice tratti da questo libro non richiede alcuna autorizzazione. La vendita o distribuzione di esempi tratti dal libro, invece, richiede un'autorizzazione. Rispondere a una domanda citando questo libro e il codice di esempio non richiede alcuna autorizzazione. Incorporare nella documentazione di un prodotto una quantità significativa di codice tratto da questo libro richiede un'autorizzazione.

Appreziamo, ma generalmente non richiediamo, l'attribuzione. Un'attribuzione, di solito, include il titolo, l'autore, l'editore e l'ISBN dell'opera originale. Per esempio: *Python for Algorithmic Trading*, by Yves Hilpisch (O'Reilly). Copyright 2021 Yves Hilpisch, 978-1-492-05335-4.

Se ritenete che l'utilizzo che state facendo degli esempi di codice vada oltre un semplice impiego locale o che richieda un'autorizzazione, non esitate a contattare l'editore del volume originale all'indirizzo permissions@oreilly.com.

Ringraziamenti

Voglio ringraziare i revisori tecnici – Hugh Brown, McKlayne Marshall, Ramanathan Ramakrishnamoorthy e Prem Jebaseelan – che con i loro commenti mi hanno aiutato a migliorare i contenuti del libro.

Un ringraziamento speciale va, come di consueto, a Michael Schwed, che mi supporta in tutte le questioni tecniche, dalle più semplici alle più complesse, con il suo ampio e approfondito know-how tecnologico.

Anche i delegati dei Certificate Programs in Python for Computational Finance and Algorithmic Trading hanno contribuito a migliorare questo libro. Il loro feedback continuo

mi ha permesso di eliminare errori e imprecisioni, e di perfezionare il codice e i notebook utilizzati nei nostri corsi di formazione online e ora, finalmente, anche in questo libro. Vorrei anche ringraziare l'intero team di O'Reilly Media, in particolare Michelle Smith, Michele Cronin, Victoria DeRose e Danny Elfanbaum, per aver reso possibile tutto questo e per avermi aiutato a perfezionare il libro.

Naturalmente, tutti gli errori rimanenti sono solo miei.

Inoltre, vorrei anche ringraziare il team di Refinitiv, e in particolare Jason Ramchandani, per avermi fornito un continuo supporto e l'accesso ai dati finanziari. I principali file di dati utilizzati in tutto il libro e messi a disposizione dei lettori provengono, in un modo o nell'altro, dalle API di Refinitiv.

Alla mia famiglia con amore. Dedico questo libro a mio padre Adolf, il cui sostegno a me e alla nostra famiglia dura ormai da quasi cinque decenni.