1
Machine Learning for Trading

Algorithmic trading relies on computer programs that execute algorithms to automate some, or all, elements of a trading strategy. Algorithms are a sequence of steps or rules to achieve a goal and can take many forms. In the case of **machine learning** (**ML**), algorithms pursue the objective of learning other algorithms, namely rules, to achieve a target based on data, such as minimizing a prediction error.

These algorithms encode various activities of a portfolio manager who observes market transactions and analyzes relevant data to decide on placing buy or sell orders. The sequence of orders defines the portfolio holdings that, over time, aim to produce returns that are attractive to the providers of capital, taking into account their appetite for risk.

Ultimately, the goal of active investment management consists in achieving alpha, that is, returns in excess of the benchmark used for evaluation. The fundamental law of active management applies the **information ratio** (**IR**) to express the value of active management as the ratio of portfolio returns above the returns of a benchmark, usually an index, to the volatility of those returns. It approximates the information ratio as the product of the **information coefficient** (**IC**), which measures the quality of forecast as their correlation with outcomes, and the breadth of a strategy expressed as the square root of the number of bets.

Hence, the key to generating alpha is forecasting. Successful predictions, in turn, require superior information or a superior ability to process public information. Algorithms facilitate optimization throughout the investment process, from asset allocation to idea-generation, trade execution, and risk management. The use of ML for algorithmic trading, in particular, aims for more efficient use of conventional and alternative data, with the goal of producing both better and more actionable forecasts, hence improving the value of active management.

Historically, algorithmic trading used to be more narrowly defined as the automation of trade execution to minimize costs as offered by the sell side, but we will take a more comprehensive perspective since the use of algorithms, and ML, in particular, has come to impact a broader range of activities from idea generation and alpha factor design to asset allocation, position sizing, and the testing and evaluation of strategies.

This chapter looks at the bigger picture of how the use of ML has emerged as a critical source of competitive advantage in the investment industry and where it fits into the investment process to enable algorithmic trading strategies.

We will be covering the following topics in the chapter:

- How this book is organized and who should read it
- How ML has come to play a strategic role in algorithmic trading
- How to design and execute a trading strategy
- How ML adds value to an algorithmic trading strategy

How to read this book

If you are reading this, then you are probably aware that ML has become a strategic capability in many industries, including the investment industry. The explosion of digital data that drives much of the rise of ML is having a particularly powerful impact on investing, which already has a long history of using sophisticated models to process information. The scope of trading across asset classes implies that a vast range of new, alternative data may be relevant in addition to the market and fundamental data that used to be the focus of the analytical efforts.

You may have also come across the insight that the successful application of ML or data science requires the integration of statistical knowledge, computational skills, and domain expertise at the individual or team level. In other words, it is essential to ask the right questions, identify and understand the data that may provide the answers, deploy a broad range of tools to obtain results, and interpret them in a way that leads to the right decisions.

Consequently, this book takes an integrated perspective on the application of ML to the domain of investment and trading. In this section, we will lay out what to expect, how it goes about achieving its objectives, and what you need to both meet your goals and have fun in the process.

What to expect

This book aims to equip you with the strategic perspective, conceptual understanding, and practical tools to add value from applying ML to the trading and investment process. To this end, it covers ML as an important element in a process rather than a standalone exercise.

First and foremost, it covers a broad range of supervised, unsupervised, and reinforcement learning algorithms useful for extracting signals from the diverse data sources relevant to different asset classes. It introduces a ML workflow and focuses on practical use cases with relevant data and numerous code examples. However, it also develops the mathematical and statistical background to facilitate the tuning of an algorithm or the interpretation of the results.

The book recognizes that investors can extract value from third-party data more than other industries. As a consequence, it covers not only how to work with market and fundamental data but also how to source, evaluate, process, and model alternative data sources such as unstructured text and image data.

It relates the use of ML to research and evaluate alpha factors to quantitative and factor-based strategies and introduces portfolio management as the context for the deployment of strategies that combine multiple alpha factors. It also highlights that ML can add value beyond predictions relevant to individual asset prices, for example to asset allocation and addresses the risks of false discoveries from using ML with large datasets to develop a trading strategy.

It should not be a surprise that this book does not provide investment advice or ready-made trading algorithms. Instead, present building blocks required to identify, evaluate, and combine datasets that suitable for any given investment objective, select and apply ML algorithms to this data, and develop and test algorithmic trading strategies based on the results.

Who should read this book

You should find the book informative if you are an analyst, data scientist, or ML engineer with an understanding of financial markets and interest in trading strategies. You should also find value as an investment professional who aims to leverage ML to make better decisions.

If your background is software and ML, you may be able to just skim or skip some introductory material on ML. Similarly, if your expertise is in investment, you will likely be familiar with some or all of the financial context. You will likely find the book most useful as a survey of key algorithms, building blocks and use cases than for specialized coverage of a particular algorithm or strategy. However, the book assumes you are interested in continuing to learn about this very dynamic area. To this end, it references numerous resources to support your journey towards customized trading strategies that leverage and build on the fundamental methods and tools it covers.

You should be comfortable using Python 3 and various scientific computing libraries like `numpy`, `pandas`, or `scipy` and be interested in picking up numerous others along the way. Some experience with ML and scikit-learn would be helpful, but we briefly cover the basic workflow and reference various resources to fill gaps or dive deeper.

How the book is organized

The book provides a comprehensive introduction to how ML can add value to the design and execution of trading strategies. It is organized in four parts that cover different aspects of the data sourcing and strategy development process, as well as different solutions to various ML challenges.

Part I – the framework – from data to strategy design

The first part provides a framework for the development of algorithmic trading strategies. It focuses on the data that power the ML algorithms and strategies discussed in this book, outlines how ML can be used to derive trading signals, and how to deploy and evaluate strategies as part of a portfolio.

The remainder of this chapter summarizes how and why ML became central to investment, describes the trading process and outlines how ML can add value. `Chapter 2`, *Market and Fundamental Data*, covers sources and working with original exchange-provided tick and financial reporting data, as well as how to access numerous open-source data providers that we will rely on throughout this book.

`Chapter 3`, *Alternative Data for Finance*, provides categories and criteria to assess the exploding number of sources and providers. It also demonstrates how to create alternative data sets by scraping websites, for example to collect earnings call transcripts for use with **natural language processing** (**NLP**) and sentiment analysis algorithms in the second part of the book.

`Chapter 4`, *Alpha Factor Research*, provides a framework for understanding how factors work and how to measure their performance, for example using the **information coefficient** (**IC**). It demonstrates how to engineer alpha factors from data using Python libraries offline and on the Quantopian platform. It also introduces the `zipline` library to backtest factors and the `alphalens` library to evaluate their predictive power.

`Chapter 5`, *Strategy Evaluation*, introduces how to build, test and evaluate trading strategies using historical data with `zipline` offline and on the Quantopian platform. It presents and demonstrates how to compute portfolio performance and risk metrics using the `pyfolio` library. It also addresses how to manage methodological challenges of strategy backtests and introduce methods to optimize a strategy from a portfolio risk perspective.

Part 2 – ML fundamentals

The second part covers the fundamental supervised and unsupervised learning algorithms and illustrates their application to trading strategies. It also introduces the Quantopian platform where you can leverage and combine the data and ML techniques developed in this book to implement algorithmic strategies that execute trades in live markets.

`Chapter 6`, The *Machine Learning Process*, sets the stage by outlining how to formulate, train, tune and evaluate the predictive performance of ML models as a systematic workflow.

`Chapter 7`, *Linear Models*, it shows how to use linear and logistic regression for inference and prediction and how to use regularization to manage the risk of overfitting. It presents the Quantopian trading platform and demonstrates how to build factor models and predict asset prices.

`Chapter 8`, *Time Series Models*, covers univariate and multivariate time series, including vector autoregressive models and cointegration tests, and how they can be applied to pairs trading strategies. `Chapter 9`, *Bayesian Machine Learning*, presents how to formulate probabilistic models and how **Markov Chain Monte Carlo** (**MCMC**) sampling and Variational Bayes facilitate approximate inference. It also illustrates how to use PyMC3 for probabilistic programming to gain deeper insights into parameter and model uncertainty.

`Chapter 10`, *Decision Trees and Random Forests*, shows how to build, train and tune non-linear tree-based models for insight and prediction. It introduces tree-based ensemble models and shows how random forests use bootstrap aggregation to overcome some of the weaknesses of decision trees. `Chapter 11`, *Gradient Boosting Machines* ensemble models and demonstrates how to use the libraries `xgboost`, `lightgbm`, and `catboost` for high-performance training and prediction, and reviews in depth how to tune the numerous hyperparameters.

`Chapter 12`, *Unsupervised Learning* introduces how to use dimensionality reduction and clustering for algorithmic trading. It uses principal and independent component analysis to extract data-driven risk factors. It presents several clustering techniques and demonstrates the use of hierarchical clustering for asset allocation.

Part 3 – natural language processing

Part three focuses on text data and introduces state-of-the-art unsupervised learning techniques to extract high-quality signals from this key source of alternative data.

`Chapter 13`, *Working with Text Data*, demonstrates how to convert text data into a numerical format and applies the classification algorithms from *part two* for sentiment analysis to large datasets. `Chapter 14`, *Topic Modeling*, applies Bayesian unsupervised learning to extract latent topics that can summarize a large number of documents and offer more effective ways to explore text data or use topics as features for a classification model. It demonstrates how to apply this technique to earnings call transcripts sourced in `Chapter 3`, *Alternative Data for Finance*, and to annual reports filed with the **Securities and Exchange Commission** (**SEC**).

`Chapter 15`, *Word Embeddings*, uses neural networks to learn state-of-the-art language features in the form of word vectors that capture semantic context much better than traditional text features and represent a very promising avenue for extracting trading signals from text data.

Part 4 – deep and reinforcement learning

Part 4 introduces deep learning and reinforcement learning.

- `Chapter 17`, *Deep Learning*, introduces Keras, TensorFlow and PyTorch, the most popular deep learning frameworks and illustrates how to train and tune various architectures.
- `Chapter 18, Recurrent Neural Networks, presents RNNs for time series data`

- `Chapter 19`, *Convolutional Neural Networks*, illustrates how to use CNNs with image and text data
- `Chapter 20`, *Autoencoders and Generative Adversarial Nets*, shows how to use deep neural networks for unsupervised learning with autoencoders and presents GANs that produce synthetic data
- `Chapter 21`, *Reinforcement Learning*, demonstrates the use of reinforcement learning to build dynamic agents that learn a policy function based on rewards using the OpenAI gym platform

What you need to succeed

The book content revolves around the application of ML algorithms to different datasets. Significant additional content is hosted on GitHub to facilitate review and experiments with the examples discussed in the book. It contains additional detail and instructions as well as numerous references.

Data sources

We will use freely available historical data from market, fundamental and alternative sources. `Chapter 2`, *Market and Fundamental Data* and `Chapter 3`, *Alternative Data for Finance* cover characteristics and access to these data sources and introduce key providers that we will use throughout the book. The companion GitHub repository (see beneath) contains instructions on how to obtain or create some of the datasets that we will use throughout and includes some smaller datasets.

A few sample data sources that we will source and work with include, but are not limited to:

- NASDAQ ITCH order book data
- Electronic Data Gathering, Analysis, and Retrieval (EDGAR) SEC filings
- Earnings call transcripts from Seeking Alpha
- Quandl daily prices and other data points for over 3,000 US stocks
- Various macro fundamental data from the Federal Reserve and others
- Large Yelp business reviews and Twitter datasets
- Image data on oil tankers

Some of the data is several GB large (e.g. the NASDAQ and SEC filings). The notebooks indicate when that is the case.

GitHub repository

The GitHub repository contains Jupyter Notebooks that illustrate many of the concepts and models in more detail. The Notebooks are referenced throughout the book where used. Each chapter has its own directory with separate instructions where needed, as well as reference specific to the chapter's content.

Jupyter Notebooks is a great tool for creating reproducible computational narratives, and it enables users to create and share documents that combine live code with narrative text, mathematical equations, visualizations, interactive controls, and other rich output. It also provides building blocks for interactive computing with data, such as a file browser, terminals, and a text editor.

You can find the code files placed at:
`https://github.com/PacktPublishing/Hands-On-Machine-Learning-for-Algorithmic-Trading`.

Python libraries

The book uses Python 3.7, and recommends `miniconda` to install the `conda` package manager and to create a `conda` environment to install the the requisite libraries. To this end, the GitHub repo contains an `environment.yml` file. Please refer to the installation instructions referenced in the GitHub repo's `README` file.

The rise of ML in the investment industry

The investment industry has evolved dramatically over the last several decades and continues to do so amid increased competition, technological advances, and a challenging economic environment. This section will review several key trends that have shaped the investment environment in general, and the context for algorithmic trading more specifically, and related themes that will recur throughout this book.

The trends that have propelled algorithmic trading and ML to current prominence include:

- Changes in the market microstructure, such as the spread of electronic trading and the integration of markets across asset classes and geographies
- The development of investment strategies framed in terms of risk-factor exposure, as opposed to asset classes

- The revolutions in computing power, data-generation and management, and analytic methods
- The outperformance of the pioneers in algorithmic traders relative to human, discretionary investors

In addition, the financial crises of 2001 and 2008 have affected how investors approach diversification and risk management and have given rise to low-cost passive investment vehicles in the form of **exchange-traded funds** (**ETFs**). Amid low yield and low volatility after the 2008 crisis, cost-conscious investors shifted $2 trillion from actively-managed mutual funds into passively managed ETFs. Competitive pressure is also reflected in lower hedge fund fees that dropped from the traditional 2% annual management fee and 20% take of profits to an average of 1.48% and 17.4%, respectively, in 2017.

From electronic to high-frequency trading

Electronic trading has advanced dramatically in terms of capabilities, volume, coverage of asset classes, and geographies since networks started routing prices to computer terminals in the 1960s.

Equity markets have led this trend worldwide. The 1997 order-handling rules by the SEC introduced competition to exchanges through **electronic communication networks** (**ECN**). ECNs are automated **Alternative Trading Systems** (**ATS**) that match buy-and-sell orders at specified prices, primarily for equities and currencies and are registered as broker-dealers. It allows significant brokerages and individual traders in different geographic locations to trade directly without intermediaries, both on exchanges and after hours. Dark pools are another type of ATS that allow investors to place orders and trade without publicly revealing their information, as in the order book maintained by an exchange. Dark pools have grown since a 2007 SEC ruling, are often housed within large banks, and are subject to SEC regulation.

With the rise of electronic trading, algorithms for cost-effective execution have developed rapidly and adoption has spread quickly from the sell side to the buy side and across asset classes. Automated trading emerged around 2000 as a sell-side tool aimed at cost-effective trade execution that spread orders over time to limit the market impact. These tools spread to the buy side and became increasingly sophisticated by taking into account, for example, transaction costs and liquidity, as well as short-term price and volume forecasts.

Direct Market Access (**DMA**) gives a trader greater control over execution by allowing it to send orders directly to the exchange using the infrastructure and market participant identification of a broker who is a member of an exchange. Sponsored access removes pre-trade risk controls by the brokers and forms the basis for **high-frequency trading** (**HFT**).

HFT refers to automated trades in financial instruments that are executed with extremely low latency in the microsecond range and where participants hold positions for very short periods. The goal is to detect and exploit inefficiencies in the market microstructure, the institutional infrastructure of trading venues. HFT has grown substantially over the past ten years and is estimated to make up roughly 55% of trading volume in US equity markets and about 40% in European equity markets. HFT has also grown in futures markets to roughly 80% of foreign-exchange futures volumes and two-thirds of both interest rate and Treasury 10 year futures volumes (FAS 2016).

HFT strategies aim to earn small profits per trade using passive or aggressive strategies. Passive strategies include arbitrage trading to profit from very small price differentials for the same asset, or its derivatives, traded on different venues. Aggressive strategies include order anticipation or momentum ignition. Order anticipation, also known as **liquidity detection**, involves algorithms that submit small exploratory orders to detect hidden liquidity from large institutional investors and trade ahead of a large order to benefit from subsequent price movements. Momentum ignition implies an algorithm executing and canceling a series of orders to spoof other HFT algorithms into buying (or selling) more aggressively and benefit from the resulting price changes.

Regulators have expressed concern over the potential link between certain aggressive HFT strategies and increased market fragility and volatility, such as that experienced during the May 2010 Flash Crash, the October 2014 Treasury Market volatility, and the sudden crash by over 1,000 points of the Dow Jones Industrial Average on August 24, 2015. At the same time, market liquidity has increased with trading volumes due to the presence of HFT, which has lowered overall transaction costs.

The combination of reduced trading volumes amid lower volatility and rising costs of the technology and access to both data and trading venues has led to financial pressure. Aggregate HFT revenues from US stocks have been estimated to drop beneath $1 billion for the first time since 2008, down from $7.9 billion in 2009.

This trend has led to industry consolidation with various acquisitions by, for example, the largest listed proprietary trading firm Virtu Financial, and shared infrastructure investments, such as the new Go West ultra-low latency route between Chicago and Tokyo. Simultaneously, startups such as Alpha Trading Lab make HFT trading infrastructure and data available to democratize HFT by crowdsourcing algorithms in return for a share of the profits.

Factor investing and smart beta funds

The return provided by an asset is a function of the uncertainty or risk associated with the financial investment. An equity investment implies, for example, assuming a company's business risk, and a bond investment implies assuming default risk.

To the extent that specific risk characteristics predict returns, identifying and forecasting the behavior of these risk factors becomes a primary focus when designing an investment strategy. It yields valuable trading signals and is the key to superior active-management results. The industry's understanding of risk factors has evolved very substantially over time and has impacted how ML is used for algorithmic trading.

Modern Portfolio Theory (**MPT**) introduced the distinction between idiosyncratic and systematic sources of risk for a given asset. Idiosyncratic risk can be eliminated through diversification, but systematic risk cannot. In the early 1960s, the **Capital Asset Pricing Model** (**CAPM**) identified a single factor driving all asset returns: the return on the market portfolio in excess of T-bills. The market portfolio consisted of all tradable securities, weighted by their market value. The systematic exposure of an asset to the market is measured by beta, which is the correlation between the returns of the asset and the market portfolio.

The recognition that the risk of an asset does not depend on the asset in isolation, but rather how it moves relative to other assets, and the market as a whole, was a major conceptual breakthrough. In other words, assets do not earn a risk premium because of their specific, idiosyncratic characteristics, but because of their exposure to underlying factor risks.

However, a large body of academic literature and long investing experience have disproved the CAPM prediction that asset risk premiums depend only on their exposure to a single factor measured by the asset's beta. Instead, numerous additional risk factors have since been discovered. A factor is a quantifiable signal, attribute, or any variable that has historically correlated with future stock returns and is expected to remain correlated in future.

These risk factors were labeled anomalies since they contradicted the **Efficient Market Hypothesis** (**EMH**), which sustained that market equilibrium would always price securities according to the CAPM so that no other factors should have predictive power. The economic theory behind factors can be either rational, where factor risk premiums compensate for low returns during bad times, or behavioral, where agents fail to arbitrage away excess returns.

Well-known anomalies include the value, size, and momentum effects that help predict returns while controlling for the CAPM market factor. The size effect rests on small firms systematically outperforming large firms, discovered by Banz (1981) and Reinganum (1981). The value effect (Basu 1982) states that firms with low valuation metrics outperform. It suggests that firms with low price multiples, such as the price-to-earnings or the price-to-book ratios, perform better than their more expensive peers (as suggested by the inventors of value investing, Benjamin Graham and David Dodd, and popularized by Warren Buffet).

The momentum effect, discovered in the late 1980s by, among others, Clifford Asness, the founding partner of AQR, states that stocks with good momentum, in terms of recent 6-12 month returns, have higher returns going forward than poor momentum stocks with similar market risk. Researchers also found that value and momentum factors explain returns for stocks outside the US, as well as for other asset classes, such as bonds, currencies, and commodities, and additional risk factors.

In fixed income, the value strategy is called **riding the yield curve** and is a form of the duration premium. In commodities, it is called the **roll return**, with a positive return for an upward-sloping futures curve and a negative return otherwise. In foreign exchange, the value strategy is called **carry**.

There is also an illiquidity premium. Securities that are more illiquid trade at low prices and have high average excess returns, relative to their more liquid counterparts. Bonds with higher default risk tend to have higher returns on average, reflecting a credit risk premium. Since investors are willing to pay for insurance against high volatility when returns tend to crash, sellers of volatility protection in options markets tend to earn high returns.

Multifactor models define risks in broader and more diverse terms than just the market portfolio. In 1976, Stephen Ross proposed arbitrage pricing theory, which asserted that investors are compensated for multiple systematic sources of risk that cannot be diversified away. The three most important macro factors are growth, inflation, and volatility, in addition to productivity, demographic, and political risk. In 1992, Eugene Fama and Kenneth French combined the equity risk factors' size and value with a market factor into a single model that better explained cross-sectional stock returns. They later added a model that also included bond risk factors to simultaneously explain returns for both asset classes.

A particularly attractive aspect of risk factors is their low or negative correlation. Value and momentum risk factors, for instance, are negatively correlated, reducing the risk and increasing risk-adjusted returns above and beyond the benefit implied by the risk factors. Furthermore, using leverage and long-short strategies, factor strategies can be combined into market-neutral approaches. The combination of long positions in securities exposed to positive risks with underweight or short positions in the securities exposed to negative risks allows for the collection of dynamic risk premiums.

As a result, the factors that explained returns above and beyond the CAPM were incorporated into investment styles that tilt portfolios in favor of one or more factors, and assets began to migrate into factor-based portfolios. The 2008 financial crisis underlined how asset-class labels could be highly misleading and create a false sense of diversification when investors do not look at the underlying factor risks, as asset classes came crashing down together.

Over the past several decades, quantitative factor investing has evolved from a simple approach based on two or three styles to multifactor smart or exotic beta products. Smart beta funds have crossed $1 trillion AUM in 2017, testifying to the popularity of the hybrid investment strategy that combines active and passive management. Smart beta funds take a passive strategy but modify it according to one or more factors, such as cheaper stocks or screening them according to dividend payouts, to generate better returns. This growth has coincided with increasing criticism of the high fees charged by traditional active managers as well as heightened scrutiny of their performance.

The ongoing discovery and successful forecasting of risk factors that, either individually or in combination with other risk factors, significantly impact future asset returns across asset classes is a key driver of the surge in ML in the investment industry and will be a key theme throughout this book.

Algorithmic pioneers outperform humans at scale

The track record and growth of **Assets Under Management** (**AUM**) of firms that spearheaded algorithmic trading has played a key role in generating investor interest and subsequent industry efforts to replicate their success. Systematic funds differ from HFT in that trades may be held significantly longer while seeking to exploit arbitrage opportunities as opposed to advantages from sheer speed.

Systematic strategies that mostly or exclusively rely on algorithmic decision-making were most famously introduced by mathematician James Simons who founded Renaissance Technologies in 1982 and built it into the premier quant firm. Its secretive Medallion Fund, which is closed to outsiders, has earned an estimated annualized return of 35% since 1982.

DE Shaw, Citadel, and Two Sigma, three of the most prominent quantitative hedge funds that use systematic strategies based on algorithms, rose to the all-time top-20 performers for the first time in 2017 in terms of total dollars earned for investors, after fees, and since inception.

DE Shaw, founded in 1988 with $47 billion AUM in 2018 joined the list at number 3. Citadel started in 1990 by Kenneth Griffin, manages $29 billion and ranks 5, and Two Sigma started only in 2001 by DE Shaw alumni John Overdeck and David Siegel, has grown from $8 billion AUM in 2011 to $52 billion in 2018. Bridgewater started in 1975 with over $150 billion AUM, continues to lead due to its Pure Alpha Fund that also incorporates systematic strategies.

Similarly, on the Institutional Investors 2017 Hedge Fund 100 list, five of the top six firms rely largely or completely on computers and trading algorithms to make investment decisions—and all of them have been growing their assets in an otherwise challenging environment. Several quantitatively-focused firms climbed several ranks and in some cases grew their assets by double-digit percentages. Number 2-ranked **Applied Quantitative Research** (**AQR**) grew its hedge fund assets 48% in 2017 to $69.7 billion and managed $187.6 billion firm-wide.

Among all hedge funds, ranked by compounded performance over the last three years, the quant-based funds run by Renaissance Technologies achieved ranks 6 and 24, Two Sigma rank 11, D.E. Shaw no 18 and 32, and Citadel ranks 30 and 37. Beyond the top performers, algorithmic strategies have worked well in the last several years. In the past five years, quant-focused hedge funds gained about 5.1% per year while the average hedge fund rose 4.3% per year in the same period.

ML driven funds attract $1 trillion AUM

The familiar three revolutions in computing power, data, and ML methods have made the adoption of systematic, data-driven strategies not only more compelling and cost-effective but a key source of competitive advantage.

As a result, algorithmic approaches are not only finding wider application in the hedge-fund industry that pioneered these strategies but across a broader range of asset managers and even passively-managed vehicles such as ETFs. In particular, predictive analytics using machine learning and algorithmic automation play an increasingly prominent role in all steps of the investment process across asset classes, from idea-generation and research to strategy formulation and portfolio construction, trade execution, and risk management.

Estimates of industry size vary because there is no objective definition of a quantitative or algorithmic fund, and many traditional hedge funds or even mutual funds and ETFs are introducing computer-driven strategies or integrating them into a discretionary environment in a human-plus-machine approach.

Morgan Stanley estimated in 2017 that algorithmic strategies have grown at 15% per year over the past six years and control about $1.5 trillion between hedge funds, mutual funds, and smart beta ETFs. Other reports suggest the quantitative hedge fund industry was about to exceed $1 trillion AUM, nearly doubling its size since 2010 amid outflows from traditional hedge funds. In contrast, total hedge fund industry capital hit $3.21 trillion according to the latest global Hedge Fund Research report.

The market research firm Preqin estimates that almost 1,500 hedge funds make a majority of their trades with help from computer models. Quantitative hedge funds are now responsible for 27% of all US stock trades by investors, up from 14% in 2013. But many use data scientists—or quants—which, in turn, use machines to build large statistical models (`WSJ`).

In recent years, however, funds have moved toward true ML, where artificially-intelligent systems can analyze large amounts of data at speed and improve themselves through such analyses. Recent examples include Rebellion Research, Sentient, and Aidyia, which rely on evolutionary algorithms and deep learning to devise fully-automatic **Artificial Intelligence** (**AI**)-driven investment platforms.

From the core hedge fund industry, the adoption of algorithmic strategies has spread to mutual funds and even passively-managed exchange-traded funds in the form of smart beta funds, and to discretionary funds in the form of quantamental approaches.

The emergence of quantamental funds

Two distinct approaches have evolved in active investment management: systematic (or quant) and discretionary investing. Systematic approaches rely on algorithms for a repeatable and data-driven approach to identify investment opportunities across many securities; in contrast, a discretionary approach involves an in-depth analysis of a smaller number of securities. These two approaches are becoming more similar as fundamental managers take more data-science-driven approaches.

Even fundamental traders now arm themselves with quantitative techniques, accounting for $55 billion of systematic assets, according to Barclays. Agnostic to specific companies, quantitative funds trade patterns and dynamics across a wide swath of securities. Quants now account for about 17% of total hedge fund assets, data compiled by Barclays shows.

Point72 Asset Management, with $12 billion in assets, has been shifting about half of its portfolio managers to a man-plus-machine approach. Point72 is also investing tens of millions of dollars into a group that analyzes large amounts of alternative data and passes the results on to traders.

Investments in strategic capabilities

Rising investments in related capabilities—technology, data and, most importantly, skilled humans—highlight how significant algorithmic trading using ML has become for competitive advantage, especially in light of the rising popularity of passive, indexed investment vehicles, such as ETFs, since the 2008 financial crisis.

Morgan Stanley noted that only 23% of its quant clients say they are not considering using or not already using ML, down from 44% in 2016.

Guggenheim Partners LLC built what it calls a supercomputing cluster for $1 million at the Lawrence Berkeley National Laboratory in California to help crunch numbers for Guggenheim's quant investment funds. Electricity for the computers costs another $1 million a year.

AQR is a quantitative investment group that relies on academic research to identify and systematically trade factors that have, over time, proven to beat the broader market. The firm used to eschew the purely computer-powered strategies of quant peers such as Renaissance Technologies or DE Shaw. More recently, however, AQR has begun to seek profitable patterns in markets using ML to parse through novel datasets, such as satellite pictures of shadows cast by oil wells and tankers.

The leading firm BlackRock, with over $5 trillion AUM, also bets on algorithms to beat discretionary fund managers by heavily investing in SAE, a systematic trading firm it acquired during the financial crisis. Franklin Templeton bought Random Forest Capital, a debt-focused, data-led investment company for an undisclosed amount, hoping that its technology can support the wider asset manager.

ML and alternative data

Hedge funds have long looked for alpha through informational advantage and the ability to uncover new uncorrelated signals. Historically, this included things such as proprietary surveys of shoppers, or voters ahead of elections or referendums. Occasionally, the use of company insiders, doctors, and expert networks to expand knowledge of industry trends or companies crosses legal lines: a series of prosecutions of traders, portfolio managers, and analysts for using insider information after 2010 has shaken the industry.

In contrast, the informational advantage from exploiting conventional and alternative data sources using ML is not related to expert and industry networks or access to corporate management, but rather the ability to collect large quantities of data and analyze them in real-time.

Three trends have revolutionized the use of data in algorithmic trading strategies and may further shift the investment industry from discretionary to quantitative styles:

- The exponential increase in the amount of digital data
- The increase in computing power and data storage capacity at lower cost
- The advances in ML methods for analyzing complex datasets

Conventional data includes economic statistics, trading data, or corporate reports. Alternative data is much broader and includes sources such as satellite images, credit card sales, sentiment analysis, mobile geolocation data, and website scraping, as well as the conversion of data generated in the ordinary course of business into valuable intelligence. It includes, in principle, any data source containing trading signals that can be extracted using ML.

For instance, data from an insurance company on sales of new car-insurance policies proxies not only the volumes of new car sales but can be broken down into brands or geographies. Many vendors scrape websites for valuable data, ranging from app downloads and user reviews to airlines and hotel bookings. Social media sites can also be scraped for hints on consumer views and trends.

Typically, the datasets are large and require storage, access, and analysis using scalable data solutions for parallel processing, such as Hadoop and Spark; there are more than 1 billion websites with more than 10 trillion individual web pages, with 500 exabytes (or 500 billion gigabytes) of data, according to Deutsche Bank. And more than 100 million websites are added to the internet every year.

Real-time insights into a company's prospects, long before their results are released, can be gleaned from a decline in job listings on its website, the internal rating of its chief executive by employees on the recruitment site Glassdoor, or a dip in the average price of clothes on its website. This could be combined with satellite images of car parks and geolocation data from mobile phones that indicate how many people are visiting stores. On the other hand, strategic moves can be learned from a jump in job postings for specific functional areas or in certain geographies.

Among the most valuable sources is data that directly reveals consumer expenditures, with credit card information as a primary source. This data only offers a partial view of sales trends, but can offer vital insights when combined with other data. Point72, for instance, analyzes 80 million credit card transactions every day. We will explore the various sources, their use cases, and how to evaluate them in detail in `Chapter 3`, *Alternative Data for Finance*.

Investment groups have more than doubled their spending on alternative sets and data scientists in the past two years, as the asset management industry has tried to reinvigorate its fading fortunes. In December 2018, there were 375 alternative data providers listed on `alternativedata.org` (sponsored by provider Yipit).

Asset managers last year spent a total of $373 million on datasets and hiring new employees to parse them, up 60% on 2016, and will probably spend a total of $616 million this year, according to a survey of investors by `alternativedata.org`. It forecasts that overall expenditures will climb to over $1 billion by 2020. Some estimates are even higher: Optimus, a consultancy, estimates that investors are spending about $5 billion per year on alternative data, and expects the industry to grow 30% per year over the coming years.

As competition for valuable data sources intensifies, exclusivity arrangements are a key feature of data-source contracts, to maintain an informational advantage. At the same time, privacy concerns are mounting and regulators have begun to start looking at the currently largely unregulated data-provider industry.

Crowdsourcing of trading algorithms

More recently, several algorithmic trading firms have begun to offer investment platforms that provide access to data and a programming environment to crowd-source risk factors that become part of an investment strategy, or entire trading algorithms. Key examples include WorldQuant, Quantopian, and, launched in 2018, Alpha Trading Labs.

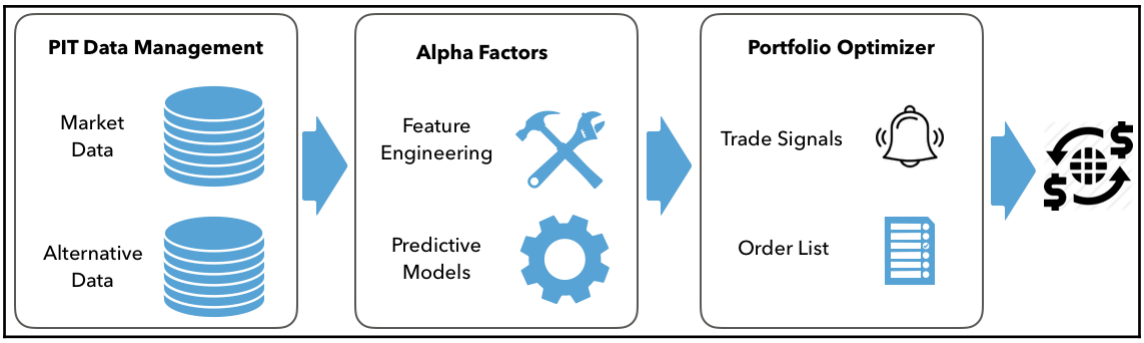
WorldQuant managed more than $5 billion for Millennium Management with $34.6 billion AUM since 2007 and announced in 2018 that it would launch its first public fund. It employs hundreds of scientists and many more part-time workers around the world in its alpha factory that organizes the investment process as a quantitative assembly line. This factory claims to have produced 4 million successfully tested alpha factors for inclusion in more complex trading strategies and is aiming for 100 million. Each alpha factor is an algorithm that seeks to predict a future asset price change. Other teams then combine alpha factors into strategies and strategies into portfolios, allocate funds between portfolios, and manage risk while avoiding strategies that cannibalize each other.

Design and execution of a trading strategy

ML can add value at multiple steps in the lifecycle of a trading strategy, and relies on key infrastructure and data resources. Hence, this book aims to addresses how ML techniques fit into the broader process of designing, executing, and evaluating strategies.

An algorithmic trading strategy is driven by a combination of alpha factors that transform one or several data sources into signals that in turn predict future asset returns and trigger buy or sell orders. `Chapter 2`, *Market and Fundamental Data* and `Chapter 3`, *Alternative Data for Finance* cover the sourcing and management of data, the raw material and the single most important driver of a successful trading strategy.

`Chapter 4`, *Alpha Factor Research* outlines a methodologically sound process to manage the risk of false discoveries that increases with the amount of data. `Chapter 5`, *Strategy Evaluation* provides the context for the execution and performance measurement of a trading strategy:



Let's take a brief look at these steps, which we will discuss in depth in the following chapters.

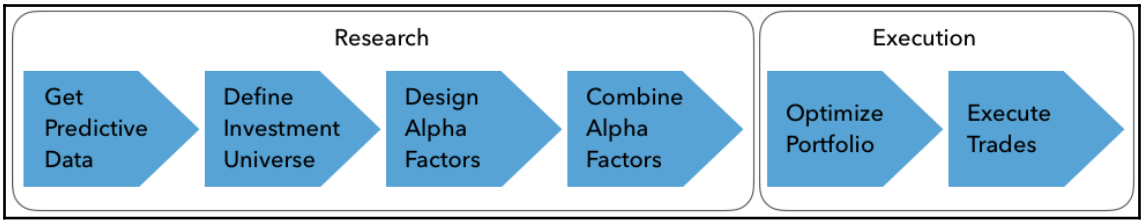
Sourcing and managing data

The dramatic evolution of data in terms of volume, variety, and velocity is both a necessary condition for and driving force of the application of ML to algorithmic trading. The proliferating supply of data requires active management to uncover potential value, including the following steps:

1. Identify and evaluate market, fundamental, and alternative data sources containing alpha signals that do not decay too quickly.
2. Deploy or access cloud-based scalable data infrastructure and analytical tools like Hadoop or Spark Sourcing to facilitate fast, flexible data access
3. Carefully manage and curate data to avoid look-ahead bias by adjusting it to the desired frequency on a **point-in-time** (**PIT**) basis. This means that data may only reflect information available and know at the given time. ML algorithms trained on distorted historical data will almost certainly fail during live trading.

Alpha factor research and evaluation

Alpha factors are designed to extract signals from data to predict asset returns for a given investment universe over the trading horizon. A factor takes on a single value for each asset when evaluated, but may combine one or several input variables. The process involves the steps outlined in the following figure:



The **Research phase** of the trading strategy workflow includes the design, evaluation, and combination of alpha factors. ML plays a large role in this process because the complexity of factors has increased as investors react to both the signal decay of simpler factors and the much richer data available today.

The development of predictive alpha factors requires the exploration of relationships between input data and the target returns, creative feature-engineering, and the testing and fine-tuning of data transformations to optimize the predictive power of the input.

The data transformations range from simple non-parametric rankings to complex ensemble models or deep neural networks, depending on the amount of signal in the inputs and the complexity of the relationship between the inputs and the target. Many of the simpler factors have emerged from academic research and have been increasingly widely used in the industry over the last several decades.

To minimize the risks of false discoveries due to data mining and because finance has been subject to decades of research that has resulted in several Nobel prizes, investors prefer to rely on factors that align with theories about financial markets and investor behavior. Laying out these theories is beyond the scope of this book, but the references will highlight avenues to dive deeper into this important framing aspect of algorithmic trading strategies.

To validate the signal content of an alpha factor candidate, it is necessary to obtain a robust estimate of its predictive power in environments representative of the market regime during which the factor would be used in a strategy. Reliable estimates require avoiding numerous methodological and practical pitfalls, including the use of data that induces survivorship or look-ahead biases by not reflecting realistic PIT information, or the failure to correct for bias due to multiple tests on the same data.

Signals derived from alpha factors are often individually weak, but sufficiently powerful when combined with other factors or data sources, for example, to modulate the signal as a function of the market or economic context.

Portfolio optimization and risk management

Alpha factors emit entry and exit signals that lead to buy or sell orders, and order execution results in portfolio holdings. The risk profiles of individual positions interact to create a specific portfolio risk profile. Portfolio management involves the optimization of position weights to achieve the desired portfolio risk and return a profile that aligns with the overall investment objectives. This process is highly dynamic to incorporate continuously-evolving market data.

The execution of trades during this process requires balancing the trader's dilemma: fast execution tends to drive up costs due to market impact, whereas slow execution may create implementation shortfall when the realized price deviates from the price that prevailed when the decision was taken. Risk management occurs throughout the portfolio-management process to adjust holdings or assume hedges, depending on observed or predicted changes in the market environment that impact the portfolio risk profile.

Strategy backtesting

The incorporation of an investment idea into an algorithmic strategy requires extensive testing with a scientific approach that attempts to reject the idea based on its performance in alternative out-of-sample market scenarios. Testing may involve simulated data to capture scenarios deemed possible but not reflected in historic data.

A strategy-backtesting engine needs to simulate the execution of a strategy realistically to achieve unbiased performance and risk estimates. In addition to the potential biases introduced by the data or a flawed use of statistics, the backtest engine needs to accurately represent the practical aspects of trade-signal evaluation, order placement, and execution in line with market conditions.

ML and algorithmic trading strategies

Quantitative strategies have evolved and become more sophisticated in three waves:

1. In the 1980s and 1990s, signals often emerged from academic research and used a single or very few inputs derived from market and fundamental data. These signals are now largely commoditized and available as ETF, such as basic mean-reversion strategies.
2. In the 2000s, factor-based investing proliferated. Funds used algorithms to identify assets exposed to risk factors like value or momentum to seek arbitrage opportunities. Redemptions during the early days of the financial crisis triggered the quant quake of August 2007 that cascaded through the factor-based fund industry. These strategies are now also available as long-only smart-beta funds that tilt portfolios according to a given set of risk factors.
3. The third era is driven by investments in ML capabilities and alternative data to generate profitable signals for repeatable trading strategies. Factor decay is a major challenge: the excess returns from new anomalies have been shown to drop by a quarter from discovery to publication, and by over 50% after publication due to competition and crowding.

There are several categories of trading strategies that use algorithms to execute trading rules:

- Short-term trades that aim to profit from small price movements, for example, due to arbitrage
- Behavioral strategies that aim to capitalize on anticipating the behavior of other market participants
- Programs that aim to optimize trade execution, and
- A large group of trading based on predicted pricing

The HFT funds discussed above most prominently rely on short holding periods to benefit from minor price movements based on bid-ask arbitrage or statistical arbitrage. Behavioral algorithms usually operate in lower liquidity environments and aim to anticipate moves by a larger player likely to significantly impact the price. The expectation of the price impact is based on sniffing algorithms that generate insights into other market participants' strategies, or market patterns such as forced trades by ETFs.

Trade-execution programs aim to limit the market impact of trades and range from the simple slicing of trades to match **time-weighted average pricing** (**TWAP**) or **volume-weighted average pricing** (**VWAP**). Simple algorithms leverage historical patterns, whereas more sophisticated algorithms take into account transaction costs, implementation shortfall or predicted price movements. These algorithms can operate at the security or portfolio level, for example, to implement multileg derivative or cross-asset trades.

Use Cases of ML for Trading

ML extracts signals from a wide range of market, fundamental, and alternative data, and can be applied at all steps of the algorithmic trading-strategy process. Key applications include:

- Data mining to identify patterns and extract features
- Supervised learning to generate risk factors or alphas and create trade ideas
- Aggregation of individual signals into a strategy
- Allocation of assets according to risk profiles learned by an algorithm
- The testing and evaluation of strategies, including through the use of synthetic data
- The interactive, automated refinement of a strategy using reinforcement learning

We briefly highlight some of these applications and identify where we will demonstrate their use in later chapters.

Data mining for feature extraction

The cost-effective evaluation of large, complex datasets requires the detection of signals at scale. There are several examples throughout the book:

- Information theory is a useful tool to extract features that capture potential signals and can be used in ML models. In `Chapter 4`, *Alpha Factor Research* we use mutual information to assess the potential values of individual features for a supervised learning algorithm to predict asset returns.
- In `Chapter 12`, *Unsupervised Learning*, we introduce various techniques to create features from high-dimensional datasets. In `Chapter 14`, *Topic Modeling*, we apply these techniques to text data.

- We emphasize model-specific ways to gain insights into the predictive power of individual variables. We use a novel game-theoretic approach called **SHapley Additive exPlanations** (**SHAP**) to attribute predictive performance to individual features in complex Gradient Boosting machines with a large number of input variables.

Supervised learning for alpha factor creation and aggregation

The main rationale for applying ML to trading is to obtain predictions of asset fundamentals, price movements or market conditions. A strategy can leverage multiple ML algorithms that build on each other. Downstream models can generate signals at the portfolio level by integrating predictions about the prospects of individual assets, capital market expectations, and the correlation among securities. Alternatively, ML predictions can inform discretionary trades as in the quantamental approach outlined above. ML predictions can also target specific risk factors, such as value or volatility, or implement technical approaches, such as trend following or mean reversion:

- In `Chapter 3`, *Alternative Data for Finance*, we illustrate how to work with fundamental data to create inputs to ML-driven valuation models
- In `Chapter 13`, *Working with Text Data*, `Chapter 14`, *Topic Modeling*, and `Chapter 15`, *Word Embeddings* we use alternative data on business reviews that can be used to project revenues for a company as an input for a valuation exercise.
- In `Chapter 8`, *Time Series Models*, we demonstrate how to forecast macro variables as inputs to market expectations and how to forecast risk factors such as volatility
- In `Chapter 18`, *Recurrent Neural Networks* we introduce **recurrent neural networks** (**RNNs**) that achieve superior performance with non-linear time series data.

Asset allocation

ML has been used to allocate portfolios based on decision-tree models that compute a hierarchical form of risk parity. As a result, risk characteristics are driven by patterns in asset prices rather than by asset classes and achieve superior risk-return characteristics.

In `Chapter 5`, *Strategy Evaluation* and `Chapter 12`, *Unsupervised Learning*, we illustrate how hierarchical clustering extracts data-driven risk classes that better reflect correlation patterns than conventional asset class definition.

Testing trade ideas

Backtesting is a critical step to select successful algorithmic trading strategies. Cross-validation using synthetic data is a key ML technique to generate reliable out-of-sample results when combined with appropriate methods to correct for multiple testing. The time series nature of financial data requires modifications to the standard approach to avoid look-ahead bias or otherwise contaminate the data used for training, validation, and testing. In addition, the limited availability of historical data has given rise to alternative approaches that use synthetic data:

- We will demonstrate various methods to test ML models using market, fundamental, and alternative that obtain sound estimates of out-of-sample errors.
- In `Chapter 20`, *Autoencoders and Generative Adversarial Nets*, we present GAN that are capable of producing high-quality synthetic data.

Reinforcement learning

Trading takes place in a competitive, interactive marketplace. Reinforcement learning aims to train agents to learn a policy function based on rewards.

- In `Chapter 21`, *Reinforcement Learning* we present key reinforcement algorithms like Q-Learning and the Dyna architecture and demonstrate the training of reinforcement algorithms for trading using OpenAI's gym environment.

Summary

In this chapter, we introduced algorithmic trading strategies and how ML has become a key ingredient for the design and combination of alpha factors, which in turn are the key drivers of portfolio performance. We covered various industry trends around algorithmic trading strategies, the emergence of alternative data, and the use of ML to exploit these new sources of informational advantages.

Furthermore, we introduced the algorithmic-trading-strategy design process, important types of alpha factors, and how we will use ML to design and execute our strategies. In the next two chapters, we will take a closer look at the oil that fuels any algorithmic trading strategy—the market, fundamental, and alternative data sources—using ML.