Preface

If you are reading this, you are probably aware that **machine learning (ML) has become a strategic capability** in many industries, including the investment industry. The explosion of digital data closely related to 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. These trends are enabling **novel approaches to quantitative investment** and are boosting the demand for the application of data science to both discretionary and algorithmic trading strategies.

The **scope of trading across asset classes** is vast because it ranges from equities and government bonds to commodities and real estate. This implies that a very large range of new alternative data sources may be relevant above and beyond the market and fundamental data that used to be at the center of most analytical efforts in the past.

You also may have 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.

Therefore, this book provides an integrated perspective on the application of ML to the domain of investment and trading. In this preface, we outline what you should expect, how we have organized the content to facilitate achieving our objectives, and what you need both to meet your goals and have fun in the process.

What to expect

This book aims to equip you with a strategic perspective, conceptual understanding, and practical tools to add value when applying ML to the trading and investment process. To this end, we cover ML as a key element in a process rather than a standalone exercise. Most importantly, we introduce an **end-to-end ML for trading (ML4T) workflow** that we apply to numerous use cases with relevant data and code examples.

The ML4T workflow starts with generating ideas and sourcing data and continues to extracting features, tuning ML models, and designing trading strategies that act on the models' predictive signals. It also includes simulating strategies on historical data using a backtesting engine and evaluating their performance.

First and foremost, the book demonstrates how you can extract signals from a diverse set of data sources and design trading strategies for different asset classes using a broad range of **supervised, unsupervised, and reinforcement learning algorithms**. In addition, it provides relevant mathematical and statistical background to facilitate tuning an algorithm and interpreting the results. Finally, it includes financial background to enable you to work with market and fundamental data, extract informative features, and manage the performance of a trading strategy.

The book emphasizes that investors can gain at least as much value from third-party data as 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 should not be a surprise that **this book does not provide investment advice** or ready-made trading algorithms. On the contrary, it in-

tends to communicate that ML faces many additional challenges in the trading domain, ranging from lower signal content to shorter time series that often make it harder to achieve robust results. In fact, we have included several examples that do not yield great results to avoid exaggerating the benefits of ML or understating the effort it takes to have a good idea, obtain the right data, engineer ingenious features, and design an effective strategy (with potentially attractive rewards).

Instead, you should find the book most useful as a guide to leveraging key ML algorithms to inform a trading strategy using a systematic workflow. To this end, we present a **framework that guides you through the ML4T process** of the following:

1. Sourcing, evaluating, and combining data for any investment objective
2. Designing and tuning ML models that extract predictive signals from the data
3. Developing and evaluating trading strategies based on the results

After reading this book, you will be able to begin designing and evaluating your own ML-based strategies and might want to consider participating in competitions or connecting to the API of an online broker and begin trading in the real world.

What's new in the second edition

This second edition emphasizes the end-to-end ML4T workflow, reflected in a new chapter on strategy backtesting (*Chapter 8, The ML4T Workflow – From Model to Strategy Backtesting*), a new appendix describing over 100 different alpha factors, and many new practical applications. We have also rewritten most of the existing content for clarity and readability.

The applications now use a broader range of data sources beyond daily US equity prices, including international stocks and ETFs, as well

as minute-frequency equity data to demonstrate an intraday strategy. Also, there is now broader coverage of alternative data sources, including SEC filings for sentiment analysis and return forecasts, as well as satellite images to classify land use.

Furthermore, the book replicates several applications recently published in academic papers. *Chapter 18*, *CNNs for Financial Time Series and Satellite Images*, demonstrates how to apply convolutional neural networks to time series converted to image format for return predictions. *Chapter 20*, *Autoencoders for Conditional Risk Factors and Asset Pricing*, shows how to extract risk factors conditioned on stock characteristics for asset pricing using autoencoders. *Chapter 21*, *Generative Adversarial Networks for Synthetic Time-Series Data*, examines how to create synthetic training data using generative adversarial networks.

All applications now use the latest available (at the time of writing) software versions, such as pandas 1.0 and TensorFlow 2.2. There is also a customized version of Zipline that makes it easy to include machine learning model predictions when designing a trading strategy.

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 an 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 in software and ML, you may be able to just skim or skip some introductory material in this area. Similarly, if your expertise is in investment, you will likely be familiar with some, or all, of the financial context that we provide for those with different backgrounds.

The book assumes that you want to continue to learn about this very dynamic area. To this end, it includes numerous end-of-chapter academic references and additional resources linked in the `README` files for each chapter in the companion GitHub repository.

You should be comfortable using Python 3 and scientific computing libraries like NumPy, pandas, or SciPy and look forward to 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. Similarly, basic knowledge of finance and investment will make some terminology easier to follow.

What this book covers

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

Part 1 – Data, alpha factors, and portfolios

The first part covers fundamental aspects relevant across trading strategies that leverage machine learning. It focuses on the data that drives the ML algorithms and strategies discussed in this book, outlines how you can engineer features that capture the data's signal content, and explains how to optimize and evaluate the performance of a portfolio.

Chapter 1, *Machine Learning for Trading – From Idea to Execution*, summarizes how and why ML became important for trading, describes the investment process, and outlines how ML can add value.

Chapter 2, *Market and Fundamental Data – Sources and Techniques*, covers how to source and work with market data, including exchange-provided tick data, and reported financials. It also demonstrates access to numerous **open source data providers** that we will rely on throughout this book.

Chapter 3, *Alternative Data for Finance – Categories and Use Cases*, explains categories and criteria to assess the exploding number of **sources and providers**. It also demonstrates how to create alternative datasets by scraping websites, for example, to collect earnings call transcripts for use with **natural language processing** (**NLP**) and sentiment analysis, which we cover in the second part of the book.

Chapter 4, *Financial Feature Engineering – How to Research Alpha Factors*, presents the process of creating and evaluating data transformations that capture the predictive signal and shows how to measure factor performance. It also summarizes insights from research into risk factors that aim to explain alpha in financial markets otherwise deemed to be efficient. Furthermore, it demonstrates how to **engineer alpha factors** using Python libraries offline and introduces the **Zipline** and **Alphalens** libraries to backtest factors and evaluate their predictive power.

Chapter 5, *Portfolio Optimization and Performance Evaluation*, introduces how to manage, optimize, and evaluate a portfolio resulting from the execution of a strategy. It presents risk metrics and shows how to apply them using the Zipline and **pyfolio** libraries. It also introduces methods to **optimize a strategy from a portfolio risk perspective**.

Part 2 – ML for trading – Fundamentals

The second part illustrates how fundamental supervised and unsupervised learning algorithms can inform trading strategies in the context of an end-to-end workflow.

Chapter 6, *The Machine Learning Process*, sets the stage by outlining how to formulate, train, tune, and evaluate the predictive performance of ML models in a systematic way. It also addresses **domain-specific concerns**, such as using cross-validation with financial time series to select among alternative ML models.

Chapter 7, *Linear Models – From Risk Factors to Return Forecasts*, shows how to use **linear and logistic regression** for inference and prediction and how to use regularization to manage the risk of overfitting. It demonstrates how to **predict US equity returns** or the direction of their future movements and how to evaluate the signal content of these predictions using Alphalens.

Chapter 8, *The ML4T Workflow – From Model to Strategy Backtesting*, integrates the various building blocks of the ML4T workflow thus far discussed separately. It presents an end-to-end perspective on the process of designing, simulating, and evaluating a trading strategy driven by an ML algorithm. To this end, it demonstrates how to **backtest an ML-driven strategy** in a historical market context using the Python libraries backtrader and Zipline.

Chapter 9, *Time-Series Models for Volatility Forecasts and Statistical Arbitrage*, covers univariate and multivariate time series diagnostics and models, including vector autoregressive models as well as ARCH/GARCH models for volatility forecasts. It also introduces cointegration and shows how to use it for a **pairs trading strategy using a diverse set of exchange-traded funds (ETFs)**.

Chapter 10, *Bayesian ML – Dynamic Sharpe Ratios and Pairs Trading*, presents 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**, for example, when evaluating **portfolio performance**.

Chapter 11, *Random Forests – A Long-Short Strategy for Japanese Stocks*, shows how to build, train, and tune nonlinear tree-based models for insight and prediction. It introduces tree-based ensembles and shows how random forests use bootstrap aggregation to overcome some of the weaknesses of decision trees. We then proceed to develop and backtest a **long-short strategy for Japanese equities**.

Chapter 12, *Boosting Your Trading Strategy*, introduces gradient boosting and demonstrates how to use the libraries XGBoost, LightBGM, and CatBoost for high-performance training and prediction. It reviews how to tune the numerous hyperparameters and interpret the model using **SHapley Additive exPlanation (SHAP)** values before building and evaluating a strategy that trades US equities based on LightGBM return forecasts.

Chapter 13, *Data-Driven Risk Factors and Asset Allocation with Unsupervised Learning*, shows how to use dimensionality reduction and clustering for algorithmic trading. It uses principal and independent component analysis to extract data-driven risk factors and generate **eigenportfolios**. It presents several clustering techniques and demonstrates the use of hierarchical clustering for **asset allocation**.

Part 3 – Natural language processing

Part 3 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 14, *Text Data for Trading – Sentiment Analysis*, demonstrates how to convert text data into a numerical format and applies the classification algorithms from Part 2 for sentiment analysis to large datasets.

Chapter 15, *Topic Modeling – Summarizing Financial News*, uses unsupervised learning to extract topics that 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* and to annual reports filed with the **Securities and Exchange Commission (SEC)**.

Chapter 16, *Word Embeddings for Earnings Calls and SEC Filings*, 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 for Trading*, introduces TensorFlow 2 and PyTorch, the most popular deep learning frameworks, which we will use throughout Part 4. It presents techniques for training and tuning, including regularization. It also builds and evaluates a **trading strategy for US equities**.

Chapter 18, *CNNs for Financial Time Series and Satellite Images*, covers **convolutional neural networks (CNNs)** that are very powerful for classification tasks with unstructured data at scale. We will introduce successful architectural designs, train a CNN on satellite data (for example, to predict economic activity), and use transfer learning to speed up training. We'll also replicate a recent idea to **convert financial time series into a two-dimensional image format** to leverage the built-in assumptions of CNNs.

Chapter 19, *RNNs for Multivariate Time Series and Sentiment Analysis*, shows how **recurrent neural networks (RNNs)** are useful for sequence-to-sequence modeling, including for univariate and multivariate time series to predict. It demonstrates how RNNs capture nonlin-

ear patterns over longer periods using word embeddings introduced in *Chapter 16* to **predict returns based on the sentiment expressed in SEC filings**.

Chapter 20, *Autoencoders for Conditional Risk Factors and Asset Pricing*, covers autoencoders for the nonlinear compression of high-dimensional data. It implements a recent paper that uses a deep autoencoder to learn both risk factor returns and factor loadings from the data while conditioning the latter on asset characteristics. We'll create a large US equity dataset with metadata and generate predictive signals.

Chapter 21, *Generative Adversarial Networks for Synthetic Time-Series Data*, presents one of the most exciting advances in deep learning. **Generative adversarial networks (GANs)** are capable of learning to reproduce synthetic replicas of a target data type, such as images of celebrities. In addition to images, GANs have also been applied to time-series data. This chapter replicates a novel approach to generate synthetic stock price data that could be used to train an ML model or backtest a strategy, and also evaluate its quality.

Chapter 22, *Deep Reinforcement Learning – Building a Trading Agent*, presents how **reinforcement learning (RL)** permits the design and training of agents that learn to optimize decisions over time in response to their environment. You will see how to create a custom trading environment and build an agent that responds to market signals using OpenAI Gym.

Chapter 23, *Conclusions and Next Steps*, summarizes the lessons learned and outlines several steps you can take to continue learning and building your own trading strategies.

Appendix, *Alpha Factor Library*, lists almost 200 popular financial features, explains their rationale, and shows how to compute them. It

also evaluates and compares their performance in predicting daily stock returns.

To get the most out of this book

In addition to the content summarized in the previous section, the hands-on nature of the book consists of over 160 Jupyter notebooks hosted on GitHub that demonstrate the use of ML for trading in practice on a broad range of data sources. This section describes how to use the GitHub repository, obtain the data used in the numerous examples, and set up the environment to run the code.

The GitHub repository

The book revolves around the application of ML algorithms to trading. The hands-on aspects are covered in Jupyter notebooks, hosted on GitHub, that illustrate many of the concepts and models in more detail. While the chapters aim to be self-contained, the code examples and results often take up too much space to include in their complete forms. Therefore, it is very important to view the notebooks that contain significant additional content while reading the chapter, even if you do not intend to run the code yourself.

The repository is organized so that each chapter has its own directory containing the relevant notebooks and a `README` file containing separate instructions where needed, as well as references and resources specific to the chapter's content. The relevant notebooks are identified throughout each chapter, as necessary. The repository also contains instructions on how to install the requisite libraries and obtain the data.

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

Data sources

We will use freely available historical data from market, fundamental, and alternative sources. *Chapter 2* and *Chapter 3* cover characteristics and access to these data sources and introduce key providers that we will use throughout the book. The companion GitHub repository just described 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
- International equity data from Stooq and using the yfinance library
- Various macro fundamental and benchmark data from the Federal Reserve
- Large Yelp business reviews and Twitter datasets
- EUROSAT satellite image data

Some of the data is large (several gigabytes), such as Nasdaq and SEC filings. The notebooks indicate when that is the case.

See the data directory in the root folder of the GitHub repository for instructions.

Anaconda and Docker images

The book requires Python 3.7 or higher and uses the Anaconda distribution. The book uses various conda environments for the four parts

to cover a broad range of libraries while limiting dependencies and conflicts.

The installation directory in the GitHub repository contains detailed instructions. You can either use the provided Docker image to create a container with the necessary environments or use the `.yml` files to create them locally.

Download the example code files

You can download the example code files for this book from your account at http://www.packtpub.com. If you purchased this book elsewhere, you can visit http://www.packtpub.com/support and register to have the files emailed directly to you.

You can download the code files by following these steps:

1. Log in or register at http://www.packtpub.com.
2. Select the **SUPPORT** tab.
3. Click on **Code Downloads & Errata**.
4. Enter the name of the book in the **Search** box and follow the on-screen instructions.

Once the file is downloaded, please make sure that you unzip or extract the folder using the latest version of your preferred compression tool:

- WinRAR or 7-Zip for Windows
- Zipeg, iZip, or UnRarX for Mac
- 7-Zip or PeaZip for Linux

The code bundle for the book is also hosted on GitHub at https://github.com/PacktPublishing/Machine-Learning-for-Algorithmic-Trading-Second-Edition. We also have other code bundles from our rich catalog of books and videos available at https://github.com/PacktPublishing/. Check them out!

Download the color images

We also provide a PDF file that has color images of the screenshots/diagrams used in this book. You can download it here: https://static.packt-cdn.com/downloads/9781839217715_ColorImages.pdf.

Conventions used

There are a number of text conventions used throughout this book.

`CodeInText` : Indicates code words in text, database table names, folder names, filenames, file extensions, pathnames, dummy URLs, user input, and Twitter handles. For example, "The `compute_factors()` method creates a `MeanReversion` factor instance and creates long, short, and ranking pipeline columns."

A block of code is set as follows:

```
from pykalman import KalmanFilter
kf = KalmanFilter(transition_matrices = [1],
                  observation_matrices = [1],
                  initial_state_mean = 0,
                  initial_state_covariance = 1,
                  observation_covariance=1,
                  transition_covariance=.01)
```

Bold: Indicates a new term, an important word, or words that you see on the screen, for example, in menus or dialog boxes, also appear in the text like this. For example, "The **Python Algorithmic Trading Library** (**PyAlgoTrade**) focuses on backtesting and offers support for paper trading and live trading."

	Informational notes appear like this.

Get in touch

Feedback from our readers is always welcome.

General feedback: Email `feedback@packtpub.com` , and mention the book's title in the subject of your message. If you have questions about any aspect of this book, please email us at `questions@packtpub.com` .

Errata: Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you have found a mistake in this book we would be grateful if you would report this to us. Please visit, `http://www.packtpub.com/submit-errata`, selecting your book, clicking on the Errata Submission Form link, and entering the details.

Piracy: If you come across any illegal copies of our works in any form on the Internet, we would be grateful if you would provide us with the location address or website name. Please contact us at `copyright@packtpub.com` with a link to the material.

If you are interested in becoming an author: If there is a topic that you have expertise in and you are interested in either writing or contributing to a book, please visit `http://authors.packtpub.com`.

Reviews

Please leave a review. Once you have read and used this book, why not leave a review on the site that you purchased it from? Potential readers can then see and use your unbiased opinion to make purchase decisions, we at Packt can understand what you think about our products, and our authors can see your feedback on their book. Thank you!

For more information about Packt, please visit packtpub.com.

Download a free PDF copy of this book

Thanks for purchasing this book!

Do you like to read on the go but are unable to carry your print books everywhere? Is your eBook purchase not compatible with the device of your choice?

Don’t worry, now with every Packt book you get a DRM-free PDF version of that book at no cost.

Read anywhere, any place, on any device. Search, copy, and paste code from your favorite technical books directly into your application.

The perks don’t stop there, you can get exclusive access to discounts, newsletters, and great free content in your inbox daily

Follow these simple steps to get the benefits:

1. Scan the QR code or visit the link below

https://packt.link/free-ebook/9781839217715

2. Submit your proof of purchase
3. That’s it! We’ll send your free PDF and other benefits to your email directly