

To my parents, Yanxia and Ximing.

# Contents

# Preface

## From Artificial Intelligence to Everyday Life

Artificial Intelligence has evolved from a speculative concept to a transformative force that shapes how we live, work, and think. At the heart of this transformation lies **machine learning**—a field that enables computers to learn patterns, make decisions, and generate insights from data. From recommendation systems and autonomous vehicles to quantitative finance and healthcare, machine learning drives the intelligent systems that define our era.

Therefore, understanding and mastering machine learning has become one of the most valuable skills of our time. It empowers individuals to not only analyze data but also to create intelligent systems that can learn, adapt, predict, and innovate.

## The Challenge of Learning Machine Learning

Yet, learning machine learning itself often feels daunting. Many students and professionals encounter the same obstacles: the mathematics appears abstract, the formulas overwhelming, and the connection between theory and practice elusive. Traditional textbooks, while rigorous, tend to emphasize symbolic derivations and statistical proofs, leaving readers with limited geometric intuition or hands-on understanding.

Many readers, despite their passion for learning machine learning, often lack the necessary mathematical foundation, especially, **linear algebra**, and **probability and statistics**. Even for those who studied these subjects in university, the knowledge often remains fragmented and disconnected from practical machine learning applications. When faced with real-world problems or algorithms, they find themselves struggling to bridge the gap between **mathematical theory** and **computational implementation**.

This book aims to close that gap. Rather than treating mathematics as a separate, abstract prerequisite, we integrate it into the learning journey itself. Each concept is introduced visually, explored interactively through code, and connected directly to machine learning algorithms. Through this approach, readers will not only revisit key mathematical ideas but also understand **why they matter** and **how they empower intelligent systems**.

## Learn by Seeing and Doing

This book was born from the belief that **machine learning can be made intuitive, visual, and practical**. The author observed that what learners often lack is not intelligence—but intuition. The absence of geometric visualization and coding practice makes even simple algorithms feel opaque. To bridge this gap, the book adopts an approach that blends **mathematical reasoning**, **geometric interpretation**, and **Python programming**.

Using clear figures, intuitive explanations, and Jupyter Notebook implementations, this book invites readers to “see” how algorithms work. Each concept is illustrated through geometric visualizations that reveal the underlying logic of learning. Each algorithm is paired with executable Python code that lets readers experiment, modify, and discover patterns themselves. Through this dual lens of visualization and practice, abstract mathematics becomes concrete understanding.

When explaining these classic machine learning algorithms, we also interweave the fundamental concepts of linear algebra, probability, and statistics that underpin them. These foundational topics are also presented using a combination of visualizations and hands-on programming, allowing readers to develop a deeper, intuitive understanding of both the mathematics and its practical application in machine learning.

## What This Book Covers

The book covers more than twenty classic algorithms across four major areas of machine learning—**regression, dimensionality reduction, classification, and clustering**. Each algorithm is explained step by step, from intuition to Python implementation, helping readers build a comprehensive mental yet practical roadmap of the field.

Along the way, we will also explore the **mathematical foundations** that underpin these algorithms—**linear algebra, and probability and statistics**. These topics are not treated as dry theory, but as living tools that reveal *why* machine learning works. Understanding these mathematical structures will allow you to look beneath the surface of algorithms and see the elegant principles that govern them.

However, this book **does not cover neural networks, deep learning, natural language processing (NLP), or large language models (LLMs)**. The focus here is on building a solid understanding of **classical machine learning algorithms**.

While neural networks and deep learning are important and rapidly growing areas, they often require more advanced mathematical background and computing resources. By concentrating on traditional machine learning techniques, this book ensures that readers develop **strong intuition, clear visual understanding, and hands-on coding experience** before moving on to more complex topics.

For readers interested in deep learning, NLP, or LLMs, the knowledge gained from this book will serve as a **foundation**, making it easier to understand and implement more advanced models in the future. This approach helps learners progress **step by step**, from fundamental concepts to state-of-the-art techniques, without feeling overwhelmed.

## Interactive Learning Resources

All accompanying code is provided as Jupyter Notebooks to ensure interactive learning and experimentation. The complete set of materials can be downloaded from the following repository:

<https://github.com/visualize-ml/Machine-Learning-Made-Visual-with-Python>

If a figure in this book is generated by the accompanying code, its caption will include the name of the corresponding Jupyter Notebook.

If you find any issues or errors in the book or its accompanying code, please feel free to open an issue on GitHub — the author will respond and provide updates promptly.

## How to Run Jupyter Notebooks

The easiest way to run Jupyter Notebooks is by installing **Anaconda**, a free and open-source distribution of Python and R for scientific computing and data science. Anaconda simplifies package management and deployment, providing everything you need to start coding immediately, including Python, commonly used libraries, and tools like **Jupyter Notebook** and **JupyterLab**.

**Jupyter Notebook** is an interactive coding environment that allows you to combine code, visualizations, and explanatory text in a single document. **JupyterLab** is the next-generation interface for Project Jupyter, offering more flexible layouts, multiple tabs, integrated terminals, and enhanced features for data exploration, visualization, and machine learning experiments.

If you are new to JupyterLab, please refer to the **official getting started guide** for installation, interface navigation, and usage instructions:

[https://jupyterlab.readthedocs.io/en/stable/getting\\_started/overview.html](https://jupyterlab.readthedocs.io/en/stable/getting_started/overview.html)

By installing Anaconda, you can launch both Jupyter Notebook and JupyterLab easily without worrying about configuring Python, packages, or dependencies manually. It is ideal for beginners and professionals who want a smooth setup for experimenting with machine learning, data analysis, and visualization.

You can download Anaconda from:

<https://www.anaconda.com/>

## How to Learn Python Programming

With the rise of tools like **ChatGPT**, learning to program has become more accessible than ever. Readers can use the **accompanying Jupyter Notebooks** provided with this book to experiment, modify, and visualize concepts interactively. Whenever you encounter a question—whether about syntax, debugging, or mathematical reasoning—you can consult ChatGPT or similar AI assistants. This **goal-oriented way of learning programming**, where coding is driven by curiosity and purpose, is in the author’s view the most effective and rewarding approach.

Therefore, this book does not focus on teaching the basics of Python programming from scratch. Instead, it assumes that readers will **learn Python as they build**, acquiring new coding skills naturally through hands-on exploration.

For those who need a quick refresher or wish to dive deeper into specific libraries used throughout this book, the following official resources are highly recommended:

- **NumPy Quickstart Guide:** <https://numpy.org/doc/stable/user/quickstart.html>
- **Pandas 10-Minute Tutorial:** [https://pandas.pydata.org/docs/user\\_guide/10min.html](https://pandas.pydata.org/docs/user_guide/10min.html)
- **Matplotlib Tutorials:** <https://matplotlib.org/stable/tutorials/index.html>
- **Scikit-learn User Guide:** [https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html)

These libraries form the **core toolkit of modern data science and machine learning**, and mastering them will significantly enhance your ability to turn mathematical ideas into working code and meaningful visualizations.

## Who This Book Is For

This book is written for a broad audience—students, educators, developers, and self-learners—anyone who seeks a hands-on, visual, and conceptually clear understanding of machine learning. Whether you are studying data science, engineering, or quantitative finance, or simply curious about how machines learn, this book will help you **build intuition before abstraction**, and **understanding before memorization**.

In essence, this book transforms learning from reading formulas to **seeing** and **creating** knowledge. By the end, readers will not only understand how algorithms work—but also why they work, and how to bring them to life through code and imagination.

## Acknowledgement

A special note of gratitude goes to **Mr. Dacheng Luan** of **Tsinghua University Press**, the executive editor of the Chinese edition of this book. His vision, professionalism, and dedication played a pivotal role in bringing this project to life.

The Chinese edition of this book is part of a larger seven-volume series published by Tsinghua University Press. This series offers a comprehensive learning path that spans **Python programming fundamentals, data visualization, algebra, geometry, calculus, linear algebra, probability and statistics, optimization, data processing, graphs and networks, and machine learning**. Together, these volumes are designed to guide

readers from the very foundations of programming and mathematics all the way to mastering modern machine learning.

The English edition you are now reading is centered around the **Machine Learning volume**, but it also **integrates foundational topics** from the previous six books. It distills essentials from the Chinese series and weaves them into a coherent framework that supports a deeper understanding of machine learning. For readers who wish to explore the broader series or access supplementary learning materials, the **Chinese drafts, Jupyter Notebooks, and source files** from all seven volumes are freely available on GitHub at:

<https://github.com/Visualize-ML>

Mr. Luan's editorial insight and unwavering support were instrumental in the successful publication and distribution of the Chinese series. His commitment not only ensured the quality and coherence of the books but also made it possible for a wide audience of learners to access clear, visual, and practical resources in this fast-evolving field. For that, I extend my deepest appreciation and heartfelt thanks.

Finally, I would like to express my heartfelt gratitude to the readers of the **Chinese edition**—both those who purchased the printed books and those who followed the free online versions on GitHub. Your encouragement, feedback, and reviews have been the **driving force** that kept this project alive and growing. Every comment, suggestion, and note of appreciation has inspired me to keep improving and expanding this series.

I also hope that the publication of this **English edition** will bring the same spirit of curiosity and collaboration to a wider audience around the world. May it help more learners—students, researchers, and professionals alike—discover the joy of understanding machine learning through visualization, mathematics, and code.

姜伟生 James Weisheng Jiang  
Toronto, Canada  
November 2025