Turn data into reliable, well-evaluated machine learning models.

This book provides a structured and practical approach to data modeling and machine learning with Python. Moving beyond isolated techniques, it presents a complete workflow—from data preparation and statistical modeling to machine learning, evaluation, and optimization.

You will learn how to build models that not only perform well, but also generalize, remain interpretable, and withstand real-world challenges.

WHAT YOU WILL LEARN

- Prepare and transform data for modeling
- Apply statistical modeling techniques, including regression and generalized linear models
- Understand core machine learning principles such as training, validation, and the bias–variance tradeoff
- Build classification, regression, and ensemble models
- Evaluate model performance using appropriate metrics and validation strategies
- Improve models through hyperparameter tuning and systematic optimization
- Interpret model behavior using modern explainability techniques

ABOUT THE AUTHOR

Dr. Shouke Wei is a researcher, scientist, and entrepreneur specializing in data analysis and modeling, wavelet-based signal processing, and AI-driven applications.

He earned his Ph.D. from Brandenburg University of Technology Cottbus–Senftenberg (Germany) and conducted postdoctoral research at Eawag (Switzerland).

He held research positions at the University of British Columbia (Canada) and served as a distinguished and adjunct professor at multiple universities (China).

https://press.deepsim.ca/

9 781067 559236

Part of the *Practical Data Science with Python* series

Practical Data Modeling and Machine Learning with Python SHOUKE WEI

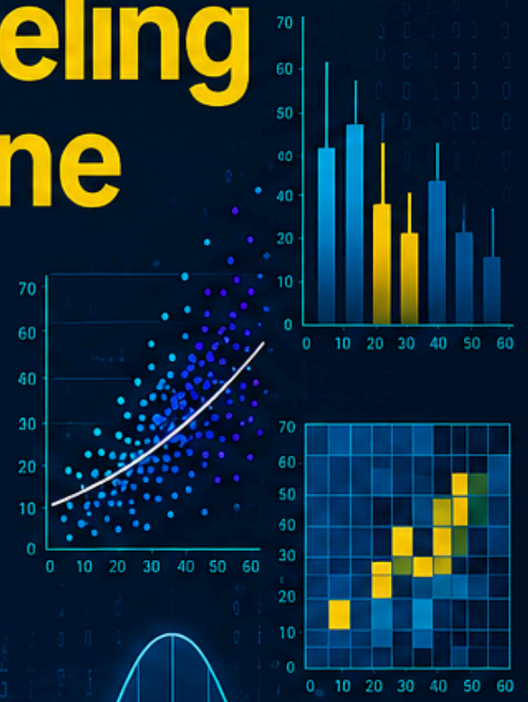
DEEPSIM PRESS

DEEPSIM PRESS

Practical Data Modeling and Machine Learning with Python

From Data Preparation to Model Evaluation and Optimization

A complete workflow for practical data modeling



SHOUKE WEI

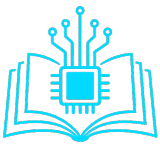
Practical Data Modeling and Machine Learning with Python

From Data Preparation to Model Evaluation and Optimization

Practical Data Modeling and Machine Learning with Python

From Data Preparation to Model Evaluation and Optimization

Shouke Wei



DEEPSIM
PRESS

Practical Data Modeling and Machine Learning with Python

First Edition, 2026

For permission requests, contact the publisher:
Email: shouke.wei@deepsim.ca

ISBN 978-1-0675592-4-3 (Hardcover)
ISBN 978-1-0675592-5-0 (Paperback)
ISBN 978-1-0675592-3-6 (eBook)
DOI 10.5281/zenodo.19753396

Published by
Deepsim Press
An independent imprint of Deepsim Intelligence Technology Inc.
Abbotsford, British Columbia, Canada
https://press.deepsim.ca

This book is intended for educational and professional readers.

For resources, updates, and companion code, visit:
https://press.deepsim.ca/data-model

About the Author

Shouke Wei, Ph.D., is a researcher, scientist, and entrepreneur specializing in intelligent IoT systems, robotics, big data analytics, modeling and forecasting, early-warning systems, and edge computing. With academic and industry experience across Europe, North America, and Asia, Dr. Wei is recognized for bridging advanced theory with real-world, production-ready systems.

Dr. Wei earned his Ph.D. in Environmental and Resource Management from the Department of Ecosystems and Environmental Informatics at Brandenburg University of Technology Cottbus–Senftenberg (Germany). He conducted postdoctoral research at the Swiss Federal Institute of Aquatic Science and Technology (Eawag, Switzerland) and held research positions at the University of British Columbia (Canada). He has also served as a distinguished and adjunct professor at multiple institutions in China.

Dr. Wei currently serves as CEO and Chief Scientist of Deepsim Intelligent Technology Inc. (Canada) and Shandong Deepsim Intelligent Technology Co., Ltd. (China). He is also a Postdoctoral Co-Supervisor at the Shandong Postdoctoral Innovation Practice Base and Director of Qilu Artificial Intelligence and Digital Manufacturing Innovation at Shandong Deepsim Intelligent Technology Co., Ltd., China.

Dr. Wei has led or contributed to 19 major international research projects and the development of numerous intelligent systems, including autonomous water-quality monitoring vessels, AI-based environmental early-warning platforms, medical image diagnosis systems, precision agriculture robots, and autonomous service robots.

His scholarly contributions include over 40 peer-reviewed publications, 8 books on practical wavelet transform applications, data analysis, and data modeling, more than 500 technical tutorial articles, six patents, and over 30 software copyrights. His work focuses on making advanced computational methods—particularly data analysis, modeling, and wavelet-based signal processing—accessible, practical, and impactful for researchers and practitioners worldwide.

For more information, visit: https://press.deepsim.ca/shouke/

Contents

Preface

Data is abundant, but understanding is not. Between raw data and meaningful decisions lies a crucial process: the ability to build, evaluate, and refine models that capture structure in the world.

This book, *Practical Data Modeling and Machine Learning with Python*, focuses on that process.

It is the second volume in the *Practical Data Science with Python* series. The first book introduced data exploration and visualization—how to observe patterns, clean data, and ask the right questions. This volume moves one step further: from understanding data to **modeling it**, and from intuition to **quantitative reasoning**.

Purpose of This Book

The central goal of this book is not simply to present algorithms, but to develop a coherent approach to **data modeling**.

In practice, modeling is not a single step. It is a system:

- defining a problem clearly
- preparing data carefully
- selecting appropriate models
- evaluating performance rigorously
- refining and improving results

This book follows that system. It integrates statistical modeling and modern machine learning into a unified workflow, emphasizing both **principles** and **practical implementation**.

What This Book Covers

This book is organized into six parts, each corresponding to a key stage in the data modeling and machine learning workflow.

Part I — Foundations of Data Modeling introduces the fundamental concepts of data modeling and analytical thinking. It covers the practical setup of a Python environment and the essential steps of data preparation and feature engineering, establishing a solid foundation for all subsequent work.

Part II — Statistical Modeling Foundations provides the necessary statistical background for modeling. Topics such as probability distributions, estimation, and hypothesis testing are presented with a focus on interpretation and practical relevance.

Part III — Statistical Modeling Techniques develops core modeling approaches, including linear regression, regularization, and generalized linear models. These methods form the bridge between classical statistics and modern machine learning.

Part IV — Foundations of Machine Learning introduces the principles that govern machine learning systems, including training and validation strategies, the bias–variance tradeoff, and the role of cross-validation and preprocessing pipelines in building reliable models.

Part V — Core Machine Learning Models presents practical machine learning methods, including classification models, regression techniques, and ensemble approaches. Emphasis is placed on understanding model behavior and comparing different methods in realistic settings.

Part VI — Model Evaluation and Optimization focuses on assessing and improving models. It covers performance metrics, validation strategies, hyperparameter tuning, and model interpretation techniques, providing a complete framework for building robust and trustworthy models.

Together, these parts form a coherent progression from data preparation to model evaluation and optimization, reflecting the full lifecycle of data-driven modeling.

A Practical Perspective

This is a practical book, but "practical" here does not simply mean applying ready-made code. It means understanding how modeling decisions are made, why certain methods work, and how different components of the workflow interact in real scenarios.

All concepts are illustrated with Python-based examples, using commonly adopted libraries and reproducible workflows. Rather than presenting isolated techniques, the book emphasizes how to build complete modeling pipelines—from data preparation and feature engineering to model training, validation, and evaluation.

Special attention is given to issues that arise in practice: data leakage, overfitting, improper validation, and the misuse of evaluation metrics. These are not treated as side topics, but as central concerns in building reliable models.

At the same time, theoretical ideas are introduced where they provide essential insight. Concepts such as the bias–variance tradeoff, statistical assumptions, and model interpretability are explained in the context of real modeling tasks, helping to connect intuition with implementation.

The goal is not only to show how to build models, but to develop the ability to judge whether a model is appropriate, reliable, and meaningful in practice.

Who This Book Is For

This book is intended for readers who:

- have basic familiarity with Python
- understand fundamental data analysis concepts

- want to move into modeling and machine learning in a structured way

Readers who have worked through the first volume of this series will find a natural continuation here. Others can still follow, provided they are comfortable with basic data manipulation and visualization.

How to Use This Book

The chapters are arranged to support a sequential reading path:

1. build foundational understanding
2. learn modeling techniques
3. understand machine learning principles
4. apply evaluation and optimization methods

More experienced readers may also use the book as a reference, focusing on specific sections such as model evaluation or ensemble methods.

What Comes Next

This book focuses on building and evaluating models.

The next volume in the series, *Advanced Data Modeling and Forecasting with Python*, extends these ideas into more specialized domains, including time series forecasting, deployment, and real-world applications.

Together, the series forms a complete pathway:

- understanding data
- building models
- applying them in real systems

Final Thoughts

All models are approximations. They simplify reality, and they inevitably make assumptions.

The goal of data modeling is not perfection, but **clarity, robustness, and usefulness**.

If this book helps you think more carefully about how models are constructed, evaluated, and applied, then it has fulfilled its purpose.

Shouke Wei, PhD
Deepsim Intelligent Technology Inc.
Deepsim Academy
Abbotsford, Canada
April 18, 2026

Acknowledgments

This book is the result of a long process of learning, practice, and reflection.

I would like to acknowledge the many sources—both direct and indirect—that contributed to its development. The ideas presented here draw upon the broader fields of statistics, machine learning, and data science, as well as years of practical experience in applying these concepts to real problems.

I am especially grateful for the open-source community, whose tools and libraries—particularly within the Python ecosystem—make modern data science both accessible and powerful. Without these contributions, much of the practical work demonstrated in this book would not be possible.

I also appreciate the role of teaching and continuous learning in shaping this material. Explaining concepts to others, refining examples, and revisiting fundamental ideas have all helped clarify the structure and presentation of this book.

Finally, I would like to thank my family for their support, patience, and understanding throughout the writing process.

Notation and Abbreviations

Symbol / Abbreviation	Meaning
Data and indices	
i	Index over observations
j	Index over predictors / features
n	Number of observations (sample size)
p	Number of predictors (excluding intercept)
Y_i	Response variable for observation i
$\mathbf{Y}$	Response vector ($n \times 1$)
X_j	j-th predictor variable
$\mathbf{X}$	Design matrix ($n \times (p+1)$)
$\mathbf{x}_i$	Feature vector for observation i ($p \times 1$)
$\mathbf{x}_{i,-j}$	Feature vector of observation i with feature j held fixed
y_i	Observed response (lower-case scalar)
$\mathcal{Y}$	Output space (label set)
c_k	Class label k
K	Number of classes
$\bar{Y}$	Sample mean of Y
$\hat{y}_i$	Fitted / predicted value for observation i
R_m	m-th leaf region of a regression tree
CV	Coefficient of variation: CV $= \sigma/\mu$
M	Number of leaf regions in a decision / regression tree
p_k	Proportion of class k in a node
n_L, n_R	Number of observations in left / right child node

Symbol / Abbreviation	Meaning
$\mathcal{S}$	Set of observations at a tree node
b	Boosting iteration index
B	Total number of trees / base learners in an ensemble
$\bar{y}$	Sample mean of the target
Parameters	
β_0	Intercept
β_1	Slope (simple regression)
$\widehat{\beta}_j$	Estimated coefficient for predictor j
Errors and residuals	
e_i	Residual: $e_i = y_i - \widehat{y}_i$
Cross-validation	
k	Number of folds in k-fold cross-validation
Abbreviations	
EDA	Exploratory Data Analysis
OLS	Ordinary least squares
GLM	Generalized linear model
IRLS	Iteratively reweighted least squares
Ridge	Ridge regression (ℓ_2-penalized OLS)
Lasso	Least absolute shrinkage and selection operator (ℓ_1-penalized OLS)
VIF	Variance inflation factor
AIC	Akaike information criterion
RMSE	Root mean squared error
MAE	Mean absolute error
MAPE	Mean absolute percentage error
ROC	Receiver operating characteristic
AUC	Area under the ROC curve (explicit form)
AP	Average Precision
BS	Brier score
OR	Odds ratio
RR	Rate ratio
CV	Cross-validation
LOOCV	Leave-one-out cross-validation

Symbol / Abbreviation	Meaning
NCV	Nested cross-validation
i.i.d.	Independent and identically distributed
PMF	Probability mass function
PDF	Probability density function
CDF	Cumulative distribution function
CLT	Central Limit Theorem
MI	Mutual information (feature selection)
RFE	Recursive feature elimination
OHE	One-hot encoding
KS	Kolmogorov-Smirnov test
Q-Q plot	Quantile-quantile plot
Bootstrap	Non-parametric resampling method
MC	Monte Carlo simulation
CI	Confidence interval
SE	Standard error
HSD	Honestly Significant Difference (Tukey)
FDR	False Discovery Rate
BH	Benjamini–Hochberg procedure
ANOVA	Analysis of Variance
kNN	k-nearest neighbours
SVM	Support vector machine
SVR	Support vector regression
CART	Classification and regression trees
RF	Random Forest
GBM	Gradient Boosting Machine
OOB	Out-of-bag (error / score)
PDP	Partial dependence plot
ICE	Individual conditional expectation
SHAP	SHapley Additive exPlanations
TimeSeriesSplit	Temporal expanding-window CV
Time series	
$y_t = T_t + S_t + R_t$	Additive decomposition of a time series
y_t	Observed value of a time series at time t
T_t	Trend component at time t

Symbol / Abbreviation	Meaning
S_t	Seasonal component at time t
m	Seasonal period
d	Order of regular differencing
D	Order of seasonal differencing
B	Backshift (lag) operator
ϕ_j	AR coefficient at lag j
$\phi(B)$	AR polynomial in the backshift operator
θ_j	MA coefficient at lag j
$\theta(B)$	MA polynomial in the backshift operator
$\Phi_P(B^m)$	Seasonal AR polynomial
$\Theta_Q(B^m)$	Seasonal MA polynomial
ρ_k	Autocorrelation at lag k
γ_k	Autocovariance at lag k
H	Forecast horizon
MAPE	Mean absolute percentage error
STL	Seasonal and Trend decomposition using Loess
ARIMA	Autoregressive Integrated Moving Average
SARIMA	Seasonal ARIMA
Unsupervised learning	
k	Number of clusters
C_c	Cluster c ($c = 1, \ldots, k$)
$\boldsymbol{\mu}_c$	Centroid of cluster c
SS_B	Between-cluster sum of squares
$\mathbf{V}$	Matrix of eigenvectors (principal component directions)
$\mathbf{V}_d$	First d eigenvectors (truncated)
λ_j	j-th eigenvalue of the sample covariance matrix
$\mathbf{S}$	Sample covariance matrix
$\mathbf{Z}$	PCA projection (scores matrix)
$f_\phi(\cdot)$	Autoencoder encoder function
$g_\theta(\cdot)$	Autoencoder decoder function
$\mathbf{z}$	Latent code (bottleneck representation)
ARI	Adjusted Rand Index

Symbol / Abbreviation	Meaning
PCA	Principal Component Analysis
KL	Kullback-Leibler divergence
Imbalanced learning	
π_1	Minority class proportion
F_β	F-score with weight β
F_1	F-score with $\beta = 1$ (special case)
MCC	Matthews Correlation Coefficient
g_i	ADASYN synthetic count for minority point i
Δ_i	Number of majority-class neighbours of minority point i
SMOTE	Synthetic Minority Over-sampling Technique
ADASYN	Adaptive Synthetic Sampling
PSI	Population Stability Index
Hybrid modeling	
$\mathcal{D}_{k(i)}$	Fold containing observation i
$\psi_{j,n}[k]$	Dilated and translated wavelet at scale j, shift n
j	Wavelet scale (resolution level)
OOF	Out-of-fold
Model deployment and monitoring	
API	Application Programming Interface
REST	Representational State Transfer
PSI	Population Stability Index
Financial time series	
σ_t	Conditional standard deviation (volatility)
ε_t	Innovation (GARCH mean equation)
$\alpha + \beta$	GARCH persistence
S	Sharpe ratio
p	Directional accuracy
T	Number of trading periods per year
$f_t(\cdot)$	Walk-forward model trained on data up to time t
ARCH	Autoregressive Conditional Heteroscedasticity

Symbol / Abbreviation	Meaning
Customer segmentation and CLV	
F (Frequency)	Number of distinct invoices / visits
y_i (CLV)	Six-month customer lifetime value proxy
Gap(k)	Gap statistic at k clusters
RFM	Recency, Frequency, Monetary
ARI	Adjusted Rand Index
BIC	Bayesian Information Criterion
End-to-end modeling framework	
SHAP	SHapley Additive exPlanations

Part I

Foundations of Data Modeling

1 Introduction to Data Modeling

Data modeling is the process of creating a mathematical or computational representation of real-world phenomena so that we can understand, predict, and make decisions based on data (Provost and Fawcett 2013). In today's data-rich world, effective data modeling has become essential across industries — from finance and healthcare to retail, energy, and technology. Good models turn raw data into actionable insights, help forecast future outcomes, optimize processes, and support evidence-based decision making.

The importance of data modeling has grown dramatically with the explosion of available data and computing power. Organizations that master modeling techniques gain significant competitive advantages through better risk management, more accurate forecasts, improved customer understanding, and more efficient operations. Whether you are a data scientist, analyst, statistician, or business professional, understanding the principles of data modeling is a foundational skill (James et al. 2023).

This chapter lays the groundwork for the rest of the book *Practical Data Modeling and Machine Learning with Python*. We will explore what data modeling really means, how it differs from simple data analysis, the contrasting perspectives of statistical modeling and machine learning, and the end-to-end workflow that turns a business problem into a deployed, useful model.

1.1 Chapter Overview

This chapter covers the following topics:

- What is Data Modeling?
- From Data Analysis to Modeling
- Statistical vs Machine Learning Perspectives
- End-to-End Modeling Workflow

By the end of this chapter, you will be able to:

- Explain the purpose and value of data modeling in practical applications
- Distinguish between exploratory data analysis and formal modeling
- Compare and contrast the statistical and machine learning approaches to modeling
- Describe the key stages of an end-to-end modeling workflow

1.2 What is Data Modeling?

Data modeling is the art and science of constructing abstract representations — *models* — that capture the underlying patterns, relationships, and structures within data. A model is a simplified approximation of reality, deliberately designed to answer specific questions or solve particular problems (Hastie et al. 2009).

The word *model* can mean different things in different communities. In statistics, a model is typically a probability distribution or set of equations describing how data were generated (Gelman et al. 2013). In machine learning, a model is often a learned function mapping inputs to outputs. In practice, both views are useful, and this book draws on both.

Models can take many forms, such as:

- **Statistical models** — linear regression, logistic regression, generalized linear models (GLMs), ARIMA
- **Machine learning models** — random forests, gradient boosting, neural networks, deep learning architectures
- **Time series models** — ARIMA, Prophet, SARIMA, exponential smoothing (Hyndman and Athanasopoulos 2021)
- **Probabilistic models** — Bayesian networks, probabilistic programming (Gelman et al. 2013)

The Map Is Not the Territory

As George E. P. Box famously noted, *"All models are wrong, but some are useful"* (Box and Draper 1987). The goal is never to perfectly replicate reality — which is almost always impossible — but to build a useful approximation that generalizes well to new, unseen data while remaining interpretable and actionable. Simpler models are often preferred when they explain the data nearly as well as more complex ones, because they tend to generalize better and are easier to maintain and explain (Hastie et al. 2009).

1.2.1 A Taxonomy of Models by Purpose

Before choosing a modeling approach, it helps to be clear about what question you are trying to answer (Provost and Fawcett 2013). The table below summarizes the four main purposes of modeling:

Table 1.1: A taxonomy of modeling purposes.

Purpose	Question	Example
Description	What structure exists in the data?	Clustering customers by behavior
Inference	What factors drive an outcome?	Which features affect churn?
Prediction	What will happen next?	Next month's sales
Decision support	What should we do?	Optimal pricing strategy

A single project may involve more than one purpose, and your choice of model should reflect the primary goal. A model built for description may not be suitable for high-stakes prediction, and a model optimized purely for predictive accuracy may not reveal the causal levers needed for decision support (Pearl and Mackenzie 2018).

1.3 From Data Analysis to Modeling

Many data practitioners begin their journey with **exploratory data analysis (EDA)** — visualizing distributions, computing summary statistics, identifying correlations, and spotting anomalies (Tukey 1977). EDA is an indispensable part of any modeling project, but it is fundamentally *descriptive* and *retrospective*.

Data modeling takes the next critical step: moving from **description** to **prediction**, **inference**, and **decision support** (James et al. 2023).

- **Data Analysis** asks: *"What happened?"* and *"What do the data look like?"*
- **Data Modeling** asks: *"Why did it happen?"*, *"What will happen next?"*, and *"What should we do about it?"* (Provost and Fawcett 2013)

A good model formalizes assumptions, quantifies uncertainty, and provides a framework for testing hypotheses and generating forecasts (Gelman et al. 2013). It transforms the intuitions gained during EDA into something that can be systematically applied, evaluated, and improved over time.

1.3.1 A Motivating Example

Consider a retail company tracking weekly sales. An analyst doing EDA might produce the following summary using pandas (McKinney 2022):

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Simulate weekly retail sales data
np.random.seed(42)
weeks = pd.date_range("2022-01-01", periods=104, freq="W")
trend = np.linspace(1000, 1400, 104)
seasonality = 200 * np.sin(2 * np.pi * np.arange(104) / 52)
noise = np.random.normal(0, 80, 104)
sales = trend + seasonality + noise
```

```
df = pd.DataFrame({"week": weeks, "sales": sales})

# EDA: descriptive statistics
print(df["sales"].describe().round(2))
```

Output:

```
count    104.00
mean    1197.42
std      217.38
min      754.21
25%     1010.55
50%     1192.84
75%     1388.97
max     1614.73
```

This tells us *what* happened — sales averaged around 1,197 units with substantial variation. Data modeling goes further: it would quantify the trend, extract the seasonal pattern, and produce a forecast with confidence intervals for the weeks ahead (Hyndman and Athanasopoulos 2021). The rest of this book will teach you exactly how to do that.

1.4 Statistical vs Machine Learning Perspectives

Although statistical modeling and machine learning overlap significantly, they traditionally emphasize different goals and philosophies (Hastie et al. 2009). Understanding these differences helps you choose the right approach for a given problem.

1.4.1 The Statistical Modeling Perspective

Statistical modeling focuses on **inference** — understanding the data-generating process and the relationships between variables (Gelman et al. 2013). Emphasis is placed on:

- **Interpretability**: each parameter has a meaning that can be communicated to stakeholders
- **Uncertainty quantification**: confidence intervals and p-values measure what we do and do not know (Gelman et al. 2013)
- **Assumption validation**: models come with explicit assumptions (e.g., normality of residuals, homoscedasticity) that must be checked (Hastie et al. 2009)
- **Hypothesis testing**: formal tests allow us to distinguish signal from noise

Common techniques include linear and logistic regression, generalized linear models (GLMs), mixed-effects models, and classical time series methods such as ARIMA and exponential smoothing (Hyndman and Athanasopoulos 2021). These models are often more transparent and easier to audit and explain to regulators or non-technical stakeholders.

1.4.2 The Machine Learning Perspective

Machine learning emphasizes **prediction accuracy** and the ability to generalize to new, unseen data (Hastie et al. 2009). The focus shifts to:

- **Performance metrics**: accuracy, RMSE, AUC on a held-out test set are the primary judges (James et al. 2023)
- **Scalability**: ML methods handle high-dimensional data and complex non-linear interactions that would break most classical statistical models
- **Automation**: hyperparameter tuning, cross-validation, and pipeline automation are first-class concerns
- **Flexibility**: fewer distributional assumptions mean the model can adapt to many data-generating processes

Common techniques include ensemble methods such as random forests and gradient boosting machines, support vector machines, and deep learning architectures such as LSTMs and Transformers. ML approaches shine when predictive accuracy is the primary goal and when datasets are large.

1.4.3 Combining Both Worlds

In practice, the best data scientists draw on both perspectives (James et al. 2023). A typical modern workflow might:

1. Use a simple statistical model (e.g., OLS regression) to quickly understand dominant relationships and identify feature candidates
2. Use a machine learning model (e.g., LightGBM) to maximize predictive performance
3. Apply statistical tools — residual diagnostics, calibration checks, SHAP values (Lundberg and Lee 2017) — to interpret the ML model and quantify its uncertainty
4. Use causal inference methods when the goal shifts from prediction to decision support (Pearl and Mackenzie 2018)

> **When to Prefer Statistical Models**
>
> Statistical models are generally preferable when you need to **explain** results to non-technical audiences, when **regulatory compliance** requires an auditable, interpretable model, when your **dataset is small** and a flexible ML model would overfit (Hastie et al. 2009), or when you want to formally **test a hypothesis** about a causal relationship.

This book embraces both worlds. Each part introduces the right philosophical lens for the task at hand, so you develop the judgment to select the right tool rather than defaulting to whichever method you know best.

1.5 End-to-End Modeling Workflow

A successful modeling project follows a structured, iterative process (Provost and Fawcett 2013). Understanding this workflow before diving into individual techniques prevents common mistakes — such as spending weeks on model tuning before truly understanding the data, or building a model that no one can deploy.

The typical end-to-end modeling workflow consists of the following stages:

1.5.1 Problem Definition & Business Understanding

Clearly define the objective, success metrics, and constraints *before* touching the data (Provost and Fawcett 2013). Key questions to answer at this stage:

- What decision will this model support?
- What does success look like, and how will it be measured?
- What are the data availability, latency, and computational constraints?
- Who are the stakeholders, and what level of interpretability do they need?

A vague problem statement leads to wasted effort. *"We want a better model"* is not a problem statement. *"We want to reduce forecast error for weekly demand by 15% compared to the current moving average, measured by MAPE on an 8-week holdout"* is.

1.5.2 Data Collection & Understanding

Identify what data is available, where it lives, and what quality issues exist. At this stage you inventory sources, assess completeness, and begin to understand the structure of the data using tools such as pandas (McKinney 2022).

Poor data quality — missing values, label errors, leakage from the future — is the most common cause of modeling failures. Investing time here pays dividends throughout the project.

1.5.3 Exploratory Data Analysis (EDA)

Before fitting any model, develop an intuitive understanding of the data (Tukey 1977). Visualize distributions, detect outliers, examine relationships between variables, and check for temporal patterns if working with time series (Hyndman and Athanasopoulos 2021; Wei 2026).

EDA often reveals transformations you need (log, differencing), features worth engineering, or problems in the data that would silently corrupt a model.

1.5.4 Data Preparation & Feature Engineering

Transform raw data into the form your models expect (McKinney 2022). This includes handling missing values, encoding categorical variables, scaling numerical features, and — critically for forecasting — constructing lag features, rolling statistics, and calendar features (Hyndman and Athanasopoulos 2021; Wei 2026).

Feature engineering is often where domain expertise has the greatest impact on model performance (Hastie et al. 2009).

1.5.5 Model Selection & Training

Choose a candidate set of models appropriate to the problem type, data size, and interpretability requirements (James et al. 2023). Start with a simple baseline — it is surprisingly common for a well-tuned linear model to outperform complex alternatives when data is limited (Hastie et al. 2009):

1.5.6 Model Evaluation & Validation

Proper evaluation requires a validation strategy that simulates real-world use (Hastie et al. 2009; James et al. 2023). For time series data, this means *time-based* splits — never randomly shuffling historical data, which would allow information from the future to leak into training (Hyndman and Athanasopoulos 2021).

Chapter 13 covers cross-validation strategies in depth, including walk-forward validation and expanding windows for forecasting models.

1.5.7 Model Interpretation & Diagnostics

Understanding *why* a model makes predictions is as important as the predictions themselves (Lundberg and Lee 2017). Residual diagnostics catch violations of modeling assumptions (Gelman et al. 2013). Feature importance plots and SHAP values (Lundberg and Lee 2017) reveal what the model has learned.

This stage is also where you verify that the model has not learned spurious shortcuts — for example, a data leakage artifact.

1.5.8 Deployment & Monitoring

A model that is not deployed creates no value. Deployment involves packaging the model, integrating it into production systems, and establishing monitoring to detect when the model's performance degrades over time — a phenomenon known as **model drift** (Provost and Fawcett 2013).

1.5.9 Iteration & Improvement

The workflow is rarely linear (Provost and Fawcett 2013). New data reveals problems with earlier feature choices. A stakeholder requirement changes the success metric. A model that worked well six months ago starts to degrade. Iteration is not a sign of failure — it is the normal rhythm of a production modeling system.

1.6 Summary

This chapter introduced the foundations that underpin the rest of the book:

- **Data modeling** is the creation of mathematical or computational representations of reality for the purposes of description, inference, prediction, or decision support (Provost and Fawcett 2013).
- **EDA** (Tukey 1977) is a necessary precursor to modeling, but modeling goes further by formalizing assumptions and enabling generalization to new data.
- **Statistical and machine learning** perspectives are complementary, not competing (Hastie et al. 2009; James et al. 2023) — effective practitioners combine the inferential rigor of statistics with the predictive power of machine learning.

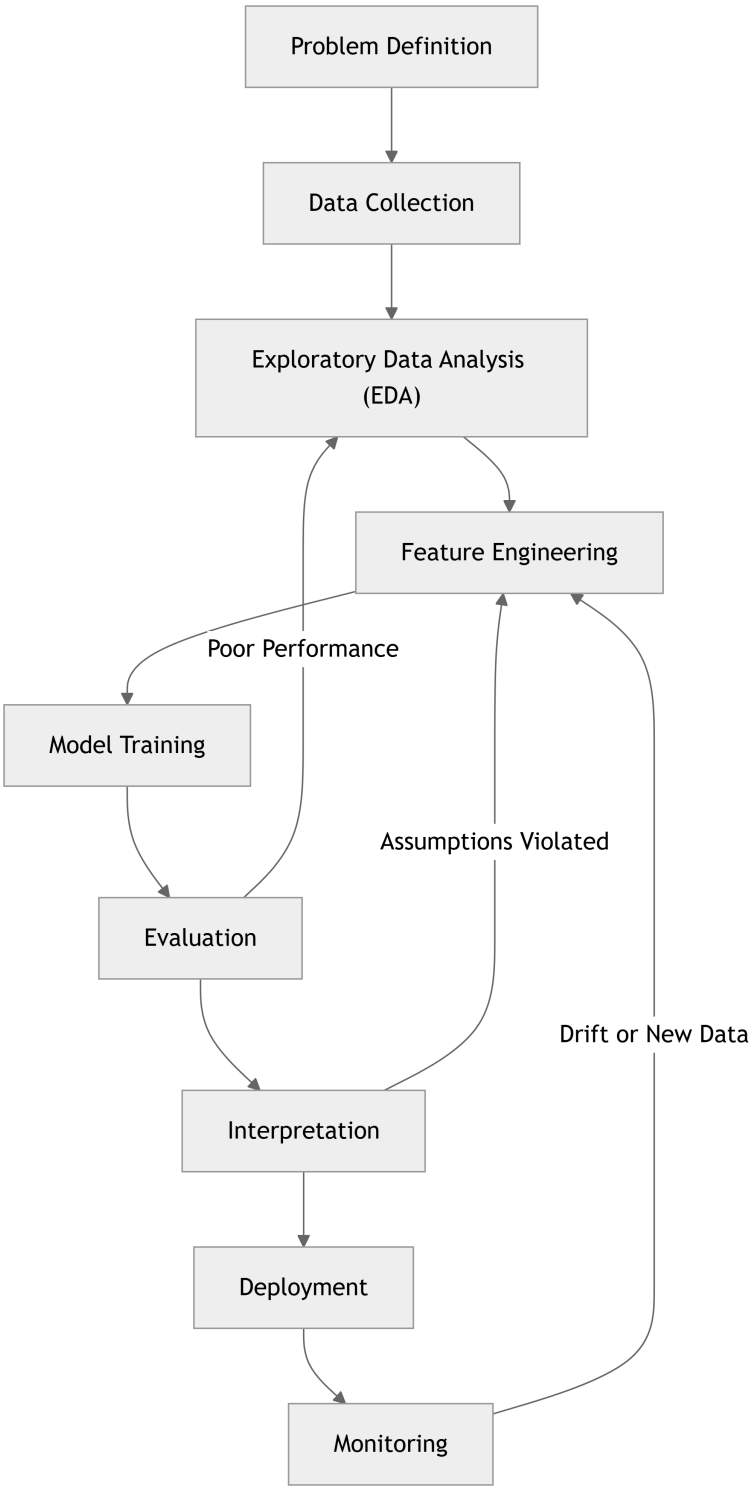


Figure 1.1: Modeling workflow illustrating a structured and iterative approach to building, evaluating, and deploying machine learning models.

- The **end-to-end modeling workflow** is a nine-stage iterative process, from problem definition through deployment and monitoring. Skipping stages or treating the workflow as linear are two of the most common causes of modeling project failures (Provost and Fawcett 2013).

In the next chapter, we turn to the data itself — how to load, clean, and prepare tabular data with pandas (McKinney 2022), and how to engineer the features that will drive model performance throughout the book.

1.7 Exercises

Exercise 1:
In your own words, explain the difference between data analysis (Tukey 1977) and data modeling (Provost and Fawcett 2013). Provide one concrete example of each from a domain that interests you.

Exercise 2:
Give three real-world situations where a predictive model would be more valuable than simple descriptive analysis. For each, identify the type of model (description, inference, prediction, or decision support) that would be most appropriate (Provost and Fawcett 2013).

Exercise 3:
A business stakeholder says: *"I just want a model that predicts sales as accurately as possible."* From a statistical perspective (Gelman et al. 2013), what additional questions would you ask before beginning? Why does each question matter?

Exercise 4:
Compare the strengths and weaknesses of statistical modeling (Hastie et al. 2009) versus machine learning approaches. For each of the following scenarios, argue which perspective is more appropriate and why:
a. A bank building a credit scoring model subject to regulatory review
b. A recommendation engine for a streaming platform with millions of users
c. An epidemiologist studying the effect of a policy intervention on disease rates (Pearl and Mackenzie 2018)

Exercise 5:
Using the simulated sales data from the motivating example in Section 1.3, write Python code (McKinney 2022) to:
a. Plot the time series
b. Compute a 4-week rolling average and overlay it on the plot
c. Identify the three weeks with the highest residual deviation from the rolling average

Exercise 6:
Why is it dangerous to evaluate a time series model by randomly shuffling all observations before splitting into train and test sets (Hyndman and Athanasopoulos 2021)? What alternative validation strategy should you use, and why?

Exercise 7:
Sketch the end-to-end modeling workflow (Provost and Fawcett 2013) for a demand forecasting system at a grocery retailer. For each stage, identify at least one decision that would be specific to this domain. Identify two points in the workflow where you would expect to iterate most frequently, and explain why.

1.8 Quiz

1. What is the primary goal of data modeling?
   A. To perfectly replicate reality
   B. To visualize data distributions
   C. To create a useful, generalizable representation for understanding, prediction, and decision-making
   D. To store data in relational databases
2. Which of the following best describes the difference between data analysis and data modeling?
   A. Data analysis focuses on prediction while modeling focuses on description
   B. They are the same thing with different names
   C. Data modeling is only used in machine learning contexts

D. Data analysis is descriptive and retrospective; data modeling moves toward prediction, inference, and decision support

3. Statistical modeling primarily emphasizes:
   A. Inference, interpretability, and understanding the data-generating process
   B. Maximum prediction accuracy on new data
   C. Using only deep neural networks
   D. Ignoring model assumptions
4. Machine learning approaches typically focus more on:
   A. Hypothesis testing and p-values
   B. Strict validation of distributional assumptions
   C. Explaining every parameter in the model
   D. Prediction accuracy and generalization performance
5. According to George Box's aphorism (Box 1976), what is true of all models?
   A. They are perfectly accurate representations of reality
   B. They are wrong, but some are useful
   C. They must be built with machine learning to be effective
   D. They never require updating once deployed
6. In the end-to-end modeling workflow, why should time series data *not* be randomly shuffled before splitting into train and test sets (Hyndman and Athanasopoulos 2021)?
   A. Because random shuffling is computationally expensive
   B. Because time series data has too many rows to shuffle
   C. Because shuffling would allow future data to leak into the training set, creating an overly optimistic evaluation
   D. Because scikit-learn does not support shuffling
7. Which of the following is the best reason to prefer a simpler statistical model over a complex machine learning model (Hastie et al. 2009)?
   A. Statistical models always perform better
   B. Machine learning is too slow
   C. Complex models always overfit
   D. The problem requires a regulatory-compliant, interpretable model and the dataset is small

Answers: 1-C, 2-D, 3-A, 4-D, 5-B, 6-C, 7-D

```
                        handle_unknown="use_encoded_value",
                        unknown_value=-1))
    ])
    return ColumnTransformer([
        ("num", num_pipe, num_cols_adult),
        ("cat", cat_pipe, cat_cols_adult)
    ])

def make_adult_pipe(clf):
    return Pipeline([
        ("preprocessor", adult_preprocessor()),
        ("clf",           clf)
    ])

gb_adult = make_adult_pipe(
    GradientBoostingClassifier(
        n_estimators=200, learning_rate=0.1,
        max_depth=4, random_state=RANDOM_SEED
    )
)
gb_adult.fit(X_atr, y_atr)

X_ate_processed = pd.DataFrame(
    gb_adult.named_steps["preprocessor"].transform(X_ate),
    columns=num_cols_adult + cat_cols_adult
)

explainer_cls = shap.TreeExplainer(gb_adult.named_steps["clf"])
shap_values_cls = explainer_cls(X_ate_processed)

fig, ax = plt.subplots(figsize=(8, 6))
shap.summary_plot(
    shap_values_cls.values,
    X_ate_processed,
    max_display=12,
    show=False,
    plot_size=None
)
plt.title("SHAP Summary — GB Classifier, Adult Income Test Set",
          pad=15)
```

```
plt.tight_layout()
plt.savefig(FIGURES / "ch17_shap_cls_summary.png", dpi=150,
            bbox_inches="tight")
plt.show()
```

Output:

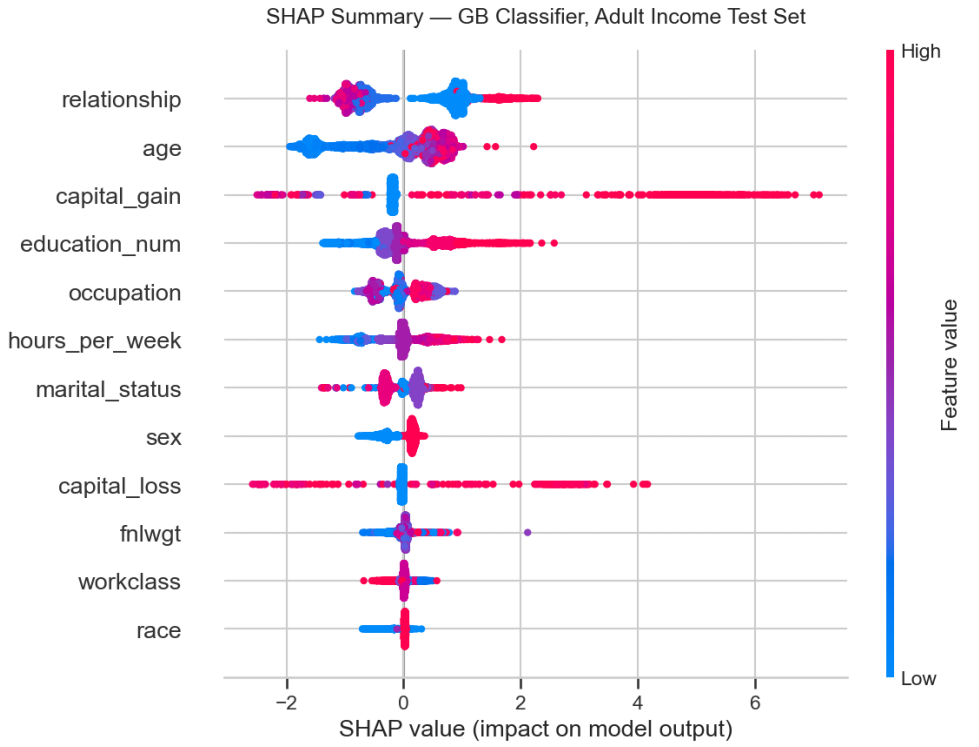


Figure 17.7: SHAP summary plot for the Gradient Boosting classifier on the Adult Income test set (log-odds scale). `relationship` is the strongest predictor, with high feature values (e.g. "Husband") pushing log-odds of >$50K upward and low values (e.g. "Own-child") pushing strongly negative. `age` has the second-widest spread, contributing positively at high values and negatively at low. `capital_gain` shows the longest positive tail: a small fraction of observations with very high capital gains receive extreme SHAP values extending past 6. `capital_loss` similarly produces a wide spread in both directions. `education_num` contributes near-monotonically — higher education consistently increases the predicted probability of high income. `fnlwgt`, `workclass`, and `race` contribute relatively little.

17.4.6 SHAP Importance vs. Gini vs. Permutation: Unified Comparison

```
def normalise(s):
    s = s.clip(lower=0)
    return s / s.sum()

shap_imp_reg = pd.Series(
    np.abs(shap_values_reg.values).mean(axis=0),
    index=X_ite_processed.columns
)
gini_series  = pd.Series(gini_imp, index=all_feat_names_ins)
perm_series  = pd.Series(perm_result.importances_mean,
index=all_feat_names_ins)

comp_df = pd.DataFrame({
    "SHAP":        normalise(shap_imp_reg),
    "Gini":        normalise(gini_series),
    "Permutation": normalise(perm_series),
}).sort_values("SHAP", ascending=False)

fig, ax = plt.subplots(figsize=(9, 4))
x = np.arange(len(all_feat_names_ins))
w = 0.28
ax.bar(x - w, comp_df["SHAP"],        width=w, label="SHAP",
       color="#AB47BC", edgecolor="white")
ax.bar(x,      comp_df["Gini"],        width=w, label="Gini",
       color="steelblue", edgecolor="white")
ax.bar(x + w, comp_df["Permutation"], width=w, label="Permutation",
       color="#F4A261", edgecolor="white")
ax.set_xticks(x)
ax.set_xticklabels(comp_df.index, fontsize=10)
ax.set_ylabel("Normalised importance")
ax.set_title("Feature Importance: SHAP vs. Gini vs. Permutation\n"
             "GB Regressor — Medical Insurance Cost Test Set")
ax.legend(fontsize=9)
plt.tight_layout()
plt.savefig(FIGURES / "ch17_importance_shap_vs_rest.png", dpi=150,
bbox_inches="tight")
```

```
plt.show()
```

Output:

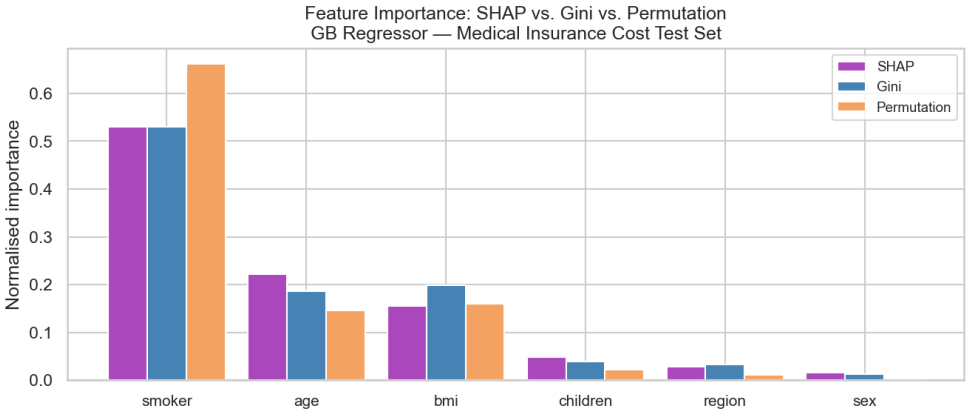


Figure 17.8: Normalised feature importance from three methods on the Medical Insurance Cost test set. All three agree that `smoker` is the dominant predictor (SHAP 53%, Gini 53%, Permutation 65%); permutation's higher estimate likely reflects correlated features redistributing importance onto the strongest predictor. The most notable within-method divergence is between `age` and `bmi`: SHAP ranks `age` clearly above `bmi` (~22% vs ~16%), while Gini reverses this, placing them nearly equal with `bmi` marginally higher (~20% vs ~19%). All three methods consistently rank `children`, `region`, and `sex` at the bottom with negligible importance.

17.5 Model Explanation Reference Framework

17.5.1 When to Use Each Explanation Method

Table 17.1: Guidance for selecting explanation methods by goal and scope.

Goal	Method	Scope
Rank features by global impact	SHAP mean	
Identify non-linear marginal effects	PDP / ICE	Global

Goal	Method	Scope
Detect interactions between features	2D PDP / SHAP dependence	Global
Explain a single prediction	SHAP waterfall / force plot	Local
Communicate quickly to non-technical stakeholders	SHAP waterfall	Local
Audit for bias/fairness across subgroups	SHAP by subgroup	Global/Local
Compare feature rankings across models	Permutation importance	Global

17.5.2 Final Evaluation and Explanation Summary

```
from sklearn.metrics import roc_auc_score, average_precision_score

print("═══ Adult Income — Final Model Comparison ═══")
print(f"{'Model':25s}  {'Acc':>6}  {'F1':>6}  {'AUC':>6}  "
      f"{'AP':>6}  {'Brier':>6}")
print("-" * 65)

from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import (
    accuracy_score, f1_score, roc_auc_score,
    average_precision_score, brier_score_loss
)

for name, clf in [
    ("Logistic Regression",
     LogisticRegression(C=0.1, max_iter=1000,
     random_state=RANDOM_SEED)),
    ("Random Forest",
     RandomForestClassifier(n_estimators=200,
     random_state=RANDOM_SEED, n_jobs=-1)),
    ("Gradient Boosting",
```

```
     GradientBoostingClassifier(n_estimators=200, learning_rate=0.1,
                                max_depth=4,
                                random_state=RANDOM_SEED)),
]:
    pipe = make_adult_pipe(clf)
    pipe.fit(X_atr, y_atr)
    y_p = pipe.predict(X_ate)
    proba = pipe.predict_proba(X_ate)[:, 1]
    print(
        f"{name:25s}  "
        f"{accuracy_score(y_ate, y_p):6.4f}  "
        f"{f1_score(y_ate, y_p):6.4f}  "
        f"{roc_auc_score(y_ate, proba):6.4f}  "
        f"{average_precision_score(y_ate, proba):6.4f}  "
        f"{brier_score_loss(y_ate, proba):6.4f}"
    )
```

Output:

```
═══ Adult Income — Final Model Comparison ═══
Model                        Acc      F1     AUC      AP   Brier
------------------------------------------------------------------
Logistic Regression       0.8279  0.5616  0.8608  0.7003  0.1195
Random Forest             0.8635  0.6940  0.9123  0.7944  0.0971
Gradient Boosting         0.8730  0.7139  0.9310  0.8360  0.0864
```

The SHAP explanations for the Gradient Boosting classifier — the best performer — reveal that `relationship` and `capital_gain` are the dominant drivers, with `education_num` third. These rankings are consistent with the permutation importance analysis and provide actionable, auditable insights for stakeholders.

17.6 Summary

This chapter developed the model interpretation framework that bridges technical performance and human understanding:

- **Feature importance** (Section 17.2): Gini importance is fast but biased toward high-cardinality features. Permutation importance is test-set-based and avoids cardinality bias but underestimates importance for correlated features. All three measures agree that `smoker` dominates in the insurance cost task (~53% for SHAP and Gini, ~65% for permutation), and diverge on the relative ranking of `bmi` versus `age` — confirming that the choice of importance measure matters, particularly for continuous versus binary features.
- **PDP and ICE plots** (Section 17.3): PDPs reveal marginal effects of features averaged over all observations. ICE plots reveal heterogeneity — diverging ICE curves signal interactions that the PDP obscures. The 2D PDP for BMI × smoker confirms the interaction identified by SHAP.
- **SHAP values** (Section 17.4): the unique fair attribution grounded in cooperative game theory. TreeSHAP computes exact values in polynomial time. The summary plot reveals global feature importance with per-observation detail; the waterfall plot provides a fully auditable local explanation; the dependence plot exposes non-linear effects and interactions without requiring pre-specified interaction terms.
- **Explanation method selection** (Section 17.5): the choice depends on whether the goal is global or local, and whether the audience is technical or non-technical. SHAP provides a unified foundation that subsumes all earlier methods while adding local and interaction capabilities.

This chapter concludes Part VI. Part VII turns to **unsupervised learning** — clustering, dimensionality reduction, and anomaly detection — where the absence of a response variable requires entirely different evaluation and explanation frameworks.

17.7 Exercises

Exercise 1: Compute SHAP values for the **Logistic Regression** classifier on the Adult Income dataset using `shap.LinearExplainer`. Compare the SHAP feature importance ranking against the model's raw standardised coefficients. For which features do they agree most and least? Explain why SHAP values and coefficients can differ for a linear model.

Exercise 2: Use `shap.TreeExplainer` to compute SHAP interaction values for the Gradient Boosting regressor on the Medical Insurance Cost test set. Extract the interaction matrix for the `bmi`–`smoker` pair. Confirm that the interaction value is near zero for non-smokers and rises sharply above BMI 30 for smokers.

Exercise 3: On the Medical Insurance Cost dataset, compute Gini importance, permutation importance, and SHAP importance for a Ridge regression model using `shap.LinearExplainer`. Do the three methods agree more closely for the linear model than for the Random Forest? Explain why.

Exercise 4: Plot ICE curves for the `age` feature in the Gradient Boosting regressor on Medical Insurance Cost. Separate the ICE curves by `smoker` status (colour them differently). Are the ICE curves parallel (homogeneous effect) or do they diverge? How does this compare to the `bmi` ICE curves from Section 17.3?

Exercise 5: Create a SHAP force plot for the five test observations with the largest prediction errors (largest |actual − predicted|) from the Gradient Boosting regressor. Do the misclassified observations share common SHAP patterns? What features appear most responsible for the errors?

17.8 Quiz

1. A SHAP value $\phi_j(\mathbf{x}_i) = +\$22{,}400$ for feature `smoker` at observation i means: A. Removing `smoker` from the model would reduce RMSE by \$22,400 B. Smoking status increases the average insurance charge across all observations by \$22,400 C. For this specific individual, being a smoker increases the predicted charge by \$22,400 relative to the model's average prediction D. The feature `smoker` has a mean absolute SHAP importance of \$22,400 across the dataset
2. Gini (impurity) importance is **biased toward high-cardinality features** because: A. High-cardinality features are always more predictive in real datasets B. More distinct values means more candidate splits at each node, inflating the total impurity reduction attributed to that feature C. Gini impurity is computed on the training set, where

high-cardinality features overfit D. The random subspace method selects high-cardinality features more frequently

3. **ICE plots** provide additional information over **PDP** by: A. Averaging predictions over more observations, reducing noise B. Showing the marginal effect of a feature for each individual observation, revealing heterogeneity that the average trend obscures C. Accounting for correlation between features D. Computing exact Shapley values for every observation
4. The **SHAP efficiency axiom** states that: A. TreeSHAP runs in polynomial rather than exponential time B. The sum of all SHAP values for an observation equals the model's deviation from the baseline prediction C. Features that never affect the model receive SHAP values of zero D. Two features with identical contributions receive equal SHAP values
5. **SHAP dependence plots** are preferred over standard scatter plots of feature value versus prediction because: A. They are faster to compute B. They show the feature's isolated contribution (SHAP value) rather than the raw prediction, which includes the effects of all other features C. They automatically detect and report all pairwise interactions D. They are based on test data rather than training data

Answers: 1-C, 2-B, 3-B, 4-B, 5-B

References

Astral. 2024. *Uv: An Extremely Fast Python Package and Project Manager*. Https://docs.astral.sh/uv/.

Bellman, Richard. 1961. *Adaptive Control Processes: A Guided Tour*. Princeton University Press.

Bergstra, James, and Yoshua Bengio. 2012. "Random Search for Hyper-Parameter Optimization." *Journal of Machine Learning Research* 13: 281–305.

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Springer.

Box, George E. P. 1976. "Science and Statistics." *Journal of the American Statistical Association* 71 (356): 791–99. https://doi.org/10.1080/01621459.1976.10480949.

Box, George E. P., and Norman R. Draper. 1987. *Empirical Model-Building and Response Surfaces*. Wiley.

Breiman, Leo. 2001. *Random Forests*. In *Machine Learning*, vol. 45. https://doi.org/10.1023/A:1010933404324.

Cohen, Jacob. 1988. *Statistical Power Analysis for the Behavioral Sciences*. 2nd ed. Lawrence Erlbaum Associates.

Cortes, Corinna, and Vladimir Vapnik. 1995. "Support-Vector Networks." *Machine Learning* 20 (3): 273–97.

De Cock, Dean. 2011. "Ames, Iowa: Alternative to the Boston Housing Data as an End of Semester Regression Project." *Journal of Statistics Education* 19 (3). https://doi.org/10.1080/10691898.2011.11889627.

Detrano, Robert, Andras Janosi, William Steinbrunn, et al. 1989. "International Application of a New Probability Algorithm for the Diagnosis of Coronary Artery Disease." *The American Journal of Cardiology* 64 (5): 304–10.

Efron, Bradley, and Robert J. Tibshirani. 1994. *An Introduction to the Bootstrap*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Fanaee-T, Hadi. 2014. "Event Labeling Combining Ensemble Detectors and Background Knowledge." *Progress in Artificial Intelligence* 2: 113–27.

Friedman, Jerome H. 2001. "Greedy Function Approximation: A Gradient Boosting Machine." *Annals of Statistics* 29 (5): 1189–232.

Friedman, Jerome H. 2002. "Stochastic Gradient Boosting." *Computational Statistics & Data Analysis* 38 (4): 367–78.

Gelman, Andrew, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. 2013. *Bayesian Data Analysis*. 3rd ed. CRC Press.

Geman, Stuart, Elie Bienenstock, and René Doursat. 1992. "Neural Networks and the Bias/Variance Dilemma." *Neural Computation* 4 (1): 1–58. https://doi.org/10.1162/neco.1992.4.1.1.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd ed. Springer. https://hastie.su.domains/ElemStatLearn/.

Ho, Tin Kam. 1998. "The Random Subspace Method for Constructing Decision Forests." *IEEE Transactions on Pattern Analysis and Machine Intelligence*

20 (8): 832–44.

Hyndman, Rob J., and George Athanasopoulos. 2021. *Forecasting: Principles and Practice*. 3rd ed. OTexts. https://otexts.com/fpp3/.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning: With Applications in R*. Springer. https://doi.org/10.1007/978-1-4614-7138-7.

James, Gareth, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. 2023. *An Introduction to Statistical Learning: With Applications in Python*. Springer. https://www.statlearning.com.

Kohavi, Ron. 1995. "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection." *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)* (San Francisco) 2: 1137–43.

Kohavi, Ron. 1996. "Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid." *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, 202–7.

Lantz, Brett. 2013. *Machine Learning with r*. Packt Publishing.

Lundberg, Scott M., and Su-In Lee. 2017. "A Unified Approach to Interpreting Model Predictions." *Advances in Neural Information Processing Systems* 30. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html.

Lundberg, Scott M., and Su-In Lee. 2020. "From Local Explanations to Global Understanding with Explainable AI for Trees." *Nature Machine Intelligence* 2: 56–67.

McCullagh, Peter, and John A. Nelder. 1989. *Generalized Linear Models*. 2nd ed. Chapman; Hall.

McKinney, Wes. 2022. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter*. 3rd ed. O'Reilly Media.

Mitchell, Tom M. 1997. *Machine Learning*. McGraw-Hill.

Nelder, John A., and Robert W. M. Wedderburn. 1972. "Generalized Linear Models." *Journal of the Royal Statistical Society: Series A (General)* 135 (3): 370–84.

Pearl, Judea, and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect*. Basic Books.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30. https://jmlr.org/papers/v12/pedregosa11a.html.

Provost, Foster, and Tom Fawcett. 2013. *Data Science for Business*. O'Reilly Media.

Tibshirani, Robert. 1996. "Regression Shrinkage and Selection via the Lasso." *Journal of the Royal Statistical Society: Series B (Methodological)* 58 (1): 267–88.

Tukey, John W. 1977. *Exploratory Data Analysis*. Addison-Wesley.

VanderPlas, Jake. 2016. *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media, Inc.

Wasserman, Larry. 2004. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics. Springer.

Wasserstein, Ronald L., and Nicole A. Lazar. 2016. "The ASA's Statement on p-Values: Context, Process, and Purpose." *The American Statistician* 70 (2): 129–33. https://doi.org/10.1080/00031305.2016.1154108.

Wei, Shouke. 2026. *Practical Data Analysis and Visualization with Python: Data Exploration, Visualization, and Scalable Data Processing*. 1st ed. Deepsim Press. https://doi.org/10.5281/zenodo.19388650.

Index