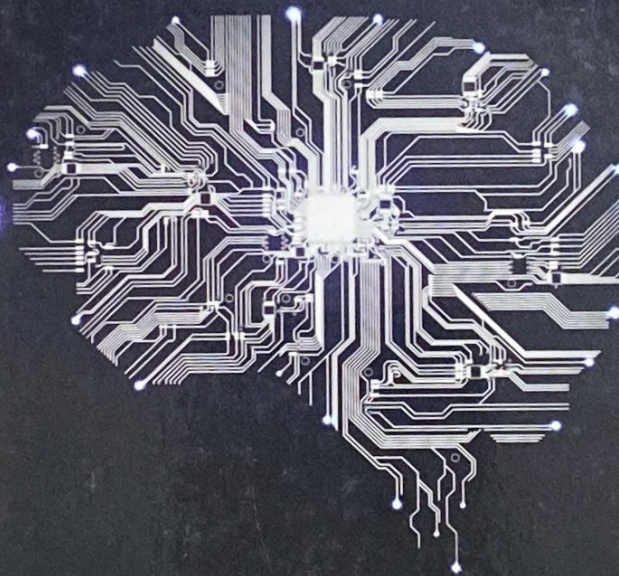


ADDISON WESLEY DATA & ANALYTICS SERIES



MACHINE LEARNING WITH PYTHON FOR EVERYONE



MARK E. FENNER

Machine Learning with Python for Everyone

Visit informa.com/databases for a complete list of available publications.

The Pearson Addison-Wesley Data & Analytics Series provides readers with practical knowledge for solving problems and answering questions with data. Titles in this series primarily focus on three areas:

1. Infrastructure: how to store, move, and manage data
2. Algorithms: how to mine intelligence or make predictions based on data
3. Visualizations: how to represent data and insights in a meaningful and compelling way

The series aims to tie all three of these areas together to help the reader build end-to-end systems for fighting spam; making recommendations; building personalization; detecting trends, patterns, or problems; and gaining insight from the data exhaust of systems and user interactions.

Make sure to connect with us!

Visit www.informaworld.com or informa.com

informa.com

Pearson
Addison-Wesley

The Pearson Addison-Wesley Data & Analytics Series



Visit informit.com/awdataseries for a complete list of available publications.

The Pearson Addison-Wesley Data & Analytics Series provides readers with practical knowledge for solving problems and answering questions with data. Titles in this series primarily focus on three areas:

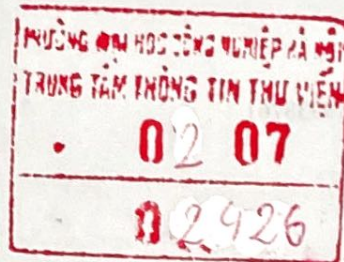
1. **Infrastructure:** how to store, move, and manage data
2. **Algorithms:** how to mine intelligence or make predictions based on data
3. **Visualizations:** how to represent data and insights in a meaningful and compelling way

The series aims to tie all three of these areas together to help the reader build end-to-end systems for fighting spam; making recommendations; building personalization; detecting trends, patterns, or problems; and gaining insight from the data exhaust of systems and user interactions.



Make sure to connect with us!
informit.com/socialconnect

Machine Learning with Python for Everyone



Mark E. Fenner

◆◆Addison-Wesley

Boston • Columbus • New York • San Francisco • Amsterdam • Cape Town
Dubai • London • Madrid • Milan • Munich • Paris • Montreal • Toronto • Delhi • Mexico City
São Paulo • Sydney • Hong Kong • Seoul • Singapore • Taipei • Tokyo

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Visit us on the Web: informit.com/aw

Library of Congress Control Number: 2019938761

Copyright © 2020 Pearson Education, Inc.

Cover image: [cono0430/Shutterstock](#)

Pages 58, 87: Screenshot of seaborn © 2012–2018 Michael Waskom.

Pages 167, 177, 192, 201, 278, 284, 479, 493: Screenshot of seaborn heatmap © 2012–2018 Michael Waskom.

Pages 178, 185, 196, 197, 327, 328: Screenshot of seaborn swarmplot © 2012–2018 Michael Waskom.

Page 222: Screenshot of seaborn stripplot © 2012–2018 Michael Waskom.

Pages 351, 354: Screenshot of seaborn implot © 2012–2018 Michael Waskom.

Pages 352, 353, 355: Screenshot of seaborn distplot © 2012–2018 Michael Waskom.

Pages 460, 461: Screenshot of Manifold © 2007–2018, scikit-learn developers.

Page 480: Screenshot of cluster © 2007–2018, scikit-learn developers.

Pages 483, 484, 485: Image of accordion, Vereshchagin Dmitry/Shutterstock.

Page 485: Image of fighter jet, 3dgenerator/123RF.

Page 525: Screenshot of seaborn jointplot © 2012–2018 Michael Waskom.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearsoned.com/permissions/.

ISBN-13: 978-0-13-484562-3

ISBN-10: 0-13-484562-5

1 2019

Contents

Foreword xxi

Preface xxiii

About the Author xxvii

I First Steps 1

1 Let's Discuss Learning 3

- 1.1 Welcome 3
- 1.2 Scope, Terminology, Prediction, and Data 4
 - 1.2.1 Features 5
 - 1.2.2 Target Values and Predictions 6
- 1.3 Putting the Machine in Machine Learning 7
- 1.4 Examples of Learning Systems 9
 - 1.4.1 Predicting Categories: Examples of Classifiers 9
 - 1.4.2 Predicting Values: Examples of Regressors 10
- 1.5 Evaluating Learning Systems 11
 - 1.5.1 Correctness 11
 - 1.5.2 Resource Consumption 12
- 1.6 A Process for Building Learning Systems 13
- 1.7 Assumptions and Reality of Learning 15
- 1.8 End-of-Chapter Material 17
 - 1.8.1 The Road Ahead 17
 - 1.8.2 Notes 17

2 Some Technical Background 19

- 2.1 About Our Setup 19
- 2.2 The Need for Mathematical Language 19

2.3	Our Software for Tackling Machine Learning	20
2.4	Probability	21
2.4.1	Primitive Events	22
2.4.2	Independence	23
2.4.3	Conditional Probability	24
2.4.4	Distributions	25
2.5	Linear Combinations, Weighted Sums, and Dot Products	28
2.5.1	Weighted Average	30
2.5.2	Sums of Squares	32
2.5.3	Sum of Squared Errors	33
2.6	A Geometric View: Points in Space	34
2.6.1	Lines	34
2.6.2	Beyond Lines	39
2.7	Notation and the Plus-One Trick	43
2.8	Getting Groovy, Breaking the Straight-Jacket, and Nonlinearity	45
2.9	NumPy versus "All the Maths"	47
2.9.1	Back to 1D versus 2D	49
2.10	Floating-Point Issues	52
2.11	EOC	53
2.11.1	Summary	53
2.11.2	Notes	54
3	Predicting Categories: Getting Started with Classification	55
3.1	Classification Tasks	55
3.2	A Simple Classification Dataset	56
3.3	Training and Testing: Don't Teach to the Test	59
3.4	Evaluation: Grading the Exam	62
3.5	Simple Classifier #1: Nearest Neighbors, Long Distance Relationships, and Assumptions	63
3.5.1	Defining Similarity	63
3.5.2	The k in k -NN	64
3.5.3	Answer Combination	64

	3.5.4	<i>k</i> -NN, Parameters, and Nonparametric Methods	65
	3.5.5	Building a <i>k</i> -NN Classification Model	66
3.6		Simple Classifier #2: Naive Bayes, Probability, and Broken Promises	68
3.7		Simplistic Evaluation of Classifiers	70
	3.7.1	Learning Performance	70
	3.7.2	Resource Utilization in Classification	71
	3.7.3	Stand-Alone Resource Evaluation	77
3.8	EOC		81
	3.8.1	Sophomore Warning: Limitations and Open Issues	81
	3.8.2	Summary	82
	3.8.3	Notes	82
	3.8.4	Exercises	83

4 Predicting Numerical Values: Getting Started with Regression 85

4.1		A Simple Regression Dataset	85
4.2		Nearest-Neighbors Regression and Summary Statistics	87
	4.2.1	Measures of Center: Median and Mean	88
	4.2.2	Building a <i>k</i> -NN Regression Model	90
4.3		Linear Regression and Errors	91
	4.3.1	No Flat Earth: Why We Need Slope	92
	4.3.2	Tilting the Field	94
	4.3.3	Performing Linear Regression	97
4.4		Optimization: Picking the Best Answer	98
	4.4.1	Random Guess	98
	4.4.2	Random Step	99
	4.4.3	Smart Step	99
	4.4.4	Calculated Shortcuts	100

	4.4.5	Application to Linear Regression	101
4.5		Simple Evaluation and Comparison of Regressors	101
	4.5.1	Root Mean Squared Error	101
	4.5.2	Learning Performance	102
	4.5.3	Resource Utilization in Regression	102
4.6		EOC	104
	4.6.1	Limitations and Open Issues	104
	4.6.2	Summary	105
	4.6.3	Notes	105
	4.6.4	Exercises	105

II Evaluation 107

5 Evaluating and Comparing Learners 109

5.1	Evaluation and Why Less Is More	109
5.2	Terminology for Learning Phases	110
	5.2.1	Back to the Machines 110
	5.2.2	More Technically Speaking . . . 113
5.3	Major Tom, There's Something Wrong: Overfitting and Underfitting	116
	5.3.1	Synthetic Data and Linear Regression 117
	5.3.2	Manually Manipulating Model Complexity 118
	5.3.3	Goldilocks: Visualizing Overfitting, Underfitting, and "Just Right" 120
	5.3.4	Simplicity 124
	5.3.5	Take-Home Notes on Overfitting 124
5.4	From Errors to Costs	125
	5.4.1	Loss 125
	5.4.2	Cost 126

5.4.3	Score	127
5.5	(Re)Sampling: Making More from Less	128
5.5.1	Cross-Validation	128
5.5.2	Stratification	132
5.5.3	Repeated Train-Test Splits	133
5.5.4	A Better Way and Shuffling	137
5.5.5	Leave-One-Out Cross-Validation	140
5.6	Break-It-Down: Deconstructing Error into Bias and Variance	142
5.6.1	Variance of the Data	143
5.6.2	Variance of the Model	144
5.6.3	Bias of the Model	144
5.6.4	All Together Now	145
5.6.5	Examples of Bias-Variance Tradeoffs	145
5.7	Graphical Evaluation and Comparison	149
5.7.1	Learning Curves: How Much Data Do We Need?	150
5.7.2	Complexity Curves	152
5.8	Comparing Learners with Cross-Validation	154
5.9	EOC	155
5.9.1	Summary	155
5.9.2	Notes	155
5.9.3	Exercises	157

6 Evaluating Classifiers 159

6.1	Baseline Classifiers	159
6.2	Beyond Accuracy: Metrics for Classification	161
6.2.1	Eliminating Confusion from the Confusion Matrix	163
6.2.2	Ways of Being Wrong	164
6.2.3	Metrics from the Confusion Matrix	165
6.2.4	Coding the Confusion Matrix	166
6.2.5	Dealing with Multiple Classes: Multiclass Averaging	168

	6.2.6	F_1	170
6.3		ROC Curves	170
	6.3.1	Patterns in the ROC	173
	6.3.2	Binary ROC	174
	6.3.3	AUC: Area-Under-the-(ROC)-Curve	177
	6.3.4	Multiclass Learners, One-versus-Rest, and ROC	179
6.4		Another Take on Multiclass: One-versus-One	181
	6.4.1	Multiclass AUC Part Two: The Quest for a Single Value	182
6.5		Precision-Recall Curves	185
	6.5.1	A Note on Precision-Recall Tradeoff	185
	6.5.2	Constructing a Precision-Recall Curve	186
6.6		Cumulative Response and Lift Curves	187
6.7		More Sophisticated Evaluation of Classifiers: Take Two	190
	6.7.1	Binary	190
	6.7.2	A Novel Multiclass Problem	195
6.8		EOC	201
	6.8.1	Summary	201
	6.8.2	Notes	202
	6.8.3	Exercises	203
7		Evaluating Regressors	205
	7.1	Baseline Regressors	205
	7.2	Additional Measures for Regression	207
		7.2.1	Creating Our Own Evaluation Metric 207
		7.2.2	Other Built-in Regression Metrics 208
		7.2.3	R^2 209

- 7.3 Residual Plots 214
 - 7.3.1 Error Plots 215
 - 7.3.2 Residual Plots 217
- 7.4 A First Look at Standardization 221
- 7.5 Evaluating Regressors in a More Sophisticated Way: Take Two 225
 - 7.5.1 Cross-Validated Results on Multiple Metrics 226
 - 7.5.2 Summarizing Cross-Validated Results 230
 - 7.5.3 Residuals 230
- 7.6 EOC 232
 - 7.6.1 Summary 232
 - 7.6.2 Notes 232
 - 7.6.3 Exercises 234

III More Methods and Fundamentals 235

8 More Classification Methods 237

- 8.1 Revisiting Classification 237
- 8.2 Decision Trees 239
 - 8.2.1 Tree-Building Algorithms 242
 - 8.2.2 Let's Go: Decision Tree Time 245
 - 8.2.3 Bias and Variance in Decision Trees 249
- 8.3 Support Vector Classifiers 249
 - 8.3.1 Performing SVC 253
 - 8.3.2 Bias and Variance in SVCs 256
- 8.4 Logistic Regression 259
 - 8.4.1 Betting Odds 259
 - 8.4.2 Probabilities, Odds, and Log-Odds 262
 - 8.4.3 Just Do It: Logistic Regression Edition 267
 - 8.4.4 A Logistic Regression: A Space Oddity 268

8.5	Discriminant Analysis	269
8.5.1	Covariance	270
8.5.2	The Methods	282
8.5.3	Performing DA	283
8.6	Assumptions, Biases, and Classifiers	285
8.7	Comparison of Classifiers: Take Three	287
8.7.1	Digits	287
8.8	EOC	290
8.8.1	Summary	290
8.8.2	Notes	290
8.8.3	Exercises	293
9	More Regression Methods	295
9.1	Linear Regression in the Penalty Box: Regularization	295
9.1.1	Performing Regularized Regression	300
9.2	Support Vector Regression	301
9.2.1	Hinge Loss	301
9.2.2	From Linear Regression to Regularized Regression to Support Vector Regression	305
9.2.3	Just Do It—SVR Style	307
9.3	Piecewise Constant Regression	308
9.3.1	Implementing a Piecewise Constant Regressor	310
9.3.2	General Notes on Implementing Models	311
9.4	Regression Trees	313
9.4.1	Performing Regression with Trees	313
9.5	Comparison of Regressors: Take Three	314
9.6	EOC	318
9.6.1	Summary	318
9.6.2	Notes	318
9.6.3	Exercises	319

10 Manual Feature Engineering: Manipulating Data for Fun and Profit 321

- 10.1 Feature Engineering Terminology and Motivation 321
 - 10.1.1 Why Engineer Features? 322
 - 10.1.2 When Does Engineering Happen? 323
 - 10.1.3 How Does Feature Engineering Occur? 324
- 10.2 Feature Selection and Data Reduction: Taking out the Trash 324
- 10.3 Feature Scaling 325
- 10.4 Discretization 329
- 10.5 Categorical Coding 332
 - 10.5.1 Another Way to Code and the Curious Case of the Missing Intercept 334
- 10.6 Relationships and Interactions 341
 - 10.6.1 Manual Feature Construction 341
 - 10.6.2 Interactions 343
 - 10.6.3 Adding Features with Transformers 348
- 10.7 Target Manipulations 350
 - 10.7.1 Manipulating the Input Space 351
 - 10.7.2 Manipulating the Target 353
- 10.8 EOC 356
 - 10.8.1 Summary 356
 - 10.8.2 Notes 356
 - 10.8.3 Exercises 357

11 Tuning Hyperparameters and Pipelines 359

- 11.1 Models, Parameters, Hyperparameters 360
- 11.2 Tuning Hyperparameters 362
 - 11.2.1 A Note on Computer Science and Learning Terminology 362
 - 11.2.2 An Example of Complete Search 362
 - 11.2.3 Using Randomness to Search for a Needle in a Haystack 368

- 11.3 Down the Recursive Rabbit Hole:
Nested Cross-Validation 370
 - 11.3.1 Cross-Validation, Redux 370
 - 11.3.2 GridSearch as a Model 371
 - 11.3.3 Cross-Validation Nested
within Cross-Validation 372
 - 11.3.4 Comments on Nested
CV 375
- 11.4 Pipelines 377
 - 11.4.1 A Simple Pipeline 378
 - 11.4.2 A More Complex
Pipeline 379
- 11.5 Pipelines and Tuning Together 380
- 11.6 EOC 382
 - 11.6.1 Summary 382
 - 11.6.2 Notes 382
 - 11.6.3 Exercises 383

IV Adding Complexity 385

12 Combining Learners 387

- 12.1 Ensembles 387
- 12.2 Voting Ensembles 389
- 12.3 Bagging and Random Forests 390
 - 12.3.1 Bootstrapping 390
 - 12.3.2 From Bootstrapping to
Bagging 394
 - 12.3.3 Through the Random
Forest 396
- 12.4 Boosting 398
 - 12.4.1 Boosting Details 399
- 12.5 Comparing the Tree-Ensemble
Methods 401
- 12.6 EOC 405
 - 12.6.1 Summary 405
 - 12.6.2 Notes 405
 - 12.6.3 Exercises 406

13 Models That Engineer Features for Us 409

- 13.1 Feature Selection 411
 - 13.1.1 Single-Step Filtering with Metric-Based Feature Selection 412
 - 13.1.2 Model-Based Feature Selection 423
 - 13.1.3 Integrating Feature Selection with a Learning Pipeline 426
- 13.2 Feature Construction with Kernels 428
 - 13.2.1 A Kernel Motivator 428
 - 13.2.2 Manual Kernel Methods 433
 - 13.2.3 Kernel Methods and Kernel Options 438
 - 13.2.4 Kernelized SVCs: SVMs 442
 - 13.2.5 Take-Home Notes on SVM and an Example 443
- 13.3 Principal Components Analysis: An Unsupervised Technique 445
 - 13.3.1 A Warm Up: Centering 445
 - 13.3.2 Finding a Different Best Line 448
 - 13.3.3 A First PCA 449
 - 13.3.4 Under the Hood of PCA 452
 - 13.3.5 A Finale: Comments on General PCA 457
 - 13.3.6 Kernel PCA and Manifold Methods 458
- 13.4 EOC 462
 - 13.4.1 Summary 462
 - 13.4.2 Notes 462
 - 13.4.3 Exercises 467

14 Feature Engineering for Domains: Domain-Specific Learning 469

- 14.1 Working with Text 470
 - 14.1.1 Encoding Text 471
 - 14.1.2 Example of Text Learning 476
- 14.2 Clustering 479
 - 14.2.1 k -Means Clustering 479

- 14.3 Working with Images 481
 - 14.3.1 Bag of Visual Words 481
 - 14.3.2 Our Image Data 482
 - 14.3.3 An End-to-End System 483
 - 14.3.4 Complete Code of BoVW Transformer 491
- 14.4 EOC 493
 - 14.4.1 Summary 493
 - 14.4.2 Notes 494
 - 14.4.3 Exercises 495

15 Connections, Extensions, and Further Directions 497

- 15.1 Optimization 497
- 15.2 Linear Regression from Raw Materials 500
 - 15.2.1 A Graphical View of Linear Regression 504
- 15.3 Building Logistic Regression from Raw Materials 504
 - 15.3.1 Logistic Regression with Zero-One Coding 506
 - 15.3.2 Logistic Regression with Plus-One Minus-One Coding 508
 - 15.3.3 A Graphical View of Logistic Regression 509
- 15.4 SVM from Raw Materials 510
- 15.5 Neural Networks 512
 - 15.5.1 A NN View of Linear Regression 512
 - 15.5.2 A NN View of Logistic Regression 515
 - 15.5.3 Beyond Basic Neural Networks 516
- 15.6 Probabilistic Graphical Models 516
 - 15.6.1 Sampling 518
 - 15.6.2 A PGM View of Linear Regression 519

15.6.3	A PGM View of Logistic Regression	523
15.7	EOC	525
15.7.1	Summary	525
15.7.2	Notes	526
15.7.3	Exercises	527

A mlwpy.py Listing 529

Index 537

Whether it is called statistics or machine learning patterns from data have been touched (or so we think) by hardware and software engineers alike. It is upon software that most people focus their attention.

While many languages are used for machine learning, including R, C, Java, Fortran, and Go, Python has proven remarkably popular. This is in large part thanks to its simplicity, which makes it easy to not only make a host of different models, but to also compare features, evaluate the model quality, and score new data. The *mlwpy.py* project has quickly become one of Python's most important and powerful machine learning libraries.

While advanced mathematical concepts underpin machine learning, it is entirely possible to train complex models without a thorough background in calculus and matrix algebra. For many people, getting into machine learning through programming, rather than math, is a more attainable goal. That is precisely the goal of this book: to use Python as a hook into machine learning and then add in some math as needed. Following in the footsteps of *R for Everyone* and *Fundamentals of Machine Learning with Python for Everyone* strives to be open and accessible to anyone looking to learn about the exciting area of math and computation.

Mark Fenner has spent years practicing his craft as teacher of science and machine learning concepts to people of varying backgrounds. He has the ability to break down complex ideas into simple components. That translation results in a form of storytelling that explains concepts while motivating people and providing concrete examples. This book is easy to read, with many code samples so the reader can follow along on their computer.

With more people than ever eager to understand and implement machine learning, it is essential to have practical resources to guide them. This book, and thoughtfully Alex, fills that need with this insightful and motivating text. *Machine Learning with Python for Everyone* lives up to its name, allowing people with all degrees of previous training to quickly improve their machine learning knowledge and build exciting machine models in this important field.

Jared Landier
Series Editor

Foreword

Whether it is called statistics, data science, machine learning, or artificial intelligence, learning patterns from data is transforming the world. Nearly every industry imaginable has been touched (or soon will be) by machine learning. The combined progress of both hardware and software improvements are driving rapid advancements in the field, though it is upon software that most people focus their attention.

While many languages are used for machine learning, including R, C/C++, Fortran, and Go, Python has proven remarkably popular. This is in large part thanks to scikit-learn, which makes it easy to not only train a host of different models but to also engineer features, evaluate the model quality, and score new data. The scikit-learn project has quickly become one of Python's most important and powerful software libraries.

While advanced mathematical concepts underpin machine learning, it is entirely possible to train complex models without a thorough background in calculus and matrix algebra. For many people, getting into machine learning through programming, rather than math, is a more attainable goal. That is precisely the goal of this book: to use Python as a hook into machine learning and then add in some math as needed. Following in the footsteps of *R for Everyone* and *Pandas for Everyone*, *Machine Learning with Python for Everyone* strives to be open and accessible to anyone looking to learn about this exciting area of math and computation.

Mark Fenner has spent years practicing the communication of science and machine learning concepts to people of varying backgrounds, honing his ability to break down complex ideas into simple components. That experience results in a form of storytelling that explains concepts while minimizing jargon and providing concrete examples. The book is easy to read, with many code samples so the reader can follow along on their computer.

With more people than ever eager to understand and implement machine learning, it is essential to have practical resources to guide them, both quickly and thoughtfully. Mark fills that need with this insightful and engaging text. *Machine Learning with Python for Everyone* lives up to its name, allowing people with all manner of previous training to quickly improve their machine learning knowledge and skills, greatly increasing access to this important field.

Jared Lander,
Series Editor

Preface

In 1983, the movie *WarGames* came out. I was a preteen and I was absolutely engrossed: by the possibility of a nuclear apocalypse, by the almost magical way the lead character interacted with computer systems, but mostly by the potential of machines that could *learn*. I spent years studying the strategic nuclear arsenals of the East and the West—fortunately with a naivete of a tweener—but it was almost ten years before I took my first serious steps in computer programming. Teaching a computer to do a set process was amazing. Learning the intricacies of complex systems and bending them around my curiosity was a great experience. Still, I had a large step forward to take. A few short years later, I worked with my first program that was explicitly designed to *learn*. I was blown away and I knew I found my intellectual home. I want to share the world of *computer programs that learn* with you.

Audience

Who do I think *you* are? I've written *Machine Learning with Python for Everyone* for the absolute beginner to machine learning. Even more so, you may well have very little college-level mathematics in your toolbox and *I'm not going to try to change that*. While many machine learning books are very heavy on mathematical concepts and equations, I've done my best to *minimize* the amount of mathematical luggage you'll have to carry. I do expect, given the book's title, that you'll have some basic proficiency in Python. If you can *read* Python, you'll be able to get a lot more out of our discussions. While many books on machine learning rely on mathematics, I'm relying on stories, pictures, and Python code to communicate with you. There *will* be the occasional equation. Largely, these can be skipped if you are so inclined. But, if I've done my job well, I'll have given you enough context around the equation to maybe—just *maybe*—understand what it is trying to say.

Why might you have this book in your hand? The least common denominator is that all of my readers want to *learn* about machine learning. Now, you might be coming from very different backgrounds: a student in an introductory computing class focused on machine learning, a mid-career business analyst who all of sudden has been thrust beyond the limits of spreadsheet analysis, a tech hobbyist looking to expand her interests, or a scientist needing to analyze data in a new way. Machine learning is permeating society. Depending on your background, *Machine Learning with Python for Everyone* has different things to offer you. Even a mathematically sophisticated reader who is looking to do a break-in to machine learning using Python can get a lot out of this book.

So, my goal is to take someone with an interest or need to do some machine learning and teach them the *process* and the most important *concepts* of machine learning in a concrete way using the Python scikit-learn library and some of its friends. You'll come

away with overall patterns, strategies, pitfalls, and gotchas that will be applicable in every learning system you ever study, build, or use.

Approach

Many books that try to explain mathematical topics, such as machine learning, do so by presenting equations as if they tell a story to the uninitiated. I think that leaves many of us—even those of us who like mathematics!—stuck. Personally, I build a far better mental picture of the process of machine learning by combining visual and verbal descriptions with *running code*. I'm a computer scientist at heart and by training. I love building things. Building things is how I know that I've reached a level where I *really* understand them. You might be familiar with the phrase, "If you really want to know something, teach it to someone." Well, there's a follow-on. "If you really want to know something, teach a computer to do it!" That's my take on how I'm going to teach you machine learning. With minimal mathematics, I want to give you the concepts behind the most important and frequently used machine learning tools and techniques. Then, I want you to immediately see how to make a computer do it. One note: we won't be programming these methods from scratch. We'll be standing on the shoulders of giants and using some very powerful, time-saving, prebuilt software libraries (more on that shortly).

We won't be covering all of these libraries in great detail—there is simply too much material to do that. Instead, we are going to be practical. We are going to use the best tool for the job. I'll explain enough to orient you in the concept we're using—and then we'll get to using it. For our mathematically inclined colleagues, I'll give pointers to more in-depth references they can pursue. I'll save most of this for end-of-the-chapter notes so the rest of us can skip it easily.

If you are flipping through this introduction, deciding if you want to invest time in this book, I want to give you some insight into things that are out-of-scope for us. We aren't going to dive into mathematical proofs or rely on mathematics to explain things. There are many books out there that follow that path and I'll give pointers to my favorites at the ends of the chapters. Likewise, I'm going to assume that you are fluent in basic- to intermediate-level Python programming. However, for more advanced Python topics—and things that show up from third-party packages like NumPy or Pandas—I'll explain enough of what's going on so that you can understand each technique and its context.

Overview

In **Part I**, we establish a foundation. I'll give you some verbal and conceptual introductions to machine learning in Chapter 1. In Chapter 2 we introduce and take a slightly different approach to some mathematical and computational topics that show up repeatedly in machine learning. Chapters 3 and 4 walk you through your first steps in building, training, and evaluating learning systems that classify examples (classifiers) and quantify examples (regressors).

Part II shifts our focus to the most important aspect of applied machine learning systems: evaluating the success of our system in a realistic way. Chapter 5 talks about general

evaluation techniques that will apply to all of our learning systems. Chapters 6 and 7 take those general techniques and add evaluation capabilities for classifiers and regressors.

Part III broadens our toolbox of learning techniques and fills out the components of a practical learning system. Chapters 8 and 9 give us additional classification and regression techniques. Chapter 10 describes *feature engineering*: how we smooth the edges of rough data into forms that we can use for learning. Chapter 11 shows how to chain multiple steps together as a single learner and how to tune a learner's inner workings for better performance.

Part IV takes us beyond the basics and discusses more recent techniques that are driving machine learning forward. We look at learners that are made up of multiple little learners in Chapter 12. Chapter 13 discusses learning techniques that incorporate automated feature engineering. Chapter 14 is a wonderful capstone because it takes the techniques we describe throughout the book and applies them to two particularly interesting types of data: images and text. Chapter 15 both reviews many of the techniques we discuss and shows how they relate to more advanced learning architectures—neural networks and graphical models.

Our main focus is on the techniques of machine learning. We will investigate a number of learning algorithms and other processing methods along the way. However, completeness is not our goal. We'll discuss the most common techniques and only glance briefly at the two large subareas of machine learning: graphical models and neural, or deep, networks. However, we will see how the techniques we focus on relate to these more advanced methods.

Another topic we won't cover is implementing specific learning algorithms. We'll build on top of the algorithms that are already available in scikit-learn and friends; we'll create larger solutions using them as components. Still, someone has to implement the gears and cogs inside the black-box we funnel data into. If you are really interested in implementation aspects, you are in good company: I love them! Have all your friends buy a copy of this book, so I can argue I need to write a follow-up that dives into these lower-level details.

Acknowledgments

I must take a few moments to thank several people that have contributed greatly to this book. My editor at Pearson, Debra Williams Cauley, has been instrumental in every phase of this book's development. From our initial meetings, to her probing for a topic that might meet both our needs, to gently shepherding me through many (many!) early drafts, to constantly giving me just enough of a push to keep going, and finally climbing the steepest parts of the mountain at its peak . . . through all of these phases, Debra has shown the highest degrees of professionalism. I can only respond with a heartfelt *thank you*.

My wife, Dr. Barbara Fenner, also deserves more praise and thanks than I can give her in this short space. In addition to the burdens that any partner of an author must bear, she *also* served as my primary draft reader *and* our intrepid illustrator. She did the hard work of drafting all of the non-computer-generated diagrams in this book. While this is not our first joint academic project, it has been turned into the longest. Her patience is, by all appearances, never ending. Barbara, *I thank you!*