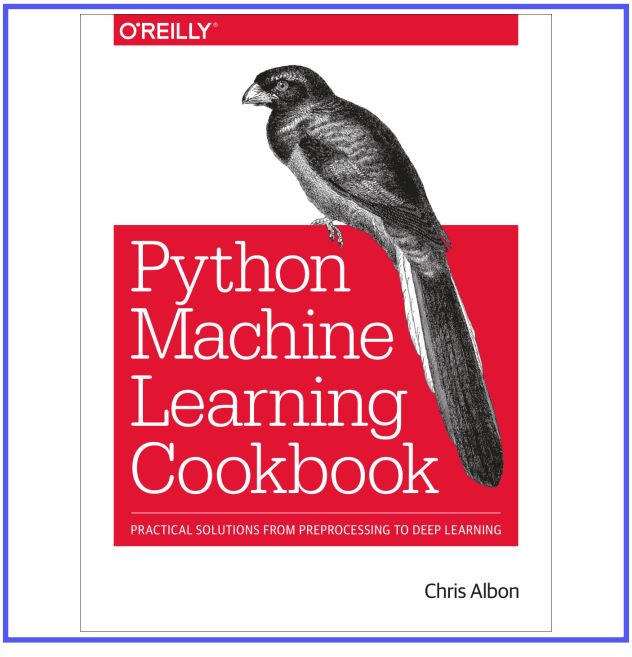


ebookbig.com

or scan the QR code to access the digital study portal.

5. Handling Categorical Data. 81
5.0 Introduction 81
5.1 Encoding Nominal Categorical Features 82
5.2 Encoding Ordinal Categorical Features 84
5.3 Encoding Dictionaries of Features 86
5.4 Imputing Missing Class Values 88
5.5 Handling Imbalanced Classes 90

6. Handling Text. 95
6.0 Introduction 95
6.1 Cleaning Text 95
6.2 Parsing and Cleaning HTML 97
6.3 Removing Punctuation 98
6.4 Tokenizing Text 98
6.5 Removing Stop Words 99
6.6 Stemming Words 100
6.7 Tagging Parts of Speech 101
6.8 Encoding Text as a Bag of Words 104
6.9 Weighting Word Importance 106

7. Handling Dates and Times. 109
7.0 Introduction 109
7.1 Converting Strings to Dates 109
7.2 Handling Time Zones 111
7.3 Selecting Dates and Times 112
7.4 Breaking Up Date Data into Multiple Features 113
7.5 Calculating the Difference Between Dates 114
7.6 Encoding Days of the Week 115
7.7 Creating a Lagged Feature 116
7.8 Using Rolling Time Windows 117
7.9 Handling Missing Data in Time Series 118

8. Handling Images. 121
8.0 Introduction 121
8.1 Loading Images 122
8.2 Saving Images 124
8.3 Resizing Images 125
8.4 Cropping Images 126
8.5 Blurring Images 128
8.6 Sharpening Images 131
8.7 Enhancing Contrast 133
8.8 Isolating Colors 135

8.9 Binarizing Images 137
8.10 Removing Backgrounds 140
8.11 Detecting Edges 144
8.12 Detecting Corners 146
8.13 Creating Features for Machine Learning 150
8.14 Encoding Mean Color as a Feature 152
8.15 Encoding Color Histograms as Features 153

9. Dimensionality Reduction Using Feature Extraction. 157
9.0 Introduction 157
9.1 Reducing Features Using Principal Components 158
9.2 Reducing Features When Data Is Linearly Inseparable 160
9.3 Reducing Features by Maximizing Class Separability 162
9.4 Reducing Features Using Matrix Factorization 165
9.5 Reducing Features on Sparse Data 166

10. Dimensionality Reduction Using Feature Selection. 169
10.0 Introduction 169
10.1 Thresholding Numerical Feature Variance 170
10.2 Thresholding Binary Feature Variance 171
10.3 Handling Highly Correlated Features 172
10.4 Removing Irrelevant Features for Classification 174
10.5 Recursively Eliminating Features 176

11. Model Evaluation. 179
11.0 Introduction 179
11.1 Cross-Validating Models 179
11.2 Creating a Baseline Regression Model 183
11.3 Creating a Baseline Classification Model 184
11.4 Evaluating Binary Classifier Predictions 186
11.5 Evaluating Binary Classifier Thresholds 189
11.6 Evaluating Multiclass Classifier Predictions 192
11.7 Visualizing a Classifier's Performance 194
11.8 Evaluating Regression Models 196
11.9 Evaluating Clustering Models 198
11.10 Creating a Custom Evaluation Metric 199
11.11 Visualizing the Effect of Training Set Size 201
11.12 Creating a Text Report of Evaluation Metrics 203
11.13 Visualizing the Effect of Hyperparameter Values 205

12. Model Selection. 209
12.0 Introduction 209

12.1 Selecting Best Models Using Exhaustive Search 210
12.2 Selecting Best Models Using Randomized Search 212
12.3 Selecting Best Models from Multiple Learning Algorithms 214
12.4 Selecting Best Models When Preprocessing 215
12.5 Speeding Up Model Selection with Parallelization 217
12.6 Speeding Up Model Selection Using Algorithm-Specific Methods 219
12.7 Evaluating Performance After Model Selection 220

13. Linear Regression. 223
13.0 Introduction 223
13.1 Fitting a Line 223
13.2 Handling Interactive Effects 225
13.3 Fitting a Nonlinear Relationship 227
13.4 Reducing Variance with Regularization 229
13.5 Reducing Features with Lasso Regression 231

14. Trees and Forests. 233
14.0 Introduction 233
14.1 Training a Decision Tree Classifier 233
14.2 Training a Decision Tree Regressor 235
14.3 Visualizing a Decision Tree Model 236
14.4 Training a Random Forest Classifier 238
14.5 Training a Random Forest Regressor 240
14.6 Identifying Important Features in Random Forests 241
14.7 Selecting Important Features in Random Forests 243
14.8 Handling Imbalanced Classes 245
14.9 Controlling Tree Size 246
14.10 Improving Performance Through Boosting 247
14.11 Evaluating Random Forests with Out-of-Bag Errors 249

15. K-Nearest Neighbors. 251
15.0 Introduction 251
15.1 Finding an Observation's Nearest Neighbors 251
15.2 Creating a K-Nearest Neighbor Classifier 254
15.3 Identifying the Best Neighborhood Size 256
15.4 Creating a Radius-Based Nearest Neighbor Classifier 257

16. Logistic Regression. 259
16.0 Introduction 259
16.1 Training a Binary Classifier 259
16.2 Training a Multiclass Classifier 261
16.3 Reducing Variance Through Regularization 262

16.4 Training a Classifier on Very Large Data 263
16.5 Handling Imbalanced Classes 264

17. Support Vector Machines. 267
17.0 Introduction 267
17.1 Training a Linear Classifier 267
17.2 Handling Linearly Inseparable Classes Using Kernels 270
17.3 Creating Predicted Probabilities 274
17.4 Identifying Support Vectors 276
17.5 Handling Imbalanced Classes 277

18. Naive Bayes. 279
18.0 Introduction 279
18.1 Training a Classifier for Continuous Features 280
18.2 Training a Classifier for Discrete and Count Features 282
18.3 Training a Naive Bayes Classifier for Binary Features 283
18.4 Calibrating Predicted Probabilities 284

19. Clustering. 287
19.0 Introduction 287
19.1 Clustering Using K-Means 287
19.2 Speeding Up K-Means Clustering 290
19.3 Clustering Using Meanshift 291
19.4 Clustering Using DBSCAN 292
19.5 Clustering Using Hierarchical Merging 294

20. Neural Networks. 297
20.0 Introduction 297
20.1 Preprocessing Data for Neural Networks 298
20.2 Designing a Neural Network 300
20.3 Training a Binary Classifier 303
20.4 Training a Multiclass Classifier 305
20.5 Training a Regressor 307
20.6 Making Predictions 309
20.7 Visualize Training History 310
20.8 Reducing Overfitting with Weight Regularization 313
20.9 Reducing Overfitting with Early Stopping 315
20.10 Reducing Overfitting with Dropout 317
20.11 Saving Model Training Progress 319
20.12 k-Fold Cross-Validating Neural Networks 321
20.13 Tuning Neural Networks 322
20.14 Visualizing Neural Networks 325

20.15 Classifying Images 327
20.16 Improving Performance with Image Augmentation 331
20.17 Classifying Text 333

21. Saving and Loading Trained Models. 337
21.0 Introduction 337
21.1 Saving and Loading a scikit-learn Model 337
21.2 Saving and Loading a Keras Model 339

Index. 341

Preface

Over the last few years machine learning has become embedded in a wide variety of day-to-day business, nonprofit, and government operations. As the popularity of machine learning increased, a cottage industry of high-quality literature that taught applied machine learning to practitioners developed. This literature has been highly successful in training an entire generation of data scientists and machine learning engineers. This literature also approached the topic of machine learning from the perspective of providing a learning resource to teach an individual what machine learning is and how it works. However, while fruitful, this approach left out a different perspective on the topic: the nuts and bolts of doing machine learning day to day. That is the motivation of this book—not as a tome of machine learning knowledge for the student but as a wrench for the professional, to sit with dog-eared pages on desks ready to solve the practical day-to-day problems of a machine learning practitioner.

More specifically, the book takes a task-based approach to machine learning, with almost 200 self-contained solutions (you can copy and paste the code and it'll run) for the most common tasks a data scientist or machine learning engineer building a model will run into.

The ultimate goal is for the book to be a reference for people building real machine learning systems. For example, imagine a reader has a JSON file containing 1,000 categorical and numerical features with missing data and categorical target vectors with imbalanced classes, and wants an interpretable model. The motivation for this book is to provide recipes to help the reader learn processes such as:

- 2.5 Loading a JSON file
- 4.2 Standardizing a Feature
- 5.3 Encoding Dictionaries of Features
- 5.4 Imputing Missing Class Values

- 9.1 Reducing Features Using Principal Components
- 12.2 Selecting Best Models Using Randomized Search
- 14.4 Training a Random Forest Classifier
- 14.7 Selecting Random Features in Random Forests

The goal is for the reader to be able to:

1. Copy/paste the code and gain confidence that it actually works with the included toy dataset.
2. Read the discussion to gain an understanding of the theory behind the technique the code is executing and learn which parameters are important to consider.
3. Insert/combine/adapt the code from the recipes to construct the actual application.

Who This Book Is For

This book is not an introduction to machine learning. If you are not comfortable with the basic concepts of machine learning or have never spent time learning machine learning, do not buy this book. Instead, this book is for the machine learning practitioner who, while comfortable with the theory and concepts of machine learning, would benefit from a quick reference containing code to solve challenges he runs into working on machine learning on an everyday basis.

This book assumes the reader is comfortable with the Python programming language and package management.

Who This Book Is Not For

As stated previously, this book is not an introduction to machine learning. This book should not be your first. If you are unfamiliar with concepts like cross-validation, random forest, and gradient descent, you will likely not benefit from this book as much as one of the many high-quality texts specifically designed to introduce you to the topic. I recommend reading one of those books and then coming back to this book to learn working, practical solutions for machine learning.

Terminology Used in This Book

Machine learning draws upon techniques from a wide range of fields, including computer science, statistics, and mathematics. For this reason, there is significant variation in the terminology used in the discussions of machine learning:

Observation
: A single unit in our level of observation—for example, a person, a sale, or a record.

Learning algorithms
: An algorithm used to learn the best parameters of a model—for example, linear regression, naive Bayes, or decision trees.

Models
: An output of a learning algorithm's training. Learning algorithms train models, which we then use to make predictions.

Parameters
: The weights or coefficients of a model learned through training.

Hyperparameters
: The settings of a learning algorithm that need to be set before training.

Performance
: A metric used to evaluate a model.

Loss
: A metric to maximize or minimize through training.

Train
: Applying a learning algorithm to data using numerical approaches like gradient descent.

Fit
: Applying a learning algorithm to data using analytical approaches.

Data
: A collection of observations.

Acknowledgments

This book would not have been possible without the kind help of a number of friends and strangers. Listing everyone who lent a hand to this project would be impossible, but I wanted to at least mention: Angela Bassa, Teresa Borcuch, Justin Bozonier, Andre deBruin, Numa Dhamani, Dan Friedman, Joel Grus, Sarah Guido, Bill Kambouroglou, Mat Kelcey, Lizzie Kumar, Hilary Parker, Niti Paudyal, Sebastian Raschka, and Shreya Shankar.

I owe them all a beer or five.