grokking Machine Learning

Luis G. Serrano

Foreword by Sebastian Thrun

MANNING
Shelter Island

For online information and ordering of this and other Manning books, please visit www.manning.com. The publisher offers discounts on this book when ordered in quantity. For more information, please contact

Special Sales Department
Manning Publications Co.
20 Baldwin Road, PO Box 761
Shelter Island, NY 11964
Email: orders@manning.com

©2021 by Manning Publications Co. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by means electronic, mechanical, photocopying, or otherwise, without prior written permission of the publisher.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in the book, and Manning Publications was aware of a trademark claim, the designations have been printed in initial caps or all caps.

♾ Recognizing the importance of preserving what has been written, it is Manning's policy to have the books we publish printed on acid-free paper, and we exert our best efforts to that end. Recognizing also our responsibility to conserve the resources of our planet, Manning books are printed on paper that is at least 15 percent recycled and processed without the use of elemental chlorine.

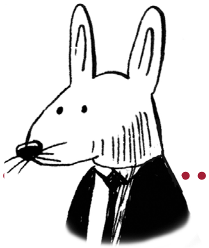
The author and publisher have made every effort to ensure that the information in this book was correct at press time. The author and publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause, or from any usage of the information herein.

Manning Publications Co.
20 Baldwin Road
Shelter Island, NY 11964

Development editor: Marina Michaels
Technical development editor: Kris Athi
Review editor: Aleksander Dragosavljević
Production editor: Keri Hales
Copy editor: Pamela Hunt
Proofreader: Jason Everett
Technical proofreader: Karsten Strøbæk, Shirley Yap
Typesetter: Dennis Dalinnik
Cover designer: Leslie Haimes

ISBN: 9781617295911
Printed in the United States of America

contents



foreword ix

preface xi

acknowledgments xiii

about this book xv

about the author xix

1 What is machine learning? It is common sense, except done by a computer 1

Do I need a heavy math and coding background to understand machine learning? 2

OK, so what exactly is machine learning? 3

How do we get machines to make decisions with data? The remember-formulate-predict framework 6

2 Types of machine learning 15

What is the difference between labeled and unlabeled data? 17

Supervised learning: The branch of machine learning that works with labeled data 18

Unsupervised learning: The branch of machine learning that works with unlabeled data 22

What is reinforcement learning? 29

3 Drawing a line close to our points: Linear regression 35

The problem: We need to predict the price of a house 37
The solution: Building a regression model for housing prices 38
How to get the computer to draw this line: The linear regression algorithm 44
How do we measure our results? The error function 60
Real-life application: Using Turi Create to predict housing prices in India 67
What if the data is not in a line? Polynomial regression 69
Parameters and hyperparameters 71
Applications of regression 72

4 Optimizing the training process: Underfitting, overfitting, testing, and regularization 77

An example of underfitting and overfitting using polynomial regression 79
How do we get the computer to pick the right model? By testing 81
Where did we break the golden rule, and how do we fix it? The validation set 84
A numerical way to decide how complex our model should be: The model complexity graph 85
Another alternative to avoiding overfitting: Regularization 86
Polynomial regression, testing, and regularization with Turi Create 95

5 Using lines to split our points: The perceptron algorithm 103

The problem: We are on an alien planet, and we don't know their language! 106
How do we determine whether a classifier is good or bad? The error function 121
How to find a good classifier? The perceptron algorithm 129
Coding the perceptron algorithm 137
Applications of the perceptron algorithm 142

6 A continuous approach to splitting points: Logistic classifiers 147

Logistic classifiers: A continuous version of perceptron classifiers 149
How to find a good logistic classifier? The logistic regression algorithm 160
Coding the logistic regression algorithm 166
Real-life application: Classifying IMDB reviews with Turi Create 171
Classifying into multiple classes: The softmax function 173

7 How do you measure classification models? Accuracy and its friends 177

Accuracy: How often is my model correct? 178
How to fix the accuracy problem? Defining different types of errors and how to measure them 179
A useful tool to evaluate our model: The receiver operating characteristic (ROC) curve 189

8 Using probability to its maximum: The naive Bayes model 205

Sick or healthy? A story with Bayes' theorem as the hero 207
Use case: Spam-detection model 212
Building a spam-detection model with real data 226

9 Splitting data by asking questions: Decision trees 233

The problem: We need to recommend apps to users according to what they are likely to download 240
The solution: Building an app-recommendation system 241
Beyond questions like yes/no 257
The graphical boundary of decision trees 261
Real-life application: Modeling student admissions with Scikit-Learn 264
Decision trees for regression 268
Applications 272

10 Combining building blocks to gain more power: Neural networks 277

Neural networks with an example: A more complicated alien planet 279
Training neural networks 292
Coding neural networks in Keras 299
Neural networks for regression 308
Other architectures for more complex datasets 309

11 Finding boundaries with style: Support vector machines and the kernel method 315

Using a new error function to build better classifiers 318
Coding support vector machines in Scikit-Learn 324
Training SVMs with nonlinear boundaries: The kernel method 326

12 Combining models to maximize results: Ensemble learning 351

With a little help from our friends 352
Bagging: Joining some weak learners randomly to build a strong learner 354
AdaBoost: Joining weak learners in a clever way to build a strong learner 360
Gradient boosting: Using decision trees to build strong learners 370
XGBoost: An extreme way to do gradient boosting 375
Applications of ensemble methods 384

13 Putting it all in practice: A real-life example of data engineering and machine learning 387

The Titanic dataset 388
Cleaning up our dataset: Missing values and how to deal with them 392
Feature engineering: Transforming the features in our dataset before training the models 395
Training our models 400
Tuning the hyperparameters to find the best model: Grid search 405
Using *K*-fold cross-validation to reuse our data as training and validation 408