

# Python Machine Learning

Unlock deeper insights into machine learning with this vital guide to cutting-edge predictive analytics

**Sebastian Raschka**



**open source**  
community experience distilled



BIRMINGHAM - MUMBAI

# Table of Contents

<b>Preface</b>	<b>vii</b>
<b>Chapter 1: Giving Computers the Ability to Learn from Data</b>	<b>1</b>
<b>Building intelligent machines to transform data into knowledge</b>	<b>2</b>
<b>The three different types of machine learning</b>	<b>2</b>
Making predictions about the future with supervised learning	3
Classification for predicting class labels	3
Regression for predicting continuous outcomes	4
Solving interactive problems with reinforcement learning	6
Discovering hidden structures with unsupervised learning	6
Finding subgroups with clustering	7
Dimensionality reduction for data compression	7
<b>An introduction to the basic terminology and notations</b>	<b>8</b>
<b>A roadmap for building machine learning systems</b>	<b>10</b>
Preprocessing – getting data into shape	11
Training and selecting a predictive model	12
Evaluating models and predicting unseen data instances	13
<b>Using Python for machine learning</b>	<b>13</b>
Installing Python packages	13
<b>Summary</b>	<b>15</b>
<b>Chapter 2: Training Machine Learning Algorithms for Classification</b>	<b>17</b>
<b>Artificial neurons – a brief glimpse into the early history of machine learning</b>	<b>18</b>
<b>Implementing a perceptron learning algorithm in Python</b>	<b>24</b>
Training a perceptron model on the Iris dataset	27
<b>Adaptive linear neurons and the convergence of learning</b>	<b>33</b>
Minimizing cost functions with gradient descent	34

Implementing an Adaptive Linear Neuron in Python	36
Large scale machine learning and stochastic gradient descent	42
<b>Summary</b>	<b>47</b>
<b>Chapter 3: A Tour of Machine Learning Classifiers Using Scikit-learn</b>	<b>49</b>
<b>Choosing a classification algorithm</b>	<b>49</b>
<b>First steps with scikit-learn</b>	<b>50</b>
Training a perceptron via scikit-learn	50
<b>Modeling class probabilities via logistic regression</b>	<b>56</b>
Logistic regression intuition and conditional probabilities	56
Learning the weights of the logistic cost function	59
Training a logistic regression model with scikit-learn	62
Tackling overfitting via regularization	65
<b>Maximum margin classification with support vector machines</b>	<b>69</b>
Maximum margin intuition	70
Dealing with the nonlinearly separable case using slack variables	71
Alternative implementations in scikit-learn	74
<b>Solving nonlinear problems using a kernel SVM</b>	<b>75</b>
Using the kernel trick to find separating hyperplanes in higher dimensional space	77
<b>Decision tree learning</b>	<b>80</b>
Maximizing information gain – getting the most bang for the buck	82
Building a decision tree	88
Combining weak to strong learners via random forests	90
<b>K-nearest neighbors – a lazy learning algorithm</b>	<b>92</b>
<b>Summary</b>	<b>96</b>
<b>Chapter 4: Building Good Training Sets – Data Preprocessing</b>	<b>99</b>
<b>Dealing with missing data</b>	<b>99</b>
Eliminating samples or features with missing values	101
Imputing missing values	102
Understanding the scikit-learn estimator API	102
<b>Handling categorical data</b>	<b>104</b>
Mapping ordinal features	104
Encoding class labels	105
Performing one-hot encoding on nominal features	106
<b>Partitioning a dataset in training and test sets</b>	<b>108</b>
<b>Bringing features onto the same scale</b>	<b>110</b>
<b>Selecting meaningful features</b>	<b>112</b>
Sparse solutions with L1 regularization	112

---

Sequential feature selection algorithms	118
<b>Assessing feature importance with random forests</b>	<b>124</b>
<b>Summary</b>	<b>126</b>
<b>Chapter 5: Compressing Data via Dimensionality Reduction</b>	<b>127</b>
<b>Unsupervised dimensionality reduction via principal component analysis</b>	<b>128</b>
Total and explained variance	129
Feature transformation	133
Principal component analysis in scikit-learn	135
<b>Supervised data compression via linear discriminant analysis</b>	<b>138</b>
Computing the scatter matrices	140
Selecting linear discriminants for the new feature subspace	143
Projecting samples onto the new feature space	145
LDA via scikit-learn	146
<b>Using kernel principal component analysis for nonlinear mappings</b>	<b>148</b>
Kernel functions and the kernel trick	148
Implementing a kernel principal component analysis in Python	154
Example 1 – separating half-moon shapes	155
Example 2 – separating concentric circles	159
Projecting new data points	162
Kernel principal component analysis in scikit-learn	166
<b>Summary</b>	<b>167</b>
<b>Chapter 6: Learning Best Practices for Model Evaluation and Hyperparameter Tuning</b>	<b>169</b>
<b>Streamlining workflows with pipelines</b>	<b>169</b>
Loading the Breast Cancer Wisconsin dataset	170
Combining transformers and estimators in a pipeline	171
<b>Using k-fold cross-validation to assess model performance</b>	<b>173</b>
The holdout method	173
K-fold cross-validation	175
<b>Debugging algorithms with learning and validation curves</b>	<b>179</b>
Diagnosing bias and variance problems with learning curves	180
Addressing overfitting and underfitting with validation curves	183
<b>Fine-tuning machine learning models via grid search</b>	<b>185</b>
Tuning hyperparameters via grid search	186
Algorithm selection with nested cross-validation	187
<b>Looking at different performance evaluation metrics</b>	<b>189</b>
Reading a confusion matrix	190
Optimizing the precision and recall of a classification model	191

---

Plotting a receiver operating characteristic	193
The scoring metrics for multiclass classification	197
<b>Summary</b>	<b>198</b>
<b>Chapter 7: Combining Different Models for Ensemble Learning</b>	<b>199</b>
<b>Learning with ensembles</b>	<b>199</b>
<b>Implementing a simple majority vote classifier</b>	<b>203</b>
Combining different algorithms for classification with majority vote	210
<b>Evaluating and tuning the ensemble classifier</b>	<b>213</b>
<b>Bagging – building an ensemble of classifiers from bootstrap samples</b>	<b>219</b>
<b>Leveraging weak learners via adaptive boosting</b>	<b>224</b>
<b>Summary</b>	<b>232</b>
<b>Chapter 8: Applying Machine Learning to Sentiment Analysis</b>	<b>233</b>
<b>Obtaining the IMDb movie review dataset</b>	<b>233</b>
<b>Introducing the bag-of-words model</b>	<b>236</b>
Transforming words into feature vectors	236
Assessing word relevancy via term frequency-inverse document frequency	238
Cleaning text data	240
Processing documents into tokens	242
<b>Training a logistic regression model for document classification</b>	<b>244</b>
<b>Working with bigger data – online algorithms and out-of-core learning</b>	<b>246</b>
<b>Summary</b>	<b>250</b>
<b>Chapter 9: Embedding a Machine Learning Model into a Web Application</b>	<b>251</b>
<b>Serializing fitted scikit-learn estimators</b>	<b>252</b>
<b>Setting up a SQLite database for data storage</b>	<b>255</b>
<b>Developing a web application with Flask</b>	<b>257</b>
Our first Flask web application	258
Form validation and rendering	259
<b>Turning the movie classifier into a web application</b>	<b>264</b>
<b>Deploying the web application to a public server</b>	<b>272</b>
Updating the movie review classifier	274
<b>Summary</b>	<b>276</b>

---

<b>Chapter 10: Predicting Continuous Target Variables with Regression Analysis</b>	<b>277</b>
<b>Introducing a simple linear regression model</b>	<b>278</b>
<b>Exploring the Housing Dataset</b>	<b>279</b>
Visualizing the important characteristics of a dataset	280
<b>Implementing an ordinary least squares linear regression model</b>	<b>285</b>
Solving regression for regression parameters with gradient descent	285
Estimating the coefficient of a regression model via scikit-learn	289
<b>Fitting a robust regression model using RANSAC</b>	<b>291</b>
<b>Evaluating the performance of linear regression models</b>	<b>294</b>
<b>Using regularized methods for regression</b>	<b>297</b>
<b>Turning a linear regression model into a curve – polynomial regression</b>	<b>298</b>
Modeling nonlinear relationships in the Housing Dataset	300
Dealing with nonlinear relationships using random forests	304
Decision tree regression	304
Random forest regression	306
<b>Summary</b>	<b>309</b>
<b>Chapter 11: Working with Unlabeled Data – Clustering Analysis</b>	<b>311</b>
<b>Grouping objects by similarity using k-means</b>	<b>312</b>
K-means++	315
Hard versus soft clustering	317
Using the elbow method to find the optimal number of clusters	320
Quantifying the quality of clustering via silhouette plots	321
<b>Organizing clusters as a hierarchical tree</b>	<b>326</b>
Performing hierarchical clustering on a distance matrix	328
Attaching dendrograms to a heat map	332
Applying agglomerative clustering via scikit-learn	334
<b>Locating regions of high density via DBSCAN</b>	<b>334</b>
<b>Summary</b>	<b>340</b>
<b>Chapter 12: Training Artificial Neural Networks for Image Recognition</b>	<b>341</b>
<b>Modeling complex functions with artificial neural networks</b>	<b>342</b>
Single-layer neural network recap	343
Introducing the multi-layer neural network architecture	345
Activating a neural network via forward propagation	347

<b>Classifying handwritten digits</b>	<b>350</b>
Obtaining the MNIST dataset	351
Implementing a multi-layer perceptron	356
<b>Training an artificial neural network</b>	<b>365</b>
Computing the logistic cost function	365
Training neural networks via backpropagation	368
<b>Developing your intuition for backpropagation</b>	<b>372</b>
<b>Debugging neural networks with gradient checking</b>	<b>373</b>
<b>Convergence in neural networks</b>	<b>379</b>
<b>Other neural network architectures</b>	<b>381</b>
Convolutional Neural Networks	381
Recurrent Neural Networks	383
<b>A few last words about neural network implementation</b>	<b>384</b>
<b>Summary</b>	<b>385</b>
<b>Chapter 13: Parallelizing Neural Network Training with Theano</b>	<b>387</b>
<b>Building, compiling, and running expressions with Theano</b>	<b>388</b>
What is Theano?	390
First steps with Theano	391
Configuring Theano	392
Working with array structures	394
Wrapping things up – a linear regression example	397
<b>Choosing activation functions for feedforward neural networks</b>	<b>401</b>
Logistic function recap	402
Estimating probabilities in multi-class classification via the softmax function	404
Broadening the output spectrum by using a hyperbolic tangent	405
<b>Training neural networks efficiently using Keras</b>	<b>408</b>
<b>Summary</b>	<b>414</b>
<b>Index</b>	<b>417</b>

---