

Python Machine Learning

Third Edition

Machine Learning and Deep Learning with Python,
scikit-learn, and TensorFlow 2

Sebastian Raschka

Vahid Mirjalili

Packt
BIRMINGHAM - MUMBAI

Chapter 4: Building Good Training Datasets – Data Preprocessing	109
Dealing with missing data	109
Identifying missing values in tabular data	110
Eliminating training examples or features with missing values	111
Imputing missing values	112
Understanding the scikit-learn estimator API	113
Handling categorical data	115
Categorical data encoding with pandas	116
Mapping ordinal features	116
Encoding class labels	117
Performing one-hot encoding on nominal features	118
Partitioning a dataset into separate training and test datasets	121
Bringing features onto the same scale	124
Selecting meaningful features	127
L1 and L2 regularization as penalties against model complexity	128
A geometric interpretation of L2 regularization	128
Sparse solutions with L1 regularization	131
Sequential feature selection algorithms	135
Assessing feature importance with random forests	141
Summary	143
Chapter 5: Compressing Data via Dimensionality Reduction	145
Unsupervised dimensionality reduction via principal component analysis	145
The main steps behind principal component analysis	146
Extracting the principal components step by step	148
Total and explained variance	151
Feature transformation	152
Principal component analysis in scikit-learn	155
Supervised data compression via linear discriminant analysis	159
Principal component analysis versus linear discriminant analysis	159
The inner workings of linear discriminant analysis	160
Computing the scatter matrices	161
Selecting linear discriminants for the new feature subspace	164
Projecting examples onto the new feature space	167
LDA via scikit-learn	168
Using kernel principal component analysis for nonlinear mappings	169
Kernel functions and the kernel trick	170
Implementing a kernel principal component analysis in Python	175
Example 1 – separating half-moon shapes	177
Example 2 – separating concentric circles	180

Table of Contents

Preface	xiii
Chapter 1: Giving Computers the Ability to Learn from Data	1
Building intelligent machines to transform data into knowledge	1
The three different types of machine learning	2
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	7
Finding subgroups with clustering	7
Dimensionality reduction for data compression	8
Introduction to the basic terminology and notations	8
Notation and conventions used in this book	9
Machine learning terminology	11
A roadmap for building machine learning systems	11
Preprocessing – getting data into shape	12
Training and selecting a predictive model	13
Evaluating models and predicting unseen data instances	14
Using Python for machine learning	14
Installing Python and packages from the Python Package Index	14
Using the Anaconda Python distribution and package manager	15
Packages for scientific computing, data science, and machine learning	16
Summary	16

Chapter 2: Training Simple Machine Learning Algorithms for Classification	19
Artificial neurons – a brief glimpse into the early history of machine learning	20
The formal definition of an artificial neuron	21
The perceptron learning rule	23
Implementing a perceptron learning algorithm in Python	26
An object-oriented perceptron API	26
Training a perceptron model on the Iris dataset	30
Adaptive linear neurons and the convergence of learning	36
Minimizing cost functions with gradient descent	37
Implementing Adaline in Python	40
Improving gradient descent through feature scaling	44
Large-scale machine learning and stochastic gradient descent	46
Summary	51
Chapter 3: A Tour of Machine Learning Classifiers	
Using scikit-learn	53
Choosing a classification algorithm	53
First steps with scikit-learn – training a perceptron	54
Modeling class probabilities via logistic regression	60
Logistic regression and conditional probabilities	60
Learning the weights of the logistic cost function	65
Converting an Adaline implementation into an algorithm for logistic regression	67
Training a logistic regression model with scikit-learn	72
Tackling overfitting via regularization	75
Maximum margin classification with support vector machines	79
Maximum margin intuition	79
Dealing with a nonlinearly separable case using slack variables	81
Alternative implementations in scikit-learn	83
Solving nonlinear problems using a kernel SVM	84
Kernel methods for linearly inseparable data	84
Using the kernel trick to find separating hyperplanes in a high-dimensional space	86
Decision tree learning	90
Maximizing IG – getting the most bang for your buck	91
Building a decision tree	96
Combining multiple decision trees via random forests	100
K-nearest neighbors – a lazy learning algorithm	103
Summary	108

Projecting new data points	183
Kernel principal component analysis in scikit-learn	187
Summary	188
Chapter 6: Learning Best Practices for Model Evaluation and Hyperparameter Tuning	191
Streamlining workflows with pipelines	191
Loading the Breast Cancer Wisconsin dataset	192
Combining transformers and estimators in a pipeline	193
Using k-fold cross-validation to assess model performance	195
The holdout method	196
K-fold cross-validation	197
Debugging algorithms with learning and validation curves	201
Diagnosing bias and variance problems with learning curves	201
Addressing over- and underfitting with validation curves	205
Fine-tuning machine learning models via grid search	207
Tuning hyperparameters via grid search	207
Algorithm selection with nested cross-validation	209
Looking at different performance evaluation metrics	211
Reading a confusion matrix	211
Optimizing the precision and recall of a classification model	213
Plotting a receiver operating characteristic	216
Scoring metrics for multiclass classification	219
Dealing with class imbalance	220
Summary	222
Chapter 7: Combining Different Models for Ensemble Learning	223
Learning with ensembles	223
Combining classifiers via majority vote	227
Implementing a simple majority vote classifier	228
Using the majority voting principle to make predictions	234
Evaluating and tuning the ensemble classifier	237
Bagging – building an ensemble of classifiers from bootstrap samples	243
Bagging in a nutshell	244
Applying bagging to classify examples in the Wine dataset	245
Leveraging weak learners via adaptive boosting	249
How boosting works	250
Applying AdaBoost using scikit-learn	254
Summary	257
Chapter 8: Applying Machine Learning to Sentiment Analysis	259
Preparing the IMDb movie review data for text processing	259

Obtaining the movie review dataset	260
Preprocessing the movie dataset into a more convenient format	260
Introducing the bag-of-words model	262
Transforming words into feature vectors	263
Assessing word relevancy via term frequency-inverse document frequency	265
Cleaning text data	267
Processing documents into tokens	269
Training a logistic regression model for document classification	272
Working with bigger data – online algorithms and out-of-core learning	274
Topic modeling with Latent Dirichlet Allocation	278
Decomposing text documents with LDA	279
LDA with scikit-learn	279
Summary	283
Chapter 9: Embedding a Machine Learning Model into a Web Application	285
Serializing fitted scikit-learn estimators	285
Setting up an SQLite database for data storage	289
Developing a web application with Flask	291
Our first Flask web application	292
Form validation and rendering	294
Setting up the directory structure	295
Implementing a macro using the Jinja2 templating engine	296
Adding style via CSS	296
Creating the result page	298
Turning the movie review classifier into a web application	300
Files and folders – looking at the directory tree	301
Implementing the main application as app.py	302
Setting up the review form	305
Creating a results page template	306
Deploying the web application to a public server	309
Creating a PythonAnywhere account	309
Uploading the movie classifier application	310
Updating the movie classifier	311
Summary	314
Chapter 10: Predicting Continuous Target Variables with Regression Analysis	315
Introducing linear regression	315
Simple linear regression	316
Multiple linear regression	317

Exploring the Housing dataset	318
Loading the Housing dataset into a data frame	318
Visualizing the important characteristics of a dataset	320
Looking at relationships using a correlation matrix	322
Implementing an ordinary least squares linear regression model	325
Solving regression for regression parameters with gradient descent	325
Estimating the coefficient of a regression model via scikit-learn	330
Fitting a robust regression model using RANSAC	332
Evaluating the performance of linear regression models	334
Using regularized methods for regression	337
Turning a linear regression model into a curve – polynomial regression	339
Adding polynomial terms using scikit-learn	340
Modeling nonlinear relationships in the Housing dataset	342
Dealing with nonlinear relationships using random forests	345
Decision tree regression	346
Random forest regression	348
Summary	350
Chapter 11: Working with Unlabeled Data – Clustering Analysis	353
Grouping objects by similarity using k-means	353
K-means clustering using scikit-learn	354
A smarter way of placing the initial cluster centroids using k-means++	358
Hard versus soft clustering	359
Using the elbow method to find the optimal number of clusters	361
Quantifying the quality of clustering via silhouette plots	363
Organizing clusters as a hierarchical tree	367
Grouping clusters in bottom-up fashion	368
Performing hierarchical clustering on a distance matrix	369
Attaching dendograms to a heat map	373
Applying agglomerative clustering via scikit-learn	375
Locating regions of high density via DBSCAN	376
Summary	382
Chapter 12: Implementing a Multilayer Artificial Neural Network from Scratch	383
Modeling complex functions with artificial neural networks	383
Single-layer neural network recap	385
Introducing the multilayer neural network architecture	387
Activating a neural network via forward propagation	391
Classifying handwritten digits	393
Obtaining and preparing the MNIST dataset	394

Implementing a multilayer perceptron	400
Training an artificial neural network	412
Computing the logistic cost function	412
Developing your understanding of backpropagation	415
Training neural networks via backpropagation	417
About the convergence in neural networks	421
A few last words about the neural network implementation	422
Summary	423
Chapter 13: Parallelizing Neural Network Training with TensorFlow	425
TensorFlow and training performance	426
Performance challenges	426
What is TensorFlow?	427
How we will learn TensorFlow	429
First steps with TensorFlow	429
Installing TensorFlow	429
Creating tensors in TensorFlow	430
Manipulating the data type and shape of a tensor	431
Applying mathematical operations to tensors	432
Split, stack, and concatenate tensors	434
Building input pipelines using <code>tf.data</code> – the TensorFlow Dataset API	435
Creating a TensorFlow Dataset from existing tensors	436
Combining two tensors into a joint dataset	437
Shuffle, batch, and repeat	439
Creating a dataset from files on your local storage disk	441
Fetching available datasets from the <code>tensorflow_datasets</code> library	445
Building an NN model in TensorFlow	450
The TensorFlow Keras API (<code>tf.keras</code>)	451
Building a linear regression model	451
Model training via the <code>.compile()</code> and <code>.fit()</code> methods	456
Building a multilayer perceptron for classifying flowers in the Iris dataset	457
Evaluating the trained model on the test dataset	461
Saving and reloading the trained model	461
Choosing activation functions for multilayer neural networks	462
Logistic function recap	463
Estimating class probabilities in multiclass classification via the softmax function	465
Broadening the output spectrum using a hyperbolic tangent	466
Rectified linear unit activation	468
Summary	470

Chapter 14: Going Deeper – The Mechanics of TensorFlow	471
The key features of TensorFlow	472
TensorFlow's computation graphs: migrating to TensorFlow v2	473
Understanding computation graphs	473
Creating a graph in TensorFlow v1.x	474
Migrating a graph to TensorFlow v2	475
Loading input data into a model: TensorFlow v1.x style	476
Loading input data into a model: TensorFlow v2 style	476
Improving computational performance with function decorators	477
TensorFlow Variable objects for storing and updating model parameters	479
Computing gradients via automatic differentiation and GradientTape	483
Computing the gradients of the loss with respect to trainable variables	483
Computing gradients with respect to non-trainable tensors	485
Keeping resources for multiple gradient computations	485
Simplifying implementations of common architectures via the Keras API	486
Solving an XOR classification problem	489
Making model building more flexible with Keras' functional API	494
Implementing models based on Keras' Model class	496
Writing custom Keras layers	497
TensorFlow Estimators	501
Working with feature columns	501
Machine learning with pre-made Estimators	506
Using Estimators for MNIST handwritten digit classification	510
Creating a custom Estimator from an existing Keras model	512
Summary	515
Chapter 15: Classifying Images with Deep Convolutional Neural Networks	517
The building blocks of CNNs	518
Understanding CNNs and feature hierarchies	518
Performing discrete convolutions	520
Discrete convolutions in one dimension	521
Padding inputs to control the size of the output feature maps	523
Determining the size of the convolution output	525
Performing a discrete convolution in 2D	526
Subsampling layers	530
Putting everything together – implementing a CNN	532
Working with multiple input or color channels	532
Regularizing an NN with dropout	536
Loss functions for classification	539

Implementing a deep CNN using TensorFlow	542
The multilayer CNN architecture	542
Loading and preprocessing the data	543
Implementing a CNN using the TensorFlow Keras API	544
Configuring CNN layers in Keras	544
Constructing a CNN in Keras	545
Gender classification from face images using a CNN	550
Loading the CelebA dataset	551
Image transformation and data augmentation	552
Training a CNN gender classifier	558
Summary	564
Chapter 16: Modeling Sequential Data Using Recurrent Neural Networks	567
Introducing sequential data	568
Modeling sequential data – order matters	568
Representing sequences	569
The different categories of sequence modeling	570
RNNs for modeling sequences	571
Understanding the RNN looping mechanism	571
Computing activations in an RNN	574
Hidden-recurrence versus output-recurrence	577
The challenges of learning long-range interactions	580
Long short-term memory cells	582
Implementing RNNs for sequence modeling in TensorFlow	584
Project one – predicting the sentiment of IMDb movie reviews	585
Preparing the movie review data	585
Embedding layers for sentence encoding	590
Building an RNN model	592
Building an RNN model for the sentiment analysis task	594
Project two – character-level language modeling in TensorFlow	600
Preprocessing the dataset	601
Building a character-level RNN model	607
Evaluation phase – generating new text passages	609
Understanding language with the Transformer model	613
Understanding the self-attention mechanism	614
A basic version of self-attention	614
Parameterizing the self-attention mechanism with query, key, and value weights	616
Multi-head attention and the Transformer block	617
Summary	618
Chapter 17: Generative Adversarial Networks for Synthesizing New Data	619
Introducing generative adversarial networks	620

Starting with autoencoders	620
Generative models for synthesizing new data	623
Generating new samples with GANs	624
Understanding the loss functions of the generator and discriminator networks in a GAN model	626
Implementing a GAN from scratch	628
Training GAN models on Google Colab	628
Implementing the generator and the discriminator networks	631
Defining the training dataset	636
Training the GAN model	638
Improving the quality of synthesized images using a convolutional and Wasserstein GAN	646
Transposed convolution	647
Batch normalization	648
Implementing the generator and discriminator	651
Dissimilarity measures between two distributions	657
Using EM distance in practice for GANs	661
Gradient penalty	662
Implementing WGAN-GP to train the DCGAN model	663
Mode collapse	667
Other GAN applications	669
Summary	670
Chapter 18: Reinforcement Learning for Decision Making in Complex Environments	671
Introduction – learning from experience	672
Understanding reinforcement learning	672
Defining the agent-environment interface of a reinforcement learning system	674
The theoretical foundations of RL	676
Markov decision processes	676
The mathematical formulation of Markov decision processes	677
Visualization of a Markov process	679
Episodic versus continuing tasks	680
RL terminology: return, policy, and value function	680
The return	680
Policy	682
Value function	683
Dynamic programming using the Bellman equation	685
Reinforcement learning algorithms	686
Dynamic programming	686
Policy evaluation – predicting the value function with dynamic programming	687
Improving the policy using the estimated value function	688

Policy iteration	688
Value iteration	689
Reinforcement learning with Monte Carlo	689
State-value function estimation using MC	690
Action-value function estimation using MC	690
Finding an optimal policy using MC control	691
Policy improvement – computing the greedy policy from the action-value function	691
Temporal difference learning	691
TD prediction	692
On-policy TD control (SARSA)	693
Off-policy TD control (Q-learning)	694
Implementing our first RL algorithm	694
Introducing the OpenAI Gym toolkit	695
Working with the existing environments in OpenAI Gym	695
A grid world example	697
Implementing the grid world environment in OpenAI Gym	698
Solving the grid world problem with Q-learning	705
Implementing the Q-learning algorithm	705
A glance at deep Q-learning	709
Training a DQN model according to the Q-learning algorithm	710
Implementing a deep Q-learning algorithm	712
Chapter and book summary	717
Other Books You May Enjoy	721
Index	725
