

# Table of Contents

<b>Preface</b>	<b>xi</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	5
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
<b>Introduction to the basic terminology and notations</b>	<b>8</b>
<b>A roadmap for building machine learning systems</b>	<b>11</b>
Preprocessing – getting data into shape	12
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 and packages from the Python Package Index	14
Using the Anaconda Python distribution and package manager	14
Packages for scientific computing, data science, and machine learning	15
<b>Summary</b>	<b>15</b>
<b>Chapter 2: Training Simple 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>
The formal definition of an artificial neuron	19
The perceptron learning rule	21

<b>Implementing a perceptron learning algorithm in Python</b>	<b>24</b>
An object-oriented perceptron API	24
Training a perceptron model on the Iris dataset	28
<b>Adaptive linear neurons and the convergence of learning</b>	<b>34</b>
Minimizing cost functions with gradient descent	35
Implementing Adaline in Python	38
Improving gradient descent through feature scaling	42
Large-scale machine learning and stochastic gradient descent	44
<b>Summary</b>	<b>50</b>
<b>Chapter 3: A Tour of Machine Learning Classifiers</b>	<b>51</b>
<b>Using scikit-learn</b>	<b>51</b>
<b>Choosing a classification algorithm</b>	<b>52</b>
<b>First steps with scikit-learn – training a perceptron</b>	<b>52</b>
<b>Modeling class probabilities via logistic regression</b>	<b>59</b>
Logistic regression intuition and conditional probabilities	59
Learning the weights of the logistic cost function	63
Converting an Adaline implementation into an algorithm for logistic regression	66
Training a logistic regression model with scikit-learn	71
Tackling overfitting via regularization	73
<b>Maximum margin classification with support vector machines</b>	<b>76</b>
Maximum margin intuition	77
Dealing with a nonlinearly separable case using slack variables	79
Alternative implementations in scikit-learn	81
<b>Solving nonlinear problems using a kernel SVM</b>	<b>82</b>
Kernel methods for linearly inseparable data	82
Using the kernel trick to find separating hyperplanes in high-dimensional space	84
<b>Decision tree learning</b>	<b>88</b>
Maximizing information gain – getting the most bang for your buck	90
Building a decision tree	95
Combining multiple decision trees via random forests	98
<b>K-nearest neighbors – a lazy learning algorithm</b>	<b>101</b>
<b>Summary</b>	<b>105</b>
<b>Chapter 4: Building Good Training Sets – Data Preprocessing</b>	<b>107</b>
<b>Dealing with missing data</b>	<b>107</b>
Identifying missing values in tabular data	108
Eliminating samples or features with missing values	109
Imputing missing values	110
Understanding the scikit-learn estimator API	111

<b>Handling categorical data</b>	<b>112</b>
Nominal and ordinal features	113
Creating an example dataset	113
Mapping ordinal features	113
Encoding class labels	114
Performing one-hot encoding on nominal features	116
<b>Partitioning a dataset into separate training and test sets</b>	<b>118</b>
<b>Bringing features onto the same scale</b>	<b>120</b>
<b>Selecting meaningful features</b>	<b>123</b>
L1 and L2 regularization as penalties against model complexity	124
A geometric interpretation of L2 regularization	124
Sparse solutions with L1 regularization	126
Sequential feature selection algorithms	130
<b>Assessing feature importance with random forests</b>	<b>136</b>
<b>Summary</b>	<b>139</b>
<b>Chapter 5: Compressing Data via Dimensionality Reduction</b>	<b>141</b>
<b>Unsupervised dimensionality reduction via principal component analysis</b>	<b>142</b>
The main steps behind principal component analysis	142
Extracting the principal components step by step	144
Total and explained variance	147
Feature transformation	148
Principal component analysis in scikit-learn	151
<b>Supervised data compression via linear discriminant analysis</b>	<b>155</b>
Principal component analysis versus linear discriminant analysis	155
The inner workings of linear discriminant analysis	156
Computing the scatter matrices	157
Selecting linear discriminants for the new feature subspace	160
Projecting samples onto the new feature space	162
LDA via scikit-learn	163
<b>Using kernel principal component analysis for nonlinear mappings</b>	<b>165</b>
Kernel functions and the kernel trick	166
Implementing a kernel principal component analysis in Python	172
Example 1 – separating half-moon shapes	173
Example 2 – separating concentric circles	176
Projecting new data points	179
Kernel principal component analysis in scikit-learn	183
<b>Summary</b>	<b>184</b>

---

<b>Chapter 6: Learning Best Practices for Model Evaluation and Hyperparameter Tuning</b>	<b>185</b>
<b>Streamlining workflows with pipelines</b>	<b>185</b>
Loading the Breast Cancer Wisconsin dataset	186
Combining transformers and estimators in a pipeline	187
<b>Using k-fold cross-validation to assess model performance</b>	<b>189</b>
The holdout method	190
K-fold cross-validation	191
<b>Debugging algorithms with learning and validation curves</b>	<b>195</b>
Diagnosing bias and variance problems with learning curves	196
Addressing over- and underfitting with validation curves	199
<b>Fine-tuning machine learning models via grid search</b>	<b>201</b>
Tuning hyperparameters via grid search	201
Algorithm selection with nested cross-validation	203
<b>Looking at different performance evaluation metrics</b>	<b>205</b>
Reading a confusion matrix	206
Optimizing the precision and recall of a classification model	207
Plotting a receiver operating characteristic	210
Scoring metrics for multiclass classification	213
<b>Dealing with class imbalance</b>	<b>214</b>
<b>Summary</b>	<b>216</b>
<b>Chapter 7: Combining Different Models for Ensemble Learning</b>	<b>219</b>
<b>Learning with ensembles</b>	<b>219</b>
<b>Combining classifiers via majority vote</b>	<b>224</b>
Implementing a simple majority vote classifier	224
Using the majority voting principle to make predictions	231
Evaluating and tuning the ensemble classifier	234
<b>Bagging – building an ensemble of classifiers from bootstrap samples</b>	<b>240</b>
Bagging in a nutshell	240
Applying bagging to classify samples in the Wine dataset	242
<b>Leveraging weak learners via adaptive boosting</b>	<b>246</b>
How boosting works	246
Applying AdaBoost using scikit-learn	251
<b>Summary</b>	<b>254</b>
<b>Chapter 8: Applying Machine Learning to Sentiment Analysis</b>	<b>255</b>
<b>Preparing the IMDb movie review data for text processing</b>	<b>256</b>
Obtaining the movie review dataset	256
Preprocessing the movie dataset into more convenient format	257

---

<b>Introducing the bag-of-words model</b>	<b>259</b>
Transforming words into feature vectors	259
Assessing word relevancy via term frequency-inverse document frequency	261
Cleaning text data	264
Processing documents into tokens	266
<b>Training a logistic regression model for document classification</b>	<b>268</b>
<b>Working with bigger data – online algorithms and out-of-core learning</b>	<b>270</b>
<b>Topic modeling with Latent Dirichlet Allocation</b>	<b>274</b>
Decomposing text documents with LDA	275
LDA with scikit-learn	275
<b>Summary</b>	<b>279</b>
<b>Chapter 9: Embedding a Machine Learning Model into a Web Application</b>	<b>281</b>
<b>Serializing fitted scikit-learn estimators</b>	<b>282</b>
<b>Setting up an SQLite database for data storage</b>	<b>285</b>
<b>Developing a web application with Flask</b>	<b>287</b>
Our first Flask web application	288
Form validation and rendering	290
Setting up the directory structure	291
Implementing a macro using the Jinja2 templating engine	292
Adding style via CSS	293
Creating the result page	294
<b>Turning the movie review classifier into a web application</b>	<b>294</b>
Files and folders – looking at the directory tree	296
Implementing the main application as app.py	298
Setting up the review form	300
Creating a results page template	302
<b>Deploying the web application to a public server</b>	<b>304</b>
Creating a PythonAnywhere account	304
Uploading the movie classifier application	305
Updating the movie classifier	306
<b>Summary</b>	<b>308</b>
<b>Chapter 10: Predicting Continuous Target Variables with Regression Analysis</b>	<b>309</b>
<b>Introducing linear regression</b>	<b>310</b>
Simple linear regression	310
Multiple linear regression	311
<b>Exploring the Housing dataset</b>	<b>312</b>
Loading the Housing dataset into a data frame	313

---

Visualizing the important characteristics of a dataset	314
Looking at relationships using a correlation matrix	316
<b>Implementing an ordinary least squares linear regression model</b>	<b>319</b>
Solving regression for regression parameters with gradient descent	319
Estimating coefficient of a regression model via scikit-learn	324
<b>Fitting a robust regression model using RANSAC</b>	<b>325</b>
<b>Evaluating the performance of linear regression models</b>	<b>328</b>
<b>Using regularized methods for regression</b>	<b>332</b>
<b>Turning a linear regression model into a curve – polynomial regression</b>	<b>334</b>
Adding polynomial terms using scikit-learn	334
Modeling nonlinear relationships in the Housing dataset	336
<b>Dealing with nonlinear relationships using random forests</b>	<b>339</b>
Decision tree regression	340
Random forest regression	342
<b>Summary</b>	<b>345</b>
<b>Chapter 11: Working with Unlabeled Data – Clustering Analysis</b>	<b>347</b>
<b>Grouping objects by similarity using k-means</b>	<b>348</b>
K-means clustering using scikit-learn	348
A smarter way of placing the initial cluster centroids using k-means++	353
Hard versus soft clustering	354
Using the elbow method to find the optimal number of clusters	357
Quantifying the quality of clustering via silhouette plots	358
<b>Organizing clusters as a hierarchical tree</b>	<b>363</b>
Grouping clusters in bottom-up fashion	364
Performing hierarchical clustering on a distance matrix	365
Attaching dendrograms to a heat map	369
Applying agglomerative clustering via scikit-learn	371
<b>Locating regions of high density via DBSCAN</b>	<b>372</b>
<b>Summary</b>	<b>378</b>
<b>Chapter 12: Implementing a Multilayer Artificial Neural Network from Scratch</b>	<b>379</b>
<b>Modeling complex functions with artificial neural networks</b>	<b>380</b>
Single-layer neural network recap	382
Introducing the multilayer neural network architecture	384
Activating a neural network via forward propagation	387
<b>Classifying handwritten digits</b>	<b>389</b>
Obtaining the MNIST dataset	390
Implementing a multilayer perceptron	396

---

<b>Training an artificial neural network</b>	<b>407</b>
Computing the logistic cost function	408
Developing your intuition for backpropagation	411
Training neural networks via backpropagation	412
<b>About the convergence in neural networks</b>	<b>417</b>
<b>A few last words about the neural network implementation</b>	<b>418</b>
<b>Summary</b>	<b>419</b>
<b>Chapter 13: Parallelizing Neural Network Training with TensorFlow</b>	<b>421</b>
<hr/>	
<b>TensorFlow and training performance</b>	<b>421</b>
What is TensorFlow?	423
How we will learn TensorFlow	424
First steps with TensorFlow	424
Working with array structures	427
Developing a simple model with the low-level TensorFlow API	428
<b>Training neural networks efficiently with high-level TensorFlow APIs</b>	<b>433</b>
Building multilayer neural networks using TensorFlow's Layers API	434
Developing a multilayer neural network with Keras	438
<b>Choosing activation functions for multilayer networks</b>	<b>443</b>
Logistic function recap	444
Estimating class probabilities in multiclass classification via the softmax function	446
Broadening the output spectrum using a hyperbolic tangent	447
Rectified linear unit activation	449
<b>Summary</b>	<b>451</b>
<b>Chapter 14: Going Deeper – The Mechanics of TensorFlow</b>	<b>453</b>
<hr/>	
<b>Key features of TensorFlow</b>	<b>454</b>
<b>TensorFlow ranks and tensors</b>	<b>454</b>
<b>How to get the rank and shape of a tensor</b>	<b>455</b>
<b>Understanding TensorFlow's computation graphs</b>	<b>456</b>
<b>Placeholders in TensorFlow</b>	<b>459</b>
Defining placeholders	459
Feeding placeholders with data	460
Defining placeholders for data arrays with varying batchsizes	461
<b>Variables in TensorFlow</b>	<b>462</b>
Defining variables	463
Initializing variables	465
Variable scope	466
Reusing variables	468

<b>Building a regression model</b>	<b>471</b>
<b>Executing objects in a TensorFlow graph using their names</b>	<b>475</b>
<b>Saving and restoring a model in TensorFlow</b>	<b>476</b>
<b>Transforming Tensors as multidimensional data arrays</b>	<b>479</b>
<b>Utilizing control flow mechanics in building graphs</b>	<b>483</b>
<b>Visualizing the graph with TensorBoard</b>	<b>487</b>
Extending your TensorBoard experience	490
<b>Summary</b>	<b>491</b>
<b>Chapter 15: Classifying Images with Deep Convolutional Neural Networks</b>	<b>493</b>
<hr/>	
<b>Building blocks of convolutional neural networks</b>	<b>494</b>
Understanding CNNs and learning feature hierarchies	494
Performing discrete convolutions	496
Performing a discrete convolution in one dimension	496
The effect of zero-padding in a convolution	499
Determining the size of the convolution output	501
Performing a discrete convolution in 2D	502
Subsampling	506
<b>Putting everything together to build a CNN</b>	<b>508</b>
Working with multiple input or color channels	508
Regularizing a neural network with dropout	512
<b>Implementing a deep convolutional neural network using TensorFlow</b>	<b>514</b>
The multilayer CNN architecture	514
Loading and preprocessing the data	516
Implementing a CNN in the TensorFlow low-level API	517
Implementing a CNN in the TensorFlow Layers API	530
<b>Summary</b>	<b>536</b>
<b>Chapter 16: Modeling Sequential Data Using Recurrent Neural Networks</b>	<b>537</b>
<hr/>	
<b>Introducing sequential data</b>	<b>538</b>
Modeling sequential data – order matters	538
Representing sequences	539
The different categories of sequence modeling	540
<b>RNNs for modeling sequences</b>	<b>541</b>
Understanding the structure and flow of an RNN	541
Computing activations in an RNN	543
The challenges of learning long-range interactions	546
LSTM units	548

---

<b>Implementing a multilayer RNN for sequence modeling in TensorFlow</b>	<b>550</b>
<b>Project one – performing sentiment analysis of IMDb movie reviews using multilayer RNNs</b>	<b>551</b>
Preparing the data	552
Embedding	556
Building an RNN model	558
The SentimentRNN class constructor	559
The build method	560
Step 1 – defining multilayer RNN cells	562
Step 2 – defining the initial states for the RNN cells	562
Step 3 – creating the RNN using the RNN cells and their states	563
The train method	563
The predict method	565
Instantiating the SentimentRNN class	565
Training and optimizing the sentiment analysis RNN model	566
<b>Project two – implementing an RNN for character-level language modeling in TensorFlow</b>	<b>567</b>
Preparing the data	568
Building a character-level RNN model	572
The constructor	573
The build method	574
The train method	576
The sample method	578
Creating and training the CharRNN Model	579
The CharRNN model in the sampling mode	580
<b>Chapter and book summary</b>	<b>580</b>
<b>Index</b>	<b>583</b>

---