

# Inhaltsverzeichnis

<b>Vorwort</b>	13
<b>Über den Autor</b>	14
<b>Danksagungen</b>	15
<b>Über die Korrektoren</b>	16
<b>Einleitung</b>	18
<b>1 Wie Computer aus Daten lernen können</b>	23
1.1 Intelligente Maschinen, die Daten in Wissen verwandeln	23
1.2 Die drei Arten des Machine Learnings	24
1.2.1 Mit überwachtem Lernen Vorhersagen treffen	24
1.2.2 Interaktive Aufgaben durch verstärkendes Lernen lösen	27
1.2.3 Durch unüberwachtes Lernen verborgene Strukturen erkennen	28
1.3 Grundlegende Terminologie und Notation	29
1.4 Entwicklung eines Systems für das Machine Learning	31
1.4.1 Vorverarbeitung: Daten in Form bringen	32
1.4.2 Trainieren und Auswählen eines Vorhersagemodells	32
1.4.3 Bewertung von Modellen und Vorhersage anhand unbekannter Dateninstanzen	33
1.5 Machine Learning mit Python	33
1.5.1 Python-Pakete installieren	34
1.6 Zusammenfassung	35
<b>2 Lernalgorithmen für die Klassifizierung trainieren</b>	37
2.1 Künstliche Neuronen: Ein kurzer Blick auf die Anfänge des Machine Learnings	37
2.2 Implementierung eines Perzeptron-Lernalgorithmus in Python	43
2.2.1 Trainieren eines Perzeptron-Modells auf die Iris-Datensammlung	46
2.3 Adaptive lineare Neuronen und die Konvergenz des Lernens	51
2.3.1 Straffunktionen mit dem Gradientenabstiegsverfahren minimieren	52

2.3.2	Implementierung eines adaptiven linearen Neurons in Python .....	54
2.3.3	Large-scale Machine Learning und stochastisches Gradientenabstiegsverfahren .....	59
2.4	Zusammenfassung .....	65
3	<b>Machine-Learning-Klassifizierer mit scikit-learn verwenden .....</b>	<b>67</b>
3.1	Auswahl eines Klassifizierungsalgorithmus .....	67
3.2	Erste Schritte mit scikit-learn .....	68
3.2.1	Trainieren eines Perzeptrons mit scikit-learn .....	68
3.3	Klassenwahrscheinlichkeiten durch logistische Regression modellieren .....	73
3.3.1	Logistische Regression und bedingte Wahrscheinlichkeiten .....	73
3.3.2	Gewichtungen der logistischen Straffunktion ermitteln ...	77
3.3.3	Trainieren eines logistischen Regressionsmodells mit scikit-learn .....	78
3.3.4	Überanpassung durch Regularisierung verhindern .....	81
3.4	Maximum-Margin-Klassifizierung mit Support Vector Machines. .	84
3.4.1	Maximierung des Randbereichs .....	85
3.4.2	Handhabung des nicht linear trennbaren Falls mit Schlupfvariablen .....	86
3.4.3	Alternative Implementierungen in scikit-learn .....	88
3.5	Nichtlineare Aufgaben mit einer Kernel-SVM lösen .....	89
3.5.1	Mit dem Kernel-Trick Hyperebenen in höherdimensionalen Räumen finden. ....	91
3.6	Lernen mit Entscheidungsbäumen .....	94
3.6.1	Maximierung des Informationsgewinns: Daten ausreizen ...	95
3.6.2	Konstruktion eines Entscheidungsbaums .....	100
3.6.3	Schwache Klassifizierer mit Random Forests zu besseren kombinieren .....	102
3.7	k-Nearest-Neighbor: Ein Lazy-Learning-Algorithmus .....	104
3.8	Zusammenfassung .....	107
4	<b>Gut geeignete Trainingsdatenmengen: Datenvorverarbeitung .....</b>	<b>109</b>
4.1	Umgang mit fehlenden Daten .....	109
4.1.1	Exemplare oder Merkmale mit fehlenden Werten entfernen .....	111

4.1.2	Fehlende Werte ergänzen .....	112
4.1.3	Die Schätzer-API von scikit-learn .....	112
4.2	Handhabung kategorialer Daten .....	113
4.2.1	Zuweisung von ordinalen Merkmalen .....	114
4.2.2	Kodierung der Klassenbezeichnungen .....	115
4.2.3	One-hot-Kodierung der nominalen Merkmale .....	116
4.3	Aufteilung einer Datensammlung in Trainings- und Testdaten ...	118
4.4	Anpassung der Merkmale .....	120
4.5	Auswahl aussagekräftiger Merkmale .....	122
4.5.1	Dünnbesetzte Lösungen durch L1-Regularisierung .....	122
4.5.2	Algorithmen zur sequenziellen Auswahl von Merkmalen .....	128
4.6	Beurteilung der Bedeutung von Merkmalen mit Random Forests .....	134
4.7	Zusammenfassung .....	136
5	<b>Datenkomprimierung durch Dimensionsreduktion .....</b>	<b>137</b>
5.1	Unüberwachte Dimensionsreduktion durch Hauptkomponenten- analyse .....	137
5.1.1	Totale Varianz und Varianzaufklärung .....	139
5.1.2	Merkmalstransformation .....	143
5.1.3	Hauptkomponentenanalyse mit scikit-learn .....	145
5.2	Überwachte Datenkomprimierung durch lineare Diskriminanz- analyse .....	148
5.2.1	Berechnung der Streumatrizen .....	150
5.2.2	Auswahl linearer Diskriminanten für den neuen Merkmals- unterraum .....	152
5.2.3	Projektion in den neuen Merkmalsraum .....	155
5.2.4	LDA mit scikit-learn .....	156
5.3	Kernel-Hauptkomponentenanalyse für nichtlineare Zuordnungen verwenden .....	157
5.3.1	Kernel-Funktionen und der Kernel-Trick .....	158
5.3.2	Implementierung einer Kernel-Hauptkomponentenanalyse in Python .....	162
5.3.3	Projizieren neuer Datenpunkte .....	169
5.3.4	Kernel-Hauptkomponentenanalyse mit scikit-learn .....	173
5.4	Zusammenfassung .....	174

<b>6</b>	<b>Bewährte Verfahren zur Modellbewertung und Hyperparameter-Abstimmung</b> . . . . .	<b>175</b>
6.1	Arbeitsabläufe mit Pipelines optimieren . . . . .	175
6.1.1	Die Wisconsin-Brustkrebs-Datensammlung . . . . .	175
6.1.2	Transformer und Schätzer in einer Pipeline kombinieren . . .	176
6.2	Beurteilung des Modells durch k-fache Kreuzvalidierung . . . . .	178
6.2.1	2-fache Kreuzvalidierung . . . . .	178
6.2.2	k-fache Kreuzvalidierung . . . . .	180
6.3	Algorithmen mit Lern- und Validierungskurven debuggen. . . . .	184
6.3.1	Probleme mit Bias und Varianz anhand von Lernkurven erkennen . . . . .	184
6.3.2	Überanpassung und Unteranpassung anhand von Validierungskurven erkennen . . . . .	187
6.4	Feinabstimmung eines Lernmodells durch Rastersuche . . . . .	189
6.4.1	Hyperparameterabstimmung durch Rastersuche . . . . .	190
6.4.2	Algorithmenauswahl durch verschachtelte Kreuzvalidierung. . . . .	191
6.5	Verschiedene Kriterien zur Leistungsbewertung . . . . .	193
6.5.1	Interpretation einer Wahrheitsmatrix. . . . .	193
6.5.2	Optimierung der Genauigkeit und der Trefferquote eines Klassifizierungsmodells . . . . .	195
6.5.3	Receiver-Operating-Characteristic-Diagramme . . . . .	197
6.5.4	Bewertungskriterien für Mehrfachklassifizierungen. . . . .	200
6.6	Zusammenfassung . . . . .	201
<b>7</b>	<b>Kombination verschiedener Modelle für das Ensemble Learning</b> . . .	<b>203</b>
7.1	Ensemble Learning . . . . .	203
7.2	Implementierung eines einfachen Mehrheitsentscheidungs-Klassifizierers . . . . .	207
7.2.1	Kombination mehrerer Klassifizierungsalgorithmen per Mehrheitsentscheidung . . . . .	213
7.3	Bewertung und Abstimmung des Klassifizierer-Ensembles . . . . .	216
7.4	Bagging: Klassifizierer-Ensembles anhand von Bootstrap-Stichproben entwickeln . . . . .	222
7.5	Schwache Klassifizierer durch adaptives Boosting verbessern. . . . .	227
7.6	Zusammenfassung . . . . .	234
<b>8</b>	<b>Machine Learning zur Analyse von Stimmungslagen nutzen</b> . . . . .	<b>237</b>
8.1	Die IMDb-Filmdatenbank . . . . .	237

175	8.2	Das Bag-of-words-Modell . . . . .	239
175	8.2.1	Wörter in Merkmalsvektoren umwandeln . . . . .	240
175	8.2.2	Beurteilung der Wortrelevanz durch das Tf-idf-Maß . . . . .	242
176	8.2.3	Textdaten bereinigen . . . . .	244
178	8.2.4	Dokumente in Token zerlegen . . . . .	246
178	8.3	Ein logistisches Regressionsmodell für die Dokumentklassi- fizierung trainieren . . . . .	248
180	8.4	Verarbeitung großer Datenmengen: Online-Algorithmen und Out-of-Core Learning . . . . .	250
184	8.5	Zusammenfassung . . . . .	254
184	9	<b>Einbettung eines Machine-Learning-Modells in eine Webanwendung.</b> . . . . .	255
187	9.1	Serialisierung angepasster Schätzer mit scikit-learn . . . . .	255
189	9.2	Einrichtung einer SQLite-Datenbank zum Speichern von Daten . . .	258
190	9.3	Entwicklung einer Webanwendung mit Flask . . . . .	261
191	9.3.1	Die erste Webanwendung mit Flask. . . . .	261
193	9.3.2	Formularvalidierung und -ausgabe. . . . .	263
193	9.4	Der Filmbewertungsklassifizierer als Webanwendung . . . . .	267
195	9.5	Einrichtung der Webanwendung auf einem öffentlich zugänglichen Webserver. . . . .	275
197	9.5.1	Updaten des Filmbewertungsklassifizierers . . . . .	276
200	9.6	Zusammenfassung . . . . .	278
201	10	<b>Vorhersage stetiger Zielvariablen durch Regressionsanalyse . . . . .</b>	279
203	10.1	Ein einfaches lineares Regressionsmodell. . . . .	279
203	10.2	Die Lebensbedingungen-Datensammlung . . . . .	280
207	10.2.1	Visualisierung der wichtigen Eigenschaften einer Datenmenge . . . . .	282
213	10.3	Implementierung eines linearen Regressionsmodells mit der Methode der kleinsten Quadrate . . . . .	286
216	10.3.1	Berechnung der Regressionsparameter mit dem Gradientenabstiegsverfahren. . . . .	287
222	10.3.2	Abschätzung der Koeffizienten eines Regressionsmodells mit scikit-learn . . . . .	291
273	10.4	Anpassung eines robusten Regressionsmodells mit dem RANSAC-Algorithmus . . . . .	293
37	10.5	Bewertung der Leistung linearer Regressionsmodelle. . . . .	295
37	10.6	Regularisierungsverfahren für die Regression einsetzen. . . . .	299

10.7	Polynomiale Regression: Umwandeln einer linearen Regression in eine Kurve .....	300
10.7.1	Modellierung nichtlinearer Zusammenhänge in der Lebensbedingungen-Datensammlung .....	302
10.7.2	Handhabung nichtlinearer Beziehungen mit Random Forests .....	306
10.8	Zusammenfassung .....	310
11	<b>Verwendung nicht gekennzeichneten Daten: Clusteranalyse.</b> .....	313
11.1	Gruppierung von Objekten nach Ähnlichkeit mit dem k-Means-Algorithmus .....	313
11.1.1	Der k-Means++-Algorithmus .....	317
11.1.2	»Harte« und »weiche« Clustering-Algorithmen .....	319
11.1.3	Die optimale Anzahl der Cluster mit dem Ellenbogenkriterium ermitteln .....	322
11.1.4	Quantifizierung der Clustering-Güte mit Silhouetten- diagrammen .....	323
11.2	Cluster als hierarchischen Baum organisieren .....	328
11.2.1	Hierarchisches Clustering einer Distanzmatrix .....	329
11.2.2	Dendrogramme und Heatmaps verknüpfen .....	333
11.2.3	Agglomeratives Clustering mit scikit-learn .....	335
11.3	Bereiche hoher Dichte mit DBSCAN ermitteln .....	335
11.4	Zusammenfassung .....	340
12	<b>Künstliche neuronale Netze für die Bilderkennung trainieren</b> .....	343
12.1	Modellierung komplexer Funktionen mit künstlichen neuronalen Netzen .....	343
12.1.1	Einschichtige neuronale Netze .....	344
12.1.2	Mehrschichtige neuronale Netzarchitektur .....	346
12.1.3	Aktivierung eines neuronalen Netzes durch Vorwärts- propagation .....	349
12.2	Klassifizierung handgeschriebener Ziffern .....	351
12.2.1	Die MNIST-Datensammlung .....	352
12.2.2	Implementierung eines mehrschichtigen Perzeptrons .....	357
12.3	Trainieren eines künstlichen neuronalen Netzes .....	365
12.3.1	Berechnung der logistischen Straffunktion .....	365
12.3.2	Trainieren neuronaler Netze durch Backpropagation .....	368
12.4	Ein Gespür für die Backpropagation entwickeln .....	371
.5	Debugging neuronaler Netze durch Gradientenprüfung .....	373

12.6	Konvergenz in neuronalen Netzen. ....	379
12.7	Weitere neuronale Netzarchitekturen ....	380
12.7.1	Konvolutionale neuronale Netze. ....	380
12.7.2	Rekurrente neuronale Netze ....	382
12.8	Abschließende Bemerkungen zur Implementierung neuronaler Netze. ....	383
12.9	Zusammenfassung ....	384
13	<b>Parallelisierung des Trainings neuronaler Netze mit Theano</b> .....	385
13.1	Erstellen, Kompilieren und Ausführen von Ausdrücken mit Theano ....	385
13.1.1	Was genau ist Theano? ....	387
13.1.2	Erste Schritte mit Theano ....	388
13.1.3	Theano konfigurieren ....	389
13.1.4	Mit Array-Strukturen arbeiten. ....	391
13.1.5	Zusammengefasst: Lineare Regression als Beispiel .....	394
13.2	Auswahl der Aktivierungsfunktionen neuronaler Feedforward-Netze. ....	398
13.2.1	Die logistische Funktion kurz zusammengefasst ....	398
13.2.2	Wahrscheinlichkeiten bei der Mehrfachklassifizierung mit der softmax-Funktion abschätzen ....	400
13.2.3	Verbreiterung des Ausgabespektrums mittels Tangens hyperbolicus ....	402
13.3	Effizientes Training neuronaler Netze mit Keras ....	404
13.4	Zusammenfassung ....	410
	<b>Stichwortverzeichnis</b> .....	413