O'REILLY®

Einführung in Machine Learning mit Python

PRAXISWISSEN DATA SCIENCE

Andreas C. Müller & Sarah Guido
Übersetzung von Kristian Rother

dessen Aufgabe es ist, eingehende Nachrichten in den Spam-Ordner zu verschieben. Sie könnten eine schwarze Liste von Wörtern erstellen, die zum Einstufen einer E-Mail als Spam führen. Dies ist ein Beispiel für ein von Experten entwickeltes Regelsystem als »intelligente« Anwendung. Bei manchen Anwendungen ist das Festlegen von Regeln von Hand praktikabel, besonders wenn Menschen ein gutes Verständnis für den zu modellierenden Prozess besitzen. Allerdings hat das Verwenden von Hand erstellter Regeln zwei große Nachteile:

- Die Entscheidungslogik ist für eine bestimmtes Fachgebiet und eine Aufgabe spezifisch. Selbst eine kleine Veränderung der Aufgabe kann dazu führen, dass das gesamte System neu geschrieben werden muss.
- Das Entwickeln von Regeln erfordert ein tiefes Verständnis davon, wie ein menschlicher Experte diese Entscheidung treffen würde.

Ein Beispiel, bei dem der händische Ansatz fehlschlägt, ist das Erkennen von Gesichtern in Bildern. Heutzutage kann jedes Smartphone ein Gesicht in einem Bild erkennen. Allerdings war Gesichtserkennung bis 2001 ein ungelöstes Problem. Das Hauptproblem dabei war, dass ein Computer die Pixel (aus denen ein Computerbild besteht) im Vergleich zu Menschen auf sehr unterschiedliche Weise »wahrnimmt«. Diese unterschiedliche Repräsentation macht es einem Menschen praktisch unmöglich, ein gutes Regelwerk zu entwickeln, mit dem sich umschreiben lässt, was ein Gesicht in einem digitalen Bild ausmacht. Mit maschinellem Lernen ist es dagegen ausreichend, einem Programm eine große Sammlung von Bildern mit Gesichtern vorzulegen, um die Charakteristiken zum Erkennen von Gesichtern auszuarbeiten.

Welche Probleme kann Machine Learning lösen?

Die erfolgreichsten Arten maschineller Lernalgorithmen sind diejenigen, die den Entscheidungsprozess durch Verallgemeinerung aus bekannten Beispielen automatisieren. In diesem als *überwachtes Lernen* bekannten Szenario beliefert der Nutzer einen Algorithmus mit Paaren von Eingabewerten und erwünschten Ausgabewerten, und der Algorithmus findet heraus, wie sich die gewünschte Ausgabe erstellen lässt. Damit ist der Algorithmus ohne menschliche Hilfe in der Lage, aus zuvor unbekannten Eingaben eine Ausgabe zu berechnen. Bei unserem Beispiel der Spam-Klassifizierung würde der Nutzer dem Algorithmus eine große Anzahl E-Mails (die Eingaben) sowie die Angabe, welche dieser E-Mails Spam sind (die erwünschte Ausgabe), zur Verfügung stellen. Für eine neue E-Mail kann der Algorithmus dann vorhersagen, ob die neue E-Mail Spam ist.

Maschinelle Lernalgorithmen, die aus Eingabe-Ausgabe-Paaren lernen, bezeichnet man als überwachte Lernalgorithmen, weil ein »Lehrer« den Algorithmus in Form der erwünschten Ausgaben für jedes Lernbeispiel anleitet. Obwohl das Erstellen eines Datensatzes geeigneter Ein- und Ausgaben oft mühevolle Handarbeit bedeu-

tet, sind überwachte Lernalgorithmen gut verständlich, und ihre Leistung ist leicht messbar. Wenn Ihre Anwendung sich als überwachte Lernaufgabe formulieren lässt und Sie einen Datensatz mit den gewünschten Ergebnissen erstellen können, lässt sich Ihre Fragestellung vermutlich durch Machine Learning beantworten.

Beispiele für überwachtes maschinelles Lernen sind:

Auf einem Briefumschlag die Postleitzahl aus handschriftlichen Ziffern zu erkennen
: Hier besteht die Eingabe aus der eingescannten Handschrift, und die gewünschte Ausgabe sind die Ziffern der Postleitzahl. Um einen Datensatz zum Erstellen eines maschinellen Lernmodells zu erzeugen, müssen Sie zuerst viele Umschläge sammeln. Dann können Sie die Postleitzahlen selbst lesen und die Ziffern als gewünschtes Ergebnis abspeichern.

Anhand eines medizinischen Bildes entscheiden, ob ein Tumor gutartig ist
: Hierbei ist die Eingabe das Bild, und die Ausgabe, ob der Tumor gutartig ist. Um einen Datensatz zum Erstellen eines Modells aufzubauen, benötigen Sie eine Datenbank mit medizinischen Bildern. Sie benötigen auch eine Expertenmeinung, es muss sich also ein Arzt sämtliche Bilder ansehen und entscheiden, welche Tumore gutartig sind und welche nicht. Es ist sogar möglich, dass zur Entscheidung, ob der Tumor im Bild krebsartig ist oder nicht, zusätzlich zum Bild weitere Diagnosen nötig sind.

Erkennen betrügerischer Aktivitäten bei Kreditkartentransaktionen
: Hierbei sind die Eingaben Aufzeichnungen von Kreditkartentransaktionen, und die Ausgabe ist, ob diese vermutlich betrügerisch sind oder nicht. Wenn Ihre Organisation Kreditkarten ausgibt, müssen Sie sämtliche Transaktionen aufzeichnen und ob ein Nutzer betrügerische Transaktionen meldet.

Bei diesen Beispielen sollte man hervorheben, dass das Sammeln der Daten bei diesen Aufgaben grundsätzlich unterschiedlich ist, auch wenn die Ein- und Ausgabedaten sehr klar wirken. Das Lesen von Umschlägen ist zwar mühevoll, aber auch unkompliziert. Medizinische Bilder und Diagnosen zu sammeln, erfordert dagegen nicht nur teure Geräte, sondern auch seltenes und teures Expertenwissen, von den ethischen und datenschutztechnischen Bedenken einmal abgesehen. Beim Erkennen von Kreditkartenbetrug ist das Sammeln der Daten deutlich einfacher. Ihre Kunden werden Sie mit den nötigen Ausgabedaten versorgen. Um die Ein-/Ausgabedaten für betrügerische und ehrliche Aktivitäten zu erhalten, müssen Sie nichts anderes tun, als zu warten.

Unüberwachte Algorithmen
: sind eine weitere Art in diesem Buch behandelter Algorithmen. Beim unüberwachten Lernen sind nur die Eingabedaten bekannt, und dem Algorithmus werden keine bekannten Ausgabedaten zur Verfügung gestellt. Es sind viele erfolgreiche Anwendungen dieser Methoden bekannt, sie sind aber in der Regel schwieriger zu verstehen und auszuwerten.

Beispiele für unüberwachtes Lernen sind:

Themen in einer Serie von Blogeinträgen erkennen
: Sie haben eine große Menge Textdaten und möchten diese zusammenfassen und die darin vorherrschenden Themen herausfinden. Wenn Sie nicht im Voraus wissen, welches diese Themen sind oder wie viele Themen es gibt, so gibt es keine bekannte Ausgabe.

Kunden in Gruppen mit ähnlichen Vorlieben einteilen
: Mit einem Satz Kundendaten könnten Sie ähnliche Kunden erkennen und herausfinden, ob es Kundengruppen mit ähnlichen Vorlieben gibt. Bei einem Online-Geschäft könnten diese »Eltern«, »Bücherwürmer« oder »Spieler« sein. Weil diese Gruppen nicht im Voraus bekannt sind, oft nicht einmal deren Anzahl, haben Sie keine bekannte Ausgabe.

Erkennung außergewöhnlicher Zugriffsmuster auf eine Webseite
: Um unrechtmäßige Nutzung oder Fehler zu erkennen, ist es oft hilfreich, Zugriffe zu finden, die sich von der Durchschnittsnutzung abheben. Jedes außergewöhnliche Muster kann sehr unterschiedlich sein, und Sie haben womöglich keinerlei Aufzeichnungen über abnorme Nutzung. Weil Sie in diesem Fall die Zugriffe einer Webseite beobachten und nicht wissen, was normale Nutzung ist und was nicht, handelt es sich hier um eine unüberwachte Fragestellung.

Sowohl bei überwachten als auch bei unüberwachten Lernaufgaben ist es wichtig, eine für den Computer verständliche Repräsentation Ihrer Eingabedaten zu haben. Oft hilft es, sich die Daten als Tabelle vorzustellen. Jeder zu untersuchende Datenpunkt (jede E-Mail, jeder Kunde, jede Transaktion) ist eine Zeile, und jede Eigenschaft, die diesen Datenpunkt beschreibt (z. B. das Alter eines Kunden oder die Menge oder der Ort der Transaktion), ist eine Spalte. Sie können Nutzer durch Alter, Geschlecht, das Datum der Registrierung und wie oft sie in Ihrem Online-Geschäft eingekauft haben, charakterisieren. Das Bild eines Tumors lässt sich durch die Graustufenwerte jedes Pixels beschreiben oder aber durch Größe, Gestalt und Farbe des Tumors.

Jede Entität oder Zeile bezeichnet man beim maschinellen Lernen als *Datenpunkt* (oder Probe), die Spalten – also die Eigenschaften, die diese Entitäten beschreiben werden – nennt man *Merkmale*.

Im weiteren Verlauf dieses Buches werden wir uns genauer mit dem Aufbau einer guten Datenrepräsentation beschäftigen, was man als *Extrahieren von Merkmalen* oder *Merkmalsgenerierung* bezeichnet. Sie sollten auf jeden Fall bedenken, dass kein maschinelles Lernverfahren ohne entsprechende Information zu Vorhersagen in der Lage ist. Wenn zum Beispiel das einzige bekannte Merkmal eines Patienten der Nachname ist, wird kein Algorithmus in der Lage sein, das Geschlecht vorherzusagen. Diese Information ist schlicht nicht in Ihren Daten enthalten. Wenn Sie

ein weiteres Merkmal mit dem Vornamen des Patienten hinzufügen, haben Sie mehr Glück, da sich das Geschlecht häufig aus dem Vornamen vorhersagen lässt.

Ihre Aufgabe und Ihre Daten kennen

Der möglicherweise wichtigste Teil beim maschinellen Lernen ist, dass Sie Ihre Daten verstehen und wie diese mit der zu lösenden Aufgabe zusammenhängen. Es ist nicht sinnvoll, zufällig einen Algorithmus auszuwählen und Ihre Daten hineinzuwerfen. Bevor Sie ein Modell konstruieren können, ist es notwendig, zu verstehen, was in Ihrem Datensatz vorgeht. Jeder Algorithmus unterscheidet sich darin, bei welchen Daten und welchen Aufgabenstellungen er am besten funktioniert. Wenn Sie eine Aufgabe mit maschinellem Lernen bearbeiten, sollten Sie folgende Fragen beantworten oder zumindest im Hinterkopf behalten:

- Welche Fragen versuche ich zu beantworten? Glaube ich, dass die erhobenen Daten diese Frage beantworten können?
- Wie lässt sich meine Frage am besten als maschinelle Lernaufgabe ausdrücken?
- Habe ich genug Daten gesammelt, um die zu beantwortende Fragestellung zu repräsentieren?
- Welche Merkmale der Daten habe ich extrahiert? Sind diese zu den richtigen Vorhersagen in der Lage?
- Wie messe ich, ob meine Anwendung erfolgreich ist?
- Wie interagiert mein maschinelles Lernmodell mit anderen Teilen meiner Forschungsarbeit oder meines Produkts?

In einem breiteren Kontext sind die Algorithmen und Methoden für maschinelles Lernen nur Teil eines größeren Prozesses zum Lösen einer bestimmten Aufgabe. Es ist sinnvoll, das große Ganze stets im Blick zu behalten. Viele Leute investieren eine Menge Zeit in das Entwickeln eines komplexen maschinellen Lernsystems, nur um hinterher herauszufinden, dass sie das falsche Problem gelöst haben.

Wenn man sich eingehend mit den technischen Aspekten maschinellen Lernens beschäftigt (wie wir es in diesem Buch tun werden), ist es leicht, die endgültigen Ziele aus den Augen zu verlieren. Wir werden die hier gestellten Fragen nicht im Detail diskutieren, halten Sie aber dazu an, sämtliche explizit oder implizit getroffenen Annahmen beim Aufbau maschineller Lernmodelle zu berücksichtigen.

Warum Python?

Python ist für viele Anwendungen aus dem Bereich Data Science die lingua franca geworden. Python kombiniert die Ausdruckskraft allgemein nutzbarer Programmiersprachen mit der einfachen Benutzbarkeit einer domänenspezifischen Skript-

sprache wie MATLAB oder R. Für Python gibt es Bibliotheken zum Laden von Daten, Visualisieren, Berechnen von Statistiken, Sprachverarbeitung, Bildverarbeitung usw. Dies gibt Data Scientists einen sehr umfangreichen Werkzeugkasten mit Funktionalität für allgemeine und besondere Einsatzgebiete. Einer der Hauptvorteile von Python ist die Möglichkeit, direkt mit dem Code zu interagieren, sei es in einer Konsole oder einer anderen Umgebung wie dem Jupyter Notebook, das wir uns in Kürze ansehen werden. Machine Learning und Datenanalyse sind von Grund auf iterative Prozesse, bei denen die Daten die Analyse bestimmen. Es ist entscheidend, diese Prozesse mit Werkzeugen zu unterstützen, die schnelle Iterationen und leichte Benutzbarkeit ermöglichen.

Als allgemein einsetzbare Programmiersprache lassen sich mit Python auch komplexe grafische Benutzeroberflächen (GUIs) und Webdienste entwickeln und in bestehende Systeme integrieren.

scikit-learn

`scikit-learn` ist ein Open Source-Projekt, Sie dürfen es also kostenlos verwenden und verbreiten. Jeder kommt leicht an die Quelltexte heran und kann sehen, was hinter den Kulissen passiert. Das `scikit-learn`-Projekt wird kontinuierlich weiterentwickelt und verbessert und besitzt eine große Nutzergemeinde. Es enthält eine Anzahl hochentwickelter maschineller Lernalgorithmen und eine umfangreiche Dokumentation (*http://scikit-learn.org/stable/documentation*) zu jedem Algorithmus. `scikit-learn` ist sehr beliebt, und die Nummer Eins der Python-Bibliotheken für Machine Learning. Es wird in Wirtschaft und Forschung eingesetzt, und im Netz existieren zahlreiche Tutorials und Codebeispiele. `scikit-learn` arbeitet eng mit einigen weiteren wissenschaftlichen Python-Werkzeugen zusammen, die wir im Verlauf dieses Kapitels kennenlernen werden.

Wir empfehlen, dass Sie beim Lesen dieses Buches auch den User Guide (*http://scikit-learn.org/stable/user_guide.html*) und die Dokumentation der API von `scikit-learn` lesen, um zusätzliche Details und weitere Optionen zu jedem Algorithmus kennenzulernen. Die Online-Dokumentation ist sehr ausführlich, und dieses Buch liefert Ihnen die Grundlagen in maschinellem Lernen, um es im Detail zu verstehen.

Installieren von scikit-learn

`scikit-learn` benötigt zwei weitere Python-Pakete, *NumPy* und *SciPy*. Zum Plotten und zur interaktiven Entwicklung sollten Sie außerdem `matplotlib`, IPython und Jupyter Notebook installieren. Wir empfehlen Ihnen, eine der folgenden Python-Distributionen zu verwenden, in denen die notwendigen Pakete bereits enthalten sind: