

nen für viele eher ein Werkzeugkasten ist, den sie im Rahmen des sogenannten Data Minings bzw. der Knowledge Discovery in Databases einsetzen. Andere wiederum sehen es als Teil der künstlichen Intelligenz, und entsprechend schauen die Weisen sehr unterschiedlich auf den gleichen Elefanten.

## ■ 2.2 Künstliche Intelligenz, Data Mining und Knowledge Discovery in Databases

Wenn man von **künstlicher Intelligenz** spricht, steht man vor dem Problem, dass ohne das Adjektiv *künstlich* Intelligenz nicht im Sinne einer mathematischen Definition scharf definiert ist. Der Mensch geht davon aus, dass er – im unterschiedlichen Maße – intelligent ist und danach wird versucht, eine Definition zu erstellen.

Der Begriff *künstliche Intelligenz* spiegelt dabei den Menschen als Maßstab wider. Eine künstliche Intelligenz soll im Wesentlichen die gleichen intellektuellen Tätigkeiten wie ein Mensch ausführen können oder ihn dabei übertreffen. In der Forschung geht man davon aus, dass eine solche künstliche Intelligenz Folgendes leisten können sollte:

1. Logisches Denken
2. Treffen von Entscheidungen bei Unsicherheit
3. Planen
4. Lernen
5. Kommunikation in natürlicher Sprache

All diese Aspekte sollen eingesetzt werden können, um Ziele zu erreichen. Allgemein muss man sagen, dass der erste Punkt *Logisches Denken* zu den härtesten gehört und auch wegen des Wortes *Denken* am schwierigsten zu überprüfen ist.

Wo sind wir dort mit dem maschinellen Lernen zu finden? Nun, augenscheinlich dient das maschinelle Lernen dazu, den Punkt 4 *Lernen* zu bearbeiten. Die Algorithmen des maschinellen Lernens helfen zumindest auch beim Punkt 5 *Kommunikation* und sind ebenfalls in der Lage, den Punkt 3 *Planen* zu verbessern. Wie wir im Laufe des Buches noch sehen werden, fällt der Punkt 2 *Entscheidungen* auch fast vollständig in diesen Bereich. Bei so großen Überschneidungen ist es kein Wunder, dass die Begriffe oft durcheinandergeraten. Es ist allerdings nicht dasselbe, denn *Planen* oder *Entscheidungen bei Unsicherheiten treffen* kann man auch anders. Bis auf den Aspekt des Lernens selbst stellt das maschinelle Lernen oft Ansätze bereit, ist aber nicht der einzige Ansatz.

In der Aufzählung oben fehlen einige besonders schwer zu greifende Begriffe, die oft mit künstlicher Intelligenz verbunden werden, wie *Bewusstsein* und *Empfindungsvermögen*. Hier ist das maschinelle Lernen in dem mir bekannten Stand vollkommen außen vor und kann zumindest aktuell noch keinen Beitrag leisten; allenfalls es vortäuschen. Wenn eine Maschine die obigen Aspekte alle beherrschen würde, spräche man von einer **starken künstlichen Intelligenz**.

Die **schwache künstliche Intelligenz** hingegen beschränkt sich auf konkrete einzelne Anwendungsfelder – ist also keine universale Intelligenz – oder darauf, in gewissen Situationen intel-

ligent zu erscheinen. Letzteres, finde ich, macht sie dann wieder sehr menschlich, denn wer war noch nicht in der Lage, einmal schlauer aussehen zu müssen, als er vermutlich ist?

Der Turing-Test wird oft erwähnt, wenn es um die Überprüfung geht, ob eine Maschine intelligent ist. Er besteht im Wesentlichen daraus, dass ein Mensch nicht mehr in der Lage ist, zu erkennen, ob das Gegenüber bei einem Telefongespräch oder einem Chat eine Maschine oder ein Mensch ist. Hierzu gab es schon viele Tests und Programme, unter anderem **Cleverbot**, der auf Small-Talk spezialisiert ist und als Unterhaltungsmodul von **Hitchbot** diente, der als Anhalter in Kanada unterwegs war.

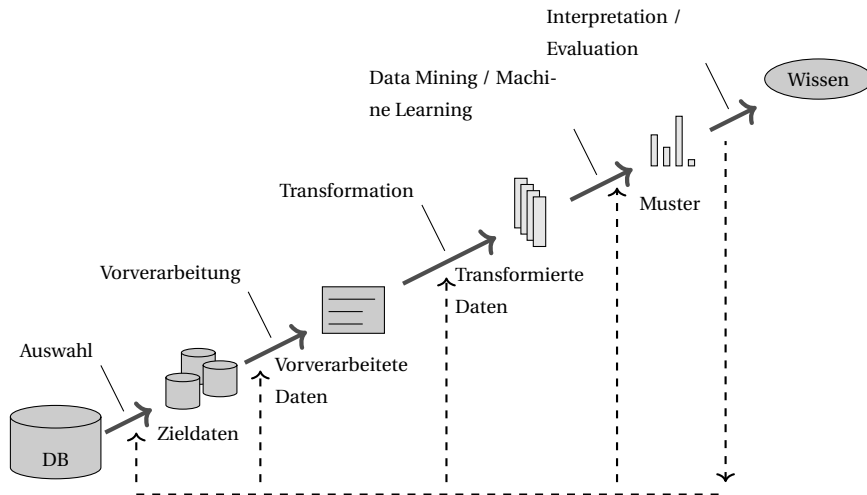
Die Frage, die Sie sich selbst beantworten müssen, ist, ob eine Maschine, die den **Turing-Test** besteht, eine starke oder eine schwache künstliche Intelligenz ist. Der amerikanische Philosoph John Rogers Searle zum Beispiel geht davon aus, dass in diesem Fall nur eine Intelligenz vorgetäuscht wird, und hat das in einem Gedankenexperiment, welches als das *Chinesische Zimmer* bekannt ist, ausgearbeitet. Die Frage ist, wie man überhaupt eine starke Intelligenz testen könnte.

Schwache künstliche Intelligenzen, welche quasi Spezialisten auf genau ihrem Gebiet sind, werden jedenfalls aktuell rapide weiterentwickelt und dringen in Bereiche der Produktion, Planung etc. vor. Natürlich sind auch Computerspiele typische Einsatzfelder schwacher künstlicher Intelligenzen. Hierbei muss man zwei Anwendungsfälle unterscheiden: Einmal die KI, die in Spielen – in der Regel mit vollständiger Information wie Schach oder GO – immer besser werden und menschliche Spieler hinter sich lassen. Zum anderen aber die künstliche Intelligenz, die in Computerspielen die Rolle von sogenannten *Non-Player-Characters* (NPC) übernimmt. Hier geht es wieder im Wesentlichen darum, ein gewünschtes Verhalten nachzuahmen und nicht darum, besser als der Spieler zu sein. Wenn dort in einem Rollenspiel ein Dorfdepp vorkommt, soll er nicht zu clever sein, sondern wie ein Schauspieler seine Rolle spielen. Bei NPCs geht es also oft darum, zu unterhalten, was hier dann die eigentliche Leistung ist. Alles andere wäre ökonomisch nicht weise, denn nur wenige von uns sind so veranlagt, dass sie Geld ausgeben, um sich von einem Computer einmal richtig demütigen zu lassen ... die meisten wollen eher selbst gewinnen. Während die Techniken für die Bewältigung von Spielen wie GO in der Regel auf maschinelles Lernen aufbauen, ist dies bei den NPCs in Computerspielen nur sehr selten der Fall.

Fassen wir einmal zusammen, dass maschinelles Lernen und künstliche Intelligenz eine große Überschneidung haben, und viele Techniken des maschinellen Lernens im Bereich der künstlichen Intelligenz eingesetzt werden. Wie sieht es mit dem anderen großen Feld aus, also der **Knowledge Discovery in Databases**?

Um das besser zu verstehen, beginnen wir mit dem Prozess der Knowledge Discovery in Databases, wie er in Abbildung 2.1 dargestellt ist. Diese Version geht auf die Veröffentlichung [FPSS96] zurück.

Am Anfang steht immer eine Sammlung von Daten, die wir einfachheitshalber mit einer großen Datenbank darstellen. Knowledge Discovery in Databases (KDD) ist dabei der Prozess der (semi-)automatischen Extraktion von Wissen aus eben dieser Datenbank. Wie man an dem Begriff (*semi-*)*automatisch* erkennt, ist hier oft ein Mensch enger Begleiter des Prozesses, und der Prozess läuft nicht immer autonom ab. In seiner (semi-)automatischen Form ist der KDD-Prozess interaktiv und iterativ, was bedeutet, dass der Anwender Entscheidungen trifft und einige Schritte ggf. wiederholt werden müssen. In Abbildung 2.1 wird dies durch die Rückwärtspeile nach jedem Hauptschritt angedeutet.



**Abbildung 2.1** Knowledge Discovery in Databases als Prozess

Beim KDD-Prozess geht es zunächst darum, die Anwendungsdomäne zu verstehen und Daten für die definierten Ziele auszuwählen. Der Grund ist, dass die Ausgangsdatensammlung nicht speziell für unser Ziel angefertigt wurde, sondern eben viele Dinge enthält, die für uns keine Rolle spielen. Im nächsten Schritt der Vorverarbeitung werden die Daten bereinigt. Das meint unter anderem den Umgang mit fehlenden Daten, wie wir ihn in Abschnitt 9.2.2 diskutieren. Als Nächstes müssen die Daten transformiert und ggf. auch reduziert werden. Die Transformation kann dabei z. B. darin liegen, Daten, die als Strings vorliegen, numerischen Werten zuzuordnen, da die meisten Methoden auf numerischen Werten basieren und eben nicht auf symbolischen Größen. Das Reduzieren meint dann z. B., Daten zusammenzulegen, wie wir es in Kapitel 9 besprechen werden. Nehmen wir als Beispiel an, in einer Datenbank sind Informationen zu Länge, Breite und Gewicht eines Fahrzeuges enthalten. Wollen wir nicht mit so vielen Merkmalen arbeiten, fassen wir diese drei Merkmale irgendwie zu einem Meta-Merkmal *Groß* zusammen. Wir haben im weiteren Verlauf nicht mehr Länge, Breite und Gewicht, sondern einen Wahrheitswert für *Groß*, der z. B. zwischen 0 und 1 liegt. Für die vorliegenden Daten wählen wir dann eine geeignete Methode aus dem Bereich des maschinellen Lernens – oder wie andere sagen würden Data Minings – aus und nutzen diese. Das Resultat sind neue Ergebnisse (Muster) in den Daten. Das wäre im Wesentlichen das Wissen. Wenn es aber rein um Muster geht, kann es auch sein, dass diese wieder von einem Menschen interpretiert werden und dann als Output dieses Prozesses verwendet werden, um z. B. Abläufe in Firmen oder Institutionen zu verbessern.

Solange es um die Erkennung von Mustern in Daten geht, fallen das unüberwachte Lernen als Teil des Machine Learnings, über das wir gleich noch sprechen werden, und das Data Mining als Schritt innerhalb des KDD-Prozesses eigentlich zusammen. Es gibt manchmal den Versuch, eine Abgrenzung über die Menge der Daten zu machen. Das bedeutet, Sie lesen vielleicht irgendwo, dass es sich um Machine Learning handle, wenn es in den Hauptspeicher passt. Wenn es hingegen sequenziell aus Datenbank-Anwendungen kommen muss, sei es Data Mining. Das Problem an diesem Ansatz ist, dass es vorkommen kann, dass eine Fragestel-

lung im Jahr 2005 Data Mining war, weil diese nicht in den Hauptspeicher passte. Aber im Jahr 2015 ist es dann Machine Learning, weil der durchschnittliche Hauptspeicher in Computern sich verändert hat. Das ist, wie ich finde, unglücklich. Es spricht nichts dagegen, Data Mining zu sagen, wenn man hervorheben möchte, dass eine klassische Datenbank Ausgangspunkt der Arbeit war und man nichts mit künstlicher Intelligenz zu tun hat. Wirklich sinnvoll zu unterscheiden sind jedoch primär der KDD-Prozess und das maschinelle Lernen als Werkzeug innerhalb dieses Prozesses.

## ■ 2.3 Strukturierte und unstrukturierte Daten in Big und Small

Beim maschinellen Lernen scheint es immer darum zu gehen, aus Daten Wissen zu generieren, und zwar unabhängig davon, ob im Umfeld der künstlichen Intelligenz Systeme trainiert werden oder im Rahmen eines KDD-Prozesses Wissen aus einer Datenbank erzeugt wird. Da also Daten der Dreh- und Angelpunkt der ganzen Angelegenheit sind, sollte man sich die unterschiedlichen Arten von Daten einmal ansehen.

Zunächst gilt es, zwischen strukturierten und unstrukturierten Daten zu unterscheiden. **Strukturierte Daten** kann man sich fast immer in Form einer Tabelle vorstellen. Jede Spalte stellt dabei ein **Merkmal** oder eben englisch **Feature** dar und jede Zeile einen Eintrag, der mehrere dieser Merkmale in einem **Datensatz** oder **Record** kombiniert. Ein Problem mit der Fachsprache und der Umgangssprache ist, dass *Datensatz* oft eher als *Datenbestand*, also mehr im Sinne von *ein Satz/eine Sammlung von Daten*, benutzt wird.

**Tabelle 2.1** Strukturierter Datenbestand in einer Tabellenform

Merkmale	$f_1$	$f_2$	$f_3$	...	$f_{n-2}$	$f_{n-1}$	$f_n$
Datensatz							
1							
2							
⋮							
$m-1$							
$m$							

Nehmen wir an, die Tabelle enthält Informationen zu PKW, dann kann jede Zeile für ein konkretes Auto stehen, und in den Spalten finden sich die Eigenschaften, wie die Anzahl der Türen ( $f_1$ ), der kombinierte Verbrauch ( $f_2$ ), der Anschaffungspreis ( $f_3$ ) etc. Solche Tabellen sind auch die Grundlage von relationalen Datenbankanwendungen wie SQL etc., sodass viele Daten, die wir in Unternehmen finden, strukturiert sind. Das bedeutet, wenn wir uns für eine Eigenschaft eines Objektes interessieren, wissen wir genau, wo wir diese finden. Die Informationen sind für uns in strukturierter Weise abrufbar.

Nun sind irgendwie alle Daten, die in einem Computer verarbeitet werden, strukturiert; also auch Bilder, die oft als Beispiel für **unstrukturierte Daten** genannt werden. Wie kommt das?



**Abbildung 2.2** Unstrukturierte Information, dass ein Hund auf dem Bild ist

Das Bild-Format als solches ist natürlich strukturiert. Wenn wir z. B. ein Bild im PNG-Format in Python laden, werden wir drei Matrizen mit RGB (Rot, Gelb, Blau)-Werten erhalten und können dadurch, dass bekannt ist, wie das Bildformat aufgebaut ist, dieses Bild auch anzeigen. Die Information, was z. B. auf dem Foto in Abbildung 2.2 zu sehen ist, können wir jedoch nicht strukturiert abgreifen. Ist eine Katze auf dem Bild oder ein Hund oder keines von beiden? Die Information ist irgendwie im Bild enthalten, jedoch nicht für uns direkt zugreifbar. Dasselbe gilt für freie Texte wie E-Mails, denn sie sind bzgl. der Informationen, die uns interessieren, unstrukturiert.

Es ist einsichtig, dass es für uns leichter ist, strukturierte Daten als Grundlagen für Lernalgorithmen zu verwenden als unstrukturierte. Ebenso muss man sich klarmachen, dass die Frage, ob etwas als strukturiert oder unstrukturiert gilt, manchmal von der Anwendung bzw. Frage abhängt. Nehmen wir an, Sie haben eine Aufnahme einer Wärmebildkamera. Wenn die gesuchte Information die Wärme an einem Bildpunkt ist, so ist dieses Bild als Informationsquelle sehr strukturiert. Wollen wir hingegen auf dem Bild erkennen, ob etwas ein Gesicht ist oder nicht, dann ist die Datenquelle unstrukturiert.

Ein anderer Begriff, der in letzter Zeit im Zusammenhang mit dem maschinellen Lernen durch die Presse wirbelt, ist **Big Data**. Was meint man damit und will man das eigentlich haben? Generell meint Big Data Datenbestände, die bzgl. ihrer Menge, Komplexität, schwachen Strukturierung und/oder Schnellebigkeit ein Problem für die herkömmliche Datenverarbeitung bzw. Datenanalyse sind. Eine recht akzeptierte Definition von Big Data bezieht das *big* auf drei Dimensionen

- Volume – großes Datenvolumen
- Velocity – große Geschwindigkeit, in der neue Daten generiert werden
- Variety – große Bandbreite der Datentypen und -quellen

Wenn man das zunächst so liest, dann ist Big Data nichts, was man haben möchte, denn das oben Aufgelistete bedeutet, dass man ein Problem hat, mit etwas umzugehen. Neben dem Punkt, dass Big Data aktuell durchaus ein Hype-Begriff ist, um Dinge zu verkaufen, ist es je-