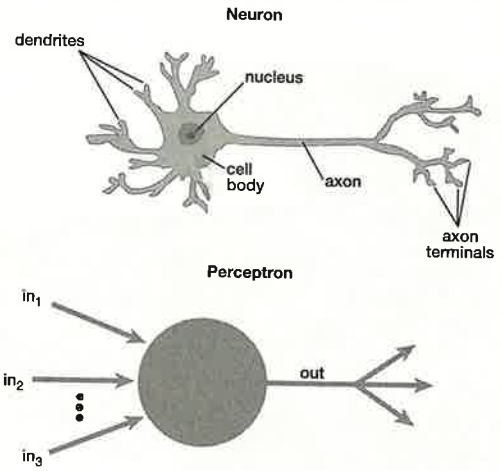
CHAPTER 8

Calling Bullshit on Big Data

The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

THUS BEGAN A JULY 8, 1958, ARTICLE IN *THE NEW YORK TIMES*. THE embryo was a simple building block called the *perceptron*.



"Perceptron." The word sounds magical—Houdini's new act—but like any good conjuring trick, its real magic lies in its simplicity. A perceptron is a simple logical circuit designed to mimic a biological neuron. It takes a set of numerical values as inputs, and then spits out either a 0 or a 1. The numerical inputs might be the pixel values of a chest X-ray; the output, whether a patient has pneumonia.

Connect enough of these perceptrons together in the right ways, and you can build a chess-playing computer, a self-driving car, or an algorithm that translates speech in real time like Douglas Adams's Babel Fish. You don't hear the term "perceptron" often these days, but these circuits are the building blocks for the convolutional neural networks and deep learning technologies that appear in headlines daily. The same old magic is still selling tickets.

The inventor of the perceptron, Frank Rosenblatt, was a psychologist by training, with broad interests in astronomy and neurobiology. He also had a knack for selling big ideas. While working at the Cornell Aeronautical Laboratory, he used a two-million-dollar IBM 704 computer to simulate his first perceptron. He described his work in grandiose terms. His machine, he told *The New York Times,* would think the way that humans do and learn from experience. Someday, he predicted, his perceptrons would be able to recognize faces and translate speech in real time. Perceptrons would be able to assemble other perceptrons, and perhaps even attain consciousness. Someday they could become our eyes and ears in the universe, sent out beyond the bounds of Earth to explore the planets and stars on our behalf.

Imagine the buzz that this kind of talk would have generated in 1958. How many science fiction writers were inspired by Rosenblatt and the sensationalized reports that followed?

Let's fast-forward fifty-five years. On December 28, 2013, *The New York Times* published yet another article about neural networks and their brain-like capabilities—retelling the same 1958 story. Though the computer hardware is vastly more powerful, the basic approach remains close to what Rosenblatt described a half century earlier. The author of the 2013 article speculated that in a very short amount of time, these brain-like machines "will make possible a new generation of artificial intelligence [AI] systems that will perform some functions that humans do with ease: see, speak, listen, navigate, manipulate and control." It was as if the original writers outslept Rip Van Winkle, waking fifty-five years later to write the same article about the same technology using the same superlatives.

So, what has changed since Rosenblatt's early experiments with perceptrons? The hype certainly hasn't diminished. The newspapers are full of breathless articles gushing about the latest breakthrough that someone promises is just around the corner. AI jobs are paying

superstar salaries. Tech firms are wooing away from campus professors with AI expertise. Venture capital firms are throwing money at anyone who can say "deep learning" with a straight face.

Here Rosenblatt deserves credit because many of his ambitious predictions have come true. The algorithms and basic architecture behind modern AI—machines that mimic human intelligence—are pretty much the same as he envisioned. Facial recognition technology, virtual assistants, machine translation systems, and stock-trading bots are all built upon perceptron-like algorithms. Most of the recent breakthroughs in machine learning—a subdiscipline of AI that studies algorithms designed to learn from data—can be ascribed to enormous leaps in the amount of data available and the processing power to deal with it, rather than to a fundamentally different approach.*

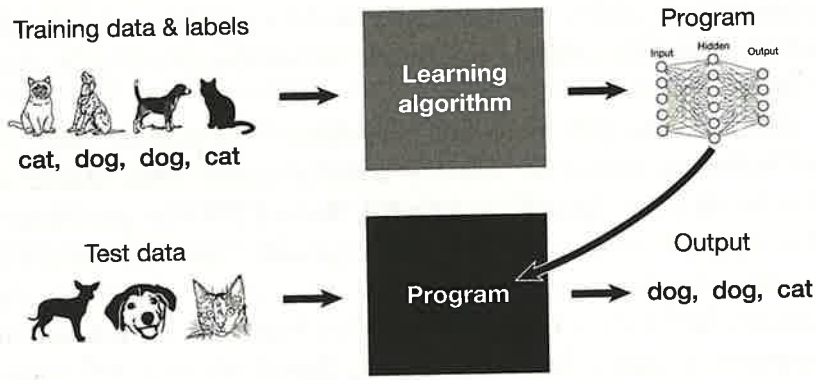
Indeed, machine learning and artificial intelligence live and die by the data they employ. With good data you can engineer remarkably effective algorithms for translating one language into another, for example. But there's no magical algorithm that can spin flax into gold. You can't compensate for bad data. If someone tells you otherwise, they are bullshitting.

How does machine learning work? It is a twist on the usual logic of computer programming. In a classical computer program, you write a program, give the computer data, and then the computer generates output:

In machine learning, you give the computer a set of *training data*. If you are teaching a computer to tell the difference between drawings of cats and dogs, these would be cat pictures and dog pictures. You

* Even though the term "big data" has already started to sound somewhat dated compared to fresher alternatives such as "machine intelligence" and "deep [anything]," we have titled this chapter around the phrase "big data" because it is data that is driving technology's current heyday. The algorithms are basically the same ones invented in the 1950s, and even computational power has started to level off over the past ten years.

also give the computer a set of *labels* for the training data that you know to be correct. In the cat and dog example, the computer would be told for each training image whether it was looking at a cat or a dog. The computer then uses a learning algorithm to produce a new program. For example, the learning algorithm might teach a neural network to distinguish cats from dogs. Then you can use the new program to label unfamiliar data, the *test data*. In the cat and dog example, you could then give the computer drawings it had never seen before, and it would tell you whether they are of cats or dogs.



Because the data are central to these systems, one rarely needs professional training in computer science to spot unconvincing claims or problematic applications. Most of the time, we don't need to understand the learning algorithm in detail. Nor do we need to understand the workings of the program that the learning algorithm generates. (In so-called deep learning models, no one—including the creators of the algorithm—really understands the workings of the program that algorithm generates.) All you have to do to spot problems is to think about the training data and the labels that are fed into the algorithm. Begin with bad data and labels, and you'll get a bad program that makes bad predictions in return. This happens so often that there is an acronym for it in computer science: GIGO—garbage in, garbage out. We'd like to deflate the hype behind AI, with a focus on the data rather than the details of the algorithm inside the black box.

The promise of AI spurs economic activity and inspires exciting science fiction plots; but it also creates unreasonable expectations, drives irresponsible research in both industry and academia, threat-

ens to extinguish any hope of personal privacy, and motivates misdirected policy. Researchers and technologists spend far too much time focusing on the sexy what-might-be, and far too little time on the important what-is. As Zachary Lipton, an AI researcher at Carnegie Mellon University, explains, "Policy makers [are] earnestly having meetings to discuss the rights of robots when they should be talking about discrimination in algorithmic decision making." Delving into the details of algorithmic auditing may be dull compared to drafting a Bill of Rights for robots, or devising ways to protect humanity against Terminator-like superintelligent machines. But to address the problems that AI is creating now, we need to understand the data and algorithms we are already using for more mundane purposes.

There is a vast gulf between AI alarmism in the popular press, and the reality of where AI research actually stands. Elon Musk, the founder of Tesla, SpaceX, and PayPal, warned US state governors at their national meeting in 2017 that AI posed a "fundamental risk to the existence of human civilization." Around the same time, *Fast Company* fueled those fears with an article headlined "AI Is Inventing Languages Humans Can't Understand. Should We Stop It?" Sounds scary, right? The story described a Facebook research project gone awry. While trying to build a chatbot that could carry on a convincing conversation, researchers tried having computer algorithms train one another to speak. But the speech that the algorithms developed was nothing like human language. *Fast Company* reported that the researchers quickly shut down the project. Skynet was officially on its way to self-awareness, but disaster had been averted—or so the story, and many others like it, suggested.

So what did this robot language look like? Let's take a look at some snippets of the chatbot conversation:

> BOB THE BOT: "I can can I I everything else."
> ALICE THE BOT: "Balls have zero to me to me to me to me to me to me to me to me to."
> BOB: "You I everything else."
> ALICE: "Balls have a ball to me to me to me to me to me to me to me to me."

It could be that AI will indeed evolve its own language and become self-aware. If it does, we hope it does so in peace and harmony with humans. But this Facebook chatbot was not headed down that particular road. The original blog post from the Facebook team simply described a research project where the chatbot language evolved the repetition of nonsensical sentences. The popular press distorted these into dramatic stories about shocked researchers scrambling to shut down the project and save the human race. But when asked by Snopes reporters about the story, the researchers commented that they had been unconcerned. It was the media response that surprised them. "There was no panic," one researcher said, "and the project hasn't been shut down." They weren't afraid for humankind. They simply observed that the chatbots were not heading toward the goal of human-like conversation, and went back to the drawing board.

We want to provide an antidote to this hype, but first, let's look at an instance in which machine learning delivers. It involves a boring, repetitive, everyday task that has received too little limelight rather than too much.

HOW MACHINES SEE

When you think about cutting-edge information technology, odds are you don't think about the US Postal Service. But in fact, few industries depend so critically upon advances in machine learning.

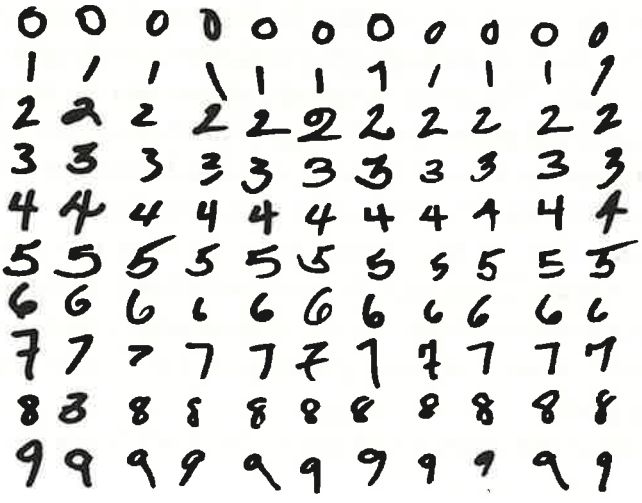
The US postal system processes half a billion pieces of mail every single day. It's a staggering number. If all seven billion people on the planet sent a letter or package, the postal service could process them all in a fortnight. This, even though the address on each piece of mail has to be read and interpreted. For typewritten addresses, this task is reasonably easy to delegate to machines. Handwriting is harder, but the US postal service has developed a remarkable handwriting recognition system that correctly interprets handwritten addresses 98 percent of the time. Those include your doctor's handwritten holiday cards, your grandma's letter to her local congresswoman, and your six-year-old's letter to the zoo pleading for a live video feed of the new baby giraffe.

What about the 2 percent of letters that can't be read by the ma-

chine? Those letters go to a massive postal complex in Salt Lake City. There, address experts decipher illegible addresses twenty-four hours a day, seven days a week, in thirty-three different shifts. The fastest of all the employees can process more than 1,800 addresses per hour, or one every two seconds! Santa's elves would have a hard time keeping up.

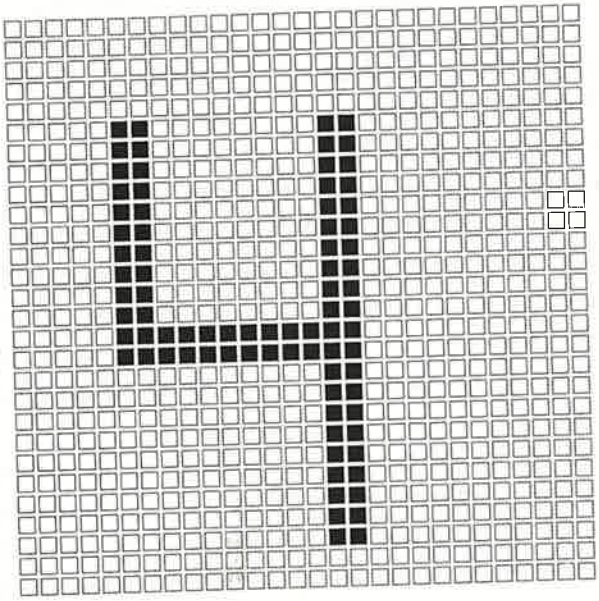
When it comes to open-ended tasks involving judgment and discretion, there is still no substitute for human intervention. Identifying fake news, detecting sarcasm, creating humor—for now, these are areas in which machines fall short of their human creators. However, reading addresses is relatively simple for a computer. The digit classification problem—figuring out whether a printed digit is a one, a two, a three, etc.—is a classic application of machine learning.

How does a computer do this? Much as we described in the cat and dog example. First we assemble a set of training data. We assemble a large collection of handwritten numbers—thousands of images—that humans have labeled as 0, 1, 2, . . . 9. Much of the time, the only limit to the computer's learning capacity is the availability of high-quality labels with which to train the machine. Fortunately for the postal service, an appropriate set of labeled, handwritten digits was created many years ago. It is called MNIST, the Modified National Institute of Standards and Technology database for handwritten digits, and it includes seventy thousand labeled images of handwritten digits, similar to those drawn below.



So how does the algorithm "see" images? If you don't have a background in computer vision, this may seem miraculous. Let's take a brief digression to consider how it works.

A computer stores an image as a matrix. A matrix can be thought of as a table of rows and columns. Each cell in this table contains a number. For simplicity, let's assume that our image is black and white. If a cell is black, the value is 0; otherwise, it is 1.*



The 28-row-by-28-column image above indicates the numeral 4. There are 784 squares, and each square is filled with a 0 or 1. Computers "see" images through these matrices. Seeing the digit 4 means identifying the unique properties of matrices that correspond to a handwritten 4 and then comparing matrices of similar images.

The known images with "4" are the training data on which machines learn.† Given enough training data and quantitative methods

* Grayscale images provide more options. Instead of just 1 or 0, the cell values can be numbers between 0 and 255, with smaller numbers corresponding to darker shades of gray. For color images, each cell has a value for red, green, and blue, separately.

† When training data with known labels is available, we call this a supervised machine learning problem. When labels with right or wrong answers are not available, we generally call it an unsupervised machine learning problem. As an example of an unsupervised learning problem, we might aim to find groups of customers with similar purchasing patterns. In this chapter, we focus on supervised machine learning problems.

for penalizing mistakes, we can teach the machine to consistently classify a handwritten "4." To evaluate how well the machine is learning, we provide it with test data: data it has never seen before. It is in dealing with test data that the rubber hits the road.

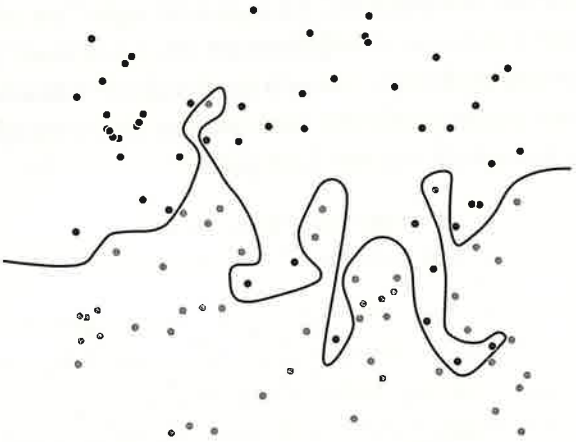
Often algorithms will perfectly classify all the training data, by essentially memorizing every data point and all its attributes. For the handwritten numerals, the machine might memorize the exact position and value of every pixel. Give it an image from that training set, and it will correctly guess the number with certainty. But that is not good enough. Training data drawn from the real world will invariably be messy. Some of the messiness comes from the idiosyncrasies of individual penmanship; some from badly scanned images; some when an image is mislabeled or drawn from the wrong data set entirely. Memorizing this noise creates problems for the computer when it is asked to label a new image that wasn't in the training set. If the accuracy of the computer's labeling drops significantly when it moves from training data to test data, the model is likely *overfitting*—classifying noise as relevant information when making predictions. Overfitting is the bane of machine learning.*

When scientists create models for predicting the positions of the planets, they don't catalog every position of every planet at every possible time; they identify the key physical laws for predicting future positions. The challenge in machine learning is to develop algorithms that can *generalize,* applying what they have learned to identify patterns they haven't seen before.

To get a better sense of how machines identify patterns and make predictions, let's walk through an example with nothing more than open and filled circles. But you could think of this as a data set of patients with and without diabetes based on various health indicators. Suppose we want to come up with a rule to predict whether a new point, not among this set, will be light or dark. The hundred points we already have are our training data. The dark circles are mostly at the top and the light circles are mostly at the bottom, so we might try to come up with a boundary that separates the light circles from the dark ones.

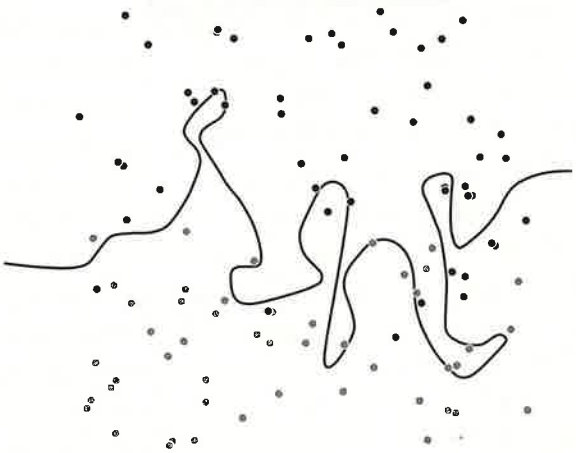
* When a model ignores both the noise and relevant patterns in the data, we call this underfitting, and it too can be a problem. In this case, the algorithm may perform about as well on the test data as on the training data—but performance on both data sets will be poor.

Miscategorized: 0 Dark, 0 Light



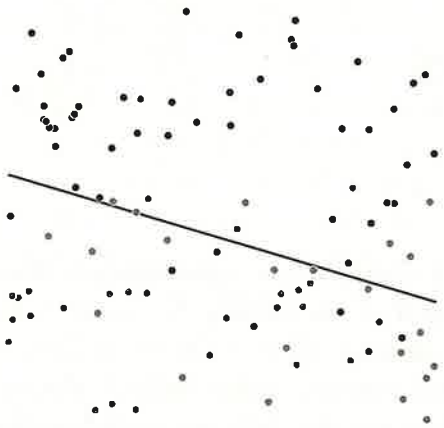
This boundary shown above—our model—does a beautiful job and gets every point of the training data correct. Every point above the boundary is dark, every point below is light. However, when given an additional hundred points (below), eleven filled points fall below the boundary and nine light points end up above it. What happened is that the boundary we picked from the training data weaved and wound its way between the light and dark points, and in doing so treated random noise in the data as if it were a meaningful pattern. Our model overfit the data.

Miscategorized: 11 Dark, 9 Light



To avoid overfitting, we could try using a simpler model. Let's try a straight line for a boundary. Though a straight line might be too simple for many data sets, it will illustrate our point here. This simpler model doesn't do perfectly—there is no straight line that separates the light and dark points perfectly—but we can find a straight line that misclassifies seven dark and ten light points.

Miscategorized: 7 Dark, 10 Light



This simpler model doesn't go twisting and winding all over the place to get every single point in the right place; it hasn't overfit the training data. Thus it performs about as well on the test data as it did on the training data. On the test data, this model misclassifies six dark points and five light points.

Miscategorized: 6 Dark, 5 Light

This is just a toy example, but the same issues arise in most machine learning applications. Complicated models do a great job of fitting the training data, but simpler models often perform better on the test data than more complicated models. The trick is figuring out just how simple of a model to use. If the model you pick is too simple, you end up leaving useful information on the table.

GARBAGE IN, GARBAGE OUT

Why is it so important to understand the role of training data in machine learning? Because it is here where the most catastrophic mistakes are made—and it is here where an educated eye can call bullshit on machine learning applications. In chapter 3, we told a story about a machine learning algorithm that was supposed to detect who was a criminal, but instead learned to detect who was smiling. The problem was with the training data. The criminal faces used to train the algorithm were seldom smiling, whereas the noncriminal faces were usually smiling. In the real world, smiling is not a good indicator of whether someone is a criminal or not, but the machine didn't know it was supposed to be finding signs of criminality. It was just trying to discriminate between two different sets of faces in the training set. The presence or absence of a smile turned out to be a useful signal, because of the way the training images were chosen.

No algorithm, no matter how logically sound, can overcome flawed training data. Early computing pioneer Charles Babbage commented on this in the nineteenth century: "On two occasions I have been asked, 'Pray, Mr. Babbage, if you put into the machine wrong figures, will the right answers come out?' . . . I am not able rightly to apprehend the kind of confusion of ideas that could provoke such a question."

As we noted earlier, good training data are often difficult and expensive to obtain. In addition, training data typically come from the real world—but the real world is full of human biases and their consequences. For various reasons, the glamorous side of machine learning research involves developing new algorithms or tweaking old ones. But what is more sorely needed is research on how to select appropriate, representative data. Advances in that domain would pay rich dividends.

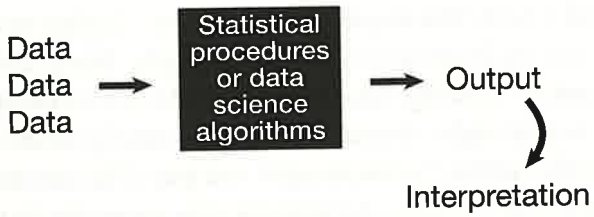
Let's now return to the post office. In this case, the data used to train the algorithm are actually quite good. The MNIST collection of

handwritten letters and numbers is comprehensive. The correct label for each image is straightforward to determine. There isn't much bullshit to call. The methods work well, sorting mail quickly and efficiently, while saving shippers and taxpayers millions of dollars.

Even relatively straightforward problems reading handwritten addresses run into issues of sampling bias, however. If our training data included only numbers written by United States residents, the algorithm might misclassify many 1s as 7s, because elsewhere in the world, a 1 is sometimes written with a hook at the top (and a 7 is written with a crosshatch to distinguish it).

We need to be sure that the training data cover the same range of variation that our algorithm will encounter in the real world. Fortunately, even when we account for international variation in writing styles, numerals afford a relatively constrained set of possibilities compared to most other data sets. Compare the challenge of teaching an algorithm to classify news stories as true or fake. This is much harder than deciding whether a handwritten numeral is a six. You don't necessarily know the answer just by looking at it; you may have to do some research. It's not clear where to do that research, or what sources count as authoritative. Once you find an answer, reasonable people still might not agree whether you are looking at fake news, hyperpartisan news, satire, or some other type of misinformation. And because fake news is continually evolving, a training set of fake news stories from 2020 might be out of date by 2021.

To call bullshit on the newest AI startup, it is often sufficient to ask for details about the training data. Where did it come from? Who labeled it? How representative is the data? Remember our black box diagram:



If the data going into the black box pass your initial interrogation, skip the algorithm and focus instead on the other end of the chain: what comes out of the black box and how it's interpreted.

GAYDAR MACHINES AND BULLSHIT CONCLUSIONS

In early September 2017, *The Economist* and *The Guardian* released a pair of oddly credulous news stories claiming that researchers at Stanford had developed AI capable of detecting a person's sexual orientation. The AI didn't need to observe a subject's behavior, examine his or her medical history (thank goodness), or ask anyone any questions. All it needed was a picture of a subject's face. The researchers announced that their AI could infer a person's sexual orientation from tiny differences in facial structure, differences too subtle for human observers to pick up at all.

This work attracted considerable attention from traditional and social media alike. Commentators mostly questioned the ethical implications of conducting such a study, given that homosexuality is often stigmatized, outlawed in a number of countries, and in some places even punishable by death. Should this sort of software have been developed at all? What were the implications of releasing it? What would this research mean for the LBGTQ community? These are all valid questions, but the first question should have been whether the study itself was any good.*

The *Economist* and *Guardian* stories described a research paper in which Stanford University researchers Yilun Wang and Michal Ko-

* We will assume that the authors reported their experiment accurately and that their algorithm performed as well as described. Indeed, their basic finding—that an algorithm can guess sexual orientation at a rate better than chance—was replicated using different training and test data in an unpublished master's thesis by John Leuner, though the algorithm had somewhat lower accuracy in that study.

sinski trained a deep neural network to predict whether someone was straight or gay by looking at their photograph. Wang and Kosinski collected a set of training images from an Internet dating website, photos of nearly eight thousand men and nearly seven thousand women, evenly split between straight and gay. The researchers used standard computer vision techniques for processing the facial images. When given pictures of two people, one straight and the other gay, the algorithm did better than chance at guessing which was which. It also did better than humans charged with the same task.

There are so many questions one could ask about the training data. What are the consequences of using training data that treats sexual preference as binary (gay or straight) instead of as a continuum? Do the photographs include a representative population? How were the photographs collected? Are the labels (straight or gay) accurate? What kinds of biases arise in the sample? These are all good questions, but here we want to look at what comes out of the black box instead of what goes in. We will ask whether the authors' two main inferences are justified based on their results.

Inference 1: The computer can detect subtle facial features that humans cannot.

The authors argue that the neural net is picking up on features that humans are unable to detect: "We show that faces contain much more information about sexual orientation than can be perceived and interpreted by the human brain." While the results are consistent with this claim, they do not come anywhere near demonstrating it. Indeed, there is a simpler explanation for why the computer does better than human subjects: The contest between the human and machine is not fair. For one thing, untrained human judges are pitted against a highly trained algorithm. The machine had thousands of images to learn from. The human players had no chance to practice, aside from exposure to a small number of images to make sure they understood the rules of the game.*

* Another concern is that the human judges were recruited through Amazon's Mechanical Turk crowdsourcing system. As such, they could have been working from anywhere in the world and may have not have been familiar with US cultural norms around how self-presentation varies with sexual orientation.

But there is a bigger problem. Humans are not good at aggregating information to make decisions, whereas computational learning algorithms do this sort of thing very well. Imagine the following: We provide a machine learning algorithm with a camera feed that shows a blackjack table. The machine then watches millions of blackjack games and learns to play. Blackjack is a simple game, and a standard learning algorithm could quickly infer good rules of thumb for playing. Once we have trained our algorithm, we compare the performance of the computer to the performance of nonexpert humans. The computer player would substantially outperform the average human player. So, what should we conclude?

One might conjecture that the machine can see some facet of the undealt cards that people cannot. Perhaps it is picking up information from the back of the facedown card atop the dealer's deck, for example. But notice that we are asking the computer and the humans to do two separate things: (1) detect cues on the game table, and (2) make good decisions about what to do given these cues. With our excellent visual systems and pattern detection abilities, humans are very good at the former task. How do we prove to websites that we are humans and not bots? By solving "CAPTCHA" visual processing challenges. But we are terrible at making probabilistic decisions based on partial information. It would be silly to conclude that the cards are somehow marked in a way visible to machines but hidden to humans. The simpler and more reasonable interpretation is that untrained humans are making stupid bets.

A parallel situation arises when we ask a computer or a human to determine, from photographs, which of two people is more likely to be gay. A human viewer gets all sorts of information from a photograph. Rather than trying to decide whether to hit or stand with a jack and a four while the dealer is showing a six,* you might see that one person has a baseball cap while the other has full sideburns. One has an eyebrow piercing, the other a tattoo. Each of these facts shifts your probability estimate of the subjects' sexual orientations. But by how much? How do these facts interact? Even trained humans are poor at making this kind of decision, and humans in this study are untrained. Of course they're no match for AI.

* Stand.

Inference 2: The differences the computer picks up are due to prenatal hormone exposure.

After concluding that the computer was picking up cues that go undetected by humans, Wang and Kosinski set out to explore what these cues might be. They observed that the faces of people with homosexual and heterosexual orientation had, on average, slightly different contours. From this finding the authors made a bizarre inferential leap to something known as the prenatal hormone theory (PHT) of sexual orientation. This theory proposes that differences in sexual orientation arise due to differences in hormone exposure prior to birth. The authors explain:

> According to the PHT, same-gender sexual orientation stems from the underexposure of male fetuses or overexposure of female fetuses to androgens [hormones associated with development of male characteristics] that are responsible for sexual differentiation. . . . As the same androgens are responsible for the sexual dimorphism of the face, the PHT predicts that gay people will tend to have gender-atypical facial morphology. . . . According to the PHT, gay men should tend to have more feminine facial features than heterosexual men, while lesbians should tend to have more masculine features than heterosexual women. Thus, gay men are predicted to have smaller jaws and chins, slimmer eyebrows, longer noses, and larger foreheads; the opposite should be true for lesbians.

Wang and Kosinski conclude that their findings "provide strong support for the PHT."

But extraordinary claims require extraordinary evidence. The idea that homosexual and heterosexual people have differently shaped faces due to prenatal hormone exposure is an extraordinary claim. Instead of extraordinary evidence, however, we get almost no evidence at all. Instead, the authors present us with the observation that the facial contours inferred by the deep neural net look a bit different. The gold standard for demonstrating differences in facial shapes requires 3D facial measurements under controlled laboratory condi-

tions; Wang and Kosinski use machine learning on self-selected 2D photographs from a dating site.

But all we really know is that a deep neural net can draw a distinction between self-selected photos of these two groups for reasons that we don't really understand. Any number of factors could be involved in the variation of these facial shapes, ranging from grooming to attire to photo choice to lighting. At the very least, the authors would need to show a statistically significant difference between face shapes. They fail to do even this.*

Suppose we were actually convinced that this study shows real structural, physiognomic differences (as opposed to differences in self-presentation) according to sexual orientation. Would these provide strong evidence for the PHT? Such results would be consistent with PHT, but, again, they don't offer strong evidence. What is the standard for strong evidence? To provide strong evidence for a hypothesis, a test has to put the hypothesis on the line—it has to have the potential to reject the hypothesis if the results come out a certain way. If this experiment had found no facial differences according to sexual preference, PHT proponents might explain that away either by pointing to the same kinds of limitations in study design that we have discussed above, or by positing that the primary effects of prenatal hormone exposure are behavioral rather than physiognomic.

On the other hand, if there are statistically significant actual physical differences in facial structure according to sexual orientation, they could arise from any of a number of mechanisms. The correlation between facial structure and sexual preference could be genetic. It could be due to some aspect of the environment other than hormone exposure. It could be due to hormone exposure outside the womb. It could be that people with different facial structures are treated differently, leading to differences in sexual orientation. Or it could be that differences in sexual orientation lead to differences in facial structure.

* The authors do find a statistically significant correlation between a measure that they call "facial femininity" and the probability of having a homosexual orientation. Moreover, the algorithm can make better-than-chance guesses based upon facial outline and nose shape. So the algorithm seems to be picking up something, but we suspect that facial contours and "facial femininity" are readily influenced by aspects of self-presentation including makeup, lighting, hairstyle, angle, photo choice, and so forth.

(This is not as silly as it sounds; facial width may be influenced by diet and exercise, which may be associated with sexual orientation.) We could go on and on. Our point is simply that the present study, even if taken at face value, does not support PHT over other possible explanations for homosexuality.

The results of the Wang and Kosinski paper are *consistent* with conclusions that the authors draw: that the possibility exists that computers can detect facial features that humans cannot, and that facial differences associated with sexual orientation arise through differences in prenatal hormone exposure. But the experiment fails to rule out numerous explanations that we consider more likely. We suspect the most likely explanation for their findings is that untrained humans are poor at integrating multiple cues to make good probabilistic estimates, and that sexual orientation influences grooming and self-presentation. Both of these are well-known facts; the former can explain why the neural network outperforms humans and the latter can explain why the neural network performs substantially better than chance. With those explanations in hand, there is no need to invoke mysterious features below the threshold of human perception, or effects of prenatal hormone exposure on facial structure. Wang and Kosinski overstepped in drawing strong conclusions from their weak results.

In the original paper and in conversations on Twitter, Wang and Kosinski expressed a wish that, for the sake of human rights, their results turn out to be wrong. We hope this analysis will help them rest easy.

HOW MACHINES THINK

It is not difficult to question the training data, as we did for the criminal faces paper in chapter 3, or the interpretation of the results, as we did for the gaydar paper in this chapter. And you can do both without opening the black box; in neither case did we have to discuss how a neural network operates.

Understanding the intricate details of how a specific algorithm makes decisions is a different story. It is extremely difficult to understand how even the most straightforward of artificial intelligence sys-

tems make decisions. Trying to explain how deep learning algorithms or convolutional neural networks do so is more difficult still. If you don't believe us, ask a computer scientist to explain how her neural network actually arrives at its results.

This opacity is a major challenge for the field of machine learning. The core purpose of the technology is to save people having to tell the computer what to learn in order to achieve its objective. Instead, the machines create their own rules to make decisions—and these rules often make little sense to humans.*

To better understand the decision-making process, researchers are finding new ways to peer into what the machines are "seeing" when they make decisions. Computer scientist Carlos Guestrin and his colleagues developed a straightforward, automated method for distinguishing photographs of huskies from photographs of wolves. In their paper they described the classification task, the methods they used, and how well their algorithm performed. Then they went one step further and looked at what information the algorithm was using in each image to draw the distinction.

Most studies don't take this extra step. It's difficult and the end users of these methods often don't care. "The method can distinguish huskies from wolves 70 percent of the time. Let's build an app." But we should care. This kind of examination allows us to better understand the limitations of AI and helps us interpret decisions that go awry in more serious applications.

Guestrin and his co-authors found that the algorithm was not paying much attention to snouts, eyes, fur, or any of the morphological features a person would use to distinguish a husky from a wolf. Instead, it was picking up on something external, a correlate of being a wolf that was present in the images. The machine learned that the wolf images but not the husky images tended to be shot in the snow, and exploited this difference in making its decisions.

* The AlphaGo program, which beat one of the best human Go players in the world, provides a good example. AlphaGo didn't start with any axioms, scoring systems, lists of opening moves, or anything of the sort. It taught itself how to play the game, and made probabilistic decisions based on the given board configuration. This is pretty amazing given the 10^{350} possible moves that Go affords. By comparison, chess has "only" about 10^{123}. Go masters have learned some new tricks from the play of AlphaGo, but good luck trying to understand what the machine is doing at any broad and general level.

(a) Husky classified as wolf

(b) Explanation

When machine learning algorithms key in on auxiliary features of this sort, they may do well at analyzing images exactly like the ones they were trained on, but they will not be able to generalize effectively to other contexts. John Zech and colleagues at California Pacific Medical Center wanted to investigate how well neural networks could detect pathologies such as pneumonia and cardiomegaly—enlarged heart—using X-ray images. The team found that their algorithms performed relatively well in hospitals where they were trained, but poorly elsewhere.

It turned out that the machine was cueing on parts of the images that had nothing to do with the heart or lungs. For example, X-ray images produced by a portable imaging device had the word PORTABLE printed in the upper right corner—and the algorithm learned that this is a good indicator that a patient has pneumonia. Why? Because portable X-ray machines are used for the most severely ill patients, who cannot easily go to the radiology department of the hospital. Using this cue improved prediction in the original hospital. But it was of little practical value. It had little to do with identifying pneumonia, didn't cue in on anything doctors didn't already know, and wouldn't work in a different hospital that used a different type of portable X-ray machine.

Machines are not free of human biases; they perpetuate them, depending on the data they're fed. In the case of criminal sentencing, *ProPublica* and others have shown that algorithms currently in use are identifying black defendants as "future" criminals at nearly twice the rate as white defendants, which leads to differences in pretrial release, sentencing, and parole deals. Algorithmic lenders charge higher inter-

est rates to both black and Latino applicants. Automated hiring software from some of the largest US employers such as Amazon have preferentially selected men over women. When we train machines to make decisions based on data that arise in a biased society, the machines learn and perpetuate those same biases. In situations like this, "machine learning" might better be called "machine indoctrination."

To address the major impact that algorithms have on human lives, researchers and policy makers alike have started to call for algorithmic accountability and algorithmic transparency. Algorithmic accountability is the principle that firms or agencies using algorithms to make decisions are still responsible for those decisions, especially decisions that involve humans. We cannot let people excuse unjust or harmful actions by saying "It wasn't our decision; it was the algorithm that did that." Algorithmic transparency is the principle that people affected by decision-making algorithms should have a right to know why the algorithms are making the choices that they do. But many algorithms are considered trade secrets.*

Perhaps the biggest problem with algorithmic transparencies comes back to the interpretability issue. Even if companies fully revealed the details of their algorithms, all their features and parameters, it may still be nearly impossible to understand why the algorithms make the decisions they do. Government policies can demand accountability, but they're not worth much if no one can understand what is going on well enough to explain what the algorithm is doing.†

Algorithmic biases can be particularly difficult to eradicate. Policy makers may require rules that forbid decisions based on race or gender, but it is often not sufficient to simply omit that information in the data provided to an algorithm. The problem is that other pieces of information may be correlated with race or gender, particularly when considered in concert. For example, if you build a machine to predict where the next domestic violence event will occur, the machine may choose an apartment over detached housing since those who share a

* Another reason to conceal the details of the algorithms is that secrecy also helps deter gaming. If Google were to release the exact details of its algorithm, search engine optimization companies would have a field day manipulating the algorithm, and the quality of Google's search results would decline.

† For example, the new General Data Protection Regulation (GDPR) in the European Union includes a "right to explanation" in Recital 71. This companion document will be influential going forward, but it is the most debated of all subjects in the GDPR, partly because of the difficulty in defining what is meant by "explain."

wall are more likely to report the event. This kind of prediction then resurfaces racial elements that you tried to remove in the first place. If you remove names from résumés as a way of eliminating gender discrimination, you may be disappointed, as Amazon was, when the machine continues to preferentially choose men over women. Why? Amazon trained the algorithm on its existing résumés, and there are features on a résumé besides a name that can reveal one's gender—such as a degree from a women's college, membership in a women's professional organization, or a hobby with skewed gender representation.

Philosophers often describe knowledge as "justified true belief." To know something, it has to be true and we have to believe it—but we also have to be able to provide a justification for our belief. That is not the case with our new machine companions. They have a different way of knowing and can offer a valuable complement to our own powers. But to be maximally useful, we often need to know how and why they make the decisions that they do. Society will need to decide where, and with which kinds of problems, this tradeoff—opacity for efficiency—is acceptable.

HOW MACHINES FAIL

In 2009, *Nature* published an article that described a new method for predicting flu outbreaks based on the search terms people use when querying Google. Terms such as "fever," "headache," "flu symptoms," and "pharmacies near me" could be used to track the spread of flu across the United States. Not only could these search frequencies and their geolocations be used to predict doctor visits, the method was both faster and cheaper than the epidemiological tracking methods employed by the Centers for Disease Control and Prevention (CDC).

The paper generated tremendous excitement and was covered by nearly every major newspaper and media outlet. Tech evangelists touted the results as an example of how big data would change the world. University professors, including one of the authors of this book, discussed the paper in their courses. Startup companies based around data analytics inserted the *Nature* paper into their pitch decks. When you have Google-scale data, argued *Wired* editor Chris Ander-

son, "the numbers speak for themselves." The scientific method was no longer necessary, he argued; the huge volumes of data would tell us everything we need to know. Data scientists didn't need years of epidemiological training or clinicians to diagnose flu symptoms. They just need enough data to "nowcast"* the flu and inform the CDC where to deliver Tamiflu. Or so we were told.

In all the excitement, we forget that if it sounds too good to be true, it probably is. And it was. By 2014, the headlines had turned from celebratory to monitory: "Google and the Flu: How Big Data Will Help Us Make Gigantic Mistakes," "Why Google Flu Is a Failure," "What We Can Learn from the Epic Failure of Google Flu Trends." The method worked reasonably well for a couple of years but, before long, the results started to miss the mark, not by a little but by a factor of two. Over time, the predictions continued to get worse. The results became so poor that Google axed the project and took down the Flu Trends website.

In retrospect, the study was doomed from the beginning. There was no theory about what search terms constituted relevant predictors of flu, and that left the algorithm highly susceptible to chance correlations in timing. For example, "high school basketball" was one of the top 100 predictors of a flu outbreak, simply because the search terms "flu" and "high school basketball" both peak in the winter. Like Hollywood relationships, spurious correlations fall apart on accelerated time scales. Search behavior and digital environments change—for example, Google introduced its "suggest" feature after this study was conducted.† Doing so likely altered people's search behavior. If you start typing "I am feeling" and the suggest feature offers "feverish," you may choose this term more often than other variants (e.g., "I am feeling hot"). This increases the frequency of certain queries, and in a vicious cycle they may become even more likely to be suggested by Google's autocomplete feature. When the frequency of search queries changes, the rules that the algorithm learned previously may no longer be effective.

If the Google Flu Trends algorithm had to predict flu cases for only

* "Nowcasting" is a flashy word made up to describe the practice of "predicting" aspects of the present or recent past using computer models, before anyone has time to measure them directly.

† The study included Web queries between 2003 and 2008. The suggest (autocomplete) functionality became broadly available on Google's website in 2008.

the first two years, we would still be writing about its triumph. When asked to extend beyond this time period, it failed. Sound familiar? Yep, this is overfitting. The machine likely focused on irrelevant nuances of that time period. This is where the scientific method can help. It is designed to develop theory that hyper-focuses on the key elements driving the spread of the flu, while ignoring the inconsequential. Search terms might be good indicators of those key elements, but we need a theory to help us generalize beyond two years of predictions. Without theory, predictions based on data rely on mere correlations.

When venturing into the black box, also consider the following. Many of the more complicated algorithms use dozens, hundreds, or even thousands of variables when making predictions. Google Flu Trends relied on forty-five key search queries that best predicted flu outbreaks. A machine learning system designed to detect cancer might look at a thousand different genes. That might sound like a good thing. Just add more variables; more data equals better predictions, right? Well, not exactly. The more variables you add, the more training data you need. We talked earlier about the cost of obtaining good training data. If you have ten thousand genes you want to incorporate into your model, good luck in finding the millions of example patients that you will need to have any chance of making reliable predictions.

This problem with adding additional variables is referred to as the *curse of dimensionality*. If you add enough variables into your black box, you will eventually find a combination of variables that performs well—but it may do so by chance. As you increase the number of variables you use to make your predictions, you need exponentially more data to distinguish true predictive capacity from luck. You might find that by adding the win-loss record of the New York Yankees to your list of a thousand other variables, you can better predict the Dow Jones index over the previous three months. But you will likely soon discover that this prediction success was the result of a chance alignment in the data—nothing more. Ask that variable to predict the index for the next three months, and the success rate will fall precipitously.

Researchers have not stopped trying to use data to help clinicians and solve health problems, nor should they. Researchers at Microsoft

Research are using search queries from Bing to detect people with undiagnosed pancreatic cancer, hopefully learning from the Google Flu Trends mistakes. Provided that it is collected legally, with consent, and with respect for privacy, data is valuable for understanding the world. The problem is the hype, the notion that something magical will emerge if only we can accumulate data on a large enough scale. We just need to be reminded: Big data is not better; it's just bigger. And it certainly doesn't speak for itself.

In 2014, TED Conferences and the XPrize Foundation announced an award for "the first artificial intelligence to come to this stage and give a TED Talk compelling enough to win a standing ovation from the audience." People worry that AI has surpassed humans, but we doubt AI will claim this award anytime soon. One might think that the TED brand of bullshit is just a cocktail of sound-bite science, management-speak, and techno-optimism. But it's not so easy. You have to stir these elements together just right, and you have to sound like you believe them. For the foreseeable future, computers won't be able to make the grade. Humans are far better bullshitters.