1 What is machine learning?

This chapter covers

- Machine-learning basics
- Advantages of machine learning over traditional approaches
- Overview of the basic machine-learning workflow
- Overview of advanced methods for improving model performance

In 1959, an IBM computer scientist named Arthur Samuel wrote a computer program to play checkers. Each board position was assigned a score based on its likelihood of leading to a win. At first, scores were based on a formula using factors such as the number of pieces on each side and the number of kings. It worked, but Samuel had an idea about how to improve its performance. He had the program play thousands of games against itself and used the results to refine the positional scoring. By the mid-1970s, the program had achieved the proficiency of a respectable amateur player.[1]

[1] Jonathan Schaeffer, *One Jump Ahead: Computer Perfection at Checkers* (New York: Springer, 2009).

Samuel had written a computer program that was able to improve its own performance through experience. It learned—and machine learning (ML) was born.

The aim of this book isn't to describe the gory mathematical details of machine-learning algorithms (although we'll peel back a few layers of the onion to provide insight into the inner workings of the most common ones). Rather, the book's primary purpose is to instruct non-experts on important aspects and common challenges when integrating machine learning into real-world applications and data pipelines. In this first chapter, we present a real business problem—reviewing loan applications—to demonstrate the advantages of using machine learning over some of the most common alternatives.

1.1 Understanding how machines learn

When we talk about human learning, we distinguish between rote learning, or memorization, and true intelligence. Memorizing a telephone number or a set of instructions is undoubtedly learning. But when we say *learning*, we frequently mean something more.

When children play in groups, they observe how others respond to their actions. Their future social behaviors are informed by this experience. But they don't rewind and replay their past. Rather, certain recognizable features of their interactions—playground, classroom, Mom, Dad, siblings, friends, strangers, adults, children, indoors, outdoors—provide clues. They assess each new situation based on the features it has in common with past situations. Their learning is more than gathering knowledge. They're building what might be called *insight*.

Imagine teaching a child the difference between dogs and cats by using flashcards. You show a card, the child makes a choice, and you place the card in one of two piles for right and wrong choices, respectively. As the child practices, his performance improves. Interestingly, it isn't necessary to first teach the child techniques for cat and dog recognition. Human cognition has built-in classification mechanisms. All that's needed are *examples*. After the child is proficient with the flashcards, he'll be able to classify not only the images on the flashcards, but also most any cat or dog image, not to mention the real thing. This ability to *generalize*, to apply knowledge gained through training to new unseen examples, is a key characteristic of both human and machine learning.

Of course, human learning is far more sophisticated than even the most advanced machine-learning algorithms, but computers have the advantage of greater capacity to memorize, recall, and process data. Their experience comes in the form of historical data that's processed—using the techniques described in this book—to create and optimize, through experience, algorithms that embody, if not true insight, at least the ability to generalize.

Analogies between human and machine learning naturally bring to mind the term *artificial intelligence* (AI) and the obvious question, "What's the difference between AI and machine learning?" There's no clear consensus on this matter, but most (not all) agree that ML is one form of AI, and that AI is a far broader subject encompassing such areas as robotics, language processing, and computer vision systems. To increase the ambiguity even further, machine learning is being applied in many of these adjacent AI fields with increasing frequency. We can say that the discipline of machine learning refers to *a specific body of knowledge and an associated set of techniques.* It's fairly clear what is, and what isn't, machine learning, whereas the same can't always be said for artificial intelligence. Paraphrasing Tom Mitchell's often-cited definition, a computer program is said to learn if its performance of a certain task, as measured by a computable score, improves with experience.[2]

Kaggle, a machine-learning consultancy, ran a competition for the most accurate program for classifying whether images depicted a dog or cat.[3] Competitors were provided 25,000 example images for training. Each was labeled to indicate the species depicted. After all the competitors had trained their algorithms, they were tested on their ability to classify 12,500 unlabeled test images.

When we explain the Kaggle competition to people, they often respond by reflecting on the sorts of rules one might apply to accomplish dog and cat recognition. Cats' ears are triangular and stand up; dogs' ears are floppy—but not always. Try to imagine how you might explain to a person who had never seen a dog or a cat how to tell the difference, without showing any examples.

People use a variety of methods involving shapes, colors, textures, proportions, and other features to learn, and to generalize, from examples. Machine learning also employs a variety of strategies, in various combinations, depending on the problem at hand.

These strategies are embodied in collections of algorithms developed over the course of recent decades by academics and practitioners in disciplines ranging from statistics, computer science, robotics, and applied mathematics, to online search, entertainment, digital advertising, and language translation. They are diverse and have various strengths and weaknesses. Some of them are classifiers. Others predict a numeric measurement. Some measure the similarity or difference of comparable entities (for example, people, machines, processes, cats, dogs). What the algorithms have in common is learning from examples (experience) and the capacity to apply what they've learned to new, unseen cases—the ability to generalize.

In the cats and dogs competition, during the learning phase, competitors' programs tried over and over to perform correct classifications using many algorithms. In each of the millions of iterations of the learning process, the programs performed the classification, measured their results, and then adjusted the process ever so slightly, searching for incremental improvements. The winner classified 98.914% of the unseen test images correctly. That's pretty good, considering the human error rate

---

2 Tom Mitchell, *Machine Learning* (McGraw Hill, 1997), 2. "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

3 See "Dogs vs. Cats" at www.kaggle.com/c/dogs-vs-cats.

is around 7%. Figure 1.1 illustrates the process. The machine-learning process analyzes labeled images and builds a model that is, in turn, used by the *recall* (prediction) process to classify unlabeled images. There's one mislabeled cat in the example.

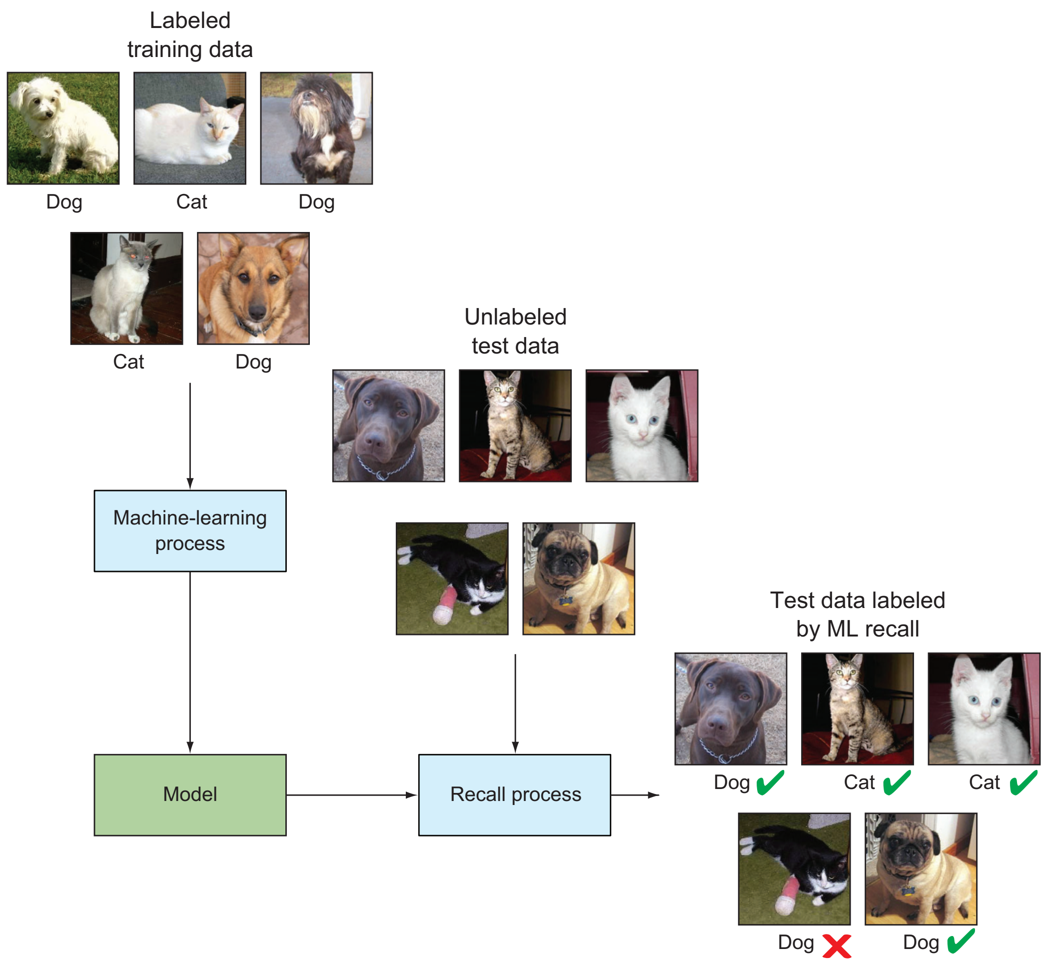

Figure 1.1 Machine-learning process for the cats and dogs competition

Please note that what we've described here is *supervised* machine learning, and it's not the only type of ML. We discuss other types later.

Machine learning can be applied to a wide range of business problems, from fraud detection, to customer targeting and product recommendation, to real-time industrial monitoring, sentiment analysis, and medical diagnosis. It can take on problems that can't be managed manually because of the huge amount of data that must be processed. When applied to large datasets, ML can sometimes find relationships so subtle that no amount of manual scrutiny would ever discover them. And when many such "weak" relationships are combined, they become strong predictors.

The process of learning from data, and subsequently using the acquired knowledge to inform future decisions, is extremely powerful. Indeed, machine learning is rapidly becoming the engine that powers the modern data-driven economy.

Table 1.1 describes widely used supervised machine-learning techniques and some of their practical applications. This isn't an exhaustive list, as the potential use cases could stretch across several pages.

Table 1.1 Use cases for supervised machine learning, organized by the type of problem

Problem	Description	Example use cases
Classification	Determine the discrete class to which each individual belongs, based on input data	Spam filtering, sentiment analysis, fraud detection, customer ad targeting, churn prediction, support case flagging, content personalization, detection of manufacturing defects, customer segmentation, event discovery, genomics, drug efficacy
Regression	Predict the real-valued output for each individual, based on input data	Stock-market prediction, demand forecasting, price estimation, ad bid optimization, risk management, asset management, weather forecasting, sports prediction
Recommendation	Predict which alternatives a user would prefer	Product recommendation, job recruiting, Netflix Prize, online dating, content recommendation
Imputation	Infer the values of missing input data	Incomplete patient medical records, missing customer data, census data

1.2 Using data to make decisions

In the following example, we describe a real-world business problem that can benefit from a machine-learning approach. We'll run through the various alternatives that are commonly used and demonstrate the advantages of the ML approach.

Imagine that you're in charge of a microlending company that provides loans to individuals who want to start small businesses in troubled communities. Early on, the company receives a few applications per week, and you're able in a few days' time to manually read each application and do the necessary background checks on each applicant to decide whether to approve each loan request. The schematic of this process is shown in figure 1.2. Your early borrowers are pleased with your short turnaround time and personal service. Word of your company starts to spread.

As your company continues to gain popularity, the number of applicants begins to increase. Soon you're receiving hundreds of applications per week. You try to stay up with the increased rate of applications by working extra hours, but the backlog of applications continues to grow. Some of your applicants grow weary of waiting and seek loans from your competitors. It's obvious to you that manually processing each application by yourself isn't a sustainable business process and, frankly, isn't worth the stress.

So what should you do? In this section, you'll explore several ways to scale up your application-vetting process to meet your increasing business needs.

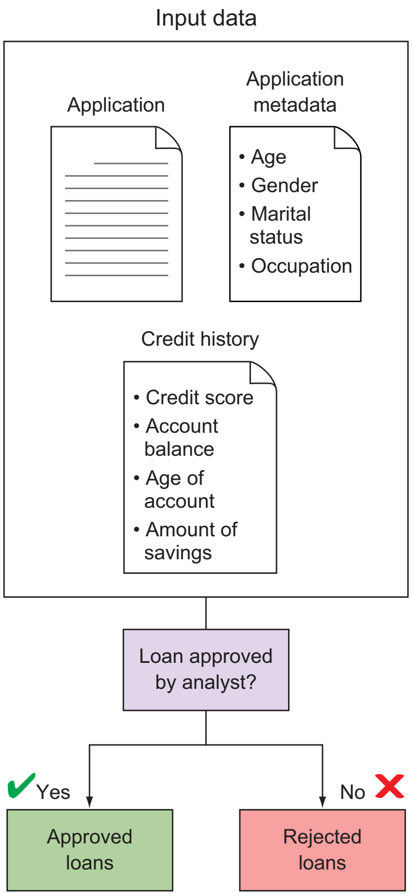

Figure 1.2 The loan-approval process for the microlending example

1.2.1 Traditional approaches

Let's explore two traditional data analysis approaches as applied to the application-vetting process: manual analysis and business rules. For each approach, we'll walk through the process of implementing the technique and highlight the ways in which it falls short of enabling you to build a scalable business.

Hire more analysts

You decide to hire another analyst to help you out. You aren't thrilled with the idea of spending some of your profit on a new hire, but with a second person vetting applications, you can process roughly twice as many applications in the same amount of time. This new analyst allows you to flush out the application backlog within a week.

For the first couple of weeks, the two of you stay up with demand. Yet the number of applications continues to grow, doubling within a month to 1,000 per week. To keep up with this increased demand, you now must hire two more analysts. Projecting forward, you determine that this pattern of hiring isn't sustainable: all of your increased revenue from new loan applicants is going directly to your new hires instead of to more-critical areas such as your microlending fund. *Hiring more analysts as demand increases hinders the growth of your business.* Further, you find that the hiring process is lengthy and expensive, sapping your business of more of its revenue. Finally, each new hire is less experienced and slower at processing applications than the last, and the added stress of managing a team of individuals is wearing on you.

Aside from the obvious disadvantage of increased cost, people bring all sorts of conscious and unconscious biases to the decision-making process. To ensure consistency, you might develop detailed guidelines for the approval process and implement an extensive training program for new analysts, but this adds still more cost and probably doesn't eliminate the bias.

EMPLOY BUSINESS RULES

Imagine that of the 1,000 loans whose repayment date has passed, 70% were repaid on time. This is shown in figure 1.3.

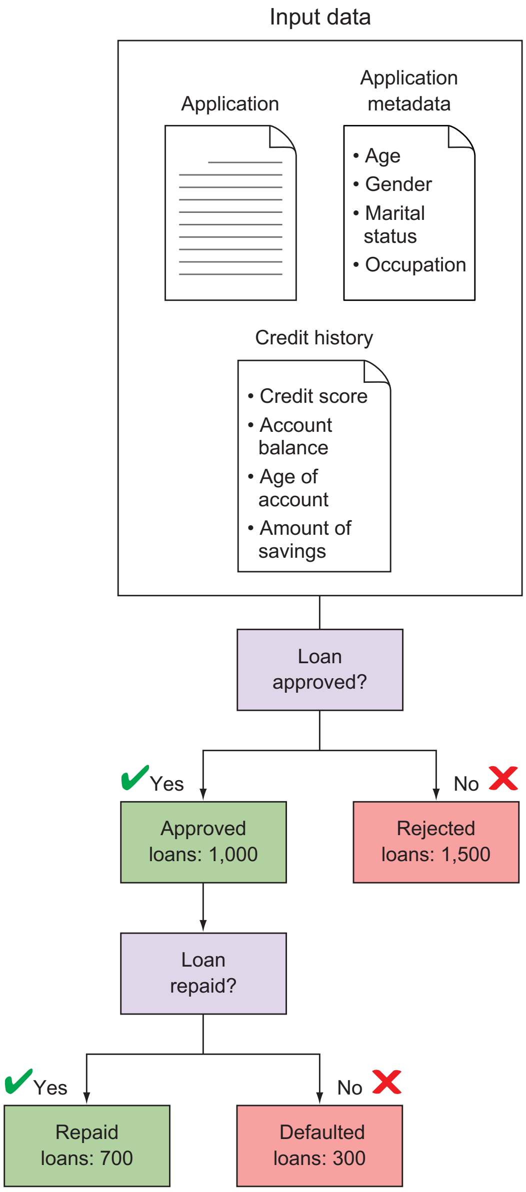

Figure 1.3 After a few months of business and 2,500 loan applications, 1,000 were approved, of which 700 applicants repaid the loan on time and the other 300 defaulted. This initial set of observed information is critical to start building automation into your loan-approval process.

You're now in a position to begin looking for trends between the applicant data and incidence of loan repayment. In particular, you perform a manual search for a set of filtering rules that produces a subset of "good" loans that were primarily paid on time. Through the process of manually analyzing hundreds of applications, you've gained extensive experience about what makes each application good or bad.[4] Through some introspection and back-testing of loan repayment status, you've noticed a few trends in the credit background checks data:[5]

- Most borrowers with a credit line of more than $7,500 defaulted on their loan.
- Most borrowers who had no checking account repaid their loan on time.

Now you can design a filtering mechanism to pare down the number of applications that you need to process manually through those two rules.

Your first filter is to automatically reject any applicant with a credit line of more than $7,500. Looking through your historical data, you find that 44 of the 86 applicants with a credit line of more than $7,500 defaulted on their loan. Roughly 51% of these high-credit-line applicants defaulted, compared to 28% of the rest. This filter seems like a good way to exclude high-risk applicants, but you realize that only 8.6% (86 out of 1,000) of your accepted applicants had a credit line that was so high, meaning that you'll still need to manually process more than 90% of applications. You need to do more filtering to get that number down to something more manageable.

Your second filter is to automatically accept any applicant who doesn't have a checking account. This seems to be an excellent filter, as 348 of the 394 (88%) applicants without a checking account repaid their loans on time. Including this second filter brings the percentage of applications that are automatically accepted or rejected up to 45%. Thus, you need to manually analyze only roughly half of the new incoming applications. Figure 1.4 demonstrates these filtering rules.

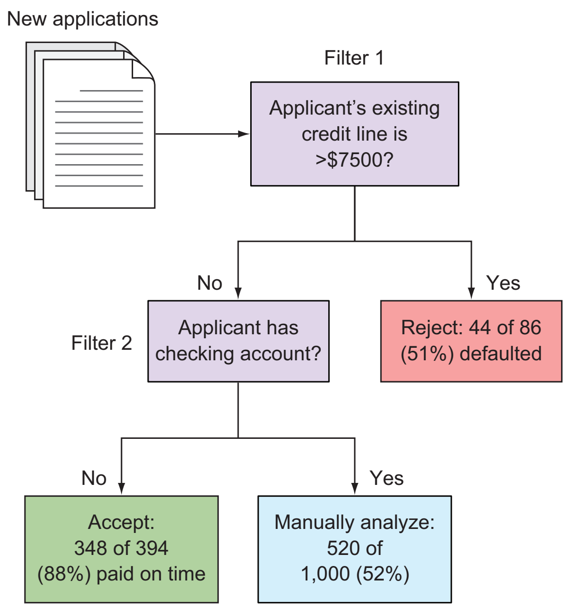

Figure 1.4 Filtering new applications through two business rules enables you to reduce manual analysis to only 52% of the incoming applications.

[4] You could also use statistical correlation techniques to determine which input data attributes are most strongly associated with the outcome event of loan repayment.

[5] In this example, we use the German Credit Data dataset. You can download this data from http://mng.bz/95r4.

With these two business rules, you can scale your business up to twice the amount of volume without having to hire a second analyst, because you now need to manually accept or reject only 52% of new applications. Additionally, based on the 1,000 applications with known outcome, you expect your filtering mechanism to erroneously reject 42 out of every 1,000 applications (4.2%) and to erroneously accept 46 of every 1,000 applications (4.6%).

As business grows, you'd like your system to automatically accept or reject a larger and larger percentage of applications without increasing losses from defaults. To do this, you again need to add more business rules. You soon encounter several problems:

- Manually finding effective filters becomes harder and harder—if not impossible—as the filtering system grows in complexity.
- The business rules become so complicated and opaque that debugging them and ripping out old, irrelevant rules becomes virtually impossible.
- The construction of your rules has no statistical rigor. You're pretty sure that better "rules" can be found by better exploration of the data, but can't know for sure.
- As the patterns of loan repayment change over time—perhaps due to changes in the population of applicants—the system doesn't adapt to those changes. To stay up to date, the system needs to be constantly adjusted.

All these drawbacks can be traced to a single debilitating weakness in a business rules approach: the system doesn't automatically learn from data.

Data-driven systems, from simple statistical models to more-sophisticated machine-learning workflows, can overcome these problems.

1.2.2 The machine-learning approach

Finally, you decide to look into an entirely automated, data-driven approach to your microlending application-vetting process. Machine learning is an attractive option because the completely automated nature of the process will allow your operation to keep pace with the increasing inflow of applications. Further, unlike business rules, ML learns the optimal decisions *directly from the data* without having to arbitrarily hard-code decision rules. This graduation from rules-based to ML-based decision making means that your decisions will be more accurate and will improve over time as more loans are made. You can be sure that your ML system produces optimized decisions with minimal handholding.

In machine learning, the data provides the foundation for deriving insights about the problem at hand. To determine whether to accept each new loan application, ML uses historical *training data* to predict the best course of action for each new application. To get started with ML for loan approval, you begin by assembling the training data for the 1,000 loans that have been granted. This training data consists of the input data for each loan application, along with the known outcome of whether each loan was repaid on time. The input data, in turn, consists of a set of *features*—numerical or categorical metrics that capture the relevant aspects of each application—such as the applicant's credit score, gender, and occupation.

In figure 1.5 historical data trains the machine-learning model. Then, as new loan applications come in, predictions of the probability of future repayment are generated instantaneously from the application data.

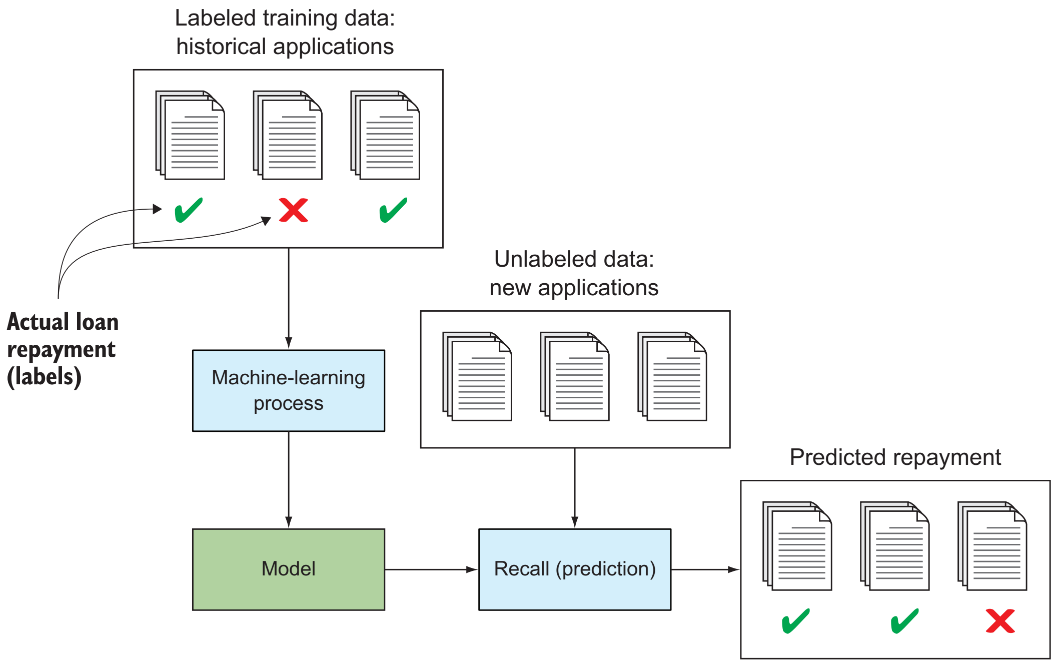

Figure 1.5 Basic ML workflow, as applied to the microloan example

ML modeling, then, determines how the input data for each applicant can be used to *best predict* the loan outcome. By finding and using patterns in the training set, ML produces a model (you can think of this as a black box, for now) that produces a prediction of the outcome for each new applicant, based on that applicant's data.

The next step is to select an ML algorithm to use. Machine learning comes in many flavors, ranging from simple statistical models to more-sophisticated approaches. Here, we compare two examples: the first is a simple parametric model, and the second a nonparametric ensemble of classification trees. Don't let the terminology scare you. Machine learning employs a lot of algorithms and lots of ways to categorize them, as you'll soon see.

Most traditional statistical business models fall into the first category. These parametric models use simple, fixed equations to express the relationship between the outcome and the inputs. Data is then used to learn the best values of the unknown terms in the equation. Approaches such as linear regression, logistic regression, and autoregressive models all fit under this category. Regression models are covered in more detail in chapter 3.

In this example, you could use logistic regression to model the loan-approval process. In logistic regression, the logarithm of the odds (the *log odds*) that each loan is repaid is modeled as a linear function of the input features. For example, if each new application contains three relevant features—the applicant's credit line, education level, and age—then logistic regression attempts to predict the log odds that the applicant will default on the loan (we'll call this y) via this equation:

Log odds that applicant will repay loan **Constant**

$$y = \beta_0 + \beta_1 * \text{Credit\_Line} + \beta_2 * \text{Education\_Level} + \beta_3 * \text{Age}$$

Coefficients

Log odds

The odds ratio is one way of expressing probability. You've undoubtedly heard someone say that a (favorite) team's chance of winning is 3 to 1. Odds are the probability of success (for example, winning) divided by the probability of failure (losing). Mathematically, this can be expressed as follows:

Odds(A) = P(A) / P(~A) = The probability of A divided by the probability of not A

So 3-to-1 odds is equivalent to 0.75 / 0.25 = 3 and log(3) = 0.47712…

If A were a fair coin toss, the odds of heads would be 0.5 / 0.5 = 1. Log(1) = 0. It turns out that the log(Odds) can take on any real-valued number. A log odds value near $-\infty$ denotes a highly unlikely event. A value near ∞ indicates near certainty, and log(1) = 0 indicates an even random change. Using log-odds instead of regular probabilities is a mathematical trick that makes certain computations easier, because unlike probabilities, they're not limited to values between 0 and 1.

The optimal values of each coefficient of the equation (in this case, β_0, β_1, β_2, and β_3) are learned from the 1,000 training data examples.

When you can express the relationship between inputs and outputs in a formula like this one, predicting the output (y) from the inputs (credit line, education level, and age) is easy. All you have to do is figure out which values of β_1, β_2, and β_3 yield the best result when using your historical data.

But when the relationship between the inputs and the response are complicated, models such as logistic regression can be limited. Take the dataset in the left panel of figure 1.6, for example. Here, you have two input features, and the task is to classify each data point into one of two classes. The two classes are separated in the two-dimensional feature space by a nonlinear curve, the *decision boundary* (depicted by the curve in the figure). In the center panel, you see the result of fitting a logistic regression model on this dataset. The logistic regression model comes up with a straight line that separates the two regions, resulting in many classification errors (points in the wrong region).

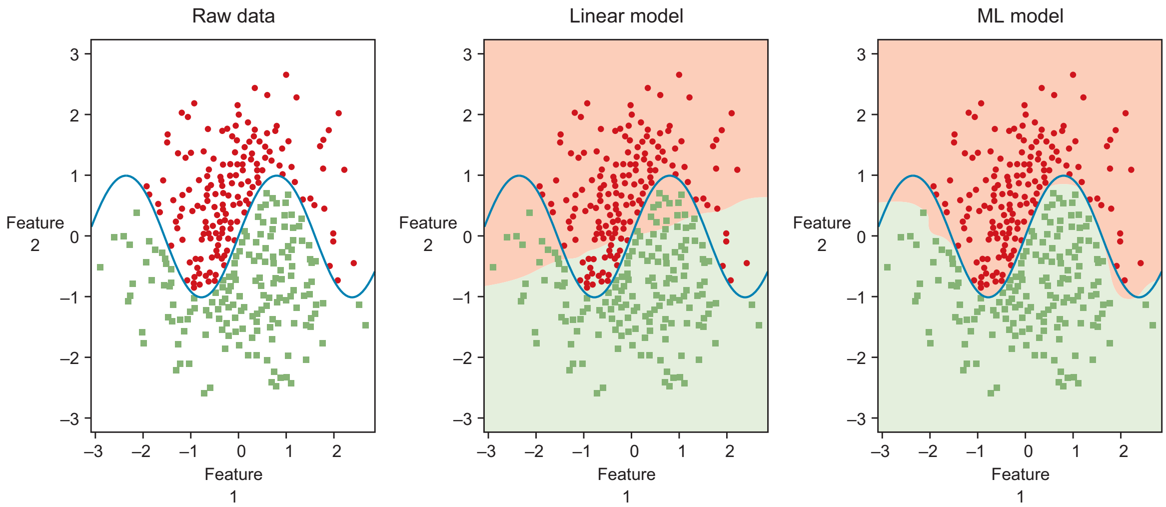

Figure 1.6 In this two-class classification, individual data points can belong to either the round class or the square class. This particular data lies in a two-dimensional feature space having a nonlinear decision boundary that separates the classes, denoted by the curve. Whereas a simple statistical model does quite poorly at accurately classifying the data (center), an ML model (right) is able to discover the true class boundary with little effort.

The problem here is that the model depicted in the center panel is attempting to explain a complicated, nonlinear phenomenon with a simple *parametric* model. The formal definition of parametric versus nonparametric models is complex and too mathematical for this book, but the gist is that parametric models work well when you have prior understanding of the relationship between your inputs and the response you're trying to predict. If you know enough about the nonlinear relationship, you may be able to transform your inputs or response variables so that a parametric model will still work. For example, if the rate at which a certain disease is observed within a population is higher for older people, you might find a linear relationship between the probability of contracting the disease and the square of the subject's age. But in the real world, you're often presented with problems for which such transformations aren't possible to guess.

What you need are more flexible models that can automatically discover complex trends and structure in data without being told what the patterns look like. This is where *nonparametric* machine-learning algorithms come to the rescue. In the right-hand panel of figure 1.6, you see the result of applying a nonparametric learning algorithm (in this case, a *random forest classifier*) to the problem. Clearly, the predicted decision boundary is much closer to the true boundary, and as a result, the classification accuracy is much higher than that of the parametric model.

Because they attain such high levels of accuracy on complicated, high-dimensional, real-world datasets, nonparametric ML models are the approach of choice for many data-driven problems. Examples of nonparametric approaches include some of the most widely used methods in machine learning, such as k-nearest neighbors, kernel smoothing, support vector machines, decision trees, and ensemble methods. We describe all of these approaches later in the book, and the appendix provides an overview of some important algorithms. Linear algorithms have other properties that make them attractive in some cases, though. They can be easier to explain and reason about, and they can be faster to compute and scale to larger datasets.

Further reading

The textbook *An Introduction to Statistical Learning* by Gareth James et al. (Springer, 2013) provides a detailed introduction to the most commonly used approaches in machine learning, at a level that's accessible to readers without a background in statistics or mathematics. A PDF version is available on the author's website (www-bcf.usc.edu/~gareth/ISL/).

Returning to the microlending problem, the best choice for scaling up your business is to employ a nonparametric ML model. The model may find the exact same rules as those you initially found manually, but chances are that they'll be slightly different in order to optimize the statistical gains. Most likely, the ML model will also automatically find other and deeper relationships between input variables and the desired outcome that you otherwise wouldn't have thought about.

In addition to providing an automated workflow, you may also attain higher accuracy, which translates directly to higher business value. Imagine that a nonparametric ML model yields 25% higher accuracy than a logistic regression approach. In this case, your ML model will make fewer mistakes on new applications: accepting fewer applicants who won't repay their loan and rejecting fewer applicants who would have repaid their loan. Overall, this means a higher average return on the loans that you do make, enabling you to make more loans overall and to generate higher revenues for your business.

We hope this gives you a taste of the power that machine learning can bring you. Before we move on to defining our basic machine-learning workflow, we'll enumerate a few advantages of machine learning, as well as a few challenges with this approach.