CHAPTER 8
Wrapping Up

You now know how to apply the important machine learning algorithms for supervised and unsupervised learning, which allow you to solve a wide variety of machine learning problems. Before we leave you to explore all the possibilities that machine learning offers, we want to give you some final words of advice, point you toward some additional resources, and give you suggestions on how you can further improve your machine learning and data science skills.

Approaching a Machine Learning Problem

With all the great methods that we introduced in this book now at your fingertips, it may be tempting to jump in and start solving your data-related problem by just running your favorite algorithm. However, this is not usually a good way to begin your analysis. The machine learning algorithm is usually only a small part of a larger data analysis and decision-making process. To make effective use of machine learning, we need to take a step back and consider the problem at large. First, you should think about what kind of question you want to answer. Do you want to do exploratory analysis and just see if you find something interesting in the data? Or do you already have a particular goal in mind? Often you will start with a goal, like detecting fraudulent user transactions, making movie recommendations, or finding unknown planets. If you have such a goal, before building a system to achieve it, you should first think about how to define and measure success, and what the impact of a successful solution would be to your overall business or research goals. Let's say your goal is fraud detection.

Then the following questions open up:

- How do I measure if my fraud prediction is actually working?
- Do I have the right data to evaluate an algorithm?
- If I am successful, what will be the business impact of my solution?

As we discussed in Chapter 5, it is best if you can measure the performance of your algorithm directly using a business metric, like increased profit or decreased losses. This is often hard to do, though. A question that can be easier to answer is "What if I built the perfect model?" If perfectly detecting any fraud will save your company $100 a month, these possible savings will probably not be enough to warrant the effort of you even starting to develop an algorithm. On the other hand, if the model might save your company tens of thousands of dollars every month, the problem might be worth exploring.

Say you've defined the problem to solve, you know a solution might have a significant impact for your project, and you've ensured that you have the right information to evaluate success. The next steps are usually acquiring the data and building a working prototype. In this book we have talked about many models you can employ, and how to properly evaluate and tune these models. While trying out models, though, keep in mind that this is only a small part of a larger data science workflow, and model building is often part of a feedback circle of collecting new data, cleaning data, building models, and analyzing the models. Analyzing the mistakes a model makes can often be informative about what is missing in the data, what additional data could be collected, or how the task could be reformulated to make machine learning more effective. Collecting more or different data or changing the task formulation slightly might provide a much higher payoff than running endless grid searches to tune parameters.

Humans in the Loop

You should also consider if and how you should have humans in the loop. Some processes (like pedestrian detection in a self-driving car) need to make immediate decisions. Others might not need immediate responses, and so it can be possible to have humans confirm uncertain decisions. Medical applications, for example, might need very high levels of precision that possibly cannot be achieved by a machine learning algorithm alone. But if an algorithm can make 90 percent, 50 percent, or maybe even just 10 percent of decisions automatically, that might already increase response time or reduce cost. Many applications are dominated by "simple cases," for which an algorithm can make a decision, with relatively few "complicated cases," which can be rerouted to a human.