knowing some of the theory behind the algorithms will make you a better data scientist. There have been many good books written about the theory of machine learning, and if we were able to excite you about the possibilities that machine learning opens up, we suggest you pick up at least one of them and dig deeper. We already mentioned Hastie, Tibshirani, and Friedman's book *The Elements of Statistical Learning* in the Preface, but it is worth repeating this recommendation here. Another quite accessible book, with accompanying Python code, is *Machine Learning: An Algorithmic Perspective* by Stephen Marsland (Chapman and Hall/CRC). Two other highly recommended classics are *Pattern Recognition and Machine Learning* by Christopher Bishop (Springer), a book that emphasizes a probabilistic framework, and *Machine Learning: A Probabilistic Perspective* by Kevin Murphy (MIT Press), a comprehensive (read: 1,000+ pages) dissertation on machine learning methods featuring in-depth discussions of state-of-the-art approaches, far beyond what we could cover in this book.

Other Machine Learning Frameworks and Packages

While `scikit-learn` is our favorite package for machine learning[1] and Python is our favorite language for machine learning, there are many other options out there. Depending on your needs, Python and `scikit-learn` might not be the best fit for your particular situation. Often using Python is great for trying out and evaluating models, but larger web services and applications are more commonly written in Java or C++, and integrating into these systems might be necessary for your model to be deployed. Another reason you might want to look beyond `scikit-learn` is if you are more interested in statistical modeling and inference than prediction. In this case, you should consider the `statsmodel` package for Python, which implements several linear models with a more statistically minded interface. If you are not married to Python, you might also consider using R, another lingua franca of data scientists. R is a language designed specifically for statistical analysis and is famous for its excellent visualization capabilities and the availability of many (often highly specialized) statistical modeling packages.

Another popular machine learning package is `vowpal wabbit` (often called `vw` to avoid possible tongue twisting), a highly optimized machine learning package written in C++ with a command-line interface. `vw` is particularly useful for large datasets and for streaming data. For running machine learning algorithms distributed on a cluster, one of the most popular solutions at the time of writing is `mllib`, a Scala library built on top of the `spark` distributed computing environment.

1 Andreas might not be entirely objective in this matter.

Ranking, Recommender Systems, and Other Kinds of Learning

Because this is an introductory book, we focused on the most common machine learning tasks: classification and regression in supervised learning, and clustering and signal decomposition in unsupervised learning. There are many more kinds of machine learning out there, with many important applications. There are two particularly important topics that we did not cover in this book. The first is *ranking*, in which we want to retrieve answers to a particular query, ordered by their relevance. You've probably already used a ranking system today; this is how search engines operate. You input a search query and obtain a sorted list of answers, ranked by how relevant they are. A great introduction to ranking is provided in Manning, Raghavan, and Schütze's book *Introduction to Information Retrieval*. The second topic is *recommender systems*, which provide suggestions to users based on their preferences. You've probably encountered recommender systems under headings like "People You May Know," "Customers Who Bought This Item Also Bought," or "Top Picks for You." There is plenty of literature on the topic, and if you want to dive right in you might be interested in the now classic "Netflix prize challenge", in which the Netflix video streaming site released a large dataset of movie preferences and offered a prize of $1 million to the team that could provide the best recommendations. Another common application is prediction of time series (like stock prices), which also has a whole body of literature devoted to it. There are many more machine learning tasks out there—much more than we can list here—and we encourage you to seek out information from books, research papers, and online communities to find the paradigms that best apply to your situation.

Probabilistic Modeling, Inference, and Probabilistic Programming

Most machine learning packages provide predefined machine learning models that apply one particular algorithm. However, many real-world problems have a particular structure that, when properly incorporated into the model, can yield much better-performing predictions. Often, the structure of a particular problem can be expressed using the language of probability theory. Such structure commonly arises from having a mathematical model of the situation for which you want to predict. To understand what we mean by a structured problem, consider the following example.

Let's say you want to build a mobile application that provides a very detailed position estimate in an outdoor space, to help users navigate a historical site. A mobile phone provides many sensors to help you get precise location measurements, like the GPS, accelerometer, and compass. You also have an exact map of the area. This problem is highly structured. You know where the paths and points of interest are from your map. You also have rough positions from the GPS, and the accelerometer and compass in the user's device provide you with very precise relative measurements. But throwing these all together into a black-box machine learning system to predict positions might not be the best idea. This would throw away all the information you