

scenes. The `scikit-learn` project is constantly being developed and improved, and it has a very active user community. It contains a number of state-of-the-art machine learning algorithms, as well as comprehensive [documentation](#) about each algorithm. `scikit-learn` is a very popular tool, and the most prominent Python library for machine learning. It is widely used in industry and academia, and a wealth of tutorials and code snippets are available online. `scikit-learn` works well with a number of other scientific Python tools, which we will discuss later in this chapter.

While reading this, we recommend that you also browse the `scikit-learn` [user guide](#) and API documentation for additional details on and many more options for each algorithm. The online documentation is very thorough, and this book will provide you with all the prerequisites in machine learning to understand it in detail.

## Installing scikit-learn

`scikit-learn` depends on two other Python packages, *NumPy* and *SciPy*. For plotting and interactive development, you should also install `matplotlib`, `IPython`, and the Jupyter Notebook. We recommend using one of the following prepackaged Python distributions, which will provide the necessary packages:

### *Anaconda*

A Python distribution made for large-scale data processing, predictive analytics, and scientific computing. Anaconda comes with `NumPy`, `SciPy`, `matplotlib`, `pandas`, `IPython`, Jupyter Notebook, and `scikit-learn`. Available on Mac OS, Windows, and Linux, it is a very convenient solution and is the one we suggest for people without an existing installation of the scientific Python packages. Anaconda now also includes the commercial Intel MKL library for free. Using MKL (which is done automatically when Anaconda is installed) can give significant speed improvements for many algorithms in `scikit-learn`.

### *Enthought Canopy*

Another Python distribution for scientific computing. This comes with `NumPy`, `SciPy`, `matplotlib`, `pandas`, and `IPython`, but the free version does not come with `scikit-learn`. If you are part of an academic, degree-granting institution, you can request an academic license and get free access to the paid subscription version of Enthought Canopy. Enthought Canopy is available for Python 2.7.x, and works on Mac OS, Windows, and Linux.

### *Python(x,y)*

A free Python distribution for scientific computing, specifically for Windows. `Python(x,y)` comes with `NumPy`, `SciPy`, `matplotlib`, `pandas`, `IPython`, and `scikit-learn`.

If you already have a Python installation set up, you can use `pip` to install all of these packages:

```
$ pip install numpy scipy matplotlib ipython scikit-learn pandas
```

## Essential Libraries and Tools

Understanding what `scikit-learn` is and how to use it is important, but there are a few other libraries that will enhance your experience. `scikit-learn` is built on top of the NumPy and SciPy scientific Python libraries. In addition to NumPy and SciPy, we will be using `pandas` and `matplotlib`. We will also introduce the Jupyter Notebook, which is a browser-based interactive programming environment. Briefly, here is what you should know about these tools in order to get the most out of `scikit-learn`.<sup>1</sup>

### Jupyter Notebook

The Jupyter Notebook is an interactive environment for running code in the browser. It is a great tool for exploratory data analysis and is widely used by data scientists. While the Jupyter Notebook supports many programming languages, we only need the Python support. The Jupyter Notebook makes it easy to incorporate code, text, and images, and all of this book was in fact written as a Jupyter Notebook. All of the code examples we include can be downloaded from [GitHub](#).

### NumPy

NumPy is one of the fundamental packages for scientific computing in Python. It contains functionality for multidimensional arrays, high-level mathematical functions such as linear algebra operations and the Fourier transform, and pseudorandom number generators.

In `scikit-learn`, the NumPy array is the fundamental data structure. `scikit-learn` takes in data in the form of NumPy arrays. Any data you're using will have to be converted to a NumPy array. The core functionality of NumPy is the `ndarray` class, a multidimensional ( $n$ -dimensional) array. All elements of the array must be of the same type. A NumPy array looks like this:

**In[2]:**

```
import numpy as np

x = np.array([[1, 2, 3], [4, 5, 6]])
print("x:\n{}".format(x))
```

---

<sup>1</sup> If you are unfamiliar with NumPy or `matplotlib`, we recommend reading the first chapter of the [SciPy Lecture Notes](#).