Catapult Launch for Python Data Science Libraries

Leon Hannah Tabak

Department of Computer Science

Cornell College

Mount Vernon, Iowa 52314

l.tabak@ieee.org

Abstract

The author shares a description of a Jupyter notebook that introduces students to features of the Python programming language that may be unfamiliar to beginning programmers and to three libraries of software that they will use in their study of machine learning. The author's course, CSC316 Machine Learning, is open to students who have completed only a single programming course. The libraries (Matplotlib, NumPy, and Pandas) have wide application in data science and machine learning. Students learn by studying and experimenting with examples in the Jupyter notebook. Students experiment by modifying parameters in function calls. In several places, the notebook shows students two ways of computing the same result and invites students to try both.

1 Challenges for our students

Students of machine learning must learn how to work with large tables of data. Skills learned in a first course in computer science are not enough to meet this challenge.

Students at Cornell College take one course at a time. Classes meet five days per week for three and a half weeks (eighteen days). Classes typically meet three or four hours per day. The compressed schedule increases the importance of finding ways to bring students quickly up to speed on essential skills so that they can engage with the most important themes of the course.

In their first course in computer science, students learned how to use assignment statements, **if** statements, and **for** loops. They learned the difference between integers and floating point numbers. The first course introduced them to lists or arrays. It gave them just a little practice importing modules (e.g., math or turtle) and using functions found in those modules.

To work efficiently with large datasets, students will need to develop their skills with Python further. They will need to acquaint themselves with additional data types and data structures. They will need to learn how to identify and use library functions.

A greater knowledge of the Python programming language and a familiarity with powerful and popular libraries enables students to express themselves more concisely. Concise expression, in turn, allows quicker experimentation, a deeper understanding of the problems that they are trying to solve, and easier communication with teammates and clients.

The author has composed a Jupyter notebook and given it to his students to use as a means of quickly gaining the skills they need for their study of machine learning.

2 Python programming skills

The author's Jupyter notebook shows students how to assign one tuple to another. By letting Python unpack a tuple, students can replace two or more assignment statements with a single assignment statement.

The notebook shows students how to use slicing to extract a subset of elements from a list and how to use list comprehensions to build lists. The notebook introduces students to functional programming with examples that use Python's **lambda**, **map()**, and **filter()** functions. These features of the programming language enable students in many cases to replace loops that require several lines of code with a single line of code.

3 Powerful and popular libraries

3.1 NumPy

The notebook shows students how to create and populate NumPy arrays. NumPy arrays can model vectors. Students get practice with combining vectors arithmetically. Students use NumPy's functions to find the minimum, maximum, and mean values in an array. Students learn the definitions and significance of vector norms, standard deviation, and root mean square error. In these exercises, students see how a function call in a single line of code can substitute for a loop that spans several lines. They get faster computation and more readable source code.

3.2 Pandas

Pandas provides DataFrames. These are two-dimensional data structures. Unlike NumPy arrays, these are heterogeneous data structures—different columns in a table can hold different types of data. Students learn how to label, add, and delete columns and rows. They learn how to select rows and columns by indices, labels, and logical conditions. Students learn how to generate statistical summaries of the contents of a DataFrame.

3.3 Matplotlib

The notebook invites students to experiment with code that draws curves, points (scatter plots), and histograms (bar charts). Exercises within the notebook ask students to change foreground and background colors, the widths of lines, labels on axes, and titles on figures. Students learn how to specify the size of a figure and how to include several plots within a single figure.

4 Practice problems

The Jupyter notebook that the author has created for his students produces:

- A list of all prime numbers less than a given integer. The code identifies these prime numbers using the Sieve of Eratosthenes algorithm. This algorithm makes an array of non-negative integers. It marks zero and one to indicate that they are not primes, then examines the other integers in turn from smallest to largest. If it finds an integer that is not yet marked, it calls it prime and marks all of its multiples to indicate that they are not prime. (2 is prime, and so 4, 6, 8, and so on cannot be prime.)
- The number of primes less than a given integer and the number of primes within a specified interval (a density function).
- The distances between successive primes. This exercise uses slices and Python's **zip()** and **map()** functions.

- A list of all twin primes (pairs of prime numbers whose difference is two) less than a given integer. This exercise uses Python's **filter()** function.
- A histogram that contains counts of the number of primes in an array whose least significant digit is 1, 3, 7, and 9. (There is only one prime number whose least significant digit is 2 and only one whose least significant digit is 5.)
- The totients of all positive integers less than a given integer. The totient of a positive integer N is the number of positive integers less than N that have no factor in common with N except for one. The algorithm for computing totients computes a product of differences. The function's arguments are the integer N and an array of prime numbers less than or equal to N. This example shows students how to connect functions that produce and consume NumPy arrays.

The notebook reads data from an image file into a NumPy array. The image is a gray scale image. Using vector arithmetic, the code produces a negative image, images with fewer shades of gray ("posterized" images), and images with false colors. The same methods work for visualizing other kinds of two dimensional data with heat maps and contour maps.

The notebook generates and plots points on a line in the plane. It also generates points that lie near the line to simulate observations that contain noise, and then uses the method of least squares to find the line that best bits the noisy data. This final exercise prepares students for later experiments. They will write code to create small, synthetic datasets for the purpose of testing machine learning functions before applying those functions to larger, real datasets.

5 The value of Jupyter notebooks

A Jupyter notebook can contain formatted text, mathematical notation, images, hyperlinks, and executable code. An author of a notebook can describe a method for solving a problem with words, equations, and a computer program. Multiple views of a method for solving a problem enable readers to more easily understand the method. Readers can execute the code with different parameters. They can annotate an author's explanations. Jupyter notebooks invite active learning.

The author did not attempt to teach his students all features of the Matplotlib, NumPy, and Pandas libraries (an impossibility, of course!). The author did not even attempt to identify the must important features. Instead, the author created a notebook that shows his students a sampling of features. It shows them a style of programming that will be unfamiliar to most. It invites them to try different ways of solving problems, and to adapt their own habits. Working with the notebook, students learn by doing. They learn how to build projects by altering a template or example in small steps until the product is their own.

References

View and download the author's Jupyter notebook at bitbucket.org/leontabak/catapult.