As a beginner how do I start learning machine learning with python?

Machine learning is a field that uses algorithms to learn from data and make predictions. Practically, this means that we can feed data into an algorithm, and use it to make predictions about what might happen in the future. This has a vast range of applications, from self-driving cars to stock price prediction. Not only is machine learning interesting, it’s also starting to be widely used, making it an extremely practical skill to learn.

In this tutorial, we’ll guide you through the basic principles of machine learning, and how to get started with machine learning with Python. Luckily for us, Python has an amazing ecosystem of libraries that make machine learning easy to get started with. We’ll be using the excellent Scikit-learn, Pandas, and Matplotlib libraries in this tutorial.

Moving forward, I make the assumption that you are not an expert in:

Machine learning

Python

Any of Python's machine learning, scientific computing, or data analysis libraries

It would probably be helpful to have some basic understanding of one or both of the first 2 topics, but even that won't be necessary; some extra time spent on the earlier steps should help compensate.

Basic Python Skills

If we intend to leverage Python in order to perform machine learning, having some base understanding of Python is crucial. Fortunately, due to its widespread popularity as a general purpose programming language, as well as its adoption in both scientific computing and machine learning, coming across beginner's tutorials is not very difficult. Your level of experience in both Python and programming in general are crucial to choosing a starting point.

First, you need Python installed. Since we will be using scientific computing and machine learning packages at some point, I suggest that you install Anaconda. It is an industrial-strength Python implementation for Linux, OSX, and Windows, complete with the required packages for machine learning, including numpy, scikit-learn, and maloti. It also includes iPython Notebook, an interactive environment for many of our tutorials. I would suggest Python 2.7, for no other reason than it is still the dominant installed version.

If you have no knowledge of programming, my suggestion is to start with the following free online book, then move on to the subsequent materials:

Python The Hard Way, by Zed A. Shaw

learn-pythonIf you have experience in programming but not with Python in particular, or if your Python is elementary, I would suggest one or both of the following:

Google Developers Python Course (highly recommended for visual learners)

An Introduction to Python for Scientific Computing (from UCSB Engineering), by M. Scott Shell (a great scientific Python intro ~60 pages)

And for those looking for a 30 minute crash course in Python, here you go:

Learn X in Y Minutes (X = Python)

Of course, if you are an experienced Python programmer you will be able to skip this step. Even if so, I suggest keeping the very readable Python documentation handy.

Foundational Machine Learning Skills

KDnuggets' own Zachary Lipton has pointed out that there is a lot of variation in what people consider a "data scientist." This actually is a reflection of the field of machine learning, since much of what data scientists do involves using machine learning algorithms to varying degrees. Is it necessary to intimately understand kernel methods in order to efficiently create and gain insight from a support vector machine model? Of course not. Like almost anything in life, required depth of theoretical understanding is relative to practical application. Gaining an intimate understanding of machine learning algorithms is beyond the scope of this article, and generally requires substantial amounts of time investment in a more academic setting, or via intense self-study at the very least.

The good news is that you don't need to possess a PhD-level understanding of the theoretical aspects of machine learning in order to practice, in the same manner that not all programmers require a theoretical computer science education in order to be effective coders.

Andrew Ng's Coursera course often gets rave reviews for its content; my suggestion, however, is to browse the course notes compiled by a former student of the online course's previous incarnation. Skip over the Octave-specific notes (a Matlab-like language unrelated to our Python pursuits). Be warned that these are not "official" notes, but do seem to capture the relevant content from Andrew's course material. Of course, if you have the time and interest, now would be the time to take Andrew Ng's Machine Learning course on Coursera.

Unofficial Andrew Ng course notes

There all sorts of video lectures out there if you prefer, alongside Ng's course mentioned above. I'm a fan of Tom Mitchell, so here's a link to his recent lecture videos (along with Maria-Florina Balcan), which I find particularly approachable:

Tom Mitchell Machine Learning Lectures

You don't need all of the notes and videos at this point. A valid strategy involves moving forward to particular exercises below, and referencing applicable sections of the above notes

and videos when appropriate. For example, when you come across an exercise implementing a regression model below, read the appropriate regression section of Ng's notes

and/or view Mitchell's regression videos at that time.

CONCLUSION:

There are many good videos lectures available on Machine learning which would accelerate the learning and it will be qualitive in nature.