# Software tools you'll need for this course

It's important to emphasize that this specialization is **not** about providing training for a specific software package. The goal of the specialization is for your effort to be spent on learning the fundamental concepts and algorithms behind machine learning in a hands-on fashion. These concepts transcend any single package. What you learn here you can use whether you write code from scratch, use any existing ML packages out there, or any that may be developed in the future.

**How this specialization was designed.** The learning approach in this specialization is to start from use cases and then dig into algorithms and methods, what we call a case-studies approach. We are very excited about this approach, since it has worked well in several other courses. The first course, Machine Learning: Foundations, was focused on understanding how ML can be used in various cases studies. The second course, Machine Learning: Regression, was focused on models that predict a continuous value from input features. The third course, Machine Learning: Classification, was focused on predicting a discrete value (class) from input features. The follow on courses will dig into more details on the algorithms and methods of other ML areas. ***We expect all learners to have taken the first, second, and third courses before taking this course.***

**Clustering & Retrieval - A Machine Learning Approach.** This course focuses on clustering and retrieval, some of the most high-impact machine learning tools out there. Retrieval is used in almost every applications and device we interact with, like in providing a set of products related to one a shopper is currently considering, or a list of people you might want to connect with on a social media platform. Clustering can be used to aid retrieval, but is a more broadly useful tool for automatically discovering structure in data, like uncovering groups of similar patients. After successfully completing this course, you will be able to use clustering and retrieval methods in practice, implement some of the most fundamental algorithms in this area, and choose the right model for your task. You will become familiar with the most successful techniques, which are most widely used in practice, including k-nearest neighbors, k-means, mixture models, and latent Dirichlet allocation. In addition, you will be able to design and implement the underlying algorithms that can learn these models at scale, using KD-trees, locality sensitive hashing, and MapReduce.

## Programming assignment format

Almost every module will be associated with one or two programming assignments. The goal of these assignments is to have hands-on experience on the techniques we discuss in lectures. To test your implementations, you will be asked questions in a quiz following the assignment.

You will be implementing core clustering and retrieval techniques or other ML concepts from scratch in most modules. In a few modules, you will use existing implementations of ML algorithms, with the goal of gaining proficiency in the ML concepts.

## Why Python

In this course, we are going to use the Python programming language to build several intelligent applications that use machine learning. Python is a simple scripting language that makes it easy to interact with data. Furthermore, Python has a wide range of packages that make it easy to get started and build applications, from the simplest ones to the most complex. Python is widely used in industry, and is becoming the de facto language for data science in industry. (R is another alternative language. However, R tends to be significantly less scalable and has very few deployment tools, thus it is seldomly used for production code in industry. It is possible, but discouraged to use R in this specialization.)

We will also encourage the use the IPython Notebook in our assignments. The IPython Notebook is a simple interactive environment for programming with Python, which makes it really easy to share your results. Think about it as a combination of a Python terminal and a wiki page. Thus, you can combine code, plots and text to explain what you did. (You are not required to use IPython Notebook in the assignments, and should have no problem using straight up Python if you prefer.)

## Useful software tools

Although you will be implementing algorithms from scratch in various assignments, some software tools will be useful in the process. In particular, there are four types of data tools that would be helpful:

- **Data manipulation**: to help you slice-and-dice the data, create new features, and clean the data.
- **Matrix operations**: in the inner loops of your algorithms, you will do various matrix operations, and libraries focus on these will speed-up your code significantly.
- **Plotting library**: so you can visualize data and models.

- **Pre-implemented ML algorithms**: in some assignments where we are focusing on exploring the output of the ML method, you will use a pre-implemented ML algorithms to help focus your efforts on the fundamentals.

## Tools for data manipulation

For data manipulation, we recommend using SFrame, an open-source, highly-scalable Python library for data manipulation. An alternative is the Pandas library. A huge advantage of SFrame over Pandas is that with SFrame, you are not limited to datasets that fit in memory, which allows you to deal with large datasets, even on a laptop. (The SFrame API is very similar to Pandas' API. Here is a doc showing the relationship between the two of them.)

## Tools for matrix operation

For matrix operations, we strongly recommend Numpy, an open-source Python library that provides fast performance, for data that fits in memory.

## Tools for plotting

For plotting, we strongly recommend you use Matplotlib, an open-source Python library with extensive plotting functionality.

## Tools with pre-implemented ML algorithms

For the few assignments where you will be using pre-implemented ML algorithms, we recommend you useGraphLab Create, which we used in the first, second, and third courses, a package we have been working on for many years now, and has seen an exciting adoption curve, especially in industry with folks building real applications. A popular alternative is to use scikit-learn. GraphLab Create is more scalable than scikit-learn and simpler to use when your data is not numeric vectors. On the other hand, scikit-learn is open-source. GraphLab Create is free on a 1-year, renewable license for educational purposes, including Coursera. This software, however, has a paid license for commercial purposes.

## For full disclosure!

GraphLab Create is very actively used in industry by a large number of companies. This package was created by a machine learning company called Dato. This company is spin off from a popular research project called GraphLab, which Carlos Guestrin, one of your two instructors, and his research group started at Carnegie Mellon University. In addition to being a professor at the University of Washington, Carlos is the CEO of Dato. The reason we suggest you use GraphLab Create for the few assignments where you will use pre-implement ML models is not because Carlos is the CEO of Dato :), but because we very strongly believe using this software will make it much easier for you to get those assignments done, and because most students are already familiar with it from the first courses.

We are happy, however, for you to use any tool(s) of your liking. As you will notice, we are only grading the output of your programs, so the specific software tool is not the focus of the course. More details on using other tools are at the end of this doc.

## Upgrade GraphLab Create

If you are using GraphLab Create and already have it installed, please make sure you upgrade to the latest version! The simplest way to do this is to:

1. Open the Dato Launcher.
2. Click on 'TERMINAL'.
3. On the terminal window, type:

```
pip install --upgrade graphlab-create
```

## Resources

These are some good resources you can explore, if you are using the recommended software tools:

- In the first course of this ML specialization, Machine Learning Foundations, we provided many tutorials and getting started guides. We recommend you go over those before tackling this course.
- There are many Python resources available online. Here is a good place for documentation.
- For SFrame & GraphLab Create, there is also a lot of information available online. Here are some starting points: the User Guide and detailed API docs.

- For Numpy, here is a getting started guide. We will also provide a tutorial when it’s time to use it.

## Installing the recommended software tools

If you choose to use the recommended tools, you have two options: downloading and installing the required software or using a prepackaged version on a free instance on Amazon EC2.

### Option 1: Downloading and installing all software on your own machine

Download and install Python, IPython Notebook, Numpy, SFrame and GraphLab Create. You can find the instructions here.

### Option 2: Using a free Amazon EC2 with all the software pre-installed

If you do not have a 64-bit computer, you will not be able to run GraphLab Create. Additionally, some of you may want a simple experience where you don't have to download the course content and install everything locally. Here, we'll address these situations!

Amazon EC2 offers free cloud computing hours with what they call micro instances. These instances are all we need to do the work for this course. We have created an image for one such instance that is easy to launch and contains all the course content. This will allow you to run everything you need for this course in the cloud for free, without having to install anything locally. (You do need to create an Amazon EC2 account and have internet access.)

You can find step-by-step instructions here:

https://dato.com/download/install-graphlab-create-aws-coursera.html

We note that installing all the software on your own local machine may be the right option for most people; especially since you can run locally everything without needing to be online to do the homeworks. But, the option using Amazon EC2 should be a great alternative.

## Github repository with starter code

In each module of the course, we have a reading with the assignments for that module as well as some starter code. For those interested, the starter code and demos used in this course are also available in a public Github repository:

https://github.com/learnml/machine-learning-specialization

## Using other software packages

We strongly encourage you to use the recommended software packages for this course, since they will allow you to learn the fundamental concepts more quickly. However, you are welcome to use others. Here are a few notes if you do so.

Installing other software tools

In the instructions above, you will be using the Dato Launcher, which will automatically install Python, IPython Notebook, Numpy, Matplotlib, SFrame and GraphLab Create. If you don't use the Dato Launcher, you will need to install each of these tools separately, by following the pages linked above. Anaconda is a good tool to help simplify some of this installation.

If you are using SFrame, but not GraphLab Create

GraphLab Create uses SFrame under the hood, but you can use just SFrame for most assignments. If you choose to do so, in the starter code for the assignments, you should change the line

```
import graphlab
```

to

```
import sframe
```

and everything should work with just some small modifications, e.g., the calls:

```
graphlab.SFrame(...)
```

will become

```
sframe.SFrame(...)
```

## If you are using other software tools out there

You are welcome to use other packages, e.g., scikit-learn instead of GraphLab Create, or Pandas instead of SFrame, or even R instead of Python. If you choose to use all these different packages, we will provide the datasets (in standard CSV format) and the assignment questions will not depend specifically on the recommended tools.