Software tools you'll need for this course

**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,](https://www.coursera.org/learn/ml-foundations) was focused on understanding how ML can be used in various cases studies, and the follow on courses will dig into the details of algorithms and methods for each of the main ML areas. ***We expect all learners to have taken the first course, before taking this course.***

**Regression - A Machine Learning Approach.** This course focuses regression, one of the most important types of data analysis, with a wide range of applications. After successfully completing this course, you will be able to use regression methods in practice, implement some of the most fundamental algorithms in this area, and choose the right model for your task. In addition, you will become proficient in core ML concepts that transcend regression settings: you will understand the practical implications of fundamental ML concepts, such as the bias-variance tradeoff, key ML data analysis techniques, such as cross validation, and the most important optimization techniques used to learn ML models, gradient descent and coordinate descent.

Programming assignment format

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.

In every module, you will be implementing a core regression technique or other ML concept from scratch. In modules where we have two programming assignments:

* the first will be focused on exploring fundamental concepts, such as regularization or cross-validation, with an existing implementation of the regression method, and
* the second will be focused on implementing the regression method for that module from scratch.

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 seldom used for production code in industry. It is possible, but discouraged to use R in this specialization.)

We will also encourage the use the Jupyter Notebook in our assignments. The Jupyter 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 Jupyter 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 fundamental ML concepts, such as cross-validation or the bias-variance tradeoff, 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 included as part of the Turi Create library. An alternative is the [Pandas](http://pandas.pydata.org/) 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.](https://turi.com/learn/translator/))

Tools for matrix operation

For matrix operations, we strongly recommend [Numpy](http://www.numpy.org/), 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](http://matplotlib.org/), 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 use [Turi Create](https://github.com/apple/turicreate), which we used in the first course, 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](http://scikit-learn.org/stable/). Turi Create is more scalable than scikit-learn and simpler to use when your data is not numeric vectors. Both Turi Create and scikit-learn is open-source. ***In this course, most of the assignments are about implementing algorithms from scratch, so this choice is more flexible than in the first course.*** 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.

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. We are happy to hear that so many of you are enjoying this approach so far!

Licenses for SFrame & Turi Create

Turi Create and the SFrame package are available in [open-source under a permissive BSD license](https://github.com/turi-code/SFrame). So, you will always be able to use Turi Create and SFrames for free. The reason we suggest you use Turi Create for this course is because this software will make it much easier for you see machine learning in action and to help you complete your assignments quickly.

Upgrade Turi Create

If you are using Turi Create and already have it installed (e.g., from the first course), please make sure you upgrade to the latest version! The simplest way to do this is to:

1. Open terminal
2. If you installed Turi Create in a virtual environment, activate that environment
3. Run # you only need to do this if you originally installed

#   Turi Create in a virtualenv (recommended)

cd <folder\_where\_you\_installed\_virtualenv>

source <virtualenv\_name>/bin/activate

# This line performs the upgrade

pip install --upgrade turicreate

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,](https://www.coursera.org/learn/ml-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](https://docs.python.org/3/).
* For SFrame & Turi Create, there is also a lot of information available online. Here are some starting points: the [User Guide](https://apple.github.io/turicreate/docs/userguide/) and [detailed API docs](https://apple.github.io/turicreate/docs/api/).
* For Numpy, here is a [getting started guide](https://docs.scipy.org/doc/numpy-1.15.1/user/quickstart.html). We will also provide a tutorial when it’s time to use it.

Installing the recommended software tools

Downloading and installing Python, Jupyter Notebook and Turi Create on your own machine

1. If you do not already have Python installed, download and install Python 3.6.7: <https://www.python.org/downloads/>. **Note: it is highly recommended that you use version 3.6.7. Installing the latest version of Python may not be compatible with Turi Create.**
2. Download and install Jupyter Notebook: <http://jupyter.org/install>. Follow the instructions for "Installing Jupyter with pip", use the commands under the section for Python 3
3. Download and install Turi Create: <https://github.com/apple/turicreate#installation>. **Note: it is not required that you use virtualenv, but it might be helpful, especially if you run into installation issues due to conflicting versions of software.**

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 other packages, e.g., [scikit-learn](http://scikit-learn.org/stable/) instead of Turi 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.