

# Introduction to Python and ML Libraries/Tools

As you may already know, we will be using the Python programming language where we'll implement all our code either via Python scripts (\*.py file extension) or in a Jupyter Notebook (\*.ipynb file extension). Additionally, for this semester, the libraries listed below are the crucial libraries we will be using throughout the semester. You will need to start becoming familiar with all these libraries/tools. **Please take some time to review the some of the following guides for any of the libraries/tools you are unfamiliar with (try to review the libraries in the order given).** Also, please make sure all these libraries are installed (if you installed Python via Anaconda then this shouldn't be a problem). You can easily check by using the command line (or **Anaconda prompt** ↗(https://stackoverflow.com/questions/47914980/how-to-access-anaconda-command-prompt-in-windows-10-64-bit) for Windows) by typing the following:

```
conda list <package name>
```

Other libraries, such as TensorFlow, will be introduced as the semester progresses. Don't worry about understanding every aspect of these libraries work right now. Simply aim to develop a general understanding of what the library is used for and begin taking a look at what their APIs look like. We will be using these libraries/tools in basically every lab so there is lots of time to practice ahead!

**Please know that the most crucial thing you can do now is to become familiar with all of the libraries documentation as you should be referring to it throughout the whole semester! We'll take a closer look at Python, NumPy, and Matplotlib in this weeks lab!**

## Python

As stated previously, Python is the language of choice for this class and the machine learning community at large. Make sure to install Anaconda with Python 3.8+. See the mini lab [install guide \(https://instructure.charlotte.edu/courses/234302/pages/mini-lab-anaconda-installation\)](https://instructure.charlotte.edu/courses/234302/pages/mini-lab-anaconda-installation) to learn how to do so. It is crucial you develop an understanding of basic Python in order to succeed in this class as we'll be programming every week!

"Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together." - [What is Python? Executive Summary](https://www.python.org/doc/essays/blurb/) ↗(https://www.python.org/doc/essays/blurb/)

- Guides
  - [What is Python?](https://www.python.org/doc/essays/blurb/) ↗(https://www.python.org/doc/essays/blurb/)
  - [Python Basics Guide](https://pythonbasics.org/) ↗(https://pythonbasics.org/)
  - [Python 3 Extensive Guide and Reference](https://docs.python.org/3/tutorial/) ↗(https://docs.python.org/3/tutorial/)
  - [Microsoft's Intro to Python Guide](https://docs.microsoft.com/en-us/learn/modules/intro-to-python/) ↗(https://docs.microsoft.com/en-us/learn/modules/intro-to-python/)
  - [Python Introduction by SciPy](https://scipy-lectures.org/intro/language/python_language.html) ↗(https://scipy-lectures.org/intro/language/python\_language.html)
  - [Inheritance in Python](https://realpython.com/inheritance-composition-python/) ↗(https://realpython.com/inheritance-composition-python/)
    - Make sure you understand the concept of inheritance as we will use this concept in our programming assignments and labs frequently! Most students tend to struggle programming in this course because this concept is not understood!

## Jupyter Notebook

Jupyter Notebook is the most popular literate programming tool that is compatible with diverse programming languages (over 40 languages including Python, R, Julia, and Scala). Jupyter Notebook's allow for interactive coding (i.e., [REPL](https://pythonprogramminglanguage.com/repl/) ↗(https://pythonprogramminglanguage.com/repl/)) which will help you explore code while also being able to summarize and write about your results.

- Guides
  - [Official Guide: Intro to Jupyter Notebook](https://jupyter-notebook.readthedocs.io/en/latest/notebook.html) (https://jupyter-notebook.readthedocs.io/en/latest/notebook.html)
  - [Brief Overview of Running Jupyter Notebook](https://docs.dea.ga.gov.au/notebooks/Beginners_guide/01_Jupyter_notebooks.html#Getting-started) (https://docs.dea.ga.gov.au/notebooks/Beginners\_guide/01\_Jupyter\_notebooks.html#Getting-started)
  - [\[Video\] Jupyter Notebook Tutorial](https://youtu.be/HW29067qVWk?t=209) (https://youtu.be/HW29067qVWk?t=209)
  - [\[Video\] Introduction to Jupyter Notebook in 5 minutes](https://www.youtube.com/watch?v=2eCHD6f_phE) ↗(https://www.youtube.com/watch?v=2eCHD6f\_phE)

## NumPy and Pandas

### NumPy

NumPy is a scientific library that focuses on the data object *ndarrays*. These arrays act as more sophisticated Python lists that allow for optimized mathematical operations to be performed such as an assortment of linear algebra computations. Learning how to create and manipulate NumPy ndarrays is one of the most important things as practically all our data will be stored implicitly or explicitly in ndarrays at somepoint.

"NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more."- [What is NumPy](https://numpy.org/doc/stable/user/whatisnumpy.html) ↗(https://numpy.org/doc/stable/user/whatisnumpy.html)

- Guides
  - [What is Numpy?](https://numpy.org/doc/stable/user/whatisnumpy.html) ↗(https://numpy.org/doc/stable/user/whatisnumpy.html)
  - [NumPy Quickstart](https://numpy.org/doc/stable/user/quickstart.html) ↗(https://numpy.org/doc/stable/user/quickstart.html)
  - [NumPy Basics for Beginners](https://numpy.org/doc/stable/user/absolute_beginners.html) ↗(https://numpy.org/doc/stable/user/absolute\_beginners.html)
  - [NumPy Quick Guide](https://www.tutorialspoint.com/numpy/numpy_quick_guide.htm) ↗(https://www.tutorialspoint.com/numpy/numpy\_quick\_guide.htm)
  - [NumPy Introduction by SciPy](https://scipy-lectures.org/intro/numpy/index.html) ↗(https://scipy-lectures.org/intro/numpy/index.html)
  - [\[Video\] NumPy in 5 minutes](https://www.youtube.com/watch?v=xECXZ3tyONo) ↗(https://www.youtube.com/watch?v=xECXZ3tyONo)

### Pandas

Pandas is a popular data science library that is built on top of NumPy. Pandas main focus is on loading datasets into objects called DataFrames which server to help quickly preprocess and neatly display datasets. While Pandas has many other uses, this is what we will mainly be using it for.

- Guides
  - [Getting started - Pandas](https://pandas.pydata.org/pandas-docs/stable/getting_started/index.html) ↗(https://pandas.pydata.org/pandas-docs/stable/getting\_started/index.html)
  - [10 minute introduction to Pandas](https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html) ↗(https://pandas.pydata.org/pandas-docs/stable/user\_guide/10min.html)
  - [NumPy vs Pandas](https://cloudxlab.com/blog/numpy-pandas-introduction/) ↗(https://cloudxlab.com/blog/numpy-pandas-introduction/)
  - [\[Video\] What is Pandas in 10 minutes](https://www.youtube.com/watch?v=dcqPhpY7tWk) ↗(https://www.youtube.com/watch?v=dcqPhpY7tWk)

## Matplotlib

Visualization is key to observe your data and analyze the experimental results produced by ML model. Matplotlib (and other plotting libraries such as Seaborn or Plotly) help you to produce the graphs and plots so you can easily observe trends in your data and results.

"Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python."

- Guides
  - [Introduction to Matplotlib Official Guide](https://matplotlib.org/stable/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py) ↗(https://matplotlib.org/stable/tutorials/introductory/usage.html#sphx-glr-tutorials-introductory-usage-py)
  - [Matplotlib PyPlot Tutorial Official](https://matplotlib.org/stable/tutorials/introductory/pyplot.html) ↗(https://matplotlib.org/stable/tutorials/introductory/pyplot.html)
    - PyPlot is a collection of functions that make Matplotlib work like MATLAB. This is the more script focused API while normal Matplotlib (using the Figure and Axes classes) uses a more object oriented approach. Frequently you'll find we use both!
  - [Difference between Matplotlib object oriented approach and the PyPlot approach](https://matplotlibblog/posts/pyplot-vs-object-oriented-interface/) ↗(https://matplotlibblog/posts/pyplot-vs-object-oriented-interface/)
  - [Matplotlib Intrductionioin by SciPy](https://scipy-lectures.org/intro/matplotlib/index.html) ↗(https://scipy-lectures.org/intro/matplotlib/index.html)
  - [Examples of different plots](https://matplotlib.org/stable/tutorials/introductory/sample_plots.html#sphx-glr-tutorials-introductory-sample-plots-py) ↗(https://matplotlib.org/stable/tutorials/introductory/sample\_plots.html#sphx-glr-tutorials-introductory-sample-plots-py)
  - [\[Video\] How to use Matplotlib in 3 minutes](https://www.youtube.com/watch?v=D4VimL3G4_o) ↗(https://www.youtube.com/watch?v=D4VimL3G4\_o)

## SciPy and Sklearn (Scikit-learn)

### SciPy

SciPy is probably the most general and extensive library out of all the libraries covered so far. SciPy implements a verity of mathematical algorithms which work in tandem with libraries like NumPy and Matplotlib. There is too much to cover when it comes to Scipy so just try to gain a general understanding of what SciPy is and become similar with how it is organized. Scipy isn't as important so there will be limited references below.

"SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data."

- Guides
  - [SciPy Introduction](https://docs.scipy.org/doc/scipy/reference/tutorial/general.html) ↗(https://docs.scipy.org/doc/scipy/reference/tutorial/general.html)
  - [Overview of SciPy](https://scipy-lectures.org/intro/scipy.html) ↗(https://scipy-lectures.org/intro/scipy.html)

### Sklearn (Scikit-learn)

Sklearn is also collection of advanced machine-learning algorithms for Python that is built upon NumPy and SciPy. Sklearn contains many algorithms we will be seeing throughout the semester. While we will be programming all the algorithms in this class by hand (for the most part), we can use Sklearn to check our algorithms are working correctly by comparing predictions/outputs!

"Scikit-learn is a machine learning library that provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python."

- Guides
  - [Introduction to Scikit Learn](https://scikit-learn.org/stable/tutorial/basic/tutorial.html) ↗(https://scikit-learn.org/stable/tutorial/basic/tutorial.html)
  - [Scikit Learn Tutorials](https://scikit-learn.org/stable/tutorial/index.html) ↗(https://scikit-learn.org/stable/tutorial/index.html)
  - [\[Video\] Intro to Sklearn in 5 minutes](https://www.youtube.com/watch?v=rvVkVsG49uU) ↗(https://www.youtube.com/watch?v=rvVkVsG49uU)

**Each Python library is super comprehensive, and you need around one week for each library if you really want to master it. You don't have enough time to learn everything for sure. So my suggestion is: to get a high-level idea of each tool/library, and search the specific command/function when it's needed in coding.**

Check the Python cheatsheets for [Numpy, Pandas, Matplotlib, and Sk-Learn](https://web.itu.edu.tr/iguzel/files/Python_Cheat_Sheets.pdf) ↗(https://web.itu.edu.tr/iguzel/files/Python\_Cheat\_Sheets.pdf).