* [Programming Resources](#gjdgxs)
  + [Python](#30j0zll)
  + [Pandas](#1fob9te)
  + [NumPy](#3znysh7)
  + [SciKit Learn](#2et92p0)

# CSE 450 - Machine Learning & Data Mining

## Programming Resources

## Programming Resources

The assignments in this course require a fair amount of Python coding, as well as the use of a few popular Python-based data science tools.

Agency Based Education

It's expected that you'll either know some Python coding coming in to this course, or be able to quickly come up to speed largely on your own. While you don't need to be an expert, or even highly skilled, the more you know about Python coding, the better off you'll be.

In addition, we'll be using a variety of machine learning and data science libraries and frameworks that you may or may not have experience with.

We don't spend much time in this class explicitly learning Python or these tools, for a few reasons:

* In industry, you'll often be asked to complete a task or project that requires quickly coming up to speed on a new technology or software library that you've never used before.
* By the time you start your first job, many of the tools and libraries will be different anyway, so we try to focus on teaching principles and helping you "learn how to learn" on your own so you can keep up with this rapidly changing field.
* As a 400-level student, we expect that at this point you have enough diligence and experience reading technical documentation and tutorials that you can handle this largely on your own.

However, we have put together this guide, as well as [a separate guide with tips for reading technical documentation](https://byui-cse.github.io/cse450-course/course/reading-technical-documentation.html) that you may find useful.

Python 2 vs 3

Python went through a major revision a few years ago. In this course, we use Python 3. You may find tutorials for (or already know) Python 2.

For the purposes of this course, there really aren't that many differences you have to think about.

You can see [a good summary of the most important differences here](https://learntocodewith.me/wp-content/uploads/2014/06/python-2-vs-3-2018.png).

### Python

Google tells me there are 387 Million hits for Python Tutorial, so obviously you have a lot of options, from [the official Python tutorial](https://docs.python.org/3/tutorial/) to [this 11 hour video course](https://www.youtube.com/watch?v=WGJJIrtnfpk).

Students often wonder which tutorial is *the best* for a given subject. Unfortunately, there is no such thing.

Some students learn better from books, others from websites, and some prefer videos. Some students want interactive tutorials, others feel like they can only learn if they take a class.

Here are some Python tutorials that I think are good for students:

* [Codecademy's Python Course](https://www.codecademy.com/catalog/language/python)
* [Google's Python Course](https://developers.google.com/edu/python/)
* [Learn Python the Hard Way](https://learnpythonthehardway.org/python3/)
* [How to Think Like a Computer Scientist](http://openbookproject.net/thinkcs/python/english3e/)

### Pandas

Pandas is a data science library that makes it easy to do common data manipulations and analysis. We'll be using it quite a bit in this course.

There aren't quite as many Pandas tutorials as there are Python tutorials, (only 16 Million hits for Pandas Tutorial).

Here are some Pandas tutorials that I think are good for students:

* [Pandas Official Getting Started Tutorials](https://pandas.pydata.org/docs/getting_started/intro_tutorials/index.html)
* [10 Minutes to Pandas](https://pandas.pydata.org/docs/user_guide/10min.html)
* [Google's Colab-based, Ultraquick Pandas Tutorial](https://colab.research.google.com/github/google/eng-edu/blob/master/ml/cc/exercises/pandas_dataframe_ultraquick_tutorial.ipynb)
* [Kaggle's Pandas Tutorial](https://www.kaggle.com/learn/pandas)

### NumPy

NumPy is a numeric computation library designed to allow Python to carry out super-optimized matrix algebra operations. It'll be rare in this course that we need to use NumPy directly, but many libraries (including Pandas and scikit-learn) are built on top of NumPy, so knowing something about it can sometimes be handy.

Here are some NumPy tutorials that I think are good for students:

* [NumPy Official Getting Started Guide](https://numpy.org/devdocs/user/absolute_beginners.html)
* [NumPy for Absolute Beginners](https://numpy.org/doc/stable/user/absolute_beginners.html)
* [Google's Colab-based, Ultraquick NumPy Tutorial](https://colab.research.google.com/github/google/eng-edu/blob/master/ml/cc/exercises/numpy_ultraquick_tutorial.ipynb)

### SciKit Learn

SciKit Learn is a machine learning library that we'll be using quite a bit in this course. It makes heavy use of both Pandas and NumPy.

My advice for this library is to get an overview of how it works, and especially learn about the pipeline functions. Then as you need to use each algorithm, read the details about that algorithm in [the User Guide](https://scikit-learn.org/stable/user_guide.html#) and [API manual](https://scikit-learn.org/stable/modules/classes.html).

* [Getting Started Guide](https://scikit-learn.org/stable/getting_started.html)
* [Working With Text Data](https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html#)
* [Some hands on SciKit-Learn Tutorials](https://scikit-learn.org/stable/tutorial/index.html)
* [The User Guide](https://scikit-learn.org/stable/user_guide.html#)