CS5785 Homework 0

Overview

Welcome to CS 5785! After completing this homework, you should be able to find a teammate, set up your preferred development environment, download and parse a dataset, and use visualization tools to help you understand what that dataset contains. Completing this homework will give you the scaffolding you need for the rest of the course.

This homework is due on **Feb 1, 2017 at 2:30 PM EST**, just before class. Upload your homework to CMS. Please upload code as a single `.zip` file and the writeup as a single `.pdf` file. A complete submission should include:

1. A write-up as a single `.pdf` file
2. Source code and data files for all of your experiments (AND figures) in `.py` files if you use Python or `.ipynb` files if you use the IPython Notebook. If you use some other language, include all build scripts necessary to build and run your project along with instructions on how to compile and run your code.

The write-up should be in **professional lab report format.** It should contain a general summary of what you did, how well your solution works, any insights you found, etc. On the cover page, include the class name, homework number, and team member names. You are responsible for submitting clear, organized answers to the questions.
Please include all relevant information for a question, including text response, equations, figures, graphs, output, etc. If you include graphs, be sure to include the source code that generated them. Please pay attention to the discussion board for relevant information regarding updates, tips, and policy changes. You are encouraged (but not required) to work in groups of 2.

If you need help

There are several strategies available to you.

- If you get stuck, we encourage you to post a question on Piazza.[1] That way, your solutions will be available to the other students in the class.
- Your TAs will offer office hours, which are a great way to get some one-on-one help.
- You are allowed to use well known libraries such as `scikit-learn`, `scikit-image`, `numpy`, `scipy`, etc. in this assignment. Any reference or copy of public code repositories should be properly cited in your submission (examples include Github, Wikipedia, Blogs).

[1] http://piazza.com/cornell/fall2016/cs5785

1 SUBMITTING HOMEWORK

All homework must be submitted via CMS, Cornell's course management system. Instructions for using CMS can be found on this semester's Piazza page, https://piazza.com/cornell/fall2016/cs5785/home.
We encourage Piazza for all homework-related discussion. If you have a question, *please do not E-mail the TAs directly*. Rather, post your question to Piazza so all students can benefit!

1. **Enroll on Piazza** and read the CMS submission instructions.
2. Please **read and follow the formatting guidelines** on the preceding page while preparing your homework.

2 SETTING UP PYTHON

1. **Find your teammate.** You are encouraged (but not required) to work in groups of 2. If you do decide to work in a team, inlcude both teammates' names on the report and in your submission email. One good way of finding teammates is to check the "*Search for Teammates!*" post on Piazza.
2. **Set up a working environment for machine learning.** You are free to use whatever language you prefer, but the TAs will only support Python.

   To set up Python on your machine, download the "Anaconda" distribution, which includes Python and several libraries that we will use in class. It is available for Windows, Linux, and Mac OS X here: http://continuum.io/downloads

   After installing Anaconda, you can develop in Python in one of many ways:

   - Anaconda includes the "IPython Notebook," which provides a web-based interface to write and run Python code. Data scientists like IPython Notebook because a notebook can include descriptions, comments, graphs, and figures in-line with the source code that generated them.

     To start the IPython Notebook, run the Anaconda "Launcher" and click "IPython Notebook." This will open up the Notebook server in a terminal and a web browser. You can then create a notebook and add Python code.
   - If you instead prefer to develop in a traditional more familiar IDE, Anaconda includes "Spyder". There are also instructions on how to point Eclipse to your new Python installation here: http://docs.continuum.io/anaconda/ide_integration

3 IRIS FLOWERS

In 1935, Edgar Anderson went to his favourite pasture and recorded the length and width of the sepals and petals on several flowers in the field. For whatever reason, this dataset became one of the oldest and most well-known "sanity-check" datasets around, being cited by countless papers. This class continues this time-honored tradition by using *Iris Flowers* to sanity-check your Python environment and plotting libraries.

1. Find and download the Iris Flowers dataset from the UC Irvine Machine Learning datasets archive at http://archive.ics.uci.edu/ml/datasets.html Hint: The `iris.names` file describes the structure of the dataset. How many features/attributes are there per sample? How many different species are there, and how many samples of each species did Anderson record?

2. Figure out how to parse the dataset you downloaded. Load the samples into an $N \times p$ array, where N is the number of samples and p is the number of attributes per sample. Aditionally, create a N-dimensional vector containing each sample's label (species).

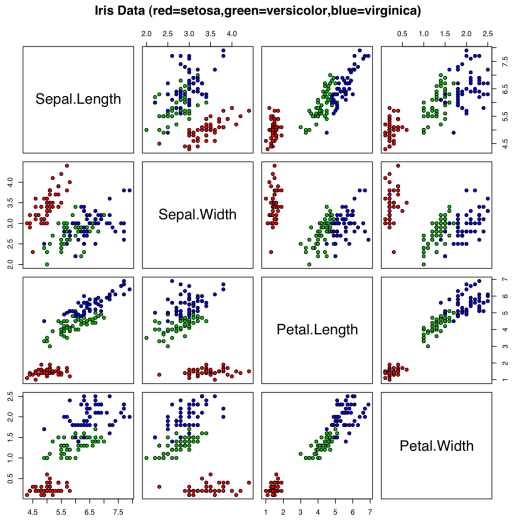
   Hint: Python has a built-in CSV parser in the `csv` library, or you can use the `"string".split(...)` method.

   Hint 2: Here is some code that prints each line in a file:

   ```
   for line in open("/path/to/filename.txt"):
       print "Line contains: "+line
   ```

3. To visualize this dataset, we would have to build a p-dimensional scatterplot. Unfortunately, we only have 2D displays so we must reduce the dataset's dimensionality. The easiest way to view the set is to plot two attributes of the data against one another and repeat for each pair of attributes.

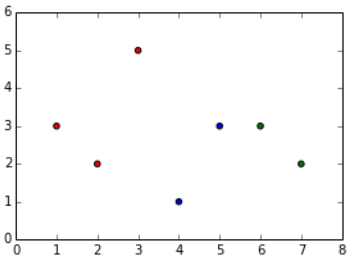
   Create every possible scatterplot from all pairs of two attributes. (For example, one scatterplot would graph petal length vs sepal width, another would graph petal length vs. sepal length, and so on). Within each scatterplot, the color of each dot should correspond with the sample species. Ideally, we're looking for something like this figure from Wikipedia:



   But your results do not have to be this ornate. Presenting six separate figures in your report is certainly fine. Be sure to include the source code for all plots!

Hint: This is one way to draw a scatterplot. Use whatever works for you.

```
from matplotlib import pylab as plt
import numpy
xs      = numpy.array([1, 2, 3, 4, 5, 6, 7])
ys      = numpy.array([3, 2, 5, 1, 3, 3, 2])
colors = ["r","r","r","b","b","g","g"]
plt.scatter(xs, ys, c=colors)
plt.savefig("plot.png")
```



Hint: If you would like plots to appear right inside of your IPython Notebook, restart the kernel and evaluate the following before running anything else:

```
%pylab inline
```

Good luck!