CS5785 / ORIE5750 / ECE5414 - Homework 0

Overview

Welcome to CS 5785 / ORIE 5750 / ECE 5414! After completing this homework, you should be able to set up your python and Jupyter Notebook 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 **Wednesday, August 30, 2023 at 11:59PM**. You have **up to 2 slip days** for late submission per assignment, and a total of 4 slip days during the entire course. Upload your solution to Gradescope. Your submission will have:

1. Source code and data files for all of your experiments (AND figures) in `.ipynb` files (file format for IPython Jupyter Notebook). These files should be placed in a folder titled `hw0` and uploaded to the `hw0-code` assignment in Gradescope.

If you include graphs, be sure to include the source code that generated them. Please pay attention to Canvas for announcements, policy changes, etc. and Piazza for homework-related questions.

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, students can help each other and instructors can provide feedback and support.
- The professor and your TAs will offer office hours, which are a great way to get some one-on-one help. You can help us select the best times for office hours by completing the Canvas Survey titled "Office Hours: Select All That Works For You" and can be found under the Quizzes tab.
- 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 Submitting Homework

All homework must be submitted via Gradescope[2]. The link to our Gradescope is available on Canvas, under the "Gradescope" tab. We encourage Piazza for all homework-related discussion. If you have a

[1] https://piazza.com/cornell/fall2023/cs5785orie5750ece5414
[2] https://www.gradescope.com/

question, *please do not E-mail the TAs directly*. Rather, post your question on Piazza so all students can benefit!

2 Setting up Python

You can find detailed instructions on installing Python and Jupyter, for either Mac and Windows by accessing the Canvas webpages titled titled "Mac Setup" and "Windows Setup" which can be found under the "Home" tab, in "Week 1".

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 https://archive.ics.uci.edu/ml/datasets/iris 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).

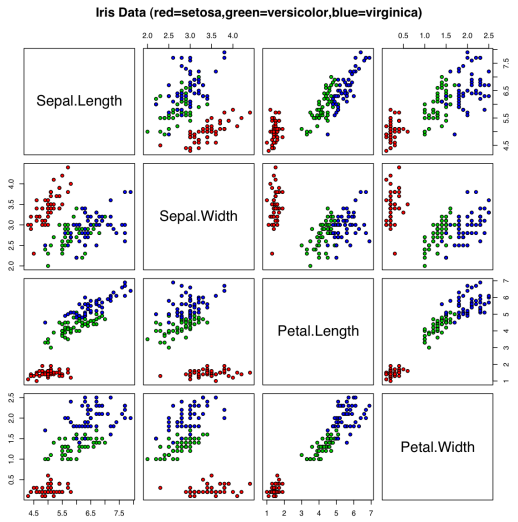
   Hint: Python has a built-in CSV parser in the `csv` library, or you can use the `"string".split(...)` method.

   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.

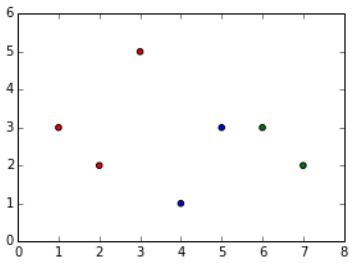
   Create every possible scatterplot from all pairs of two attributes. (For example, one scatterplot would graph petal length vs sepal width, another would graph petal length vs. sepal length, and so on). Within each scatterplot, the color of each dot should correspond with the sample species. Ideally,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



But your results do not have to be this ornate. Presenting 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 pyplot 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 Jupyter Notebook, restart the kernel and evaluate the following before running anything else:

```
%matplotlib inline
```

Good luck!