Northeastern University, College of Computer and Information Science

CS 6220 Data Mining — Assignment 0
Due Date: Month 0th, 0000 at 11:59pm ET

Setting-up your programming environment

Month 00, 0000

This assignment requires you to setup, in your local machine, the programming environment that you will be using for the remainder of the course. All of the topics that we'll cover throughout the semester have readily available implementations in Python that are simple to use and quite efficient. Because it is a lot more fun to use great code that has been written and tested than to re-invent the wheel, we encourage you to leverage as best as you can all of these great Python modules. Reading and processing datasets, training and evaluating models, and plotting the results of complex analysis requires a lot of interaction with your code, and to that end we will be relying very heavily on Jupyter (former IPython) Notebooks. This tools provides you the ability to write Python code in your browser using a phenomenal interface that allows you to create graphs, add images and videos, include markdown content, and a lot more. It is also very popular among professional data scientists and an overall great skill to have.
The three steps involved in this introductory assignment are:

1. Install Anaconda Python locally
2. Install some of the modules that we will be using
3. Create a test Notebook file to ensure everything went well

1 Installing Anaconda

Anaconda greatly simplifies things when working with data (and everything else) in Python. It is an open-source distribution of Python specifically designed for large-scale data processing, predictive analytics, and scientific computing. It also comes pre-loaded with solutions for package management, allowing us to extend its functionally by easily installing new external modules.

Installing Anaconda is quite simple regardless of the operating system you are using. You can follow the instructions here to get yourself setup. We recommend using Python 3, but you are also free to use version 2 if you prefer.

Additional instructions on that process, as well as some information on how to create your first IPython Notebook can be found on Lesson 1 here. Take a couple of minutes and watch those two short videos.

2 Installing modules

While you will likely be installing other packages throughout the course, there are a few that will be used frequently. These are listed below.

- NumPy
- SciPy
- Matplotlib
- pandas
- IPython
- scikit-learn

Install the above modules by going to your terminal (cmd on windows) and typing:

```
conda install numpy
```

Repeat that for the other modules. This will use Anaconda's package manager (conda) to complete the installation process. Depending on what features of Anaconda you installed, you may also have access to a graphic interface to conda, which you are free to use if you prefer that over running terminal commands.

3 Create a test Notebook

To make sure your environment is correctly setup and running, follow the instructions given in the *Writing and running Python in the iPython notebook* video from Lynda.com (link given in section 1) to create your first Notebook file.
Use the code snippets given in *Listing 1* to test the modules you just installed. Your notebook will ideally look like here this.
Finally, add a couple of personal touches to it. It can be anything. You can change the plot colors, add an image to your notebook file, a cat picture, include some comments or markdown. Use your creativity. Save your notebook file and submit that as your solution.

Listing 1: Sample Python code for testing required modules

```
# Testing pandas
%pylab inline
import pandas as pd
ts = pd.Series(np.random.randn(1000),
    index=pd.date_range(’1/1/2000’, periods=1000))
ts = ts.cumsum()
ts.plot()

# Testing NumPy
import numpy as np
np.arange(15).reshape(3, 5)

# Testing SciPy
import scipy as sp
sp.linspace(0, 10, 5000)

#Testing matplotlib
import matplotlib.pyplot as plt
x = np.linspace(0, 1)
y = np.sin(4 * np.pi * x) * np.exp(-5 * x)
plt.fill(x, y, ’r’)
plt.grid(True)
plt.show()

# Testing Scikit Learn
from sklearn.svm import SVC
from sklearn.datasets import load_digits
from sklearn.feature_selection import RFE

# Load the digits dataset
digits = load_digits()
X = digits.images.reshape((len(digits.images), -1))
y = digits.target

# Create the RFE object and rank each pixel
svc = SVC(kernel="linear", C=1)
rfe = RFE(estimator=svc, n_features_to_select=1, step=1)
rfe.fit(X, y)
ranking = rfe.ranking_.reshape(digits.images[0].shape)

# Plot pixel ranking
matshow(ranking)
colorbar()
title("Ranking of pixels with RFE")
show()
```