**Assignment 1– Introduction to Python for Machine Learning**

***Assignment overview.*** This assignment is designed to introduce you to coding in Python for the purposes of writing machine learning code. Please follow the steps as indicated and complete the tasks outlined below. You are expected to figure out some details of syntax by consulting Python’s Help (command?). Since the material builds on each other, it is recommended to follow the exercises in order. Please answer each question as needed by copying and commenting on the material that you produce in the IPython console as well as all scripts you are asked to create, or use Jupyter Notebook and download it as a Python file.

***Submission.*** Create a folder called ML\_Assignment1 and put all the files inside the folder. Compress this folder to create either ML\_Assignment1.zip or ML\_Assignment1.rar. Submit this compressed folder as your assignment submission on Brightspace.

***Submission deadline:*** Thursday, 14 Sep, 10:00 pm.

***Late submission policy:*** Submissions after the due date are penalized by a 10% grade reduction per day.

***Academic Integrity:*** Dalhousie academic integrity policy applies to all submissions in this course. You are expected to submit your own work. Please refer to and understand the academic integrity policy, available at <https://www.dal.ca/academicintegrity>

***Python:*** We will be using Python for the programing exercises based on scientific Python libraries like

* [**numpy**](http://www.numpy.org/) - mainly useful for its N-dimensional array objects.
* [**matplotlib**](http://matplotlib.org/) - 2D plotting library producing publication quality figures.

Moreover, the following link covers the required materials:

* [**Scipy Lecture Notes**](http://www.scipy-lectures.org/), by Gaël Varoquaux, Emmanuelle Gouillart, and Olav Vahtras

***If you have question:*** Teaching Assistants (TAs) will be present during the labs to help you with any questions you may have. If you still have questions, feel free to email me at [tt@cs.dal.ca](mailto:tt@cs.dal.ca).

**Questions:**

1. **[90 marks]** This question requires you to write a Python script file called sol1.py. Perform the operations below on the fisher-iris dataset by entering commands in the IPython console. (The fisheriris.data file gives Ronald Fisher's measurements of type, petal width (PW), petal length (PL), sepal width (SW), and sepal length (SL) for a sample of 150 irises. The lengths are measured in centimeters. Type 0 is Setosa; type 1 is Verginica, and type 2 is Versicolor. For more information, please read the fisheriris.txt file).
   1. **[8 marks]** Please download the fisheriris.zip file and extract it to the directory for this assignment. Load the ***iris*** dataset from the fisheriris.data file and place them in an array iris\_data (Hint: You can use *numpy.loadtxt* function). Load the attributes from the attributes.txt file and place them in a list feature\_names. Then load the feature names from the feature.txt file and place them in a vector target\_names. (Hint: You can use *numpy.genfromtxt* and *numpy.ndarray.tolist* functions.) **What** is the difference between *loadtxt* and *genfromtxt* functions?
   2. **[10 marks]** Pull out the column related to the class from the irirs\_data array and place it in an integer vector named *target*. Then remove the class column from the irirs\_data array (hint: you can use *numpy.delete* function). Remove the ‘Class’ from the feature\_nameslist*.* Find the size of the Fisher's measurements. Show the number of elements of the second dimension of the irirs\_data array. Capture the output to show your work.
   3. **[8 marks]** Find the sums of each of the columns in the irirs\_data array. Find the sums of just the second and the fourth columns. Pull out samples (rows) 27 through 48 and find the maximum values in each of the four columns of this set of samples. Pull out the odd-numbered samples (rows) between the third and 33rd rows and find the minimum values in the first two columns. Capture the output to show your work. Hint: you can use *numpy.ndarray* functions.
   4. **[6 marks]** Add the first and the third columns and place the results in a vector r13rd, then sum up r13rd. Compute the value each value of r13rd cubed
   5. **[8 marks]** Extract the first 4 rows of the first two columns as mat1 and first 4 rows of the last two columns as mat2. Add mat1 to mat2. Multiple the respective elements of mat1 and mat2. Capture the output to show your work.
   6. **[3 marks]** Calculate the inner product of mat1 and mat2 and store it in mat3. Write mat3 out in CSV format to a file call mat3.CSV (hint: you can use *numpy.savetxt* function).
   7. **[7 marks]** Use a bar plot to show the mean and standard deviation of each value of Fisher's measurements. Hint: use *matplotlib.pyplot* module. Add proper axis label and title for the plot.
   8. **[10 marks]** Plot the three species class examples as different colour points in a scatter plot as a function of the first two variables in the data array. Add axis labels and legend for the scatter plot using the target\_names and feature\_names. Add a proper title for the plot.
   9. **[15 marks]** Using a for loop and *matplotlib.pyplot.subplot* and *itertools* modules, plot 6 scatter plots of all combinations of two features in one plot. Hint: you should adjust proper blank spaces between subplots.
   10. **[10 marks]** Random sampling is a very important core technique in machine learning. Look up seed, RandomState, rand, randn, and randint functions in *numpy.random* module to briefly **explain** about the properties and capabilities of these commands. **Explain** a scenario that you must use RandomState before the random generator function. Create a 5 x 5 matrix call cent consisting of random values drawn from the normal distribution. **Explain** why there are negative values in cent. Create another 5 x 5 matrix call cent\_fix consisting of random values drawn from the normal distribution with fix pseudo random number using RandomState method. Try different seeds and check the values of the cent **and** cent\_fix arrays and capture the output to show your work.
   11. **[5 marks]** Generates a row vector of 5 random numbers drawn from the uniform distribution using rand() and assign the vector to a variable v. Then draw another 5 random numbers and concatenate them as additional column values to v, and display the new v.
2. **[10 marks]** This question requires you to write and run a second script called sol2.py.
   1. **[3 marks]** Create a function called forever that times the multiplication of two matrices of rank 500 consisting of random numbers using “for” loops. Output the time.
   2. **[3 marks]** Create a function called mat\_fast that times the multiplication of two matrices of rank 500 consisting of random numbers using a *numpy.ndarray* function. Output the time.
   3. **[2 marks]** Calculates and prints the difference between the computation times of the two functions.
   4. **[2 marks]** Run the script and capture the output to show your work.