Project 1 : La Vie En Rose

Due 2/22/2019 @ 12pm (noon)

TOPICS:

- Python/IPython
- Perceptron
- Adaline
- Gradient Descent Algorithm
- Training/Test Sets
- Standardizing Data
- NumPy, Sci-kit Learn, Matplotlib

BACKGROUND:
During class, we discussed how to analyze data to make a prediction based on a set of features using a perceptron and Adaline. Using algorithms such as gradient descent, one can obtain an optimal set of parameters to show relationships between variables.

DIRECTIONS:
In this project, you will use a perceptron and Adaline and analyze the results on a Breast Cancer Wisconsin (Diagnostic) dataset: (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)) to predict whether breast tissues are benign or malignant.

You will implement both perceptron and Adaline using scikit-learn starting from a blank Jupyter Notebook. The implementation of the perceptron is given in the book, and I will leave it up to you to figure out how to implement Adaline (hint: SGDClassifier). You may use any three (3) features in the dataset to make a prediction after standardizing the data and splitting them into training and test sets. You will calculate 1) the number of misclassifications and 2) the accuracy. *It is highly encouraged that you try to obtain the lowest misclassifications and highest accuracy as a matter of personal pride.*

I will leave it up to you to choose an appropriate fraction of training/test sets, learning rate, tolerance threshold, and number of iterations. In fact, you must choose at least several of each for full credit (hint: use a list and a loop). All other parameters for these implementations must be the same as the definition we discussed in the textbook.

IMPLEMENTATION NOTES:
Any program that does not execute completely without errors will not be graded.

COMMENTS AND STYLE:
Although there will be no formal policy on commenting and style, the reader should able to easily follow the main purpose of the code. Each set of code that does something significant must be commented. The variable names should be easily recognizable and acronyms should be avoided if possible.

Do not be surprised if help is not forthcoming if your code is poorly commented and/or difficult to follow. You have been warned.

PROJECT SUBMISSION:
You will turn in the modified IPython notebook with a different set of features.

The programs and graphs should be in a single directory named "BC". The contents of the directories must be archived in a tarball that is gzipped called Proj1.tar.gz.

Place the gzipped tarball in your Drop Box on Sakai before it is due.

PLEDGED WORK POLICY:
Assignments in Computer Science courses may be specified as "pledged work" assignments by the professor of the course. When an assignment is specified as "pledged work" the only aid that the student may seek is from either the course professor or TAs (including CS Center tutors) that the professor has explicitly specified. On "pledged work" assignments the student may not use the services of a tutor.

For this project, you and your partner will develop code together into shared repositories, so you will see and share work. In addition, you may discuss only **basic programming language syntax** and general computer science concepts with everyone else. Any other communications of the project (e.g., giving your code to someone else or seeing someone else's code) are strictly prohibited except with the professor and TAs of the course. Your code and your implementation of the project must be the product of your own work and that of your partner.