# ID5030 Assignment 2

## Logistic Regression and Classification

Due 12/02/2018

Estimated Time : 6-8 hours

(NOTE: The assignment is spread over two weeks)

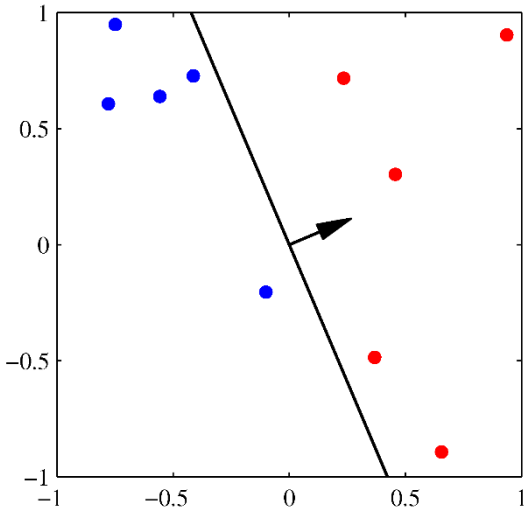
This assignment involves programming logistic regression using gradient descent from ground up and testing it on a given data set. We have broken up the problem into several parts. Answer all of them and include a short report within a .docx file as with Assignment 0 and 1. Our process for this assignment is as follows;

1. **Code generation**: We write a general purpose code and create some test cases on our own.
2. **Benchmarking**: We try out our code on a standard, benchmark problem – Digit Classification
3. **Real Life Deployment**: We now employ our code on a real life, brain tumour patient's survival prediction problem.

We will do the first part two parts this week and the real test case in the next assignment.

**Initial work** : Review class notes and your Assignment 1 submission. The same preliminaries work here as well.

1. Logistic Regression with gradient descent: Implement logistic regression **using batch or stochastic gradient descent** in Python for linearly separable data sets like the one given below. Have a strategy for systematically testing your code.




   a. **Simplest case – Linearly Separable**: Start with generating code and data for a two-class, linearly separable, 2 feature classification problem. The final code should take in two inputs (say $x_1$ and $x_2$) and give out the class. Carry out the following steps (**Estimated Time** : 1 hr, if you cleverly reuse Assignment 1)
      i. Create your own classification boundary (similar to how you generated the best fit line in Assignment 1) by assuming arbitrary weights

ii. Generate random data that lie on each side of this line. Ensure that you generate roughly the same number of data points for each class
iii. Create the corresponding ground truth label
iv. Using the binary cross-entropy loss function write a gradient descent based logistic regression classifier. Test it for various classification boundaries by modifying the weights in part i. Note that this part of the code can be written quickly if you modify your gradient descent code from Assignment 1.

**b. Nonlinear features:** (**Estimated Time**: 1 hr) Modify your dataset and code to account for nonlinearly separable data. For this exercise, assume that your classification boundary is an ellipse. Repeat all the steps of part (a) for this case

**c. Report**: Write a short report for this part including the following: (**Estimated Time**: 45 mins)
i. Your most general code
ii. Why you are sure your code works. That is, what test cases did you use and why are they general? Show some representative plots for both the linear and non-linear case
iii. What is the major modification that is required for the nonlinear case over the linear one?
iv. Which one of batch or stochastic gradient descent do you think will work better here? Why?

2. Benchmark Case – Binary Digit classification. The aim is to classify images of digits from the MNIST dataset as either a '0' or a '1'. The MNIST dataset can be downloaded from the following website. http://yann.lecun.com/exdb/mnist/
i. **Loading Images**: Write a program to load the images. Note that the images of all the digits from 0 to 9 will be present in this set. Find a way to select only those digits for your dataset that are either labelled as '0' or '1'. (**Estimated Time**: 30 min)
ii. **Conversion of input to an appropriate format**: The logistic regression algorithm you would have written would recognize only inputs expressed as vectors. The images given will expressed as matrices. Unroll these matrices into vectors so that they can act as inputs for the logistic regression code. (**Estimated Time**: 30 min)
iii. **Classification:** Use the logistic regression code developed and tested in part 1 to classify the images as belonging to class 0 or class 1, corresponding to digit 0 and digit 1 respectively. (**Estimated Time**: 1 hours)
iv. **Normalization**: Repeat the above exercise by rescaling the input values so that they always lie between 0 and 1. (**Estimated Time**: 1 hours)

v. **Report**: Write a short report for this part including the following: (**Estimated Time**: 45 mins)
    1. Your most general code
    2. Why you are sure your code works. That is, what test cases did you use and why are they general?
    3. Report your accuracy of digit classification both without and with normalization.