# Fundamentos de Data Science

## LAB N. 5

*Note that the instructions below not very "prescriptive" on the exact steps that you should carry out in this lab. It is important that you think by yourself about what you should do in order to perform a meaningful analysis of the data.*

In this lab, you will implement a Fisher Linear Discriminant, the Perceptron Algorithm and Logistic regression. For logistic regression you will implement both the stochastic gradient descent as well as the Iterative Least Squares Algorithm.

**DATASET DESCRIPTION**

You will be using 3 different datasets, that you will find on the Eclass page for this lab:

1) *FisherData.mat*: this file contains 3 different artificial datasets of points in 2D (data1, data2, data3). Each of these datasets is a cell array, and contains data from 2 different classes in its two cells.

2) *PerceptronData.mat*: this dataset is constituted by 18 points, in 2D, of which 10 in the "red class" and 8 in the "blue" class. The points are linearly separable.

3) *Occupacy* dataset: points are in 5 dimensions. Each point refers to a room, which is described by 5 attributes (first 5 columns). Each room belongs to one of possible 2 classes (last column). Note that for your convenience you will find a training set as well as 2 test sets.

   The description of the attributes is as follows:

   Temperature, in Celsius
   Relative Humidity, in percentage
   Light, in Lux
   CO2, in ppm
   Humidity Ratio, in kgwater-vapor/kg-air

   The classes to predict are 0/1 denoting the room occupancy, that is: was a person in there or not? (Occupancy detection in buildings has been estimated to save energy in the order of 30 to 42%, when used with control systems).

**YOUR TASKS**

**A) Implement the Fisher Linear Discriminant** to separate the 3 datasets in FisherData. Here it will be important that you draw your points together with the Fisher Discriminant you obtain.

**B) Implement the Perceptron algorithm** for classifying the point in the PerceptronData.mat dataset. Here it will be important that, AT EACH ITERATION, you draw your points together with current separating plane (determined by the current perceptron weights). You will be able to look at the perceptron while it learns!

**C) Build a Logistic Regression model**. You will need to implement it twice:

i. using the stochastic gradient descent.
ii. using Iterative Least Squares Algorithm.

TIP 1: it will be interesting to plot the Cross Entropy error vs the iteration.

TIP 2: you will need to initialize the weights. Start by choosing them randomly, gaussian distributed, with a very small variance…

Run some experiments and think about these questions: Do the two implementations always converge to the same solution? What happens if the step size of your stochastic gradient descent is too large? What happens if you initialize your weights to values that are too large?

Today you should aim at finishing a few scripts that implement the above descriptions.

During next week you should try to:

- divide your dataset into training and testing;
- measure the error on the training and testing dataset (which error measure should you use?);
- make your code more modular, factoring it into functions;
- generate fancier plots (with title, variable names on the axes, etc).

Have fun ! ☺