**Lab instructions 2**

Download the Bean dataset from

<https://archive.ics.uci.edu/dataset/602/dry+bean+dataset>

It contains 13611 samples from seven different types of beans. For each bean 17 different features or attributes were measured. You can read the details of the features in the detailed description given in the link above. So, we essentially have 13611 samples belonging to seven different classes and each sample is a vector in a 17 dimensional space.

Your first task is to read the data using pandas library.

For the sake of simplicity we will reduce the number of classes to 2. So, keep all samples from the classes Seker and Barbunya and remove (drop) samples belonging to the other classes.

A further simplification happens when we normalize the data. For this purpose, for each feature, find the maximum and minimum values. Let us say that for the ith feature the maximum and minimum are given by max(i) and min(i). So, for the jth sample, the normalized value for ith feature will be

The problem is that we want to find a linear decision boundary that separates the two classes using a single perceptron. Recall that the algorithm for perceptron learning is as follows:

Initialize the weights randomly

while (! converged)

for each training sample with feature vector

calculate

if() **w** = **w** + **x**

if() **w** = **w** – **x**

end

end

In the above P can be one of the classes, say Seker and N can be Barbunya.

Recall that convergence of the perceptron learning is guaranteed only if a linear decision boundary exists that separates the samples of the two classes. Since we do not have that guarantee, so it is good to change the while loop above to a fixed number of iterations (say 100).

During training, set aside 30% of the samples for testing and use only 70% for training. The split between training and testing set should be random. During training, note the number of misclassifications in each iteration. Plot a graph of number of misclassifications versus the iteration number. After exiting from the while loop you should test the model that you have built (no further weight change happens at this stage). Test your model using the 30% test samples that you had set aside. Report the error with the testing set.