**Mini-Project Number 1.** **What Breast cancer will recur?**

**Deadlines:**

Given May 20; due June 3. (i.e. two 2 weeks).

If submitted between June 3 and June 10 (-5 points). Between June 11 and June 18 (-15 points). June 18 final deadline. (So submit on time.) If you don’t finish everything by June 3 submit a report on what you have done so far.

------------------------------------------------------------------------------------------------------------------------

**Collaboration:** 1) You should work in teams of two- three. I will try to speak with each member of the team.

2) **External Collaboration**. You may freely discuss the project with other members of the class. I recommend you even arrange a zoom meeting on e.g. Sunday evenings. You should not share codes, but you can ask about problems and hear others suggestions. Of course you can not share *code* with each other. If you find help on the web or the internet, \*YOU MUST REFERENCE IT\*. If a particular student helped you, reference him and explain what help you received.

---------------------------------------------------------------------------------------------------------------------------

This work will require you to download a "real" (i.e. not toy) data set relating to Breast Cancer recurrence prediction. The data set has values taken from a mass extracted from a patient suffering from breast cancer and then indicates whether or not the cancer recurs and how long it takes.

There are approximately 151 patients that did not recur; and 47 that recurred. (This is not a "balanced" set ;i.e. different amounts of data for each of the two cases, and you will have to take that into account when considering the quality of your results. Be sure to explain)

Your goal is to implement some of our algorithms and report on how accurate they should be for generalization.

You will implement the following algorithms:

(1) \*\*\***CANCELED\*\*\*** perceptron algorithm . (Your own code. The sets are not linearly separable; but you should indicate how you discover this when running the perceptron.)

(2) **Adaline algorithm** . (Your own code. After training the adaline, you can impose a cut-off to decide which class the data point belongs to. For example, if you try to train the adaline to +1 for Recurring and -1 for Non-recurring then you can use 0 as a cut-off; i.e. a positive result means recurring and negative non-recurring.)

(3) **Backpropagation algorithm**. The first two algorithms should be your own code; the third one can be from a package (like MATLAB or (for those who took the course in deep learning (Tensorflow)). Other packages are also OK probably (I am not sure what is in WEKA, for example.) Of course you can use your own code also for (3) but that is not necessary for this exercise.

The data set can be found at

https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wpbc.data

Explanation regarding the dataset:

<https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/wpbc.names>

note: you will need to filter the dataset using field (i.e. column) “2”:

R = recurrent, N = nonrecurrent

Additionally, you need to: leave out the first column (participant’s ID) and the third column (the recurrence time) from the data when implementing your algorithms.

Training and testing.

1. You should divide the set into two classes; say 66% for training and 33% for testing. (You need to think about the balancing problem here, describe what you did …) Indicate how long you trained and what the results were for the training class and the testing class.
2. Having accomplished 1) if you change the 33% choice 3 times, you obtain 3 results. This is called "cross-validation". You should report the average of these 3 results and the standard deviation.

**Reporting the results:**

A detailed word/pdf report file along with documented code is expected. Besides the code I expect about 4 pages or so. *You will be graded on completeness and clarity of the report as well as the quality of the implementation.*

In your report you should refer to the following items (i.e. **I would like to see about 4 sections in your report**):

1. Time it took to train the final model (in sec or mins on your PC/Laptop) i.e. after all the cross validation processes.
2. Ease (are the parameters can be adjusted intuitively or not) of parameter search/adaptation (e.g. learning parameter, architecture of feed-forward network (if any)) in order to find the best model for classification
3. Performance on the training set and on the testing set (in %) – explain the results
4. Summary and discussion of process/ problems.

Note: you might meet missing data (features) in some data points - you should think and decide what to do / how to deal with it and of course explain your decision (it doesn’t need to be some “scientific” decision).

The results should be reported in clearly. Be sure to include a tables like the followingtwo ways:

* Confusion matrix for each model evaluation

|  |  |  |
| --- | --- | --- |
|  | Predicted: yes | Predicted: no |
| Actual: yes | True Positive (mean+/-std) | False Negative |
| Actual: no | False Positive | True Negative |

**Any additional charts and graphs will help**.