MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

158.333 Applied Machine Learning and Big Data Processing
S2 2020

Assignment 1

In this assignment your task is to *implement and test* your own version of **AdaBoost** and compare it to the version implemented in *scikit-learn*. You should use a Jupyter notebook to implement, run and visualise the results.

The algorithm is to be implemented as a *binary classifier*. The weak classifiers can be simple thresholds, similar to what we already did in class. Small adjustments in the algorithm can be made to improve speed or other aspects of the algorithm (for example, knowing when to stop training rather than explicitly set T rounds, using a different weak classifier etc).

Using the following UCI datasets, you need to measure training errors and test errors and report in the notebook:
https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits
https://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits
https://archive.ics.uci.edu/ml/datasets/Iris
https://archive.ics.uci.edu/ml/datasets/Spambase

UCI dataset	Classes	Features	Instances	Pre-split?	Skewed?
Iris	3	4	150	No	No
Spambase	2	57	4601	No	Yes
Optical digits	10	64	5620	No	No
Pen-based digits	10	16	10992	No	No

For the binary problem (spambase), use the standard version for AdaBoost. For multiclass problems, you need to train **one-against-all N times** (for N classes), and choose the final prediction using individual classifier's results. If there is a draw (e.g., two or more classifiers classify a new sample as their respective class), then use margins to resolve the draw.

Use a 3-fold cross-validation, i.e., each dataset should be **randomly split** into training and test, **70% for training**, and **30% for test**. Be careful to split into non-skewed sets, e.g., Iris contains 50 samples of each of the three species, then the random split should contain the same number of samples for each one of the classes, 35 in the training set and 15 in the test set. Scikit-learn has its own function for k-fold cross-validation. You should run each training/test cycle three times (the random splits will create three different classifiers). When showing the accuracy results, use the average and use error bars to show the confidence in the accuracy (the error can be adopted as +- **standard deviation**).

Compare your results with the AdaBoost implemented in scikit-learn. You should show: the **training accuracy and the test accuracy** (in a graph of any type you choose, e.g., a bar graph) separately for each of the datasets.

A final figure with the classification results for each UCI dataset should show the accuracy for each **test classification accuracy** for each dataset and for both versions of AdaBoost (your own code and the one in scikit-learn).

Submit a **one file** (notebook) on Stream by Friday **28th August 2020.**
This assignment is worth 10% of the total marks for the course.
This assignment is to be completed **individually**.