CA Assignment 1
Data Classification
Implementing Perceptron algorithm

Assessment Information

Assignment Number	1 (of 2)
Weighting	15%
Assignment Circulated	11 Feb 2022
Deadline	4 March 2022 at 17:00
Submission Mode	Electronic via Canvas
Purpose of assessment	The purpose of this assignment is to demonstrate: (1) the understanding of the Perceptron algorithm; (2) the ability to implement the Perceptron algorithm for binary classification; (3) the ability to evaluate a classification algoirthm; (4) the ability to turn a binary classification algorithm to a multi-class classification algorithm using the 1-vs-rest approach; (4) the ability to incorporate regularisation into classification algorithm.
Learning outcome assessed	(1) A critical awareness of current problems and research issues in data mining. (3) The ability to consistently apply knowledge concerning current data mining research issues in an original manner and produce work which is at the forefront of current developments in the sub-discipline of data mining.

Objectives

This assignment requires you to implement the Perceptron algorithm using the Python programming language.

NOTE
No credit will be given for implementing any other types of classification algorithms or using an existing library for classification instead of implementing it by yourself. However, you are allowed to use
- numpy library for accessing data structures such as numpy.array;
- random module; and
- pandas.read_csv, csv.reader, or similar modules **only** for reading data from the files.

However it is not a requirement of the assignment to use any of those modules.
You must provide a README file describing how to run your code to re-produce your results.

Assignment description

Download the *CA1data.zip* file. Inside, you will find two files: *train.data* and *test.data*, corresponding respectively to the train and test data to be used in this assignment. Each line in the file represents a different train/test instance. The first four values (separated by commas) are feature values for four features. The last element is the class label (class-1, class-2 or class-3).

Questions/Tasks

1. **(15 marks)** Explain the Perceptron algorithm for the binary classification case, providing its pseudo code.
2. **(30 marks)** Implement a binary perceptron.
3. **(15 marks)** Use the binary perceptron to train classifiers to discriminate between
   - class 1 and class 2,
   - class 2 and class 3, and
   - class 1 and class 3.

   Report the train and test classification accuracies for each of the three classifiers after training for 20 iterations. Which pair of classes is most difficult to separate?
4. **(30 marks)** Extend the binary perceptron that you implemented in part 3 above to perform multi-class classification using the 1-vs-rest approach. Report the train and test classification accuracies after training for 20 iterations.
5. **(10 marks)** Add an ℓ_2 regularisation term to your multi-class classifier implemented in part 4. Set the regularisation coefficient to 0.01, 0.1, 1.0, 10.0, 100.0 and compare the train and test classification accuracies.

Submission Instructions

Submit via Canvas the following **three** files (**please do NOT zip files into an archive**)

1. the source code for all your programs (**do not provide ipython/jupyter/colab notebooks, instead submit standalone code in a single .py file**),
2. a README file (plain text) describing how to compile/run your code to produce the various results required by the assignment, and
3. a PDF file of **no more than 2 pages** providing the answers to the questions. Your answers should be succinct, but complete and clear. The clarity and the presentaiton of the report will be assessed.

It is extremely important that you provide all the files described above and not just the source code!