Name: Jack Stawasz
Course: CSCI 312 Principles of Programming Languages
Assignment Deadline: April 30, 2025

Question 1 (perceptron.h)

Make a new directory called `Assignment5` in your ppl repo. Use
`ppl-25s-01/05_ABSTRACT_TYPES/perceptron/perceptron.h`
as the starting point:

1. Deprecate `#define DIMENSIONS 3`
   (a) Your ADT should be capable of handling any number of records and fields in a `.dat`
   (b) You may not assume a `.dat` will contain three fields and 20 records
   (c) You may assume that each line will contain N floating point values separated by one space, and each line corresponds to one example where the first N-1 values/fields correspond to features and the Nth value/field corresponds to the target
   (d) You may assume $2 \leq N < INT\_MAX$ (so N can be handled as an int)
2. Change `Data new_Data(int number_of_examples);` to `Data new_Data(const char *fname);`. The modified function should accept a filename and read the data in the file into a Data instance
3. Deprecate `load_data`
4. Change `Model new_Model();` to `Model new_Model(const Data data);`. The modified function should accept a Data instance from which it can deduce the dimensionality of the weight vector[1] and initialize the weight vector using the simple routine in `initialize_model`
5. Deprecate `initialize_model`
6. Change `void fit_model(Model model, Data xcoords, Data ycoords, Data targets, int number_of_examples);` to `void fit_model(Model model, Data data);`
7. Change `void run_scoring_engine(Model model);` to `void run_scoring_engine(const Model model);`

[1] Recall (1) the dimensionality of the weight vector for a 2D perceptron is 3 (the number of inputs plus one for w_0) and (2) $x_0 = 1$

Question 2 (main.c)

Use
`ppl-25s-01/05_ABSTRACT_TYPES/perceptron/main.c`
as the starting point:

1. Deprecate `int number_of_examples = atoi(argv[2]);`
2. Replace lines 11-14 with one line `Data data = new_Data(fname);`
3. Replace lines 28-30 with one line `free(data);`

Question 3 (perceptron.c)

Use
`ppl-25s-01/05_ABSTRACT_TYPES/perceptron/perceptron.c`
as the starting point:

1. Change `struct data` so it has at least three members:[2]
   (a) A two-dimensional array of doubles for the matrix of inputs
   (b) A one-dimensional array of ints for the vector of targets
   (c) A new `struct shape` that stores the number of examples and the number of features
2. Change `struct model` so it has at least two members:[3]
   (a) A one-dimensional array of doubles for the vector of weights
   (b) A new `struct shape` that stores the dimensionality of the weight vector
3. Since you deprecated `load_data` in the interface you should move the functionality to `new_Data` or make `load_data` static
4. Since you modified the interface for `new_Model`, the new implementation should use the Data instance to deduce the dimensionality of the weight vector
5. Since you deprecated `initialize_model` in the interface you should move the functionality to `new_Model` or make `initialize_model` static
6. Change `static void sgd(Model model, double xcoord, double ycoord, double target)` to `static void sgd(Model model, Data data)`
7. Change `void fit model(Model model, Data xcoords, Data ycoords, Data targets, int number_of_examples)` to
   `void fit_model(Model model, Data data)`

[2] You may add more members if you chose to
[3] You may add more members if you chose to

Question 4

Compile and run your evolved program and select five random points in [0,1]-by-[0,1] and report your predictions in the following table:

Table 1: Predictions

x	y	Prediction
0.500	**0.500**	**+1**
0.001	**0.001**	**-1**
0.900	**0.900**	**+1**
0.100	**0.900**	**+1**
0.900	**0.100**	**-1**
0.426	**0.741**	**+1**

Question 5

After you complete the questions above, categorize each and every operation in the interface as either a constructor function, access function, or manipulation procedure, and record your categorization in the following table:

Table 2: Categorization

Operation	Category
`new_Data`	**Constructor Function**
`new_Model`	**Constructor Function**
`fit_model`	**Manipulation Procedure**
`run_scoring_engine`	**Access Function**