Week 7 Project: Machine Learning

ACADEMIC HONESTY

As usual, the standard honor code and academic honesty policy applies. We will be using automated **plagiarism detection** software to ensure that only original work is given credit. Submissions isomorphic to (1) those that exist anywhere online, (2) those submitted by your classmates, or (3) those submitted by students in prior semesters, will be detected and considered plagiarism.

INSTRUCTIONS

There are three parts to this assignment. Please read all sections of the instructions carefully. In particular, note that you have a total of **5 submission** attempts for each question.

I. Perceptron Learning Algorithm (50 points)
II. Linear Regression (50 points)
III. Classification (100 points)

You are allowed to use Python packages to help you solve the problems and plot any results for reference, including **numpy**, **matplotlib**, and **scikit-learn**.

I. Perceptron Learning Algorithm

In this question, you will implement the perceptron learning algorithm ("PLA") for a linearly separable dataset. In your starter code, you will find `input1.csv`, containing a series of data points. Each point is a comma-separated ordered triple, representing **feature_1**, **feature_2**, and the **label** for the point. You can think of the values of the features as the x- and y-coordinates of each point. The label takes on a value of positive or negative one. You can think of the label as separating the points into two categories.

Implement your PLA in a file called `problem1.py`, which will be executed like so:

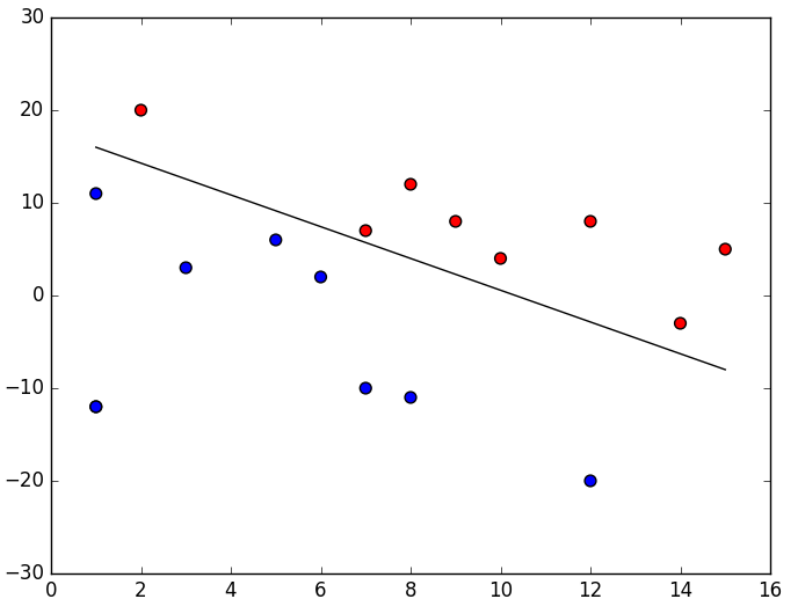
```
$ python problem1.py input1.csv output1.csv
```

If you are using Python3, name your file `problem1_3.py`, which will be executed like so:

```
$ python3 problem1_3.py input1.csv output1.csv
```

This should generate an output file called `output1.csv`. With each iteration of your PLA, your program must print a new line to the output file, containing a comma-separated list of the weights **w_1**, **w_2**, and **b** in that order.
Upon convergence, your program will stop, and the final values of w_1, w_2, and b will be printed to the output file (see **example**). This defines the **decision boundary** that your PLA has computed for the given dataset.

What To Submit. `problem1.py` or `problem1_3.py`, which should behave as specified above. Before you submit, the **RUN** button on Vocareum should help you determine whether or not your program executes correctly on the platform.



Optional. To visualize the progress and final result of your program, you can use **matplotlib** to output an image for each iteration of your algorithm. For instance, you can plot each category with a different color, and plot the decision boundary with its equation. An example is shown above for reference.

MACHINE LEARNING #1 (AI.A) (EXTERNAL RESOURCE)

(50.0 / 50.0 points)

Your email address will be used to identify your submission entry.

Go To Question #1