Winter 2026 DSC 240: Introduction to Machine Learning

Machine Problem 1

Due: **Monday, Jan 26th, 11:59 pm PST**

Notes:

1. **This assignment is to be done individually.** You may discuss the problems at a general level with others in the class (e.g., about the concepts underlying the question, or what lecture or reading material may be relevant), but the work you turn in must be solely your own.
2. Be aware of the late policy in the course syllabus – i.e., *no late days for all the assignments,* so **it is your responsibility to turn in your assignment to Gradescope by the due time.**
3. Any updates or corrections will be posted on Piazza, so check there occasionally.
4. You can refer to online resources and cite exact references. **No copying code**.
5. Be sure to re-read the **"Academic Integrity"** on the course syllabus. You must complete the section below. If you answered Yes to either of the following two questions, give corresponding full details.

Did you receive any help whatsoever from anyone in solving this assignment?
Did you give any help whatsoever to anyone in solving this assignment?

Code Instructions

Change the Python 3 program `mp1.py` that implements a three-class linear classifier.

Specially, change the function `run_train_test(training_input, testing_input)` defined in `mp1.py`. The function should include the following process:

- Training (using the training data set):
  - Compute the centroid of each class (e.g. A, B, and C).
  - Construct a discriminant function between each pair of classes (e.g. A/B, B/C, and A/C), halfway between the two centroids and orthogonal to the line connecting the two centroids. This is the "basic linear classifier" that we have discussed.
- Testing (using the testing data set):
  - For each instance, use the discriminant function to decide "A or B" and then (depending on that answer) to decide "A or C" or "B or C." (Ties should give priority to class A, then B, then C)
  - More specifically, first use A/B discriminant function, and then:
    - If classified as A, then decide A or C
    - If classified as B, then decide B or C
  - Keep track of true positives, true negatives, false positives, and false negatives.

The training and testing data sets are available in this starter package, along with a description of their formats in the `MP1_data.md`

This function is where you implement the assignment. Feel free to define additional functions, but **DO NOT** change this function signature and **DO NOT** change the `mp1.py` module/file name. We will call this as the entry point to your code.

Input Instructions

The inputs `training_input` to the `run_train_test` function should be in the form of a list from the training file. An example of how the list will look like with the provided dataset file `data/trainint1.txt` is:

```
[[3, 500, 500, 500], [4.6073, 1.585, 0.64941], [1.2675, 2.9307, 2.1275], ...]
```

NOTE: The numbers in the above example are for illustration purposes only.

The inputs `testing_input` to the function should also be in the same format. The python script `evaluate.py` may be helpful in understanding on how the provided dataset files can be converted to the required format.

Output Instructions

As output, the program should return a dictionary of averages over all three classes of the true positive rate, the false positive rate, the error rate, the accuracy, and the precision:

```
>> print(run_train_test(training_input, testing_input))
{
    "tpr": 0.80 # true positive rate
    "fpr": 0.27 # false positive rate
    "error_rate": 0.44
    "accuracy": 0.60
    "precision": 0.90
}
```

NOTE: These numbers are made up, for purposes of illustration only.

The `run_train_test` function should **return results using the same keys as shown**.

Information on computing these averages is included in the `MP1_extra.pdf`.

Evaluation Instructions

You can test your program with the `data/training1.txt` and `data/testing1.txt` provided in this starter package. You can also use the provided `evaluation.py` for checking:

```
$ python evaluate.py
```

Submission Instructions

You should upload `mp1.py` to Gradescope for grading.