General guidelines:

- This assignment is tentatively due at the end of week 8. But the final deadlines are **as posted on gradescope**
- Given that this is a new class, welcome to provide feedback on any problems with the spec!
- **This assignment has 5 tasks, each worth 5% of your grade, for a total of 25%**
- Assignments must be completed individually
- Assignment stubs are available here

Overall description

This assignment will require you to implement various fairness interventions to improve fairness outcomes, while approximately maintaining accuracy, of a particular classifier.

In this assignment, we will predict default of credit card clients using the UCI “default” dataset:
https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients

Our label and sensitive attribute are as follows:
y = default payment next month (“y” == “1” in the dataset)
z = has graduate education (“EDUCATION” == “1”)

Files

The file **autograder.py** contains code to read the data and extract the label and sensitive attribute. This is the same file the actual autograder uses, so should not be modified; it can be run as-is to test your submission.

The file **submission.py** is for you to modify as needed. You should implement each of the functions in this file. Existing implementations have already been provided to give you a sense of how the assignment works. This is the only file you should upload to the autograder.

Tasks

You are required to complete the following 5 tasks, each worth 5 marks:

1. Make a solution that is as accurate as possible; you are allowed to use the sensitive attribute.
2. Apply a **dataset-based intervention** to improve the fairness (demographic parity) of your model; that is, you can modify the dataset however you'd like, but cannot alter any other part of the modeling pipeline.

3. Apply a **model-based intervention** to improve the fairness of your model; here, the sensitive attribute may be used during training but is not available at test time (i.e., it cannot be a model feature).
4. Apply a **post-processing intervention** to improve the fairness of your model; here, you are given the output probabilities of an existing classifier, along with the sensitive attribute, and must generate output labels.
5. Apply **any** combination of interventions.

Baselines

The following baselines are provided for each task:

1. Just extend a trivial classifier (concatenation of all features) with the sensitive attribute.
2. Duplicate training instances with z=0
3. Assign higher instance weights to instances with z=0
4. Manually perturb the per-group thresholds a bit

Grading

Generally speaking, it should be possible to design interventions that satisfy *demographic parity* (almost) exactly; partial grades are provided mostly for solutions that don't quite make it. A **tentative** grading criterion is as follows (same for Tasks 2-5; Task 1 will be graded more trivially):

1 mark: Upload any *valid* solution (i.e., which doesn't throw any error)
2 marks: Obtain a solution with *roughly* equal per-group TPRs
3 marks: Obtain a solution with *roughly* equal per-group TPRs without *significantly* reducing performance compared to a trivial[1] model
4 marks: Obtain a solution with *almost identical* per-group TPRs without *significantly* reducing performance compared to a trivial model
5 marks: Obtain the *most accurate possible* solution with *almost identical* per-group TPRs (i.e., accuracy almost the same as the topmost solution in the class)

The terms "roughly", "significantly", "almost identical", and "most accurate possible" will be defined in a few weeks, once several valid solutions have been uploaded.

[1] Provided in the autograder.py file