6.867 Project: Milestone 3

(Sapna Kumari, Archer Wang, Joe O'Connor)

To evaluate the success of adversarial debiasing on the Compas Recidivism Risk Dataset, we will need to train two feed-forward neural network models, one in a non-adversarial way and in an adversarial way. We will write everything in Python and run our networks on Google Colab.

Some libraries and packages will be used to run our code. We will import and use the SQLite library to query data from the compas.db file, which contains all the personal information about the convicts. To represent the names via word embeddings, we will use the NLP library Spacy which has pre-trained word embedding models (ex. en_core_web_md library). In addition, we will use the Pytorch library to set up our non-adversarial neural network and make use of the NN module to adjust hyperparameters. For the adversarial neural network, there is already existing code but for a different scenario. Our objective is to modify the network's architecture, specifically the input and output layers, to suit it for our purposes of finding the probability of a convict committing another crime.

As for the division of labor, we will all independently experiment with the neural network architectures (different numbers of layers, different numbers of neurons per layer, and different activation functions for the hidden layers), and later compare our results to select the most viable option. Joe will initially retrieve the code for the adversarial neural network and figure out and modify the parts of the architecture that need to be changed. Archer will initially set up a working version of a skeletal architecture for the non-adversarial neural network. Sapna will retrieve the data and figure out the best word embedding model to use in our neural networks.

We still have some open questions. Should we use other libraries than Spacy for pre-trained word embedding models such as Gensim? Are there any constraints to the hyperparameters for the neural network models?