# CIS 523 Bias Bounties Project

Karan Sampath and David Feng

March 29 2022

**Members**

Karan Sampath: Wrote Project Write-Up, Developed Overall Model

David Feng: Wrote Project Markdown, Found Optimal Individual group Improvements

1. 1.1 The minimum amount of improvement is a mathematical function of the improvement on each group. Given that we have at least 5% of the dataset being improved by 5% for 5 consecutive times, we know that overall error x is being reduced by $0.05 * 0.05$ every time, and therefore minimum improvement in the overall set must be 1.25%.

   1.2 For this condition to hold, we know that the model must have become a Bayes optimal classifier. In this case, since we only want models with an improvement of 5% on the previous model, we know that after $1/0.05 * 0.05 = 400$ rounds we are guaranteed to not be able to find updates that meet the improvement criteria.

2. 2.1 1. h5

   2. h3

   3. h2

   2.2 No, we don't know whether there will necessarily need to be any repairs. We know that h4 requires repairs due to it not doing well on red squares, but we don't if this would necessarily hold true for the new model as well. For example, red little squares are currently classified by h5 in the PDL without requiring repairs. This means that we don't have enough information to say whether repairs would be required for the new model that has been added.

3. 3.1 The groups $(g, h)$ are (in the order they were accepted):

   1. College Age People (AGE=18 to AGE=22): A simple decision tree of depth 10 was used with the simple_updater because of time constraints; the fully trained RandomForestClassifier was taking too long to update.

   2. Age 19 People (AGE=19): This overlaps with college age people, but we were able to improve a specific subset by a good margin using the default decision tree of depth 10.

3. Pre-Retirement People (AGE=57 to AGE-62): Again, a simple decision tree of depth 10 was used for time reasons. We bundled them together to increase the training size for the model.

4. Non-Institutionalized Group Quarters (RELP=17): A decision tree of depth 10 was used with simple_updater.

5. Age 68 People (AGE=68): A decision tree of depth 10 was used.

6. Age 65 People (AGE=65): A decision tree of depth 10 was used again.

7. Age 21 People (AGE=21): A specially trained RandomForestClassifier was used for this group. We first used a RandomSearchCV to one, minimize overfitting through cross-validation and two, find the range of optimal hyperparameters for the model. Then, we used a GridSearchCV to find the exact best hyperparameters for this specific model and dataset. Then, the updater was used.

8. Age 20 People (AGE=20): A similar process with a tuned RandomForestClassifier was employed.

9. Age 63 People (AGE=63): Same as group 7 - a well-tuned RandomForestClassifier.

3.2 We first used an error classifier on each group to find out the best candidate groups for improvement. We then implemented individual classifier for each of these groups, aiming to develop classification methods based on our intuition and checking whether they played out.

After this initial combination of classification and trial-and-error, we then shifted to developing an overall model using more complex classifiers like Random Forests or SVMs. We believed them to be better than the given base decision tree classifier since they allowed the different group weights to vary more and give a more accurate overall prediction. Although we ran into updating errors, we were able to find a model that finally worked and then continued with appending on individual groups using our earlier strategy.

3.3 We initially looked at developing complex models including neural networks and random forests with a high number of estimators but this took too long to run on the data and hence we were forced to give this overall model strategy. However, more than just running we faced problems with updating the new model to the PDL, which we believe is a function of both the computing power available to us and the size of the dataset.

This meant that our biggest challenge became that models that beat errors in the group couldn't update the PDL because the updater simply wouldn't run on them, leading us to find often exceedingly simple models which could be accepted. Not only add a lot of time and stress, but was extremely frustrating.

4. 4.1 Groups are only accepted when they represent a certain minimum proportion of the dataset, and would actually reduce error in the dataset. In other words, we cannot improve groups without ensuring that they are actually present in the dataset and hence will need to work on the various other group models. This is a feature of the mini-group fairness understanding underlying the bias bounty group project, where each group must be under a certain error threshold if it meets the criteria of being a minimum percentage of the overall dataset. If there weren't minimum percentage restrictions, the model could become one of individual fairness, with an error rate of 0 or 1 for all cases. This would make the tradeoff between fairness and accuracy extremely severe to the point of being a potential circuit breaker, making it impossible to reach a viable model.

   4.2 Other notions of fairness discussed in class also suffer from the same problem, as they are unable to accept updates which improve groups that are not represented in the dataset. This is a feature of the group fairness dynamic, where models prioritize equalizing error across groups to ensure notions of fairness are maintained. Fundamentally, all group notions of fairness are relative between groups, and hence we cannot compare them without a certain group being accurately represented in the model.

   4.3 It would theoretically be possible to incorporate the data in our model by first finding the differences between the new dataset D' and D such that we weigh the group compositions of the combined datasets equally. We cannot naively combine the datasets since that would cause certain groups to be not well represented and may bias the model. Instead, we must train the model on both datasets, while weighing for differences. This is a difficult problem for a few reasons:

   1) Other groups g' may not be well represented in the bounty hunters' dataset, and that may cause an increase in overall error for the final model

   2) There is no easy way to integrate the dataset and ensure that both the group $g$ is represented and no other groups have been biased