$$FP = 1 - |FP_{Black} - FP_{White}|$$

$$Fair\ and\ Accurate = (1 - BS)(FP)$$

When training and tuning the models, this fair and accurate metric was used to evaluate each candidate model. During the hyperparameter tuning phase described in the next section, I found that when tuning against this metric, the best models favored predictions that very rarely exceeded the .5 probability threshold which led to very low false positive rates in both white and black parolees. This meant that the fairness penalty was not very meaningful in preventing racial bias, as there were almost never any false positives for either black or white parolees.

4.3. Parameter Tuning

To help guard against overfitting, the training set was first randomly split into a new training and hold-out set where 33% of the data was allocated to the hold-out set. For convenience, these new datasets will be referred to as the dev training set and the hold-out set respectively. XGBoost requires the tuning of several hyperparameters which control how robust the model is against overfitting. The dev training set was used to tune these hyperparameters, while the hold-out set was only used later for performance evaluation of candidate models.

I applied a grid search of the following XGBoost hyperparameters on the dev training set: *N_estimators, Depth, Min Child Weight, Learning Rate, Gamma,* and *Colsample by Tree*. To evaluate each set of hyperparameters, I randomly divided the dev training set into a sub-training and sub-test set with 33% of the data used for the sub-test set. The model was trained on this sub-training set and then evaluated against the sub-test set where the "fair and accurate" metric defined in the previous section was used to measure the model's performance. When evaluating each set of hyperparameters, I randomized the sub-training and sub-test set ten separate times and averaged the model's "fair and accurate" performance value on each randomized sub-test set. This repeated randomization approach was done to ensure that the best hyperparameters on average were chosen.

After completing a grid search, the hyperparameters with the best average performance were chosen as the final model hyperparameters. I then trained a model using these hyperparameter values on the dev training set and evaluated it on the hold-out set. This method provided an unbiased estimate of the model performance, since the hold-out set was never used in the tuning phase. These are the final hyperparameter values chosen for the XGBoost model:

- *N_estimators* = 800
- *Depth* = 4
- *Min Child Weight* = 20
- *Learning Rate* = .005
- *Gamma* = .001
- *Colsample by Tree* = .9
- *Booster* = 'gbtree'

This resource was prepared by the author(s) using Federal funds provided by the U.S. Department of Justice. Opinions or points of view expressed are those of the author(s) and do not necessarily reflect the official position or policies of the U.S. Department of Justice.