# Ensemble Classification as an Effective Method of Determining Racial Bias in Incarceration Data

April 15, 2020

**Abstract**

Machine Learning is becoming a more prevalent tool in the world of criminal justice. Often it is used to predict who will commit a crime or where crimes may occur. Seldom is it used to regulate the criminal justice system, however. In this report I examine prison inmate data and determine what machine learning techniques are effective at detecting the racial bias that has been shown to exist in this data. In this report I find that ensemble methods, especially LightGBM, are effective in classifying inmates by race, thus showing that racial bias is detectable by machine learning techniques and that machine learning is an effective tool in determining if criminal justice reform should be considered.

## 1 Problem Statement and Motivation

Last semester I took a look a data set containing the information of more than 7.5 million individuals that have been processed by the criminal justice system. I found that racial minorities were more likely to receive extreme sentences, agreeing with existing research around bias in the criminal justice system[8]. In this report I will be exploring the data from a machine learning perspective. My goal is to determine if this data can be classified in such a way that is predictive of race. The idea is that perhaps racial bias can be detected in various systems by seeing how effective different machine learning techniques are at classifying an inmate's by race given their data.

This is an unconventional way to approach criminal justice data with machine learning. Often we see machine learning being used to attempt to determine who might be a criminal or where criminal activity may occur using social media data and other public information, which may include data the government owns, but which is not available to the public. These approaches often ignore or discount the ways that these techniques may disproportionately affect people of color and the poor. Many organizations have made official statements regarding the use of machine learning in this way, often called predictive policing. The ACLU for example released a statement listing civil rights related concerns about predictive policing which was signed by several civil rights organizations including the NAACP[6].

My objective is to go against the predictive policing paradigm and use machine learning to benefit these negatively affected classes of people by using machine learning as a diagnostic tool. If it can be shown that certain machine learning techniques are effective at classifying inmates by race given incarceration related information, then we can inform policies that will attempt to correct for these systemic racial biases.

# 2 Data

## 2.1 Source and Credibility

The data that I will be using in this analysis is from one source. It is a database hosted on Data.gov and maintained by the State of Connecticut Department of Corrections. This source is highly credible because it is a primary source for the data. This organization is an official government agency which collects, maintains, and reports on this data.

## 2.2 Gathering and Cleaning

All the data which I am using in this report are freely available to the public. Collection and cleaning was relatively simple as the source data was well maintained. The file that I obtained from the Connecticut Department of Corrections is a very well maintained database. The largest issue I had with this file was mild inconsistency with the way in which certain data was encoded (ex. race was encoded as both `WHITE` and `WHITE\t`). The file is

```
individuals.csv.
```

## 2.3 About the Data

This data set contains individual information for 7.77 million people that have been processed by the justice system and recorded by the Connecticut Department of Corrections. Each individual is recorded along with their age, gender, race, offense, and sentence length, among other things.

Because there is so much to consider in what is found in the data set, I chose not to feature engineer as to avoid unneeded complexity.

The sample sizes among different races that are found in the Connecticut Department of Justice data are not similar. The sample size for American Indians and Asians is much smaller than that of Whites, Hispanics, and Blacks, hence we may see some irregular outcomes in the analysis related to these racial groups.

```
Sample size for Blacks: 3287596
Sample size for Whites: 2393949
Sample size for Hispanic: 2039297
Sample size for American Indian: 21133
Sample size for Asian: 35660
```

# 3 Methods

Before I begin discussing the methods that I did use, I will talk about some of the methods that I did not use. There are many techniques that are not applicable to this data. For example, since this data set is not a time series models like ARMA and HMM are not applicable here.

## 3.1 Dimension Reduction

A method that I attempted to use, but found little success with were dimension reduction tools like PCA, T-SNE, and UMAP. This dimension reduction would have been helpful, especially because once one-hot encoded this data becomes extremely high dimensional. However PCA, T-SNE, and

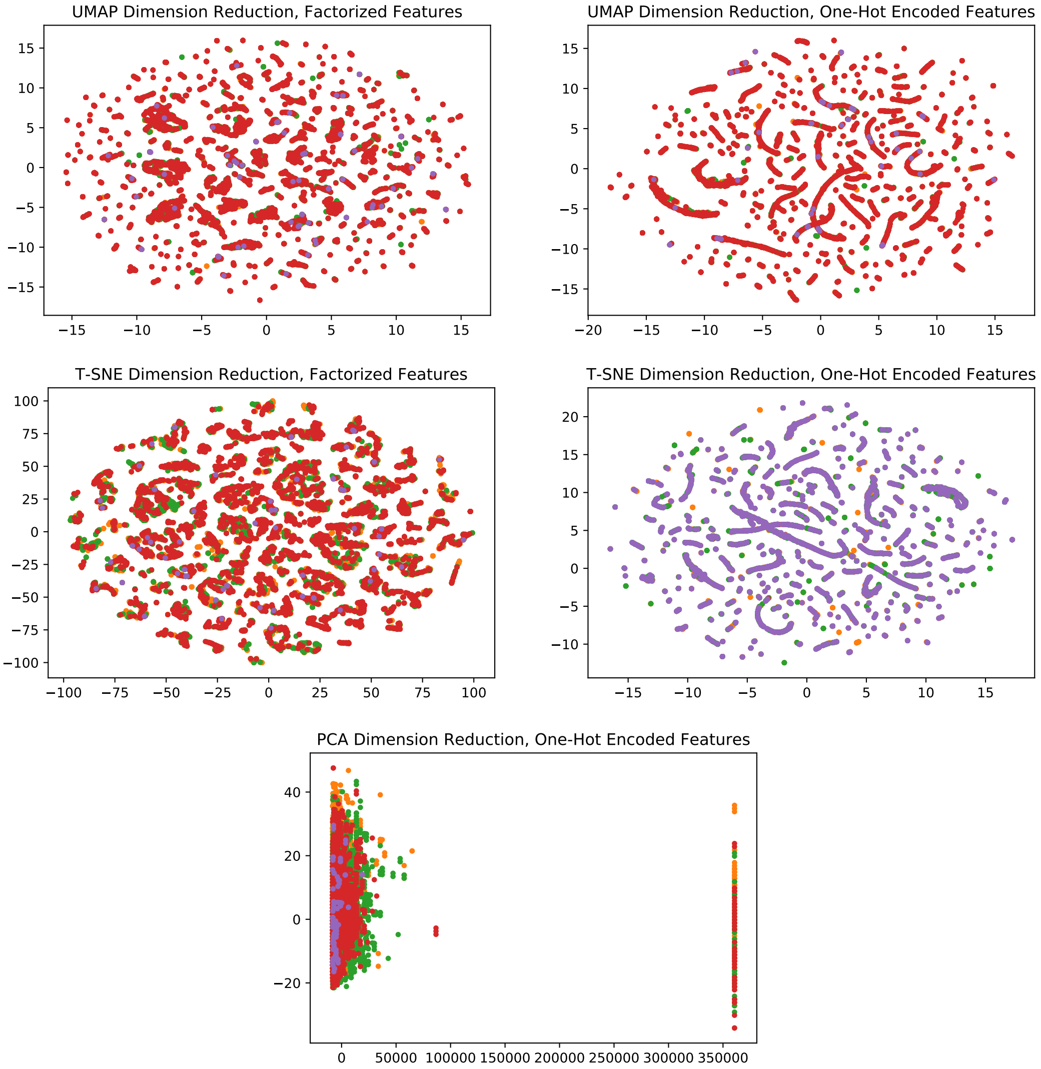

Figure 1: Dimension reduction algorithms fail to reveal significant clusters

UMAP all failed to create meaningful clusters, so I abandoned the the pursuit of clustering early on. Perhaps some kernel methods would have been helpful in this endeavor, however I could not find a kernel that could create a metric on crimes and I do not feel qualified to write one myself.

Below I have images of my attempt to use PCA, T-SNE, and UMAP to cluster the data. It is apparent that these methods are simply not effective.

This is only a small sample of things that I tried to do to get useful clustering given the dimension reduction, though it is representative of the results I found.

## 3.2 Ensemble Methods

Ensemble methods seemed immediately like the methods with the highest chance of success. I did not initially expect to successfully classify this data, however as I learned about how ensemble classifiers worked I became more confident that I would get decent results.

The primary reason that I though I would not get good results has to do with how I identified racial bias in the data. In my previous report I determined that there was racial bias in the criminal justice system based on the kurtosis of the distribution of sentence lengths, when blocking inmates by race. Interpreted this means that minorities are much more likely to receive extreme sentence lengths than a white people are. This condition is very subtle and I did not think that it would be easily detectable by machine learning methods. However, ensemble methods only need each member to do slightly better than random, so there was hope that I could get good results with these methods.

The methods that I attempted were the following: random forest classifier, gradient descent boosted classifier, XGBoost, and LightGBM.

### 3.2.1 LightGBM

LightGBM is an attempt to improve upon the efficiency, both temporally and spatially, of different gradient boosted decision tree (GBDT) algorithms such as XGBoost. This algorithm was developed by a team at Microsoft and it uses two novel techniques to improve GBDT algorithms. The baseline comparison used was against XGBoost, since the team found this method to be the best performer of the commonly used GBDT algorithms[2].

By analyzing GBDT algorithms the team found that the most expensive parts of the process is learning the decision trees and the most expensive part of learning the decision trees is finding the best split points. They decided to use a histogram based approach for efficiency. This process is dominated by the histogram building which has a complexity of $O(\#data \times \#feature)$[1]. Now the goal is to reduce the feature number or the number of data points.

**Gradient-based One-Side Sampling** This is the first novel technique proposed by the Microsoft team. It is a sampling method that is meant to reduce the number of data instances while maintaining accuracy. The main idea here is that data points with small gradient are usually ignored, since the model is already will trained on those data instances. However, the changes that occur by ignoring the data will small gradient may reduce the accuracy of the model once it is learned. Therefore GOSS examines all of the high gradient data and a random sample of the small gradient data. This method can maximize the amount of relevant data used in the training of the model without handicapping the accuracy completely.

**Exclusive Feature Bundling** This is the second novel technique and its goal is to reduce the number of features. This technique relies on the fact that high dimensional data tend to be sparse, and therefore there are likely large bundles of features that are mutually exclusive are nearly mutually exclusive. These features can be bundled into a single feature and their histograms can be combined. This reduces the complexity of building histograms from $O(\#data \times \#feature)$ to $O(\#data \times \#bundle)$ and if $\#bundle << \#feature$ then the total complexity is greatly reduced.

[1] LightGBM is not the only GDBT classifier that uses a histogram approach to finding splitting points. XGBoost also can be programmed to use this method.

# 4 Results

## 4.1 Random Forest Classifier

The random forest classifier was an easy place to begin in my attempt to find a successful classifier. It is a simple method and would likely give me a good lower bound on the success that I would have.

In using the random forest classifier I did a grid-search for the best parameters. The parameters I decided to search over are the number of trees in the classifier and the maximum depth of the trees in the classifier. I chose this because having more trees in the classifier will improve the accuracy, however if the depth of each of the trees in unbounded then overfitting of the individual trees may become an issue.

In my grid-search I found the best parameters to be the following

```
Max Depth = 85
Number of Estimators = 2100.
```

My results with the random forest classifier were rather disappointing, especially given the amount of time it took to train which was 5.32 hours on the following parameter grid

```
param_grid = {
    'n_estimators': np.arange(100,5100,500),
    'max_depth': np.arange(5,100,10)
}
```

The out-of-box score was rather promising at .70 however the method scored barely better than chance, which is .2 given that there are 5 race classes in the data. Here is a scoring of the model I ran:

```
score on a test size of 500000 is 0.286932.
```

The scoring here is lackluster to say the least though it did give me hope for more complex methods to preform better.

## 4.2 Gradient Descent Boosted Classification

This method is an obvious next step after a random forest. With the ability to alter the subsample rate and the learning rate, I expected to get better results with this model. I began by doing another grid search, but I used some of the results from the previous search on the random forest model to save time. Fitting the following grid

```
param_grid = {
    'learning_rate': np.linspace(.01,1,5),
    'subsample': np.linspace(.05,1,5)
}
```

I found the following parameters

```
Learning Rate = 0.01
Subsample Rate = 0.525.
```

The model scored no better than the random forest classifier, which surprised me:

```
score on a test size of 5000000 is 0.279797.
```

### 4.3 XGBoost

XGBoost is considered one of the best ensemble classifiers generally available, so it was the obvious conclusion to my exploration of ensemble classifiers. Its inclusion of regularization parameters led me to believe that it would score much better than the generic GBDT or random forests.

The grid I used was the following

```
param_grid = {
    'reg_alpha':np.linspace(.01,1,10),
    'reg_lambda':np.linspace(.01,1,10),
    'gamma':np.linspace(.01,1,10)
}
```

and I found the following parameters

```
L1 Regularization = 0.2575
L2 Regularization = 0.2575.
Minimum loss reduction (gamma) = 0.01.
```

XGBoost also scored no better than the other methods that I had used, which again surprised me.

```
score on a test size of 5000000 is 0.2765248.
```

### 4.4 LightGBM

Because I was very new to LightGBM when I began, I did a grid search on the following grid

```
param_grid = {
    'boosting_type': ['gbdt','dart','goss'],
    'learning_rate': np.linspace(.01,1,10),
    'n_estimators': np.arange(100,1100,100),
    'max_depth': np.arange(0,10)
}
```

and I found the following best parameters

```
Boosting Type = gbdt
Learning Rate = 0.23
Number of Estimators = 1000
Maximum Depth = 0.
```

Note that a max depth <= 0 indicates that the depth is unbounded. Searching over this grid of size 4000 is something that I would have never even tried for XGBoost or any other GBDT method, though it still did take slightly more than 20 hours to fit

After this I did a grid search on the following grid

```
param_grid = {
    'reg_alpha': np.linspace(.1,1,5),
    'reg_lambda': np.linspace(.1,1,5)
}
```

and found

```
L1 Regularization = 0.1
L2 Regularization = 0.1
```

LightGBM scored much better than any of the previous methods that I used. For the sake of training time, the model that produced this score only used 500 estimators instead of the planned 1000. This makes me hopeful that models with more estimators could be much more successful than this one.

```
score on a test size of 5000000 is 0.4282534.
```

### 4.5 Feature Importance

Finally we will examine the different assigned feature importance. They are as follows:

Figure 2: Feature importance, determined by each machine learning method[2]

| | Random Forest | Gradient Boosted | XGBoost | LightGBM |
|---|---|---|---|---|
| Gender | 0.009 | 0.01 | 0.320 | 104 |
| Age | 0.280 | 0.275 | 0.130 | 4034 |
| Offense | 0.229 | 0.214 | 0.140 | 3835 |
| Facility | 0.149 | 0.177 | 0.100 | 2147 |
| Detainer | 0.027 | 0.031 | 0.170 | 520 |
| Sentence Days | 0.307 | 0.293 | 0.130 | 4360 |

## 5 Analysis

LightGBM was a breakthrough in determining if machine learning could be used to counter bias within criminal justice systems. It scored much better than I expected, especially given the success of the other methods. The fact that any machine learning algorithm can correctly classify this data significantly better than chance leads to important conclusions. There are certainly proxies for race within this data set. This is both exciting and disturbing. Exciting because it means that racial bias is certainly quantifiable and therefore can be addressed in direct, measurable ways; disturbing because of what it implies about the nature of reality.

The existence of proxies for race within this data set strongly supports other research into the state of bias in the criminal justice system. The feature importance reported by the models is important to understanding the dimension of this bias[3]. According to the LightGBM report, age, offense, and sentence length are the most important features to consider when attempting to determine the race of an inmate. This also means that we will find the greatest racial bias in these aspects of the criminal justice system.

[2] Note that LightGBM records feature importance differently from the other methods. The rule, however, is still the same: the higher the number the greater the importance.

[3] Since LightGBM reported the best score I will primarily use its feature importance rating, however it is important to note that only XGBoost disagrees with LightGBM's feature rankings

There are countless studies that record racial disparities in arrest rates for certain offenses. One of the most well documented is drug related crimes. As far back as 1995 the Bureau of Justice Statistics reports that blacks are well over represented in drug related arrests[3]. In fact in 2015 the BJS reported that 52% of inmates are convicted of a drug related offense, and 38% of those inmates are black, supporting the 1995 claim[7].

Some may not expect age to be as important as it is. This reflects how poorly the general population understands the depth of the racial injustices that exist in the American criminal justice system. The Sentencing Project reported in 2017 that black children are 5 times more likely to be detained or committed than white children[1]. This little known disparity becomes a major proxy for race in inmate data and this report therefore confirms these findings.

The most damning result from the feature importance is the importance of sentence length. Sentence length is overwhelmingly the most important feature to examine when attempting to determine the race of an inmate. Myriad studies[4] and reports[5] describe the existence of this disparity and attempt to quantify it, including a report that I wrote last year using this very same data set. This well documented fact becomes the most important feature in building a race proxy in the data.

## 6 Conclusion

The process of realizing that I was successful in finding a method to classify inmates by race was full of conflict. I was happy and excited that my model worked, and much better than I expected at that. What came after though was a firm sense of sorrow. Yes, I was successful, but what does it say about that world that I live in? I submit that this report outlines areas within the criminal justice system where we, as Americans, ought to focus our efforts. There are many aspects the inequality, and simply treating the symptoms of racism will not solve every problem, however well defined domains and goals do a lot to providing stability in progress. These racial disparities and biases are affecting millions of Americans; it is our duty to work to eliminate racism, and addressing its effects is a good place to start.

## References

[1] *FACT SHEET: BLACK DISPARITIES IN YOUTH INCARCERATION.* 2017. URL: https://www.sentencingproject.org/wp-content/uploads/2017/09/Black-Disparities-in-Youth-Incarceration.pdf.

[2] *LightGBM:A Highly Efficient Gradient Boosting Decision Tree.* Advances in Neural Information Processing Systems 30 (NIPS 2017). 2017. URL: http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf.

[3] Patrick A. Langham. *The Racial Disparity in U.S. Drug Arrests.* 1995. URL: https://www.bjs.gov/content/pub/pdf/rdusda.pdf.

[4] David B. Mustard. "Racial, Ethnic, and Gender Disparities in Sentencing: Evidence from the U.S. Federal Courts." In: (2001). DOI: 10.1086/320276.

[5] *Report of The Sentencing Project to the United Nations Special Rapporteur on Contemporary Forms of Racism, Racial Discrimination, Xenophobia, and Related Intolerance*. 2018.

[6] *Statement of Concern About Predictive Policing by ACLU and 16 Civil Rights Privacy, Racial Justice, and Technology Organizations. (n.d.).* URL: https://www.aclu.org/other/statement-concern-about-predictive-policing-aclu-and-16-civil-rights-privacy-racial-justice. Accessed Apr 13, 2020.

[7] Sam Taxy, Julie Samuels, and William Adams. *Drug Offenders in Federal Prison: Estimates of Characteristics Based on Linked Data.* 2015. URL: https://www.bjs.gov/content/pub/pdf/dofp12.pdf.

[8] Ethan Walker. "Race and Incarceration in America". In: (2019). URL: https://github.com/EthanMWalker/incarceration_research/blob/master/tex/Race%5C%20and%5C%20Incarceration%5C%20in%5C%20America.pdf.

## Annotations

[3]This report uses data and analysis of the time to understand the racial disparity in drug arrests. The main point of information that I am using is the figure quoted on page 7, claiming that the racial representation disparity is between 23% and 13%. Blacks, at the time made up 36% of drug related arrests but only represented 13% of drug users. The 26% figure comes from an analysis of drug related behavior that lead to a greater chance of arrest, to which black drug users are partial.