Recidivism Risk: Algorithmic Prediction and Racial Bias

CS109 Project

YouTube Link: https://youtu.be/iPDPrKbT4ug

Sauren Khosla

March 2020

Introduction

Recidivism, defined as "the tendency of a convicted criminal to reoffend," [1] is an important topic to consider when evaluating the effectiveness of our criminal justice system. More specifically, the recidivism *rate* – the *rate* at which criminals reoffend – as a metric can reveal a lot about core criminal justice programs that involve deterrence and rehabilitation. In fact, the United States government itself evaluates the quality of the prison system based on the recidivism rate.[2] Unfortunately, the United States has one of the highest recidivism rates out of any country on the planet.[3] In comparison to Norway, boasting an impressive 20% recidivism rate, the United States stands at an embarrassing 76.6% re-arrest rate within just five years of release. Clearly, the United States has much to discuss when it comes to recidivism rates. Many reforms to the prison system have been floated about, one of which has been widely implemented already.

In 2018, California (and many other states) ended cash bail – a system that allowed defendants to avoid jail while awaiting trial by paying a fine – and instead replaced it with something that may prove to be even more damning: algorithmic profiling, the process by which an algorithm determines whether someone is "at risk" of recidivating and is then issued a decision based on

[1] https://www.lexico.com/en/definition/recidivism
[2] https://www.bjs.gov/content/pub/pdf/18upr9yfup0514.pdf
[3] https://www.businessinsider.com/why-norways-prison-system-is-so-successful-2014-12

that estimation.[4] As opposed to posting cash bail, defendants are now algorithmically assigned a "risk of recidivism" score through a system developed by private companies. People are classified as "low-risk", "medium-risk", or "high-risk" depending on a number of factors. "Low-risk" individuals are likely to be released until their court date; "medium-risk" individuals may be released depending on local laws; "high-risk" individuals are likely to be jailed. Ultimately, however, the burden of decision falls upon the judge overseeing the case. The algorithm provides assessments based on past defendants and their tendencies and attributes, then assigns a value to the current defendant based on matching characteristics and probability. There are a number of arguments in favor of cash bail relative to algorithmic profiling and vice versa. I outline a few here, first against the use of cash bail.

One argument against cash bail is that it can appear arbitrary. The judge takes the factors of the trial into account and dishes out a decision based on their intuition, feel for the situation, and likely with a bit of past precedence in mind. However, this system is by no means deterministic nor objective – it leaves plenty of room for variation between defendants. Secondly, the system can be discriminatory with regards to income. Some may be forced into pretrial jail simply because they cannot afford cash bail. This creates another avenue by which income inequality can be exacerbated in a society in which inequality already runs rampant.

On the other hand, one advantage of cash bail is that it is more easily scrutinized because there is a human-to-human component taking place during the issuing of the bail. In other words, it is much easier to question the decision of a human than it is to question the decision of a machine. There is no scapegoat that a judge may point to in a world absent algorithmic decision-making – the blame for the decision falls entirely on them, leaving them with the sole responsibility for the final decision. Secondly, if people have sufficient cash, cash bail is a relatively easy mechanism by which someone can temporarily escape jail: the process is simple and can be quickly transacted. Finally, "objectively" or algorithmically deciding if someone can make bail may potentially have dangerous consequences, including allowing systemic racial bias to permeate yet another facet of the criminal justice system. This final argument is where I will focus my efforts in this paper and project.

These algorithmic "risk assessments" outlined above have expanded quickly

[4] https://www.thenation.com/article/archive/california-ended-cash-bail-but-may-have-replaced-it-with-something-even-worse/

to a number of states across the United States.[5] Fortunately, or unfortunately, these algorithms have come under heavy fire for their potentially dangerous implications for people of color.[6]

One study conducted a meta-analysis and found that most studies do not even consider validity of predictions, and that a lot of studies were conducted by the same people that wrote the algorithms.[7] Other studies have explored the racial biases in these private-company-generated algorithms. For example, Skeem and Lowenkamp found that African-Americans were more likely to be classified as "at risk of recidivism", but the difference in risk assessment was not attributable to race.[8] The largest and most reputable assessment, however, comes from Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin of ProPublica in May of 2016. They analyzed the COMPAS (Correctional Offender Management Profiling for Alternative Sanctions) algorithm using a data set of $10,000$ criminal defendants in Broward County, Florida. They concluded:

> Black defendants were often predicted to be at a higher risk of recidivism than they actually were. Our analysis found that black defendants who did not recidivate over a two-year period were nearly twice as likely to be misclassified as higher risk compared to their white counterparts (45 percent vs. 23 percent).
>
> White defendants were often predicted to be less risky than they were. Our analysis found that white defendants who re-offended within the next two years were mistakenly labeled low risk almost twice as often as black re-offenders (48 percent vs. 28 percent).
>
> The analysis also showed that even when controlling for prior crimes, future recidivism, age, and gender, black defendants were 45 percent more likely to be assigned higher risk scores than white defendants.[9]

Their analysis was restricted to the COMPAS algorithm and scores given to each of the defendants in Broward County. To get a better understanding of how the scores are assessed, we should dissect the COMPAS algorithm itself. The authors of the COMPAS algorithm write that for the Violent Recidivism Risk Scale, the algorithm takes into account five main characteristics:

1. History of Noncompliance Scale

2. Vocational Education Scale

[5] https://www.vox.com/future-perfect/2018/10/17/17955306/bail-reform-criminal-justice-inequality

[6] https://www.themarshallproject.org/2015/08/04/the-new-science-of-sentencing

[7] https://csgjusticecenter.org/wp-content/uploads/2020/02/Risk-Assessment-Instruments-Validated-and-Implemented-in-Correctional-Settings-in-the-United-States.pdf

[8] https://ssrn.com/abstract=2687339

[9] https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

3. Current age
4. Age-at-first-arrest
5. History of Violence Scale

Each item in this list is then multiplied by a weight, where the size of the weight is "determined by the strength of the item's relationship to person offense recidivism that we observed in our study data." The components are then weighted and added together to calculate the overall score:

$$\begin{aligned}\text{Violent Recidivism Risk Score} = &(\text{Age} \cdot -w_0) \\ &+ (\text{Age-at-first-arrest} \cdot -w_1) \\ &+ (\text{History of Violence Score} \cdot w_2) \\ &+ (\text{Vocational Education Scale} \cdot w_3) \\ &+ (\text{History of Noncompliance Score} \cdot w_4)\end{aligned}$$

Their justification for using these specific categories includes: first, other researchers in criminal justice have found that these five categories are good indicators of violent recidivism and second, insurance companies conduct similar analyses when predicting the risk of their customers of being engaged in an accident.[10] This justification in-and-of-itself may prove to be problematic, as some of these statistics may be skewed at birth depending on the race of the defendant. This problem is the one I intend to explore.

In this paper, using the same (though slightly-filtered) data set as the authors of the ProPublica study (looking at $10,000$ defendants in Broward County, Florida), I will explore the racial bias associated with common algorithmic approaches to classifying individuals at risk of recidivism independent of the COMPAS algorithm. My results can then be used to identify whether the "risk of recidivism" assessment that is now commonly used to dictate whether individuals will "make bail" is problematic specifically when it comes to the COMPAS algorithm, or whether the system and process itself is inherently flawed. I find that, using three different algorithmic approaches, African-Americans are, on average (across the three approaches), approximately 1.44x more likely than Caucasians to be falsely classified as someone that will recidivate, while Caucasians are, on average, 1.34x more likely to be falsely classified as NOT going to recidivate in comparison to African-Americans.

[10] https://assets.documentcloud.org/documents/2840784/Practitioner-s-Guide-to-COMPAS-Core.pdf

Data Set and Cleaning

The data set I use comes from the Broward County, Florida data used in Larson & Mattu's paper. The data set contains information on thousands of violent crime perpetrators. I chose to use the violent crime data set as opposed to the non-violent crime data set since violent crimes tend to be associated with harsher sentencing, so we must be even more careful in our assessment of these defendants. Hopefully my work will shed some light on the changes to the system that must take place to bring us closer to making just decisions in the criminal justice system.

The violent crime data collected includes: name, COMPAS screening date, sex, date of birth, age, age category, race, juvenile felony count, decile score, juvenile misdemeanor count, priors count, time of jail entrance, time of jail departure, case number, offense date, charge degree, whether they recidivated, charge description, risk assessment, and more. For the purposes of my investigation, I focus on factors that may clearly influence recidivism rates. This includes juvenile felony count, juvenile misdemeanor count, priors count, charge degree, and whether they recidivated (which will be excluded and revisited). I choose to ignore sex, age category, and race, as all three of these are protected characteristics, a concept which we will revisit in the discussion section (while they may appear in the data table, they are not used in any calculations until the race-aware section).

I choose to eliminate rows where the recidivism category contains a value of -1, as it does not reflect the binary of whether or not someone recidivated. I also choose to drop rows in which the charge degree is F7, F6, F5, F4, M7, M6, M5, or M4 as the number of rows containing these values is not significant enough to create an accurate prediction and these values are often associated with an arrest case absent an actual charge. I end up not including M3 either, as none of the rows contain M3 as their charge degree.

For the other variables, I turn the values into one-hot encodings after bucketing the values. For example, consider the data column that gives the number of prior offenses the person committed. This could range from 0 to as high as 19. I turned this one column into four: zero priors, one prior, two priors and three-or-more priors. A person would have a 1 in the column that corresponds to their prior count, and a 0 in all the others.

Guiding Questions

Through my data analysis, I hope to answer a few questions pertaining to recidivism rates and the criminal justice system. These include:

- Is the issue with the "risk of recidivism" assessment specific to the COMPAS algorithm, or is the process inherently flawed?
- Is crime data biased?
- How can we reform the bail process to maximize justice?

While I will likely be unable to provide a comprehensive policy suggestion that will entirely fix the criminal justice system, I hope that my investigation can shed some light on the work that needs to be done to reform the system.

Brief Data Visualization

Following my data cleaning, below is a picture of the DataFrame:

	is_recid	sex	race	age_cat	juv_fel_count_0	juv_fel_count_1	juv_fel_count_2plus	juv_misd_count_0	juv_misd_count_1	juv_misd_count_2plus
0	0	Male	Other	Greater than 45	1.0	0.0	0.0	1.0	0.0	0.0
1	0	Male	Other	Greater than 45	1.0	0.0	0.0	1.0	0.0	0.0
3	1	Male	African-American	25 - 45	1.0	0.0	0.0	1.0	0.0	0.0
4	1	Male	African-American	Less than 25	1.0	0.0	0.0	1.0	0.0	0.0
5	1	Male	African-American	Less than 25	1.0	0.0	0.0	1.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...
18311	0	Female	African-American	25 - 45	1.0	0.0	0.0	1.0	0.0	0.0
18312	0	Male	Other	Greater than 45	1.0	0.0	0.0	1.0	0.0	0.0
18313	0	Female	African-American	25 - 45	1.0	0.0	0.0	1.0	0.0	0.0
18314	1	Female	Hispanic	Less than 25	1.0	0.0	0.0	1.0	0.0	0.0
18315	1	Female	Hispanic	Less than 25	1.0	0.0	0.0	1.0	0.0	0.0

juv_misd_count_0	juv_misd_count_1	juv_misd_count_2plus	priors_count_0	priors_count_1	priors_count_2	priors_count_3plus	M1	M2	F1	F2	F3
1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0
...	...	...	...	...	...	...	...	...	...	...	...
1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0
1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0

Technical Analysis

Naïve Bayes Analysis

We can now begin the data analysis. I start by implementing the Naïve Bayes algorithm. As mentioned, we drop the "race", "sex", and "age" columns to ensure our data from here on out is personal information-blind.

We want to next use Naïve Bayes to calculate the probability someone will recidivate. Specifically, we will want to calculate

$$P(R|X)$$

where X is all of the attributes we are considering (and R denotes recidivism – NR will denote not recidivating). By using Bayes' Theorem, we have that

$$P(R|X) = \frac{P(R)P(X|R)}{P(X)}.$$

However, it is important to note that

$$P(NR|X) = \frac{P(NR)P(X|NR)}{P(X)}.$$

Therefore, when comparing these two probabilities to make a decision, we need not worry about the denominator, since it is the same in both cases. So, we turn our attention to the numerator.

The numerator is the same as the joint probability: $P(R)P(X|R) = P(X, R)$. Using the chain-rule, we can write this as

$$P(X, R) = P(X_1|X_{2:m}, R)P(X_2|X_{3:m}, R) \cdots P(X_m|R)P(R)$$

where $X_{k:m}$ denotes the kth through mth component of X (and m is the total number of features). However, calculating each of these distributions is computationally infeasible. Since each of our random variables is binary, if we condition on k things, then to describe that conditional probability distribution we would need to keep track of 2^k different values (for a given configuration of the conditional variables, of which there are 2^k, we need to keep track of the probability that the query variable is a 1). Summing from $k = 0$ to $k = m$ makes this explode in the number of parameters we would have to remember.

To simplify this, we make the Naïve-Bayes assumption: conditioning on class

(recidivate or not recidivate), the features are independent. Therefore, we can write $P(X_l|X_{l+1:m,R}) = P(X_l|R)$. All we must do is calculate the prior, $P(R)$, as well as $P(X_i|R)$ for each feature X_i (as well as conditioning on NR).

Before doing anything, we split our data into training and testing datasets, using an 80-20 split.

```
NUM_TRAINING_EXAMPLES = int(.8 * cleaned_data.shape[0])
```

Next, we calculate the probability that someone recidivates from our training data. Since all of our values are either 0 or 1, this is relatively easy:

```
recid_total = training_target.sum()
p_recid = recid_total / training_target.shape[0]
# p_recid = 0.4816507107629823
```

To calculate the rest of the probabilities, we split our training set into those that did recidivate, and those that did not. We want to calculate the probability of each column being "1" within each table:

```
recid_dict = dict()
for (columnName, columnData) in recid_table.iteritems():
    total = columnData.sum()
    recid_dict[columnName] = total / recid_table.shape[0]
# recid_dict:
'''
{'is_recid': 1.0,
 'juv_fel_count_0': 0.9320885408823972,
 'juv_fel_count_1': 0.04411986146664659,
 'juv_fel_count_2plus': 0.023791597650956182,
 'juv_misd_count_0': 0.9060382472519198,
 'juv_misd_count_1': 0.062038849570847765,
 'juv_misd_count_2plus': 0.03192290317723234,
 'priors_count_0': 0.16277669025749134,
 'priors_count_1': 0.14817045625658787,
 'priors_count_2': 0.1130853787080259,
 'priors_count_3plus': 0.5759674747778949,
 'M1': 0.18732118656828792,
 'M2': 0.06941725643728354,
 'F1': 0.00933594338202078,
 'F2': 0.0828188525824424,
 'F3': 0.6392109622044873}
'''
```

We must now calculate the conditional probabilities for the people that did not recidivate (i.e. $P(X_i = x|NR)$. The process is analogous:

```
no_recid_dict = dict()
for (columnName, columnData) in no_recid_table.iteritems():
    total = columnData.sum()
    no_recid_dict[columnName] = total / no_recid_table.shape[0]
# no_recid_dict:
'''
```

```
{'is_recid': 0.0,
 'juv_fel_count_0': 0.9713166363509165,
 'juv_fel_count_1': 0.01874912550720582,
 'juv_fel_count_2plus': 0.00993423814187771,
 'juv_misd_count_0': 0.9641807751504128,
 'juv_misd_count_1': 0.02742409402546523,
 'juv_misd_count_2plus': 0.008395130824122009,
 'priors_count_0': 0.3321673429410942,
 'priors_count_1': 0.21631453756821042,
 'priors_count_2': 0.12228907233804394,
 'priors_count_3plus': 0.32922904715265144,
 'M1': 0.23352455575766057,
 'M2': 0.07709528473485379,
 'F1': 0.024625717084091225,
 'F2': 0.0910871694417238,
 'F3': 0.5638729536868616}
'''
```

We now have all of the data we need to calculate the probabilities we want. The information we require is stored in recid_dict, no_recid_dict and p_recid. As done in lecture, we will apply logarithm rules to make computation cleaner. Our expressions are:

$$P(R)\Pi_i P(X_i = x|R)$$

$$P(NR)\Pi_i P(Y_i = y|NR)$$

Applying the logarithm gives us:

$$\log(P(R)) + \sum_i \log(P(X_i = x|R))$$

$$\log(P(NR)) + \sum_i \log(P(Y_i = y|NR))$$

If the first expression is greater, we label the person as going to recidivate, whereas if the latter expression is greater, we label the person as not going to recidivate. We turn this into code as follows:

```
risks = dict()
first_term = np.log(p_recid)
second_term = np.log(1 - p_recid)
for (index, row) in testing_data.iterrows():
    cols = row.index
    recid_sum = first_term
    for i in range(len(row)):
        if row[i] > 0:
            recid_sum += np.log(recid_dict[cols[i]])
        else:
            recid_sum += np.log(1 - recid_dict[cols[i]])

```

```
        no_recid_sum = second_term
        for i in range(len(row)):
            if row[i] > 0:
                no_recid_sum += np.log(no_recid_dict[cols[i]])
            else:
                no_recid_sum += np.log(1 - no_recid_dict[cols[i]])

        if (recid_sum > no_recid_sum):
            risks[index] = True
        else:
            risks[index] = False
```

Now that we have categorized people based on their risk of recidivism and stored the answers in our risks dictionary, we can finally answer the question: is this algorithm racially biased? We will now analyze how the race-blind algorithm categorizes people by looking at race. We want to categorize our results for true positives, true negatives, false positives, and false negatives so that we may calculate the false positive rate and false negative rate for African-Americans versus Caucasians.

```
# True positive, true negative, false positive, false negative
Caucasians = [0, 0, 0, 0] # Initially
African_Americans = [0, 0, 0, 0] # Initially
```

False positive rate is calculated as

$$\frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}.$$

Hence, we calculate:

```
AA_NB_FP_Rate = African_Americans[2] / (African_Americans[2] + ↘
    African_Americans[1])
Caucasian_NB_FP_Rate = Caucasians[2] / (Caucasians[2] + Caucasians[1])
AA_vs_Caucasian_NB_FP = AA_NB_FP_Rate/Caucasian_NB_FP_Rate
# AA_vs_Caucasian_NB_FP = 1.4117032392894462
```

False negative rate is calculated as

$$\frac{\text{False Negatives}}{\text{False Negatives} + \text{True Positives}}.$$

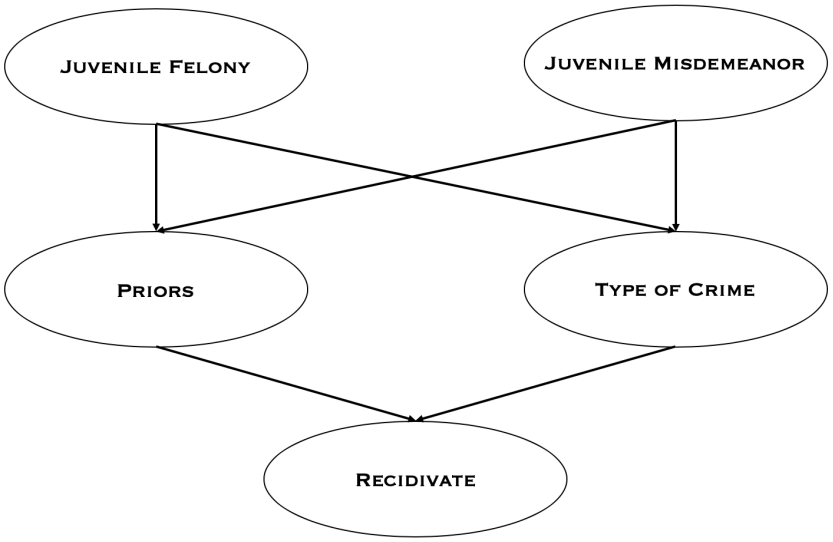
Hence, we calculate:

```
AA_NB_FN_Rate = African_Americans[3] / (African_Americans[3] + ↘
    African_Americans[0])
Caucasian_NB_FN_Rate = Caucasians[3] / (Caucasians[3] + Caucasians[0])
Caucasian_vs_AA_NB_FN = Caucasian_NB_FN_Rate/AA_NB_FN_Rate
# Caucasian_vs_AA_NB_FN = 1.3314986270078493
```

This indicates that, using the Naïve Bayes algorithm, someone that is African-American is 1.41x more likely to be falsely classified as someone that is going to recidivate, while someone that is Caucasian is 1.33x more likely to be falsely classified as someone that is NOT going to recidivate.

Bayesian Network Analysis

We now turn to creating a Bayesian Network to conduct a similar analysis. Our Bayesian Network is constructed as follows:



We start by re-cleaning our data to better fit our model. In this case, we change a few parameters: if the defendant has any juvenile felonies, we assign their juvenile felony value as 1 – otherwise it is 0. This is analogous for the juvenile misdemeanor value. For prior count, if the defendant has 0 priors, we assign them a value of 0 here; if they have 1 prior, we assign them a value of 1; if they have 2 priors, we assign them a value of 2; if they have more than 2 priors, we assign them a value of 3. We assign the type of crime value depending on the crime type (M1, M2, F1, F2, or F3). If the defendant is likely to recidivate, we assign their recidivate value as 1 – otherwise it is 0.

We now have to calculate the appropriate probability distributions. For the sake of notation, we write juvenile felony as F, juvenile misdemeanor as M,

priors as C (to avoid confusion with P for probability – think "priors count"), type of crime as T, and recidivate as R. We need to calculate the following probability distributions:

$$P(R = r|C = c, T = t)$$
$$P(C = c|F = f, M = m)$$
$$P(T = t|F = f, M = m)$$
$$P(F = f)$$
$$P(M = m)$$

We will store these probability distributions in dictionaries that map the parent values to the appropriate probability distributions. The probability distributions are calculated in a manner very similar to that in the first section, just taking extra care to condition on all the different combinations of a node's parents.

Now that we have all of the probability distributions we require, we can begin calculating changes of recidivism for our testing data. We store the risks in a dictionary for each index in our testing data:

```
BN_risks = dict()
```

For each row in our testing data, we calculate the probability that the defendant will recidivate by multiplying the conditional probabilities outlined previously.

```
for (index, row) in BN_testing_data.iterrows():
    juv_fel = row['juv_fel']
    juv_misd = row['juv_misd']
    priors = row['priors']
    crime_type = row['crime_type']
    probs = []
    if juv_fel == 1:
        probs.append(juv_fel_prob)
    else:
        probs.append(1 - juv_fel_prob)
    if juv_misd == 1:
        probs.append(juv_misd_prob)
    else:
        probs.append(1 - juv_misd_prob)
    crime_dist = type_poss[(juv_fel, juv_misd)]
    crime_p = crime_dist[crime_type]
    probs.append(crime_p)
    priors_dist = priors_poss[(juv_fel, juv_misd)]
    priors_p = priors_dist[priors]
    probs.append(priors_p)

    recid_dist = recid_poss[(priors, crime_type)]
```

```
        recid_p = recid_dist[1]
        no_recid_p = recid_dist[0]

        total = 1
        for elem in probs:
            total *= elem

        will_recid = total * recid_p
        will_not_recid = total * no_recid_p

        if will_recid > will_not_recid:
            BN_risks[index] = True
        else:
            BN_risks[index] = False
```

Note that this calculation is actually just comparing recid_p and no_recid_p because the only nodes that influence the bottom node are priors and crime type since we have all the data. Juvenile felony and juvenile misdemeanor do not directly impact the probability someone recidivates given that we have the values for priors and crime type, but the structure of our Bayesian Network is sound: it is, firstly, temporal, as it tracks someone's behavior over time (juvenile crime through all of their priors up until their current defense case), and, secondly, if there were missing data, I could utilize exact inference to predict the values of the missing data.

Since we have now categorized people based on their risk of recidivism using our Bayesian Network, we can again answer the question: is this algorithm racially biased? We will now analyze how the race-blind algorithm categorizes people by looking at race. Similarly to above, we will look at the false positive rate and false negative rate:

```
# True positive, true negative, false positive, false negative
BN_Caucasians = [0, 0, 0, 0] # Initially
BN_African_Americans = [0, 0, 0, 0] # Initially
```

Using our lists of true positive, true negative, false positive, and false negative values, we can calculate the racial bias factors.

```
AA_BN_FP_Rate = BN_African_Americans[2] / (BN_African_Americans[2] + ↘
    BN_African_Americans[1])
Caucasian_BN_FP_Rate = BN_Caucasians[2] / (BN_Caucasians[2] + ↘
    BN_Caucasians[1])
AA_vs_Caucasian_BN_FP = AA_BN_FP_Rate/Caucasian_BN_FP_Rate
# AA_vs_Caucasian_BN_FP returns 1.3880855986119143

AA_BN_FN_Rate = BN_African_Americans[3] / (BN_African_Americans[3] + ↘
    BN_African_Americans[0])
Caucasian_BN_FN_Rate = BN_Caucasians[3] / (BN_Caucasians[3] + ↘
    BN_Caucasians[0])
Caucasian_vs_AA_BN_FN = Caucasian_BN_FN_Rate / AA_BN_FN_Rate
```

```
9  # Caucasian_vs_AA_BN_FN = 1.3055740088492247
```

This indicates that African-Americans are 1.39x more likely to be falsely classified as going to recidivate and that Caucasians are 1.31x more likely to be falsely classified as NOT going to recidivate under this Bayesian Network model.

Logistic Regression Analysis

For the sake of completeness, I include a logistic regression analysis of the data in this section. Logistic regression is another classification algorithm that is widely used because of its simplicity. The goal of logistic regression is to find a weight vector $w \in \mathbb{R}^m$ so that $\sigma(w^T X)$ can accurately predict whether or not someone will recidivate, where $\sigma(z) = \frac{1}{1+e^{-z}}$, called the sigmoid function. Without going too much into detail, we give a brief overview of how this algorithm is trained. As just mentioned, we assume that $P(R|X) = \sigma(w^T X)$. Since we are working with a binary variable, we can write that

$$P(R = r|X) = \sigma(w^T X)^r (1 - \sigma(w^T X))^{1-r}.$$

Therefore, the likelihood of the data as a function of our weight vector w is given by

$$L(w) = \Pi_{i=1}^{N} \sigma(w^T X^{(i)})^{r^{(i)}} (1 - \sigma(w^T X^{(i)}))^{1-r^{(I)}}.$$

We want to find the w that maximizes this likelihood. However, instead, we can maximize the log-likehilood, since the log function is a monotonic function (maximizing log-likelihood finds the same result as maximizing likelihood). As such, we wish to find w that maximizes

$$LL(w) = \sum_{i=1}^{N} r^{(i)} \log \sigma(w^T X) + (1 - r^{(i)}) \log(1 - \sigma(w^T X)).$$

A very common way to optimize a differentiable function such as this one is gradient ascent. For some number of iterations, we will calculate the gradient of LL with respect to our current vector w, then take a small step in that direction. Given enough time, we will converge and get our best w vector.

We first define our targets – the is_recid column:

```
1  targets = cleaned_data['is_recid']
```

We now pull all of the data after the personal information in cleaned_data into DataFrames containing our testing data and our training data.

We conduct a logistic regression analysis for completeness using scikitlearn's logistic regression model and calculate the score of our testing data.

```
lr = LogisticRegression()
lr.fit(training_data, training_target)
lr.score(testing_data, testing_target)
# returns 0.6255800464037123
```

Now that we've trained our model and assessed our testing data, we can analyze how this model fared for African-Americans versus Caucasians using a confusion matrix[11] to calculate the false positive rate and false negative rate for each group.

```
AA_Matrix = confusion_matrix(african_americans_target, lr.predict(
    african_americans_data))
Caucasion_Matrix = confusion_matrix(caucasians_target, lr.predict(
    caucasians_data))
AA_FP_Rate = AA_Matrix[0, 1]/(AA_Matrix[0, 0] + AA_Matrix[0, 1])
Caucasian_FP_Rate = Caucasion_Matrix[0, 1]/(Caucasion_Matrix[0, 0] +
    Caucasion_Matrix[0, 1])
AA_vs_Caucasian_FP = AA_FP_Rate/Caucasian_FP_Rate
AA_vs_Caucasian_FP
# AA_vs_Caucasian_FP = 1.5290517508521848

AA_FN_Rate = AA_Matrix[1, 0]/(AA_Matrix[1, 1] + AA_Matrix[1, 0])
Caucasian_FN_Rate = Caucasian_Matrix[1, 0]/(Caucasian_Matrix[1, 1] +
    Caucasian_Matrix[1, 0])
Caucasian_vs_AA_FN = Caucasian_FN_Rate / AA_FN_Rate
# Caucasian_vs_AA_FN = 1.3784975034750553
```

This tells us that African-Americans are 1.53x more likely than Caucasians to be incorrectly classified as going to recidivate and Caucasians are 1.38x more likely to be incorrectly classified as NOT going to recidivate.

Race-Aware Logistic Regression Analysis

We want to see if taking race into account when training our model will significantly alter the racial bias factor in our predictions. Using the same cleaned_data DataFrame as above, we can create a DataFrame that contains the same information as previously, but additionally contains a column for race.

We want to transform our data set such that we have only binary values for each column (only race is not already like this), so we call get_dummies:

```
new_training_data = pd.get_dummies(training_data)
new_testing_data = pd.get_dummies(testing_data)
```

[11] https://en.wikipedia.org/wiki/Confusion_matrix

Our DataFrame now contains a binary value in each column (including the individual race columns: African-American, Asian, Caucasian, Hispanic, Native American, Other). We call upon the logistic regression model from before:

```
lr = LogisticRegression()
lr.fit(new_training_data, training_target)
```

Now, we analyze the results (predictions) of training with a dataset that involves race.

```
prediction = lr.predict(african_americans)
c_prediction = lr.predict(caucasians)
c_true_recid = c_test_prediction['is_recid']
true_recid = test_prediction['is_recid']
AA_Matrix = confusion_matrix(true_recid, prediction)
C_Matrix = confusion_matrix(c_true_recid, c_prediction)
```

Again, we calculate the false positive rate and false negative rate for African-Americans versus Caucasians:

```
AA_FP_Rate = AA_Matrix[0, 1]/(AA_Matrix[0, 0] + AA_Matrix[0, 1])
C_FP_Rate = C_Matrix[0, 1]/(C_Matrix[0, 0] + C_Matrix[0, 1])
AA_vs_Caucasian_FP = AA_FP_Rate/C_FP_Rate
# AA_vs_Caucasian_FP returns 1.575844202251206

AA_FN_Rate = AA_Matrix[1, 0]/(AA_Matrix[1, 1] + AA_Matrix[1, 0])
C_FN_Rate = C_Matrix[1, 0]/(C_Matrix[1, 1] + C_Matrix[1, 0])
Caucasian_vs_AA_FN = C_FN_Rate/AA_FN_Rate
# Caucasian_vs_AA_FN = 1.7435776537332501
```

This model falsely classifies African-Americans as going to recidivate at 1.58x the rate of Caucasians and falsely classifies Caucasians as NOT going to recidivate at 1.74x the rate of African-Americans. This may indicate that even when two people have the same attributes aside from race, they're more likely to be labeled as someone that is going to recidivate if they are African-American.

Discussion

Model Shortcomings

While my results are indeed comprehensive in that they take into account almost all of the factors that might play into whether someone will recidivate in the data set, there are still a few shortcomings I'd like to address, given sufficient time. First, with regards to the Bayesian Network and Naïve Bayes models, the "priors count" and "juvenile misdemeanor/felony" columns have a causal relationship. If the defendant has already committed a juvenile misdemeanor/felony, this is noted in their priors count. This means that if the

defendant has committed both a juvenile misdemeanor and a juvenile felony, then the value of their "priors count" is restricted to 2 or 2+, or if either juvenile misdemeanor or juvenile felony is 1, then the "priors count" cannot possibly be 0. What this means for my Bayesian analyses is that the relationship is not necessarily independent (or simply correlation).

In my results, I find that taking race into account reveals that the chance an African-American is falsely classified as going to recidivate is 1.58x the chance a Caucasian is falsely classified as going to recidivate. In the other, race-blind models, the chance of false-classification for an African-American is still more likely than the false-classification of a Caucasian (the same analysis applies to the false negative rate). Since the false-classification multiplier between races is not reduced to 1 when we do not account for race (meaning not accounting for race does not completely nullify the racial bias in the data set), it is possible that another variable(s) is a pseudo-race indicator. In other words, there could be a category (or categories) that has a high correlation with race, so even if we eliminate race from the equation, we are still leaving a footprint of the defendant's race in the data set. In order to investigate this, I look at the correlation between all of the columns.

```
1  correlation_cleaned_data = pd.get_dummies(cleaned_data)
2  correlation_cleaned_data.corr()
```

Doing so reveals that the race_African-American column and the priors_count_3plus column have a correlation of 0.18, which is relatively high compared to other columns' correlations. Unfortunately, this only examines correlation between every pair of columns when, in fact, the pseudo-race indicator could very well be comprised of a linear combination of columns. Future work should entail taking these factors into account, making sure to assess the potential multicollinearity[12] in the data to better understand the ways in which the data collected is racially-biased.

Lastly, while the data analysis I do strictly compares African-Americans and Caucasians, I could potentially conduct similar analysis comparing other races.

Racial Bias and Fairness

To best create "fair" machine learning models, we must first define the concept of "fairness." Over the years, mathematical definitions of "fairness" have been proposed, which often contain, but are not limited to, the following:

[12] https://en.wikipedia.org/wiki/Multicollinearity

- anti-classification: protected attributes, such as race, age, sex, gender, and other personal characteristics, should not be a factor in making machine-based decisions
- classification parity: common measures of prediction (such as false positive rates and false negative rates, as is discussed in this paper) should not vary across groups defined by protected attributes
- calibration: "conditional on risk estimates, outcomes are independent of protected attributes"[13]

In their paper, *The Measure and Mismeasure of Fairness: A Critical Review of Fair Machine Learning*,[14] Davies and Goel contend that these components of fairness "suffer from significant statistical limitations" in that it is preferable to treat "similarly risky people similarly," without concern for "fairly" taking account of protected characteristics. While this may seem best in theory, my data analysis and paper demonstrates the danger of this theory in practice. There are inherent biases in data that cannot be accounted for by a machine learning model (or if they can be, it is extremely difficult to do so). For example, following the death of Michael Brown, St. Louis police embraced crime-prediction software that would indicate to them where they should patrol – a practice termed "predictive policing." Unfortunately, this led to increased patrolling in predominantly African-American neighborhoods because the data already in use was skewed against African-American neighborhoods.[15] This, in turn, leads to data such as the recidivism data I use above also being heavily skewed against African-Americans, creating a perpetual cycle of discriminatory practices.

Conclusion

After running our race-blind data analysis, we find that the probability of falsely classifying an African-American as "at risk of recidivism" versus someone that is Caucasian is:

- 1.39x more likely for the Bayesian Network Model
- 1.41x more likely for the Naïve Bayes Model

[13] https://arxiv.org/abs/1808.00023
[14] Ibid.
[15] https://www.themarshallproject.org/2016/02/03/policing-the-future?ref=hp-2-111.RSudzP5mZ

- 1.53x more likely for the Logistic Regression Model

We additionally find that the probability of falsely classifying a Caucasian as "NOT at risk of recidivism" versus an African-American is:

- 1.31x more likely for the Bayesian Network Model
- 1.33x more likely for the Naïve Bayes Model
- 1.38x more likely for the Logistic Regression Model

Clearly, the issue does not lie strictly with the COMPAS algorithm, but with the data itself – the process by which we determine "risk of recidivism" is inherently flawed.

To return to the guiding question: "How can we reform the bail process to maximize justice?" I believe the first step is to acknowledge the problem with the current system. Even in situations where we do not display any discriminatory intent, it can be incredibly harmful to accept the values a machine prediction spits out at face value. Evidently, this dangerous practice of accepting the data without concern for the protected characteristics (as discussed above) leads to a never-ending cycle of racial bias. In essence, before asserting broadly that we should assign values without concern for protected characteristics, we should first consider the context in which we are assigning those values. In the case of bail, prison sentencing, arrest rates, and recidivism rates, it is clearly *unfair* to turn a blind eye to the protected characteristics in our assessments, attribute the racial bias to the machine learning algorithm, and excuse ourselves of the resulting consequences.