Introduction: Background and Data

To appreciate how bias can arise in predictive Machine Learning models, we will rely on the contentious example of COMPAS, which is a proprietary software designed by Northpointe Inc. (now part of Equivant) and used throughout the United States in bail and sentencing decisions. For more background about COMPAS and the dispute initiated by journalists from ProPublica, you can refer to these articles:

- https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
- https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

We are using the same dataset published by ProPublica of its Github site, with small modifications to the names of the features to make them easier to understand. The data has information on defendants from Broward County (Florida), and includes:

- `age` : The defendant's age
- `crime_type` : The type of crime for which the most recent arrest was made. This has two potential values: **Misdemeanor**, which is a less serious crime or **Felony**, which is a more serious crime.
- `race` : The defendant's race; to simplify our task, we only kept records that were either **African American** or **Caucasian**.
- `sex` : Two values, **Male** or **Female**
- `number_prior_offenses` : The total number of prior convictions for the defendant
- `reoffend_within_2_years` : Whether the defendant actually reoffended within the next two years (1 for YES, 0 for NO).

Similar to COMPAS, we will build a machine learning algorithm that can predict **whether recidivism occurs within two years** (i.e., the field `reoffend_within_2_years`) based on the other information.

---

1. Understand the Data

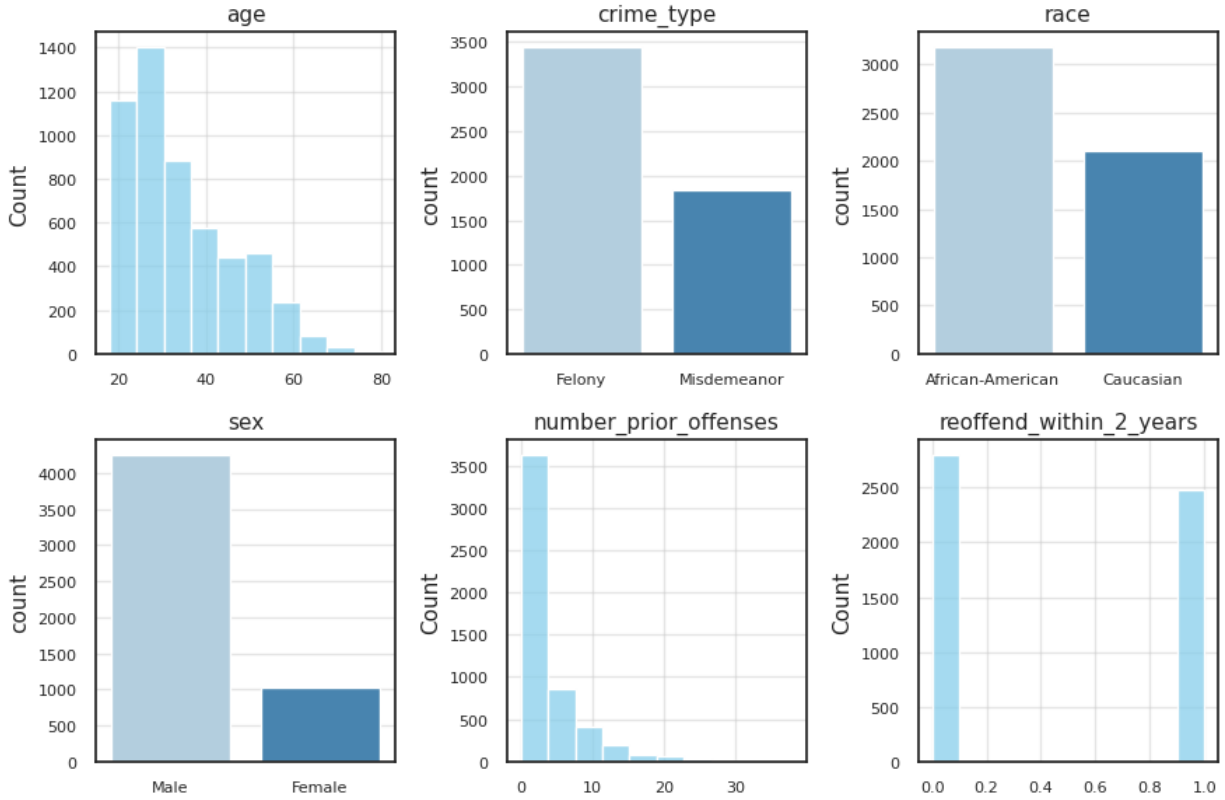
Explore the data using some of the visualizations below. We suggest expanding the hidden cells as needed to see the output.

Q1: How would you characterize:

- **the relationship between recidivism (`reoffend_within_2_years`) and other features?**
- **the relationship between race (`race`) and other features?**

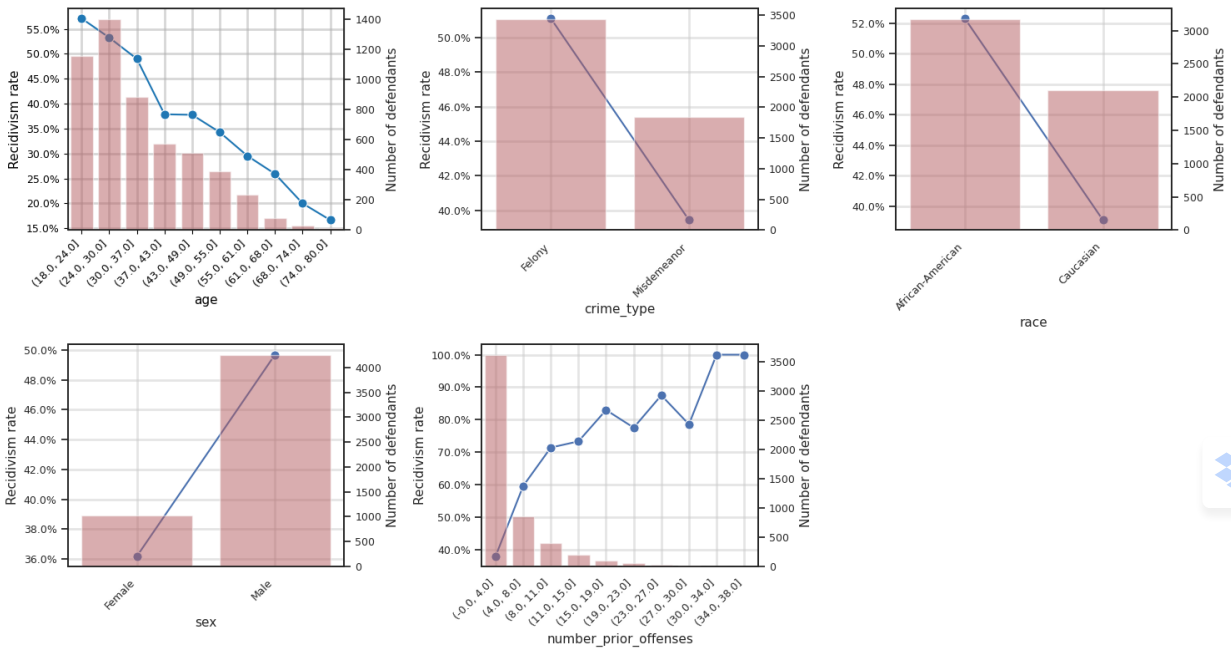
Visualize each data field

In [13]:



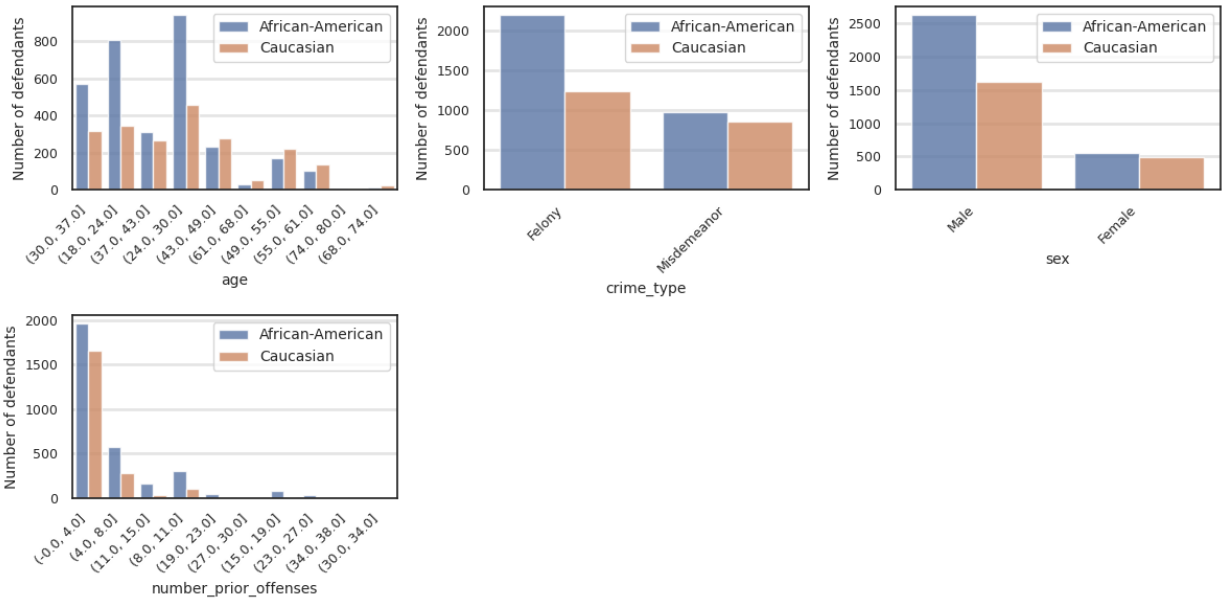
Relationship between recidivism (**reoffend_within_2_years**) and other data

In [3]:



Relationship between **race** and other data

In [11]:



---

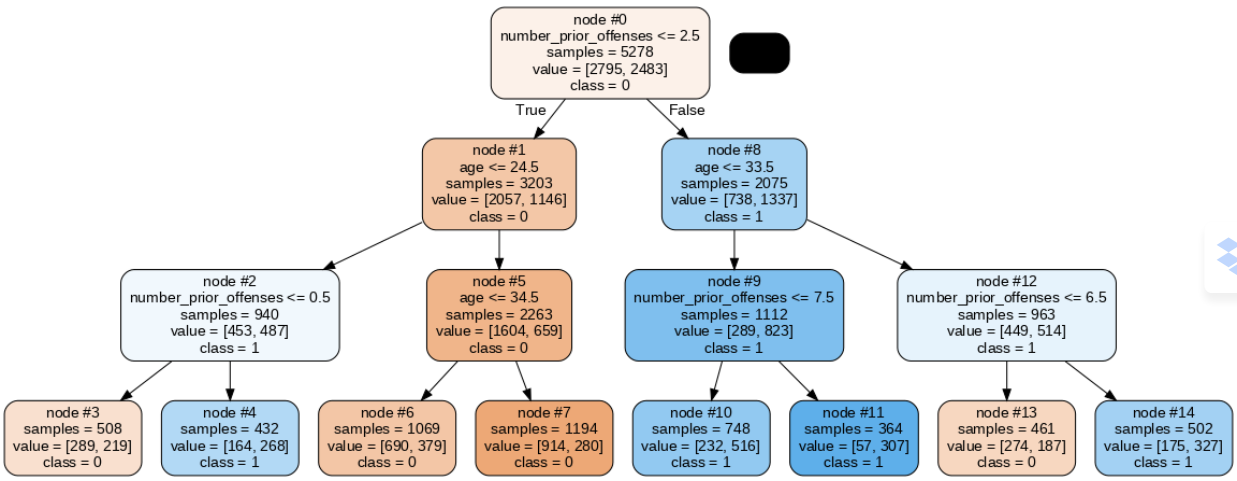
2. Construct and Inspect Our Own Model for Bias

Here, we will use a "White-Box" (i.e., Interpretable) Machine Learning Model, namely a **Decision Tree** with depth 3.

Q2: By inspecting the information below (e.g., visualizing the model and evaluating its predictive performance), would you say that the Decision Tree Model discriminates based on race?

In [5]:

Out[5]:



To help interpret the tree above, note that each node has several pieces of information:

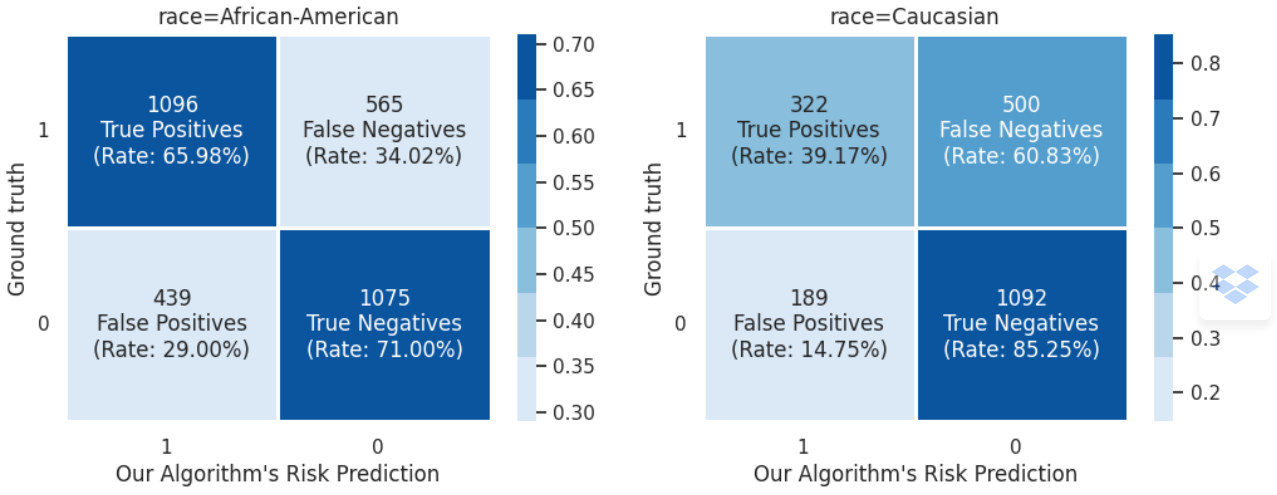
- A *node ID* to make it easy to identify the node. E.g., *"node #0"* is the root.
- A *logical condition* that compares one of the features with a value. E.g., in node #0, we are comparing the value of `number_prior_offenses` with 2.5. The left (respectively, right) subtree has all the data points where that condition is true (respectively, false).
- *samples* counts how many data points fall in the respective node. E.g., node #0 has 5,278 points.
- *value* shows two values: the first is the number of data points (in that node) that have `reoffend_within_2_years` 0, and the second value is the number of data points that have `reoffend_within_2_years` 1 (the sum of the two values always equals *samples*). For instance, node #0 has 2,795 data points that have `reoffend_within_2_years` equal to 0 and 2,483 data points that have `reoffend_within_2_years` equal to 1.

Each node is color-coded in shades of **orange** and **blue**. The two colors correspond to the two possible values of `reoffend_within_2_years`: **orange** corresponds to 0, and **blue** corresponds to 1, so each node's color tells you the label for the majority of the data points in that node: orange nodes have more zeros, blue nodes have more ones. The **shade** tells you how dominant the label is: the deeper orange (respectively, blue) a node becomes, the larger the fraction of zeros (respectively, ones).

Predictions are done by following all the conditions based on the features until a final leaf is reached. With classification problems, it is common to predict the **probability** of the target taking a value of 1, which is done by taking the **average** of the target variable for all the training samples in that leaf node. One can also obtain a prediction of the actual target **label** by taking **a majority vote** in the leaf node (which is the same as comparing the predicted probability with the threshold of 0.5).

Evaluate the predictive performance separately for African-Americans and Caucasians

In [6]:



To remember and interpret the output above, it is useful to think of a specific example that we may all be very familiar with, such as running a COVID test to detect if someone is actually infected. In this case, the **prediction** corresponds to the result of the test, and the **ground truth** corresponds to whether or not the person is infected with the virus. The following terminology is salient:

> - use **true** when the prediction is correct (i.e., it matches the ground truth) and **false** when the prediction is wrong;
> - use **positive** when the prediction is 1 (*e.g., the test result comes back "positive"*) and **negative** when the prediction is 0 (*e.g., the test result comes back "negative"*).

This gives rise to the following four cases and associated terminology:

> - **True Positive (TP)**: the test result is positive (i.e., model predicts 1) and that matches the truth.
> - **True Negative (TN)**: the test result is negative (i.e., model predicts 0) and that matches the truth.
> - **False Positive (FP)**: the test result is positive (i.e., model predicts 1), but the ground truth is negative (0).
> - **False Negative (FN)**: the test result is negative (i.e., model predicts 0), but the ground truth is positive (1).

It is common to list these four values in a 2x2 matrix called a **confusion matrix**, which is a very concise way to summarize the performance of a binary classifier. And these can be used as a starting point to calculate other performance metrics, as follows:

- **True Positive Rate (TPR)** (also known as **sensitivity**): the fraction of positive labels correctly identified as positive **TP/(TP+FN)**
- **True Negative Rate (TNR)** (also known as **specificity**): the fraction of negative labels correctly identified as negative, **TN/(TN+FP)**
- **False Positive Rate (FPR)**: the fraction of negative labels incorrectly identified as positive, **FP/(FP+TN) = 1 - TNR**
- **False Negative Rate (FNR)**: the fraction of positive labels incorrectly identified as negative, **FN/(FN+TP) = 1 - TPR**

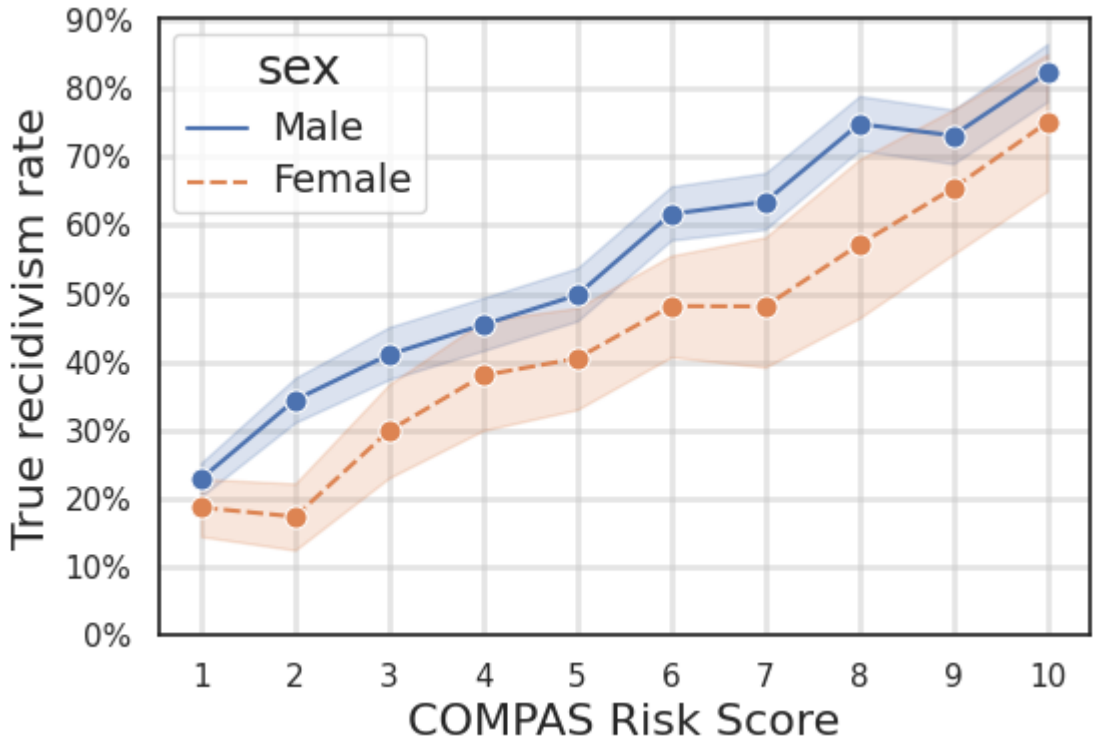
3. Evaluate a Proprietary / "Black-Box" Model

Here, we will evaluate the **COMPAS algorithm itself** based solely on its outputs, without any need to understand its inner workings.

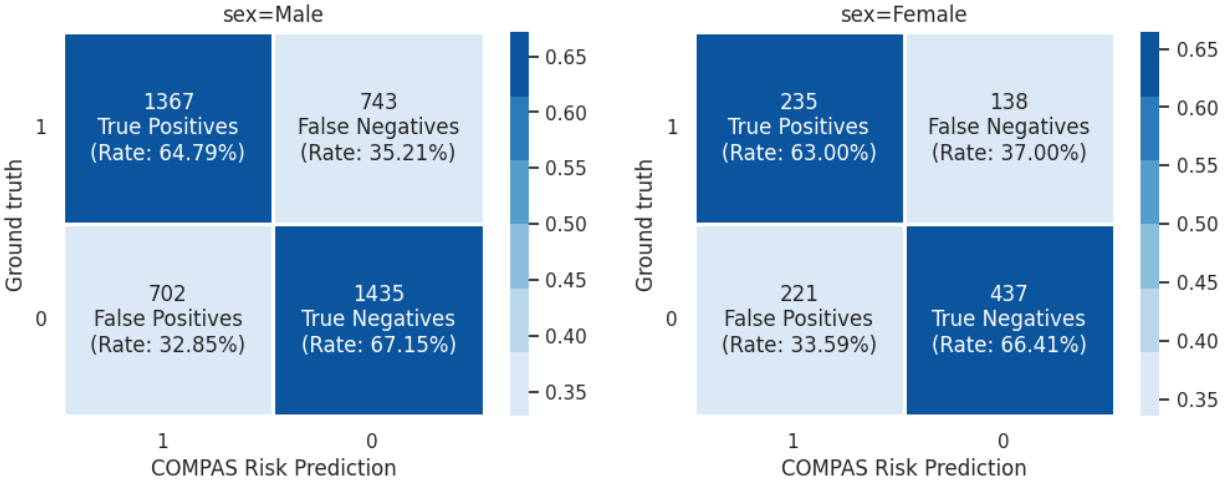
Does **COMPAS** exhibit **gender** bias?

Q3: Based on the two pieces of evidence below, would you say that COMPAS exhibits gender bias?

In [7]:



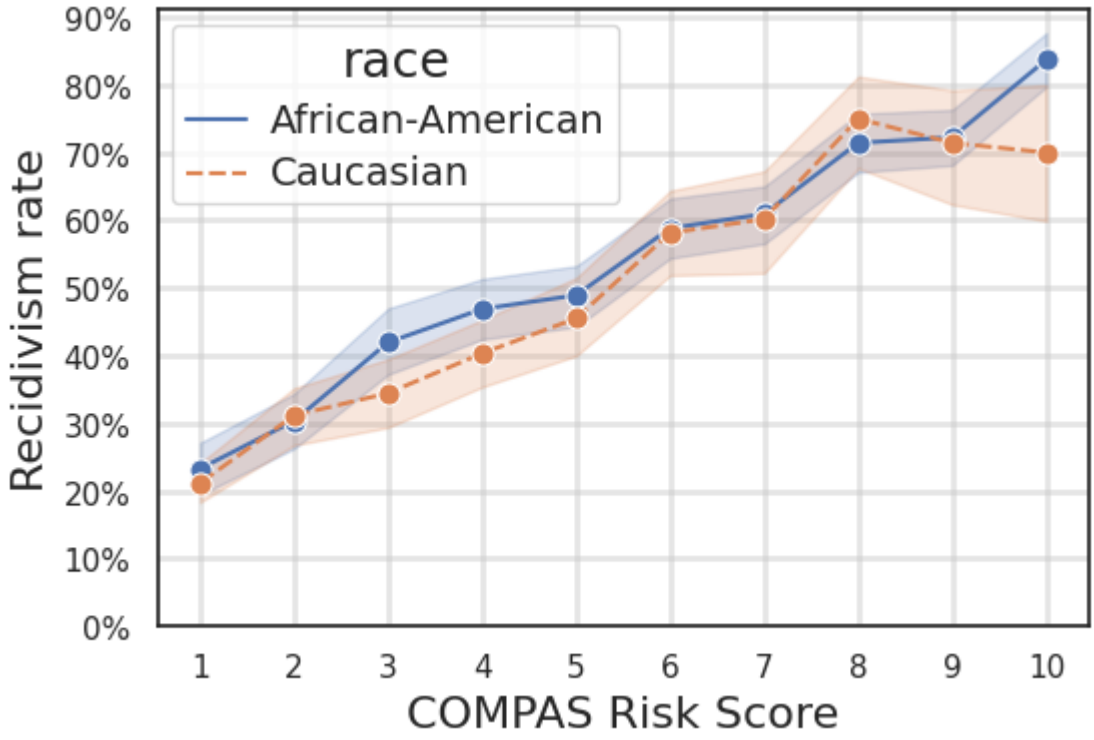
In [8]:



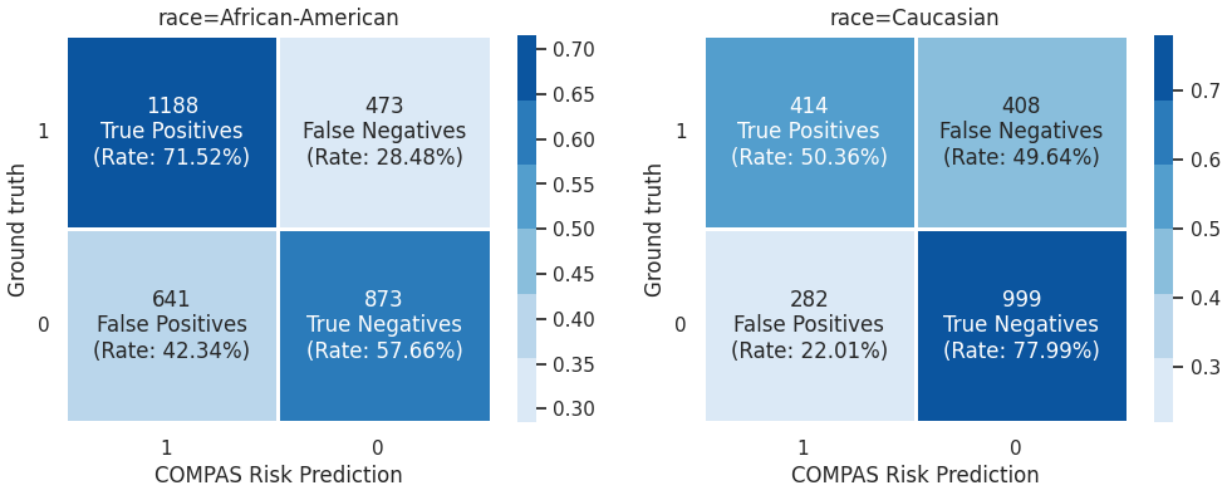
Does **COMPAS** exhibit **racial** bias?

Q4: Based on the two pieces of evidence below, would you say that COMPAS exhibits racial bias?

In [9]:


race
African-American
Caucasian
Recidivism rate
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1
2
3
4
5
6
7
8
9
10
COMPAS Risk Score

In [10]:


race=African-American
1188
True Positives
(Rate: 71.52%)
473
False Negatives
(Rate: 28.48%)
641
False Positives
(Rate: 42.34%)
873
True Negatives
(Rate: 57.66%)
Ground truth
1
0
1
0
COMPAS Risk Prediction
0.70
0.65
0.60
0.55
0.50
0.45
0.40
0.35
0.30
race=Caucasian
414
True Positives
(Rate: 50.36%)
408
False Negatives
(Rate: 49.64%)
282
False Positives
(Rate: 22.01%)
999
True Negatives
(Rate: 77.99%)
Ground truth
1
0
1
0
COMPAS Risk Prediction
0.7
0.6
0.5
0.4
0.3