

Part 1 - Description

The purpose of this website is to see a better visualization of how bias affects machine learning models through the lens of the COMPAS dataset. Specifically, it lets a user look at how Black defendants are more likely to be classified as higher risk than their White counterparts.

The information that is conveyed is not only the visuals, but also some guidelines on how to design around these constraints.

In order to make the website interesting, the beginning graph is interactive. If you brush over the graph's y axis you can see how the data is spread out for different variables. For example, you can see that when you brush over the risk score ('decile_score') of 9 and 10, the heavier strokes that connect to the race of African American shows that it's more likely to give African American defendants a higher risk score.

The target audience for a website like this would be anyone who is struggling to grasp how the models we use are racist if the data it's built upon is biased as well. However, it also is for students who are interested in learning more about how to design to counteract these biases.

Part 2 - Interactions

- Brushing to look at specific data points: If you drag over the different y-axes on the first graph, you can specifically see only the lines that touch those points.
- Accordion layout for design considerations: If you click on a given design consideration, it will expand and give more information.
- By scrolling down, content on the page will animate and come down with it.

Part 3 - External Tools

- Bootstrap
 - I chose to use Bootstrap because it's what we learned in class and makes responsive design very easy with it's grid layout. In addition, it's very widely used and so I am sure I will run into it later so it's good to learn.

- I used it to add the rows and columns (the grid layout mentioned earlier) for almost all content on the page. In addition, I also used it to make the accordion feature.
- Specifically with the accordion feature, I'm able to only show the design consideration overview and not go into specifics unless a user clicks into it, establishing a hierarchy that doesn't overwhelm the user.
- D3.js
 - I chose to use d3.js because I wanted to learn data visualization and it seemed to be the popular choice.
 - I used it to implement all the graphs that are shown within the website, including the interactive component. It also grabs the csv files and parses them.
 - It adds the main visualization to the website, making it easier for users to see data in a way that is more intuitive and interactive.
- Animate.css
 - I chose to use animate.css because I thought it would be a good way to add motion into the project.
 - I used it to let information slide into view as the user scrolls down the website.
 - It allows users to be able to see what they should be looking at because as they scroll the content comes into view. It establishes a "next" for the website.

Part 4 - Iterations from previous designs

I changed my designs during implementations by getting rid of the tabs and focusing on one particular case study, the COMPAS dataset. In addition, I decided to make it so that you can scroll through instead of having to click on next pages to make it more seamless. It would also make it such that a user doesn't have to reload graphs that take a bit of time to process). I changed the font as well, since previously I was going for a more academic feeling but decided that machine learning was too modern of a topic for that context, so I changed it to a sans-serif font.

Part 5 - Challenges

The biggest challenge that I experienced while implementing the website was definitely using d3.js. I haven't had much experience with it, but the biggest challenge was getting around how different websites were using different versions and so the syntax would change and I wouldn't understand why certain things weren't working. Another challenge

I faced was making the animations appear after scrolling, but I was able to mitigate this by using JavaScript and adding the animation classes to the elements if they were in the viewport.