figure eight

8 Ways to Prevent Bias in Your AI Projects

ALGORITHMIC BIAS IN AI IS A PERVASIVE PROBLEM, BUT IT'S ALSO A SOLVABLE ONE. AT FIGURE EIGHT, WE UNDERSTAND JUST HOW TO FIX IT. HERE ARE EIGHT WAYS TO PREVENT BIAS FROM CREEPING INTO YOUR MODELS.

1

Define and narrow the business problem you're solving

Trying to solve for too many scenarios often means you'll need a ton of labels across an unmanageable amount of classes. Narrowly defining a problem, to start, will help you make sure your model is performing well for the exact reason you've built it.

2

Structure data gathering that allows for different opinions

There are often multiple valid opinions or labels for a single datapoint. Gathering those opinions and accounting for legitimate, often subjective, disagreements will make your model more flexible.

3

Understand your training data

Even academic datasets like ImageNet can have classes and labels that introduce bias into your algorithms. The more of your data you understand and own, the less likely you are to be surprised by objectionable labels.

4

Gather a diverse ML team that asks diverse questions

We all bring different experiences and ideas to the workplace. People from diverse backgrounds–not just race and gender, but age, experience, etc.–will inherently ask different questions and interact with your model in different ways. That can help you catch problems before your model is in production.

5

Think about *all* of your end users

Likewise, understand that your end users won't simply be like you or your team. Be empathetic. Anticipate how people who aren't like you will interact with your technology and what problems might arise in their doing so.

6

Annotate with diversity

When you use humans-in-the-loop to annotate your data, it's best to draw from a diverse pool. Don't use students from a single college or even labelers from one country. The larger the pool, the more diverse your viewpoints. That can really help reduce bias.

7

Test and deploy with feedback in mind

Models rarely remain static. One of the worst mistakes you can make is deploying your model without a way for end users to give you feedback on how the model is applying in the real world.

8

Have a concrete plan to improve your model with that feedback

You'll want to keep humans-in-the-loop to react to changes, edge cases, instances of bias you might've missed, and more. You want to get feedback from your model and give it feedback of your own to improve its performance, iterating constantly towards higher accuracy.