

# DATA FLOO

## Coding Tolerance



[Nayef Al-Rodhan](#)

3:37 PM CET - January 20, 2020

As we increase the role of technology in our daily lives, and allow it to make an increasing number of decisions for us, the need to code software with an 'ethical' mindset becomes ever more important.

Whether using an algorithm to sift through CVs, provide financial recommendations, procure [predictive analytics](#) for the police, profile biometrics, evaluate college admissions, or decide the right medical treatment, there is always a danger of bias or discrimination.

Part of the reason many industries are trying to transfer decision-making to algorithms is an awareness of human bias – so how do we ensure that the technology humans create does not include, or even exacerbate, these biases?

In October of this year, a [study](#) revealed that an algorithm used in US hospitals to allocate [healthcare](#) to patients was discriminating against black Americans. The algorithm assigned risk scores to patients based on the total [healthcare](#) costs amassed in a single year, and then referred patients to the appropriate care programme. The study found black patients were assigned lower risk

scores than equally sick white people, and as such, they had to be far sicker than a white person to be recommended the relevant healthcare.

This was no small hiccup – the algorithm responsible and others like it managed 200 million people’s health care in the United States each year. Getting these things wrong can have disastrous consequences.

However, fixing discrimination in algorithms mirrors many of the same complications surrounding discrimination in society. One issue that many pick up on is a lack of diversity in those making key decisions. Just as decision makers in general need to be from a diverse range of backgrounds to lessen bias, the logic goes, there must also be greater diversity amongst programmers and algorithm designers to prevent human biases becoming built into our technology systems.

But ultimately, no matter how diverse the programmers are, biases remain because bias is bigger than any group of people. Data collection, the quality of data, the underlying purpose of the algorithm, all of these things are intrinsic to the outcome of the algorithm and varying the background of designers can only go so far in mitigating their impact.

While bias is not hardwired into the human brain, we are nevertheless predisposed to it. When we encounter people that we perceive to be different to ourselves in some fundamental way, the brain responds in a way distinct from other situations.

It only takes 170 milliseconds to identify someone else as "the other"; it is an in-built primordial system designed to promote our survival. The good news however is that our behaviour subsequent to that instantaneous reaction is all *learned*. From the time we are children, as a result of formal education, acquired familiarity with social and cultural norms, experience of politics, or even exposure to the entertainment industry, we are able to learn that the ‘other’ needn’t be threatening.

Neuroscientific studies have shown that when people are asked to make judgments about those they find to be similar to them, the areas in their ventromedial prefrontal cortex become active, whereas when engaging with those we consider ‘different’, another part of the brain - the dorsomedial prefrontal cortex - becomes active. There is a biological phenomenon that occurs when a person encounters someone they find to be different to themselves. Bias isn’t merely psychological or social, it has a neurobiological and neurochemical basis that creates distinct responses.

This is all to say that increasing the diversity of programmers, while a good thing in itself, is not sufficient to counter the social and neurobiological forces that facilitate bias.

Given that we know that programmers will be biased, through no fault of their own, it is the [job](#) of tech firms to educate and safeguard against unintended consequences. Rather than waiting for a mistake, firms should look for biases early, and consider ways to ensure every part of the development chain, from the intention, to the collection and processing of the data, to the outcome, are not producing unintended harm.

These same lessons apply to everyone, but they have especial relevance to tech firms. A key selling point for the use of algorithms in decision-making is that they can improve on the processes and inefficiencies of humans. What is clear from recent studies however is that, while there is little doubt that the algorithms save time and money, they are little better than humans at producing biased

outcomes. Worse still, because of the widespread belief that algorithms are less fallible than humans, these biases and errors can go unchecked for longer than they might if a human were to make the same mistakes.

For the tech industry, good governance is not going to be secured by new innovations. To address bias, firms will need some good old-fashioned people management, with involvement by HR and the wider company ensuring that the firm's is aware of these cultural issues and active in addressing them. Ultimately it is the people that make the algorithms, and that is where the change needs to come from.

Algorithms may be able to make decisions for us, but they do not free us from bias, they are merely very successful at efficiently replicating that bias on a wider scale. This is a human problem, and ultimately it needs a human solution.

*Image Credit: [\(Columbia Business School\)](#)*