Module 4.2

Hello and welcome back to the second video lecture for week four.

Now, in the last video lecture I introduced this idea of algorithmic accountability reporting, and in this video I want to go into more detail on some of the methods used to find stories on the algorithms beat.

I'm also going to show you a number of examples of stories found using these methods, so let's jump right into looking at these techniques. Now, effective algorithmic accountability reporting demands all of the traditional skills that journalists need in reporting and interviewing as well as domain knowledge of a beat, how to file public records requests, and of course being able to write clearly and compellingly.

Plus it involves a host of new abilities and skills like how to scrape and collect data, how to design audit studies, and in some cases how to use advanced statistical techniques. So, the algorithms beat will definitely benefit from having a team with a very diverse range of skills and abilities.

Now, to get the story there are a variety of new methods that are coming into journalism practice in order to deal with trying to investigate algorithms. These include things like reverse engineering, auditing, poking and prodding crowdsourcing, and even code reviews.

So, reverse engineering gets at this idea of trying to uncover the technical specifications for how an algorithm works. This might involve looking at trying to figure out what are the variables that the algorithm is paying attention to, and how does the algorithm weight those variables in its decision making process.

Another technique is auditing, and this involves looking at the inputs and the output of an algorithm to try to understand, and correlate, and connect those inputs to the outputs. This can involve looking for errors, so if for a given input the output is somehow a mistake or an error, this can be interesting an interesting story. It can also involve looking and comparing the inputs and outputs against benchmarks to understand if the outputs are somehow skewed, or biased, or discriminatory in some way.

A third technique involves what I would call poking and prodding, and this is really about trying to expose some kind of algorithmic reaction to a user. So to poke it and see how does the algorithm respond, how does it react, and this can provide an interesting experience that can then help illustrate how the algorithm is reacting to people.

Another technique involves crowdsourcing. Sometimes this is referred to as data donation drives. So, an investigator might ask the crowd to send in some data about their own experiences of an algorithm, or send in let's say the scores of how an algorithm has scored an individual, and then the investigator can collect all of that information together and analyze it as a set to see how the algorithm maybe is treating different people.

The last technique here I referred to as code review, and this involves actually getting the source code and reading it to try to understand if the algorithm has been implemented correctly and accurately. It might actually only be useful in a limited number of scenarios, but I wanted to include it here just for completeness.

So now I want to show you some of the stories that were found with each of these methods. So here's an example of a story that was found using a reverse engineering technique. The story is about predictive policing algorithm used in Chicago called the Strategic Subjects List or SSL, and what the SSL is it's a score that's calculated for people in order to determine what their risk is of being involved in a violent crime or being the victim of a violent crime.

Now, the way that the reporters found this story is they actually got some data that was released by the Chicago Police Department which included the scores for a number of people that had been calculated as well as a number of variables that had been used to calculate those scores.

So what the reporters then did was they built their own statistical model using the data that the Chicago Police Department had released, and then they built their own model, and they were able to predict the scores with pretty high accuracy using just eight fields of data that were in this public data release.

So you can see this in the lower right corner here showing those eight different fields of data and the numbers that are shown there indicate sort of the weight or the importance of each of those fields of data in coming up with the final SSL score. So, by essentially recreating the statistical model that the Chicago Police Department uses their reporters were able to understand sort of the importance of these different factors in the model.

Another technique that's useful for studying algorithms is auditing, and now auditing has actually been used for decades to study social science questions like housing discrimination, but it's also recently been applied to studying algorithms as well. So, here's a story that was found using an auditing technique. This was actually published a few years ago in The Washington Post.

The idea of the story was to study dynamic pricing online, and so the goal was to try to audit dynamic prices and study how prices varied by location across the United States. So the way they studied this was they picked a website like say staples.com, and they simulated different users visiting that website from different locations.

They then reported the price on the website for a variety of different products for those different simulated users coming from different locations, and in that way they were able to study how the price was different based on the location of the user that was visiting the site, and they were able to find a correlation between location and price offered online.

Now in comparison to reverse engineering and auditing which can be quite technical approaches another technique here which is a little bit lighter weight is what I would call a poking and prodding. This is really about trying to get the algorithm to react to you and then using that as a way to understand people's experiences of the algorithm.

So here's an example that was published in BuzzFeed News, and what's interesting here is that the reporter was able to illustrate and critique one of the Facebook news feed algorithm changes in particular with the news feed algorithm change related to prioritizing posts that generate a lot of discussion and conversation.

So, by posting something really annoying in her own news feed the reporter was able to generate a lot of comments. She then observed the stickiness of that post in the news feeds of her friends who were then annoyed and they were provoked to continue commenting and continue reacting to the post which just propelled the algorithm further. So it created this feedback loop where more and more conversation got people annoyed and commenting further and further which kept boosting the post and her friends news feeds.

So, the story I think made the meaning of that Facebook algorithm change a lot more salient and visceral to readers in terms of how it impacted their news feed experience. Here's an example of a crowd sourced approach to studying an algorithm. In particular, it is an investigation into one of the credit scores that's used in Germany called the Schufa Score.

This investigation was done by the Spiegel and The Bavarian Radio and they refer to this technique as a data donation because really what they're essentially doing is they're asking people to donate their data to the investigation so that it could be analyzed in order to understand trends and differences in patterns between different people.

So, this type of crowdsourced audit allowed these investigators to get a glimpse into how these credit scores varied across individuals and how the scores are correlated to various demographic factors like age and gender.

Now the downside to this kind of crowdsourced approach is that the sample that you collect is often biased. That really kind of starts to limit your abilities to really reverse engineer what's going on inside the scoring systems technical blackbox. Now, the final technique that I want to talk about is called Code Review.

Now, this example here didn't actually come about through a code review, but it's an example of where I think a code review might have actually led to a story being found sooner. So the story is about an algorithmic system that's in use in the state of Arkansas in the U.S., and what the algorithm does is it determines how many hours of caretaker assistance a person gets from the state.

In this case someone with cerebral palsy went from having 56 hours of caretaker support to 32 hours of support per week based on what the algorithm had calculated. Now, another family had actually sued the state because they had been negatively impacted by the system.

And in the court case and you know, in the actual examination of the case in court, it was revealed that the algorithm had been implemented by a third party, so the state had actually not even implemented the system, they had affirmed it out to a third party contractor.

And in fact when they went through and read the code they realized that this third party implementer had implemented the wrong version of the algorithm, and that it was actually incorrect. So, that was contributing to the negative impacts that some people were feeling from this system.

So, I would say in cases where some kind of public policy could get mistakenly implemented or mistakenly written into code, it could actually be worthwhile for journalists to try to get the code either through a public records request to try to read that code and see if it's accurate.

So to recap, I covered five methods here today. Again, not all of these methods are super technical, but some of them are like reverse engineering, and auditing, and code reviews like you may actually need higher levels of technical ability, or you could consider partnering with academics or with other experts in order to get that methodological expertise on your team.

Again, if you're looking for more information on how to cover algorithms I'll remind you again to check out www.algorithmtips.org, and actually also have a whole long chapter on algorithmic accountability reporting in my book which could provide you with more ideas and details for covering algorithms.

So, that's what I wanted to cover for now. In the next video, we're actually going to flip things around and I'm going to talk about how to be more ethical and accountable with the algorithms that you may be using in your own work. So I'll see you there.