Stop pretending technology is value neutral

15 Mar 2020

This post was prompted by the Open Cities AI Challenge, which involved using machine learning on aerial imagery to detect buildings for improved planning for disaster resilience.

Kicking the can

About six years ago, I was arguing with a wiser friend in a bar in Ghent. I was adamant that science and technology were neutral creations: what mattered was what people did with them. As an engineer, I could wash my hands of those problems. She knew better: you can't design cigarettes and pretend you don't know what cigarettes are for.

This is especially true in artificial intelligence and machine learning in particular: they are very easily applied to morally questionable practises, and due to their inscrutability, are very easily misused, even by practitioners with good intentions. The most obvious example of the former: computer vision. The same advances that allow me to easily map buildings from drone imagery, allow powerful actors to track people's faces as they move around the world. If you're working on this technology, it's crucial to think about the implications of this work. At least one person (Joe Redmon, a pioneer in AI object detection) recently did so, which sparked an interesting conversation.

The issue of scrutability is more nebulous. Machine learning (and especially deep learning) creates models that are extremely difficult to truly understand. But it's a powerful hammer, and when you're holding it, everything starts looking like a nail. Data scientists are tempted to think they can just wade into a field they know nothing about, such as disaster resilience, and fix it with a well-trained neural network. Remember that gradient descent (how models self-improve) only operates on the loss function (how models know how badly they've done) you define! If that function doesn't include ethical and real-world concerns (which unfortunately probably isn't possible) then you might be missing all sorts of minima that are better at avoiding unintended consequences.

$$\textit{Loss} = f(y) \quad \rightarrow \quad \textit{Loss} = f(y, \textit{ethics}) \quad \rightarrow \quad \textit{profit??}$$

that is, if you can make real-world concerns a core part of your modelling process, then you're doing well.

So if we don't want to kick the can down the road to politicians and decision-makers, technologists need to dial down the hubris and think about what they're doing.

Remote sensing

The topic of this AI challenge provides an excellent case study. Machine learning has brought huge capabilities to remote sensing, making it easier than ever to track and map houses, water bodies, agriculture and anything else that can be picked up by a satellite or drone camera. But there are massive implications for privacy, and for hard-to-detect bias as soon as this data is used to make decisions.

Both of these are especially true when dealing with remote, poor communities, who are typically less likely to be consulted. High-resolution aerial or street imagery can easily include individuals. Less obviously, huge datasets from disparate sources can often become uniquely identifying in unexpected ways. This isn't likely to be an issue with free satellite imagery and OpenStreetMap, but is certainly possible as soon as orthogonal data sources are incorporated, such as apps, census data, electricity use, mobile network data…

Bias is just as important. Humans don't have a great track record with bias, but with rigorous oversight and careful design, it's possible to create systems that are as bias-free as possible. If you know you're out to protect disadvantaged communities from disaster, you can monitor every step of your decision process to ensure you aren't accidentally throwing these people into the path of a tornado. As soon as a neural network is involved, you can give up any pretence of being in control of the results. Whatever biases exist in your data selection, labelling and training processes will propagate and pop up in all sorts of difficult to predict ways. A hand-built model can certainly have mistakes, but it's usually possible to work from input to output and understand what is happening. Non-deep ML approaches (such as decision trees) can be inscrutable, but it's still possible to inspect the resulting models to and guess how they might behave in different scenarios. Neural networks are much harder: a model might have tens of layers and thousands of weights, making it nearly impossible to comprehend its every nuance. There *will* be edge cases that you didn't foresee, and it's very possible those most in need of help will get thrown under the bus at some point.

Playing with scores

Let's look at a less dramatic, but very real-world example of bias in data science. The Open Cities challenge used the Jaccard index to score submissions. So the first thing I did? Searched for 'loopholes' in this index that would allow me to improve my score without really improving my model.

This prompted the following as a very simplistic example of how choosing a scoring metric can drastically impact what you're optimising for. I decided to play with some example imagery to see how four different testing criteria would push a model in different directions.

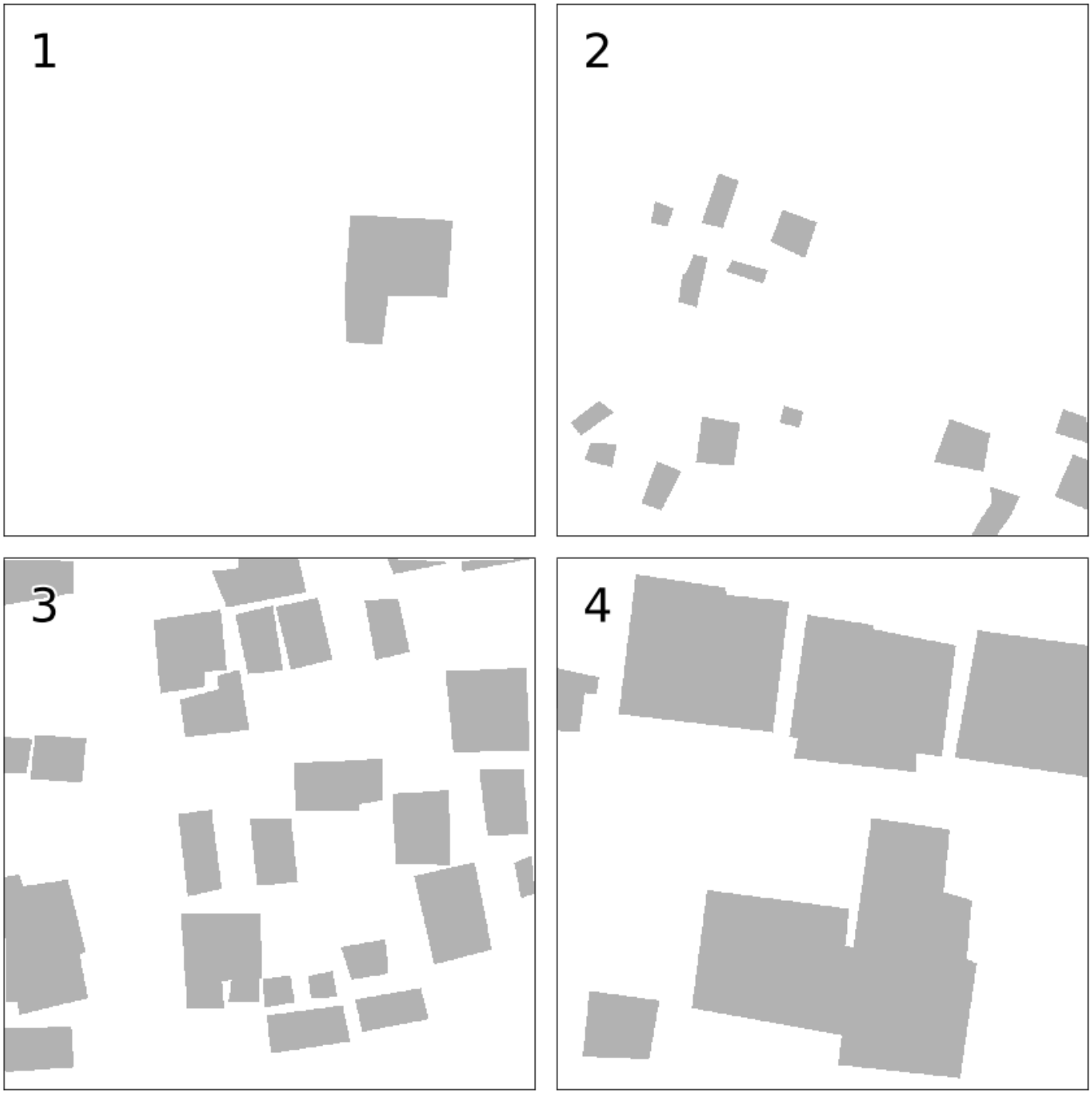
- **accuracy**: percentage of predicted pixels that are correct
- **precision**: percentage of positive pixels that are correct
- **recall**: percentage of true pixels that are predicted
- **Jaccard**: true positives divided by union of predicted positives and actual positives

To illustrate the shortfalls of these different scores, I'm going to do some mock predictions for the following four images, which come from the Open Cities challenge (I hope this is okay!). I've made some not-great building labels for them so we can get started. As you can see, they cover a range of different resolutions and building types.

Four test images with labels overlaid in blue. These labels were made by me (by hand) on a moving train, and do not represent the state-of-the-art of labelling!

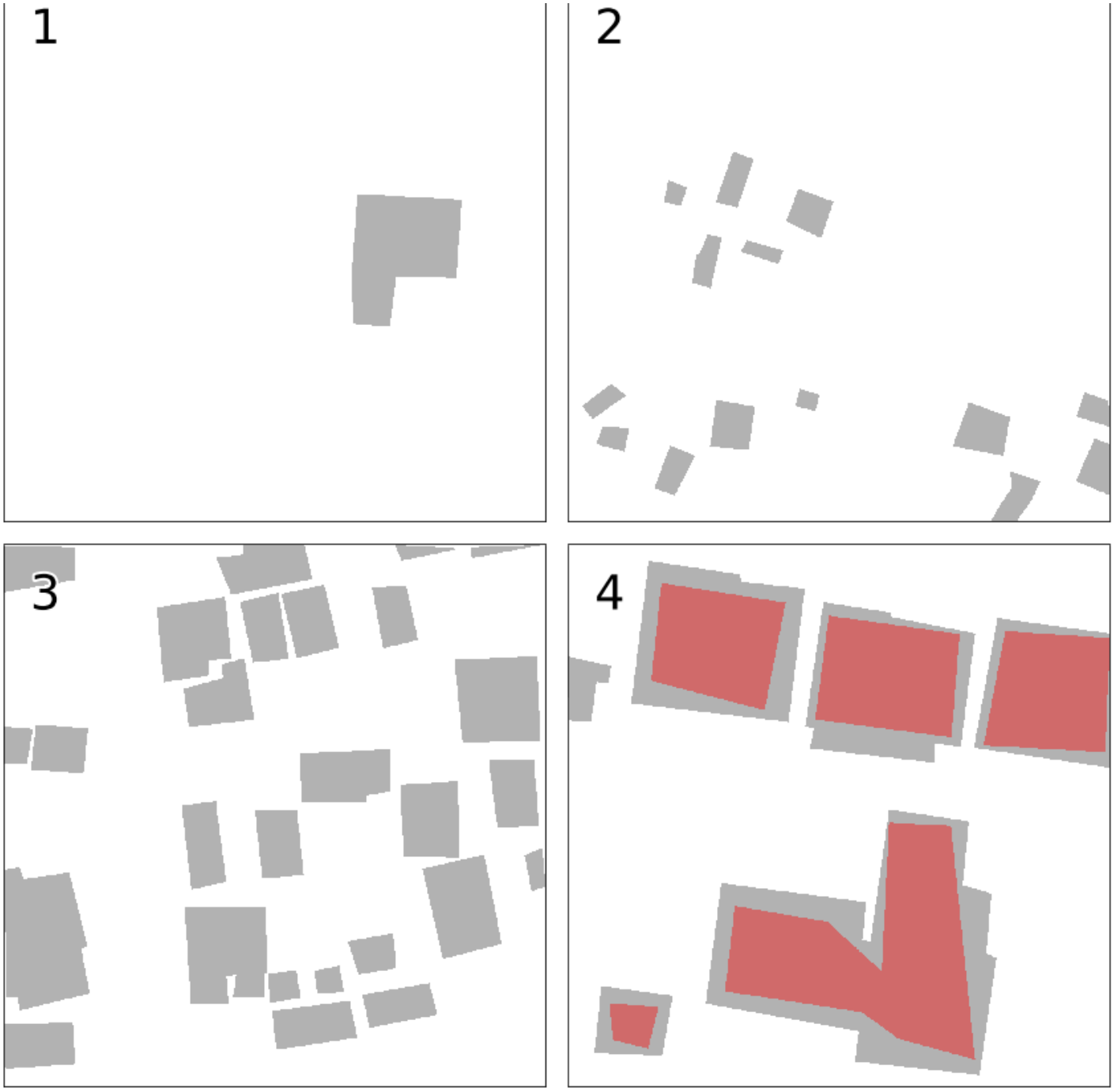
So, if we're being judged by accuracy alone, what should our model do? As the example below shows, nothing! Because the classes are unbalanced, i.e. there is much more not-building than

building, a lazy model can get pretty far by doing nothing at all. In this case, 81% accuracy!



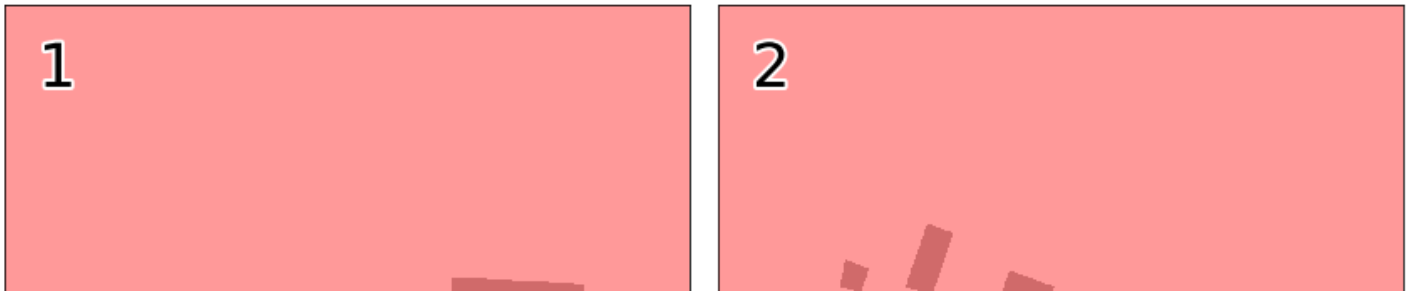
Ground-truth in grey (the hand-made labels from above), a contrived model prediction overlaid in red (i.e. no buildings predicted). Accuracy of this horrible effort? 81%!

So clearly accuracy is not very useful. What about precision and recall? The image below shows a model prediction with 100% precision: this is because precision only measures the 'positive' predictions, so it encourages the model to only make guesses when it's completely confident.



Ground-truth in grey, model prediction in red. For a precision of 100%

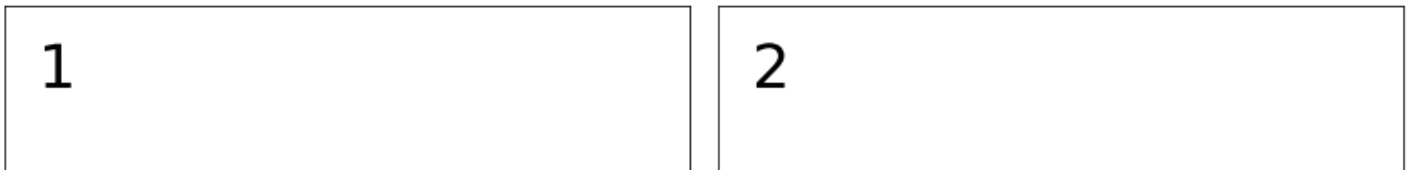
On the other hand, recall shouts “you miss 100% of the shots you don’t take!” as it rewards the model for finding any positive cases, regardless of how much it overpredicts.

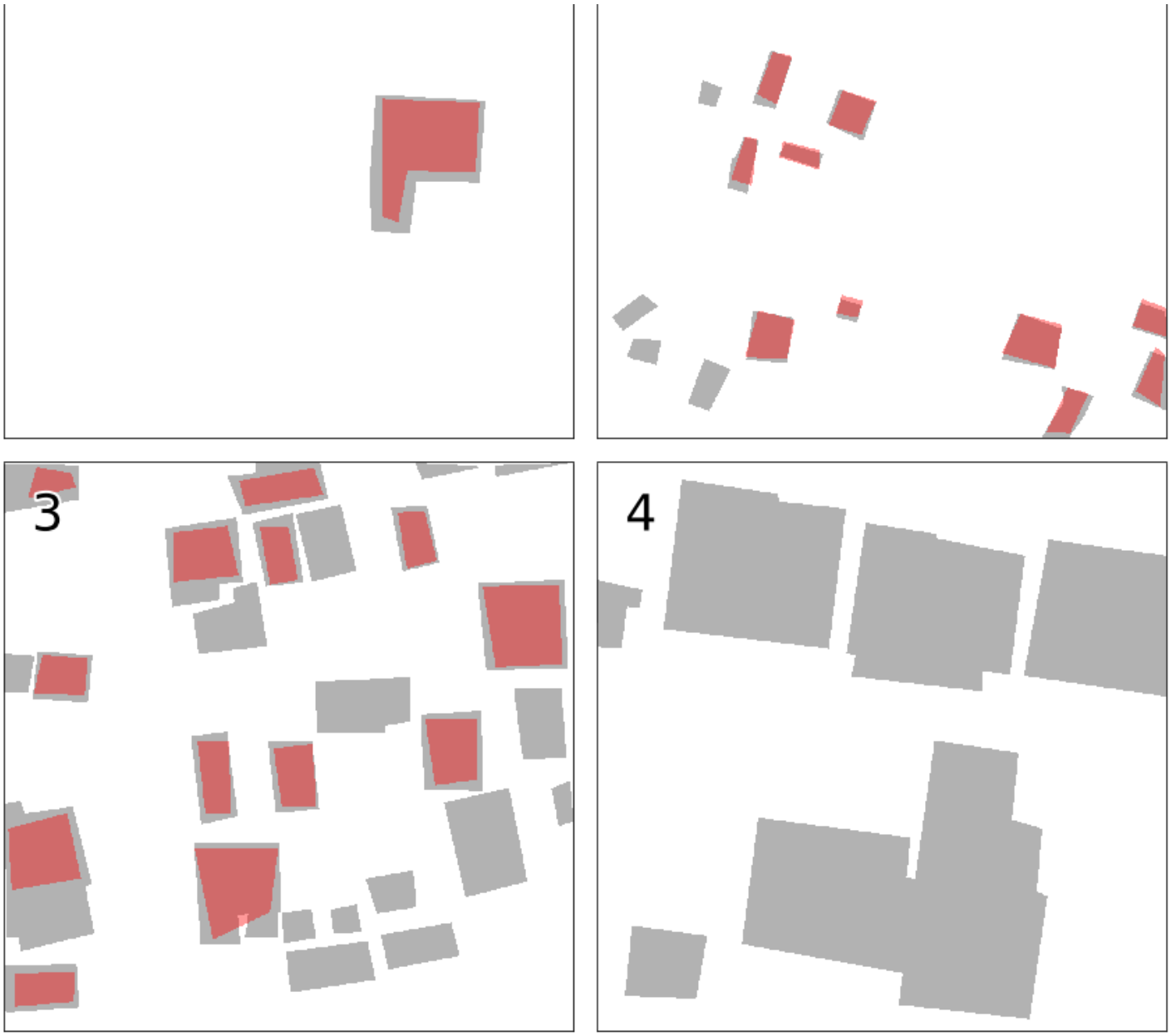


Ground-truth in grey, model prediction in red. The recall is an impressive 100%.

The preceding examples are a bit contrived, because no one uses these scores in practise. But they represent real potential issues: a disaster planner might get someone to map huge areas using AI and then treat the results as gospel. But if the modeller has been lazy and the planner isn't careful, they could go to work with meaningless data. The cutting-edge analysis would end up suggesting huge interventions in some areas and completely ignoring others.

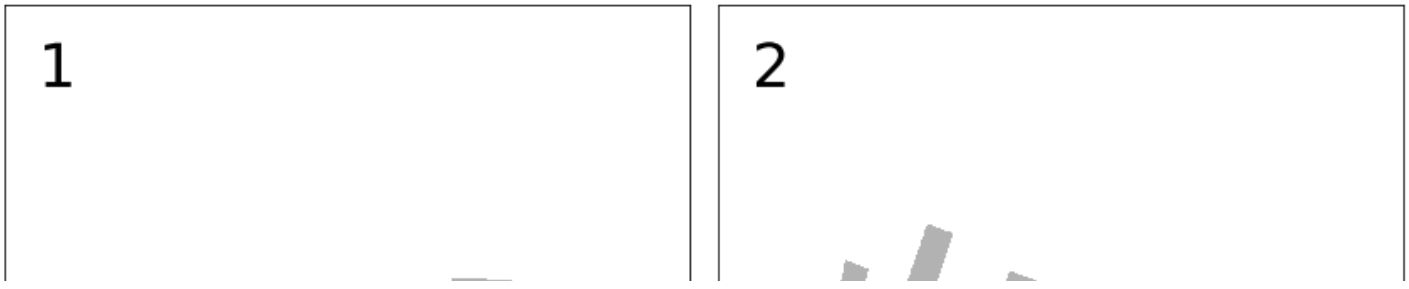
Luckily there are better systems that are used more often (but aren't panaceas). The F1 score combines the above two, leaning towards whichever score is worse. The Jaccard index (or IoU, intersection over union), which was used in this challenge, is similar, and avoids the pitfalls outlined above. However, *they're still measuring pixels*, which is the key point I want to make.

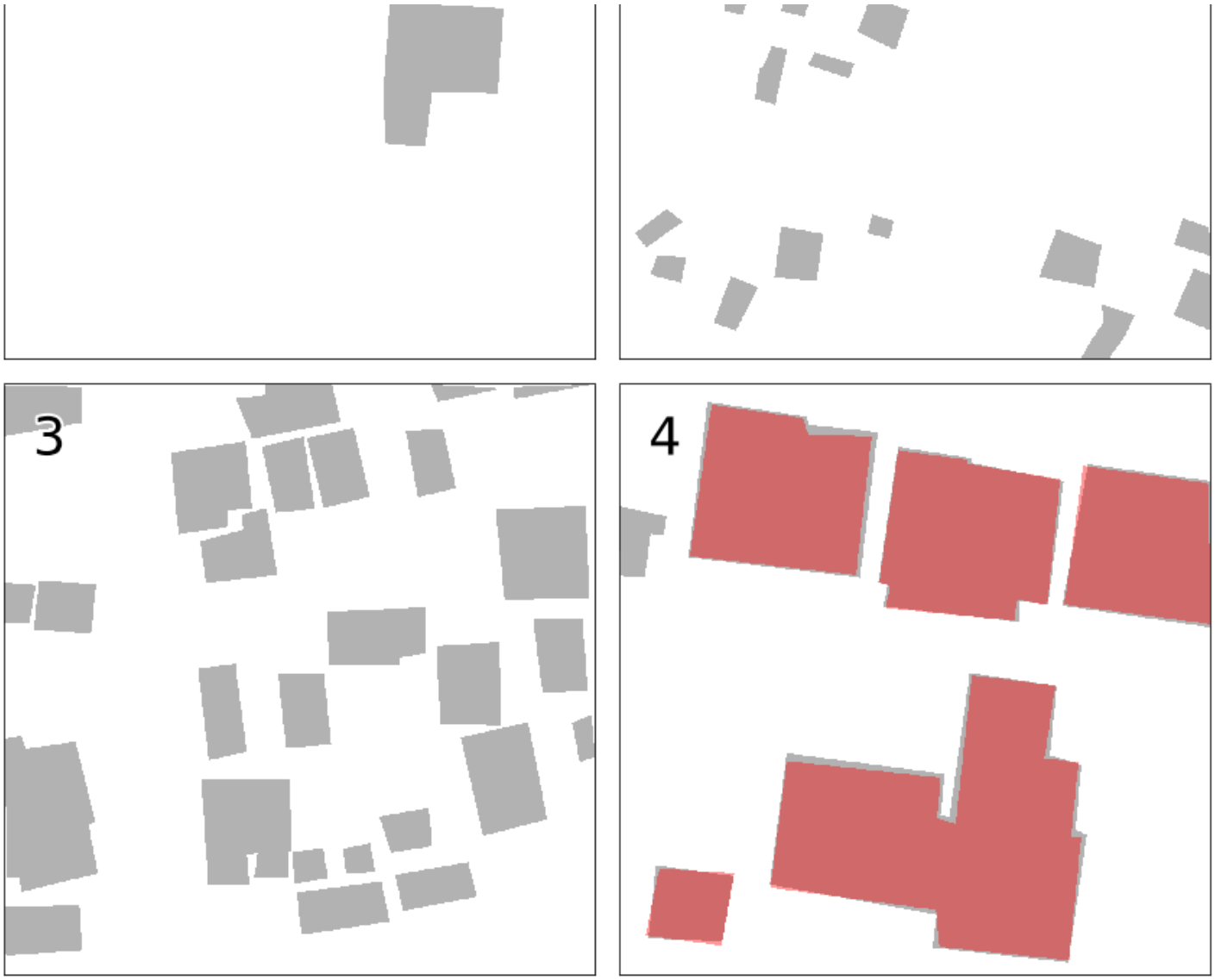




Ground-truth in grey, model prediction in red. Although we've done pretty well, the IoU is 22%.

Consider the images above and below. The first successfully identifies more than half of the buildings, but only scores 22%. Meanwhile, the second only identifies five buildings , but scores more than double at 47%. The way the competition is set up encourages both me and my models to focus on images with higher resolution, bigger houses, and worry less about sparsely populated areas with low-resolution imagery.





Ground-truth in grey, model prediction in red. Most would say this is worse than the previous attempt, but the IoU has jumped to 47%.

This was a rather roundabout example of a simple point: seemingly basic technical decisions can have large ramifications if they aren't properly thought through in the context of the full problem. In this case, the scoring criteria unwittingly encourages modellers to focus on wealthier citizens, those that are probably the least in need of whatever interventions these maps are supposed to support. Could this be improved? Possibly by weighting for the actual population in each image, or by weighting images of poorer areas higher. But we might end up just gamifying modellers into some other weird response!

Okay wrap it up now

This is just one tiny example where an innocuous (and pretty standard) decision creates an outcome that may or not be desirable. There are many others: how and by whom the training labels were created, what quality of imagery is available for different areas, what opportunity the affected people had to be involved in the process.

When it comes to buildings, I have previously advocated for sticking to human mappers. These can be local people who know the area, can differentiate between houses, sheds and schools, and whose contributions can go straight into OpenStreetMap, where they're immediately accessible to all. The most useful place for machine learning in this pipeline is to identify areas that probably have buildings, and leave the rest up to the mappers. And then one step further: go to the area and show people how to access the maps from their phones, and how to edit and contribute to them!

But there problems are where machine learning excels and humans are not great. Recognising signals of past disaster impacts in multiple data sources is a great example. We just need to be sure we know what we're looking for and don't kick the can down the road for someone else to worry about. If funds will be allocated based on the model, and real humans will be impacted, we better be certain we're predicting what we think we are, not simply making up numbers or worse, systematically excluding disadvantaged people.