Helping Your Future Self Understand the Explanations You Generated Today

Once you get an explanation technique up and running, it's tempting to cash in on your hard work and generate as many explanations as you can right away. Unfortunately, you often find yourself squinting at many charts in an old Jupyter notebook or trying to explain the context for the explanation you sent along to colleagues a few months ago.

To save yourself future frustration, we've found it useful to always embed the following information in your explanations:

1. Exact technique and parameters used, e.g., was it SHAP or Captum's sampled Shapley?
2. Model version, training configuration, hyperparameter values, and dataset version, to be able to trace the source of the explanation.
3. Timestamp the explanation was generated, which is useful for knowing if the explanation is stale.
4. Input and inference values. You would be surprised how few techniques include this information in their visualizations.

Assuming Causality

Very few, if any, explanation techniques are able to establish causality in any sufficiently complex model. Techniques can only describe correlations between what influenced the model and the prediction. For example, Integrated Gradients may highlight a single pixel as highly influential in the model's prediction, but the technique does not guarantee that the pixel caused (even in part) the prediction.

At odds with explanation techniques' ability to provide correlations is the strong human desire to explain consequences due to causality. Causality is an important part of storytelling and narratives, and often you will find that consumers try to fit an explanation into a broader narrative to justify, attack, or just comprehend a model's actions. It is very difficult to work around this need for causality, and you will not get far trying to change your consumer's instinctive behavior. Instead, there are two strategies you can use to mitigate the tendency to fall back to causative descriptions:

- Language matters. Whenever introducing an explanation, whether with text, verbally, or in a presentation, be careful to not introduce or imply causation. This can be very difficult! For example, with feature attributions, it is tempting to say a particular feature caused the model to behave a certain way. Instead, try to use words like "influence" or "suggest."

- Avoid "this-then-that" narratives. Often explanations, with good intentions, try to present a logical flow of information and narrative. "This is the input to the model, then the model generated this prediction, here is the explanation" is a common narrative. Unfortunately, this narrative also implies a causal chain of reasoning from inputs to explanations. Instead, you may want to try inverting this narrative: "The model gave this prediction, which is explained by X. Additionally, here are the inputs."

Overfitting Intent to a Model

When given a sufficiently compelling explanation, consumers are tempted to extrapolate from the explanation to concepts learned by the model. Except for those focused on concepts, e.g., TCAVs, it is difficult to say that most explanation techniques are able to reveal semantic concepts the model has learned. In our earlier example of an image classifier given a prediction for a photo of a cat, it is accurate to say what pixels influenced the prediction, but it is not accurate to reach further and say, "Now we know the model has learned how to recognize cat ears." It certainly could have, but a pixel attribution technique gives explanations based on the pixels in the image, not the semantic concepts related to those pixels.

To avoid this overfitting, make explanations clear and constrained.

Overreaching for Additional Explanations

Once given a sufficient explanation, it is not unusual for ML consumers to reach for another explanation technique to augment their understanding. However, these techniques, even if good on their own, may not actually increase the power of the original explanation. For example, a common reach is for a user who has received a feature attribution explanation to try and find a counterfactual explanation to prove the validity of the feature attribution by finding a prediction for a data sample with a different value for the most influential feature and a different predicted class. The consumer may then declare this proves the influence of the top-ranked feature. While the counterfactual can enhance our understanding of the model's behavior, and even back-and-forth between looking at feature attributions and corresponding counterfactuals can tell us about different facets of the model, it is important to not treat each explanation as a validation of the other. Each explanation has its own gotchas and nuances that constrain what they can tell us about the model. Together, they may widen our understanding of the model, but may not necessarily deepen our understanding of one particular aspect.

Preventing explanation overreach is difficult because users often take matters into their own hands to find new explanations. Strongly discouraging this behavior rarely works in practice either, as the ML consumers will genuinely believe they are proactively contributing to the overall quality of the Explainable AI. Instead, to prevent