Chapter 7. Interacting with Explainable AI

Explanations cannot exist in a vacuum. They are consumed, used, and acted upon by ourselves, our colleagues, auditors, and the public to gain an understanding of why an AI acted the way it did. Without explainability (and interpretability), Machine Learning (ML) is a one-way street of information and predictions. We may see an ML do something astounding, such as translating a paragraph from one language to another, but it is rare for us to unequivocally trust technology.

Fundamentally, we are in a working relationship with every AI we use. Imagine machine learning as your coworker. Even if this coworker did an amazing job, we would find them difficult to work with if, when we asked them to perform a task, they went off to another room, returned with the answer, and then promptly left again, never answering our questions or responding to a thing we said! This silent coworker problem is what explainability tries to address by starting a two-way dialogue, as in Figure 7-1, between the ML system and its users. However, this dialogue is very limited given how novel explainability is, which makes your choices around how to construct that dialogue even more important.

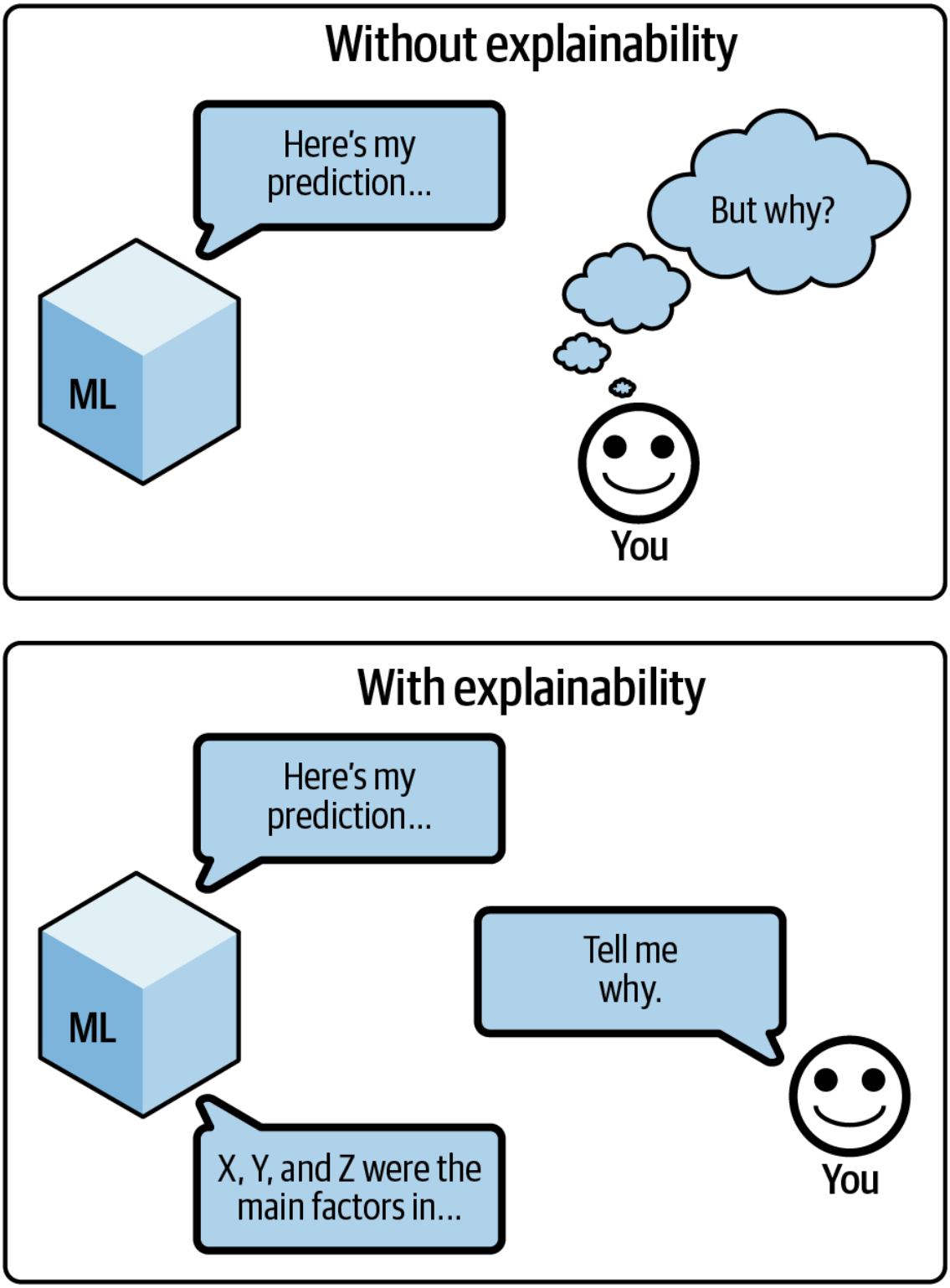

Figure 7-1. Explainability creates a dialogue between an ML and its users.

In this chapter, we will review the needs of different ML consumers and what to keep in mind when designing Explainable AI (XAI) to work best for each of these groups. We will also explore how to display explanations, and what the trade-offs are between different types of visualizations. No explanation is perfect, so we will also discuss common pitfalls in how explanations can be misinterpreted, and how to preemptively design your explainability to mitigate these issues. Finally, we will also discuss what happens after an explanation has been created and communicated, diving into the actions taken after an explanation.

Who Uses Explainability?

In Chapter 2, we discussed who consumes explanations: ML practitioners, observers, and end users. Each of these groups has a different need for explanations, and different levels of knowledge about machine learning, so we cannot treat them all the same. To better understand these groups, we can categorize them in terms of their expertise and their intent. Expertise can be in how the ML itself works, the broader environment the ML operates in, or the additional factors for the ML's input features (or inference). Intent is defined by what actions a consumer may take in reaction to receiving an explanation.

There are three common types of expertise:

Domain

Has knowledge of the environment the ML operates in, but not necessarily how the ML itself functions. For example, a banker may understand the broader economic environment for loans.

Model inputs

Understands or has access to additional information related to the inference inputs but may not be able to alter the inputs. While this could be the training data, it is more often the data sample used for a deployed prediction, or the dataset features in general. For example, a consumer has a deeper intrinsic understanding of their shopping preferences that informs a product recommendation system.

Machine learning

Understands the model architecture and how the model works; however, there may be limited understanding of the dataset or implications of predictions made. For example, this could be a data scientist who was hired as a consultant to build an ML model for predicting how long it would take for robots to assemble parts in a factory, but does not have previous experience with robots, which unpredictably break down, or assembly lines, which have complex, cascading cause and effects when estimating production times.

The following are common types of intent:

Model improvement

Based on the explanation, the user will take action to increase the quality of the model. This could include refining the training dataset, using different features, or changing the model architecture.

Build trust in the model

Increase confidence that the model's predictions are accurate and reliable.

Verify

Confirm that the model behaves as expected against a set of standards. For example, that a credit-rating model adheres to financial antidiscrimination regulations.[1]

Remediation

Understand what actions to take to alter a prediction in the future by changing the inputs to the model. For example, a consumer could remediate, in this case improve, a credit-rating prediction by reducing their debt.

Understand model behavior

Construct a simplified model in the user's mind, which can be used as a surrogate for understanding the model's performance.

Monitoring

Ongoing assessment that a model's performance remains acceptable.

An explanation consumer can have multiple simultaneous expertises and intents. In some cases, these intents can even appear contradictory. A motorist who receives a speeding ticket from an AI that tracks vehicle speeds and decides to issue tickets may simultaneously want to verify the performance of the model, and also understand what actions they could take in the future to avoid another ticket.

In practice, we have often found that people within the three main groups of explanation consumers are united by their primary expertise and needs, as shown in Table 7-1.

Table 7-1. The primary expertise and intents of explanation consumers

Consumer	Subgroup	Expertise(s)	Intent(s)
ML practitioners	Data scientist	ML	Model improvement
	ML engineer	ML	Monitoring
Observers	Business stakeholder	Domain, inputs	Trust, verify, understand
	Regulator	Domain, inputs	Verify, understand
End users	General	Inputs	Trust, remediation
	Specialist	Domain, inputs	Trust, understand

Often, as consumers become more familiar with an ML system, their expertise and intents will change. A common pattern is for data scientists to become domain experts over time, and end users, particularly in business and industries, will shift from validating a system to working to understand the model so they can anticipate and be better prepared for prediction results and common failures.

Each combination of these expertises and intents can result in a different type of explanation being more or less useful. Our advice is to think of *expertise* as the background you can assume for an explanation, and *intent* for where you want the explanation to guide the consumer toward.

PICKING THE RIGHT EXPLANATION TECHNIQUE FOR YOUR AUDIENCE

The act of choosing which explanation technique is best for your audience can be difficult due to the lack of definitive guidance and vague information in the ML community. If one believes the research literature, every new technique is strictly a vast improvement over all previous explanation techniques. Conversely, there is also a corresponding number of papers demonstrating how various explanation techniques are very bad at their job. With this conflicting advice, how does one pick the right technique? By answering the following questions, you should be able to filter many possible techniques to just one or two that meet your audience's needs:

1. What needs to be explained? A single inference, a cohort of predictions, or the global behavior of the model?
2. What is the audience's expertise? Does the technique require an understanding of how ML works to interpret the results?
3. What action will they take after an explanation? Should the technique generally inform, or should it present specific details?
4. Is this ML model being used in a critical or high-risk situation? Different explanation techniques offer varying levels of guarantees and robustness.
5. How quickly do they need an explanation? The latency of generating explanations can range from milliseconds to minutes or longer.

For widely used combinations of expertise and intent, we have put together a guide of possible techniques to use. However, view these recommendations as a starting point rather than a definitive list—the best technique for your situation may vary! See Table 7-2 for a list of suggested pairings.

Table 7-2. Suggested techniques for the different expertises and intents of an ML consumer

Expertise	Intent	Explainability technique
ML	Model improvement	Sampled Shapley, Integrated Gradients, example-based, layer-based
Domain	Model monitoring	Feature attributions
	Build trust	Example-based, independent conditional expectations, distillation
	Verify	Feature attributions, concept activations, pixel attributions
Inputs	Build trust	Example-based, region-based attributions
	Verify	Partial-design plots, feature attributions
	Remediation	Example-based, concept activations

Finally, another important group of explanation consumers is other technological systems. As an example, Google has had great success in using feature attributions to perform automated model monitoring. When abnormal drift or skew is detected in feature attributions for a newly released model, it is automatically rolled back and the previous version of the model is used. However, the topic of using explanations in an automated fashion is outside the scope of this book.

Now that you understand how XAI techniques can be used by different types of consumers and their intentions, let's turn our attention to how to visualize and present these explanations.

How to Effectively Present Explanations

Despite the whole point of explainability being to bring clarity to an ML system, explanations are often poorly presented to ML consumers. This can lead to false assumptions, misplaced confidence in an ML system, and the wrong actions being taken. Presenting explanations through a visualization, user interface, or even plain text, should serve three goals:

- Clarify what, how, and why an ML system performed the way it did.
- Accurately represent what is known in the explanation.
- Start from the ML consumer's place of understanding and build upon it.

Information visualization is an entire field of practice and research. If you want to learn more, we suggest reading *The Visual Display of Quantitative Information* by Edward Tufte (Graphics Press, 2001).

We're going to avoid the topic of explanations which themselves are interactive via UI. Although there have been research initiatives to create interactive explanations, all of these approaches are in their infancy and have not been proven to deliver at scale in the broader ML community. This comes with a caveat that we expect the future of explanations *is* interaction and collaboration, so we recommend you continue to look for opportunities in using interactive explanations.

Clarify What, How, and Why the ML Performed the Way It Did

Each explanation should provide the consumer with all the key pieces of information they need to interpret the explanation. Often, this requires additional auxiliary information to be displayed so the intent of the explanation can be understood. At a minimum, the explanation should represent:

- What is the range for a value in the explanation?
- When was the explanation generated?

- If the explanation is visual, use a recommended color scheme for good contrast in gradients, and be color-blind friendly. ColorBrewer has designed and tested color palettes that are easily perceived.
- Show the predicted value(s). If it is a cohort, provide some information about how the cohort was defined.

For consumers with ML expertise, or those with an intent to verify, the explanation should also include any relevant versioning info for the dataset and model (e.g., training/validation split, randomization seed, hyperparameters, etc.) and details of the parameters chosen for the explanation technique.

We all enjoy stories, and it is natural to convey explanations through narratives. This can be a great way to convey highly technical information to a nontechnical audience. That being said, there are two risks to using narratives. First, it is easy to get swept away in trying to make the story more compelling by reaching for more impactful terms and verbiage. Second, stories almost always make use of a timeline, implying causality. In both situations, you are inadvertently conveying unfounded accuracy or certainty in the explanation to your ML consumers. Using a more journalistic style of writing, or the inverted pyramid[2] format, can be a good structure for presenting explanations as stories to users.

Accurately Represent the Explanations

The obvious goal of any explanation is to provide an accurate understanding of the model's behavior. It is equally important, but less intuitive, to be conscientious of ensuring the accuracy of the explanation itself. Raw explanations are almost always a set of numbers, and it is vital to ensure they are accurately represented when the explanation is translated into other formats (e.g., as a saliency map, bar chart, token highlighting, generated text, etc.). For example, when explaining the predictions of an image model, Integrated Gradients highlight the specific pixels that contributed to the explanation. In contrast, techniques like XRAI and LIME visualize regions that contributed to a prediction, which is less precise. Depending on your audience, either technique may be more or less appropriate. For example, a regulator seeking to verify an ML classifier for X-rays may wish to see evidence that the model was attentive to the actual X-ray scan and not ancillary, or leaked, information such as text

overlaid in the X-ray image. Displaying the exact attributions of different pixels will be useful in verifying the system is behaving as expected.

In contrast, for a patient, it may be best to use a technique that does not mislead users into overconfidence in the accuracy of the system. For our X-ray example, we may know the system has high accuracy, but displaying exact pixel-level attributions may overcommunicate the confidence of the ML system and be counterproductive to aiding patients and medical professionals in understanding the diagnosis. In situations like these, where there is a lack of ML or domain expertise, a regional explanation technique can best convey the accuracy of the explanation, erring on the side of caution.

A common critique of explainability is that the techniques do not faithfully, or accurately, represent the actual decision-making process learned by the ML model. Many papers have been written stating that explainability should not be used due to the perceived inaccuracy of the techniques (see the Appendix for a list of suggested papers to read). However, explainability is often a necessity, not a luxury, so in many situations, you must present explanations when there is no good alternative, such as using an inherently interpretable model. It is important to not minimize the contribution of these papers to our understanding of Explainable AI, but instead view them as guides for when, and how, explainability will fail.

The majority of research on how XAI techniques fail can be grouped into two categories: the inaccuracy of the explanations, and how brittle the techniques are. The first group critiques the technical accuracy, or faithfulness, of a method to represent the model's true behavior. The second group of criticisms is focused on how robust explanations are to manipulation and noise.

Technical accuracy of explanations

Technical accuracy can be measured in several ways, such as how well the technique represents the way the model works, the numerical accuracy of the explanation value, or an independent benchmark, such as how precisely a salience map technique for an image classifier correctly outlines the shape in the image. Technical accuracy is perhaps the most utilitarian, and uncontroversial, set of research to look at when thinking about the appropriateness of a method for a given consumer. Some types

of inaccuracies are mostly irrelevant to a user. For example, an end user may not expect, or be able to differentiate, between an explanation that is 90% accurate from one that is 100% accurate. This premise is what makes sampled Shapley (covered in 2 and 3) still useful, even though they represent an approximation of the true Shapley values for attributing influence to dataset features. In contrast, Grad-CAM's (Chapter 4) inaccuracy in attributing pixel influence for a multiclassification model renders it unsuitable for almost all consumers who do not have ML expertise and would understand this important exception when viewing an explainability result.

Ironically, for all the focus on evaluating the technical accuracy of explainability methods, little research has been performed on another type of accuracy aligned to our ultimate goal of having ML consumers accurately comprehend the model. We call this the *presentation efficacy*: how well the presentation of an explanation conveys information to and is understood by ML consumers. Representing an explanation with 100% efficacy means the ML consumer perfectly understood all the information conveyed by the visualization. In contrast, a presentation efficacy of 50% would mean half of the understanding was lost due to the presentation.

Consider the two explanations in Figure 7-2; both have the same underlying explanation. By using high-contrast colors, the explanation on the left has used good information visualization practices, making it easy to understand what pixels are influencing the model's classification of the bird as a cockatoo. In contrast, the explanation on the right uses the same saliency map, but rather than using a combination of high-contrast colors to represent pixel attribution, it uses white. This makes it nearly impossible to distinguish characteristics of the explanation once it is overlaid onto the original image of a white cockatoo.

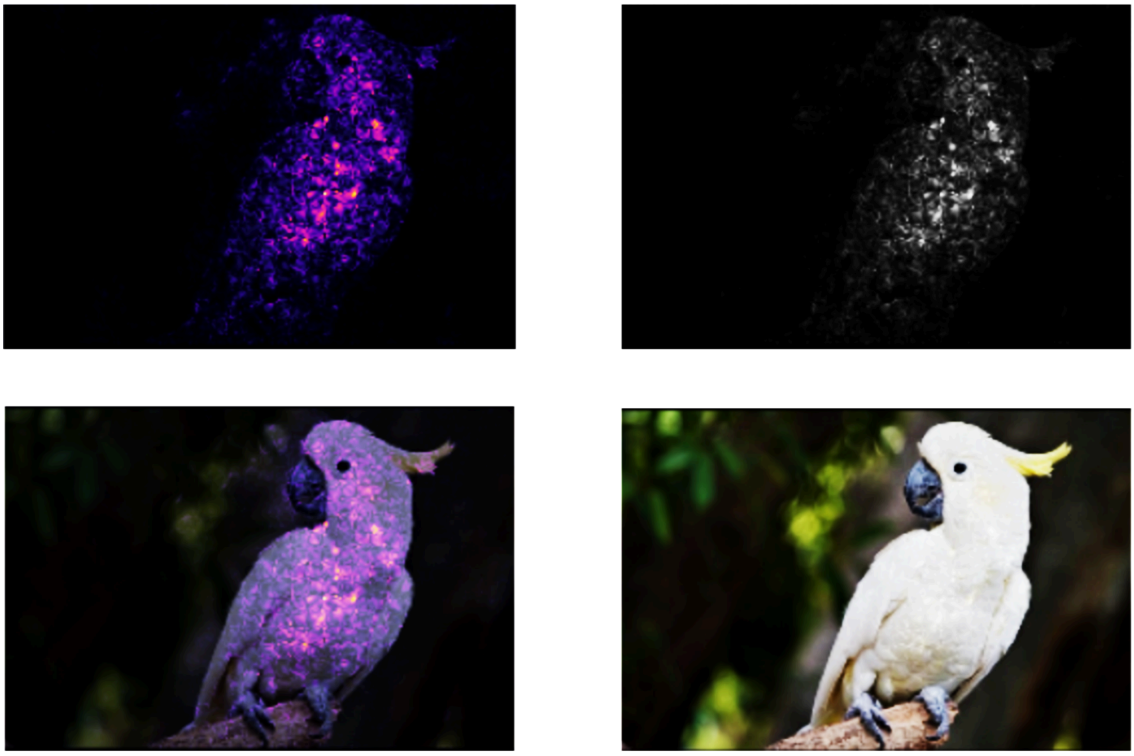

Figure 7-2. The same underlying explanation is portrayed in two different ways to illustrate how a bad visualization has very low presentation efficacy.

As of 2022, we are not aware of any research that has studied this presentation efficacy to understand topics such as thresholds for comprehension or how it may differ between types of ML consumers. In our opinion, presentation efficacy is the largest potential danger you will encounter as an ML practitioner in using, sharing, and embedding Explainable AI in your ML systems. Many Explainable AI techniques come with visualization methods that often violate core information visualization principles or, in the pursuit of aesthetics, overrepresent how accurate their explanations are.

As an example of the overrepresentation flaw, many techniques for image models mistakenly display explanation masks (heatmaps, outlines, etc.) at a higher resolution than what was generated by the technique. As these masks are often overlaid on top of the image input, the mask displays attributions to pixels that were never even seen by the model or the explanation method. Understanding how this occurs, and why it is often done, are vital to giving explainability consumers accurate explanations.

This overrepresentation comes from trying to apply an explanation mask to the original, unprocessed image used as an input to the model. Many image models crop and downsample an image to the input dimensions, often dramatically given the size of many images today compared to the common resolutions used by models (e.g., 224×224, 128×128, or 64×64 pixels). The resulting explainability mask is the same size as the input di-

mension (i.e., 224×224), but the designer of the explainability technique or the ML practitioner wishes to overlay the mask onto the original image, which is often at resolutions in the megapixels (e.g., more than 1024×1024). The easiest way to do this is to use an image transformation library to upsample (also known as upscale) the explanation mask using a standard image library to interpolate to the size of the original image. Modern interpolation methods are designed to "fill in the blanks" in a visually pleasing way. Figure 7-3 shows an example of upsampling on an image from the MNIST dataset using the nearest neighbors algorithm, the most basic of interpolation methods.[3] All MNIST images are purely black and white, so any gray areas in the upsampled image are approximations by the algorithm of what the pixels should look like in that region.

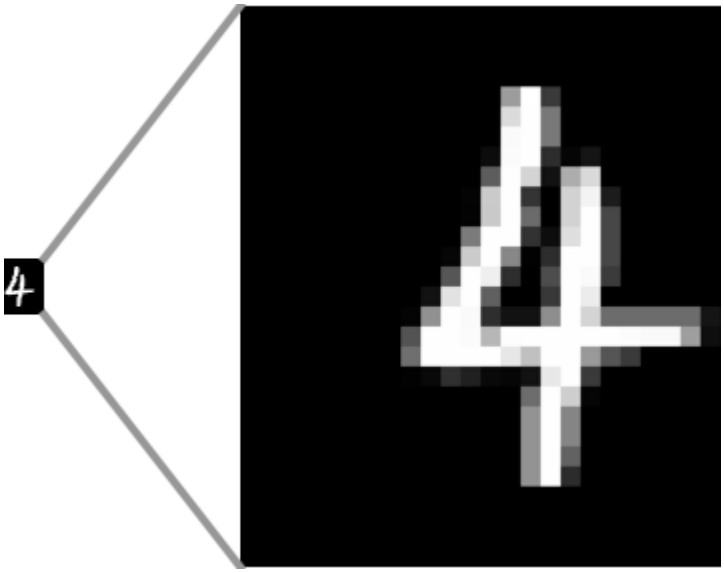

Figure 7-3. Upsampling an image of a 4 from the MNIST dataset to 10× its size.

However, this interpolation is based on assumptions on what the new pixels should look like, rather than deriving their values from an authoritative source. As a result, upsampled explanation masks will highlight pixel attributions that never existed. Figure 7-4 shows an example of this issue. For the upscaled saliency map, erroneous attributions are introduced, resulting in light green pixels. The problem can most clearly be seen when we overlay the saliency map on top of the original image. Whereas the original pixel attribution technique showed the model as only being influenced by four pixels in a low-resolution diamond, the upscaled saliency map begins to imply that there was a fuzziness to the pixel attribution technique. This is most problematic when interesting features, such as edges or patterns, were lost in downscaling or cropping the original feature.

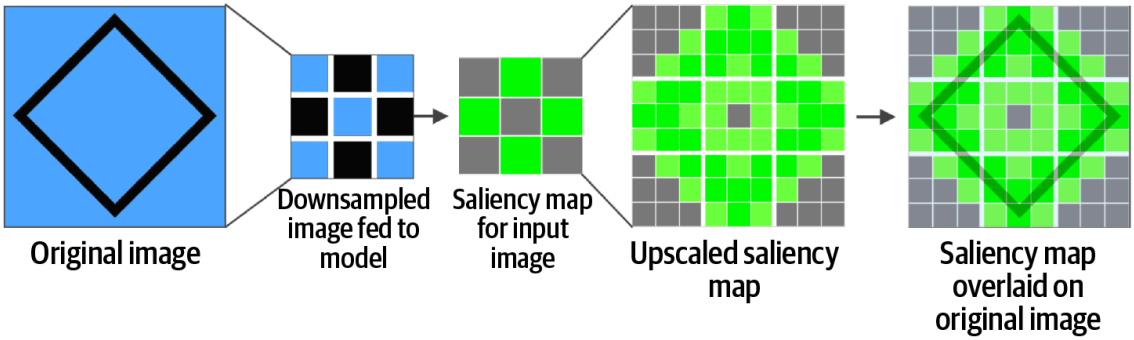

Figure 7-4. Example of an upsampled image mask and what pixel attributions were interpolated.

To avoid this problem, avoid upsampling explanation masks using interpolation. Two options, seen in Figure 7-5, are to upsample with no interpolation, which results in a disjointed but still faithful explanation mask, or to interpolate with some sort of pattern or shading to indicate "we don't know what the explanation would be for these pixels."

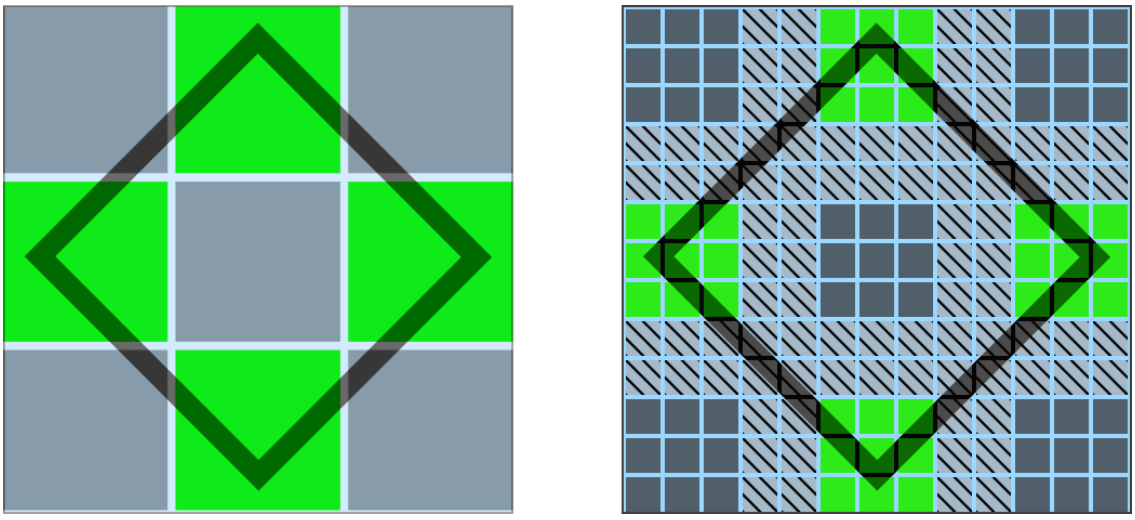

Figure 7-5. Illustrative examples, not meant for actual use, of how one could accurately show a saliency map that is resized to the original image.

Another common technical inaccuracy results from explainability techniques that do not perform per-element attributions of the input features. While this rarely occurs for structured data models, it is common to do this for feature attributions in time-series models and for region-based explanation techniques for image or text models. For example, many image explanation techniques that generate heatmap explanation masks will display a gradient to interpolate between different regions of the heatmap. This is inaccurate because the interpolation is introduced *after* the explanation has been calculated and conveys a smooth transition in model influence that may not actually exist.

A common rationale for using these gradients is that CNN models utilize a similar smoothing kernel in their convolution (downsampling) layers, so

it is okay to use a similar smoothing approach in the explanation. However, the ML community has not studied the effectiveness of this claim as of the summer of 2022.

Brittleness in explanations

There is a growing body of research focused on the brittleness of different explanation techniques. *Brittleness* is the inability of a system to perform well outside of its original design parameters. For explainability, this takes the form of adversarial attacks and artificial noise injected into inputs.[4] For many techniques, it is clearly demonstrated they do not handle brittle attacks with much grace. For example, researchers were able to inject noise into an image that resulted in the image mask generated by the explanation technique spelling out different words, as shown in Figure 7-6.[5]

Figure 7-6. From the work of Dombrowski et al., the original image of a dog on the left can be imperceptibly manipulated to the image on the right to such an extent that words can be created in the explanation output.

WHAT ARE ADVERSARIAL ATTACKS?

In machine learning, an adversarial attack is an attempt to trick or influence a model's prediction with deceptive or manipulated data. Evasion tactics are the most common and typically involve some data modification in the form of adversarial examples; i.e., an instance with small, imperceptible changes that can fool the model into making the wrong prediction. For example, glasses or clothing that have been designed to evade facial recognition software or reflective tape that is used to trick license plate readers. These kinds of adversarial examples and attacks can pose a real problem for real-world ML systems. In 2017, scientists at MIT's LabSix, an AI research group, caused Google's image recognition model to classify a 3D-printed toy turtle modified with a slight texture as a rifle and classify a cat as guacamole.

Adversarial attacks can also take the form of model poisoning or model extraction. An extreme example of model poisoning would be when the Microsoft Twitter chatbot Tay (now defunct) was corrupted by users' input from producing light, playful conversation as intended to instead creating misogynistic and racist rants. As an example of model extraction, it's been shown that large language models[6] like GPT-3 can actually leak details from the training data and can be prompted to return private and personally identifiable information such as names, email addresses, and phone numbers when certain manipulated keywords or phrases are sent for prediction.

In other cases, researchers have been able to demonstrate[7] how image-based explanations can be manipulated into misexplanations through fine-tuning the model, with no significant loss in accuracy; an example is shown in Figure 7-7.

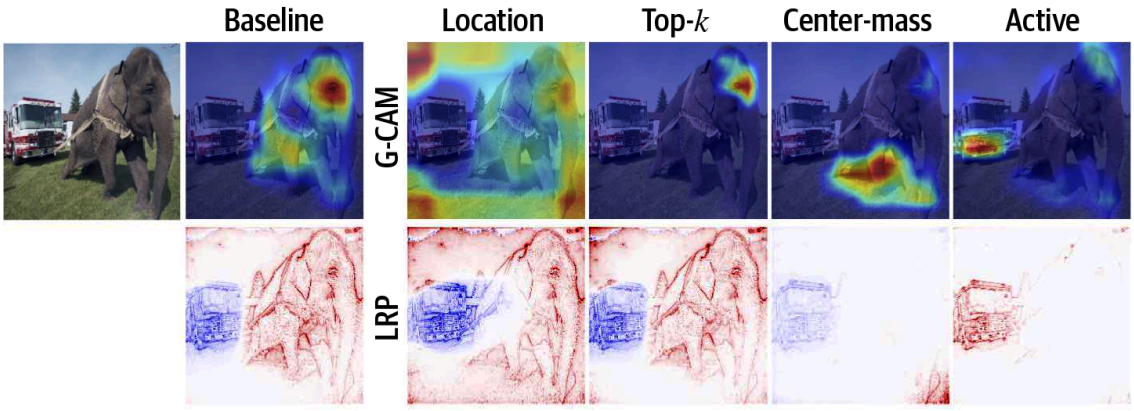

Figure 7-7. By fine-tuning the model's weights, Heo et al. were able to change the results of both Grad-CAM (Chapter 4) and LRP (Chapter 5), with no meaningful loss in the model's performance. (Print readers can see the color image at https://oreil.ly/xai-fig-7-7.)

Research into the brittleness of explainability techniques has primarily focused on image models and image-focused explanation techniques, but it is reasonable to assume many of these adversarial attacks could be adapted to other modalities. What is also not clear, as of 2022, is whether the explanation techniques are themselves performing poorly, or whether they are surfacing turmoil within the model.

While it is tempting to draw the conclusion that these explainability techniques should never be used due to their shortcomings, you should also consider how likely it is these issues may be encountered in your ML system when it is deployed in the real world. For example, how susceptible are the inputs of your model to adversarial attack? If your model draws entirely from factual internal sources, e.g., sales data or camera images on a factory floor, it is unlikely that the explanation will be exposed to adversarial attacks (or, if it is, this may be the least of your problems). Conversely, an AI that flags user comments in a forum is highly exposed to adversarial attacks, and we could expect any accompanying explanations to be vulnerable as well.

WITH HIGH-RISK SYSTEMS, ENSURE EXPLANATIONS DO NOT OVERREPRESENT THEIR VALIDITY

If your AI is being used in a high-risk system, e.g., in the medical domain or justice system, be careful to ensure your target audience does not place more emphasis on the explanation than is warranted by the technique. In such cases, we often observe that explainability is used with an intent to improve trust in the system with end users. However, these explanations are often treated as absolute truths by ML consumers and used to justify their conclusions, rather than a *probable* explanation for a model's behavior.[8]

For other, more primitive, explanation techniques such as PDP plots (discussed in Chapter 3), the risk of conveying inaccurate explanations through poor visualizations is much lower. Issues to be aware of include changing scales in the axes between explanations, which consumers may not notice, and plots that are so small they can be overinterpreted.

Build on the ML Consumer's Existing Understanding

The most useful explanations for consumers are those that build on their existing knowledge, either of the inputs, prediction, or ML model, to gain a more sophisticated understanding. To understand why this is, we must first understand a few aspects of human-computer interaction: mental models, situational awareness, and satisficing.

Mental models are very similar to ML models: they represent a framework that a person has learned that lets them quickly and efficiently reach conclusions and make decisions. Like ML, mental models often do not truly represent the actual system they model. As an example, most people's mental model of how to drive a car is that pressing the gas pedal makes the car go faster. In reality, the gas pedal controls the amount of air and fuel flowing into a car engine, which then causes a larger combustion, and in turn, causes the engine to exert more force through gears in a transmission. The gearing in this transmission allows the engine to exert more or less force depending on the speed the wheels are rotating at. While the true model of the car is more accurate, and explains why pressing a gas pedal at different speeds does not make the car accelerate the same amount, reasoning through this process every time would make driving much more onerous. Most of the time, it is sufficient to use a simpler mental model that pushing the gas pedal makes the car go faster.

For Explainable AI, the best explanations match the consumer's mental model of the ML system, or are able to help them build a sufficiently accurate mental model. It's most effective to evaluate how well the frameworks of the explanation and the mental model match each other when deciding between different families of XAI techniques. Determining user's mental models for different ML systems is done primarily through conducting user interviews and research. This is a time-consuming process that requires a trained UX researcher. Assuming your ML system is replacing an existing process, one shortcut to discovering this pairing is to ask users how they build confidence in decisions without an AI. For example, when classifying cancer in cell tissue slides, pathologists often justify their decision by referencing textbook images that represent canonical examples of cancers in cell tissue. In this case, using an example-based (or counterfactual) technique would be the best pairing for the

pathologist end user, as shown by the SMILY app in Figure 7-8, developed by Google Health.

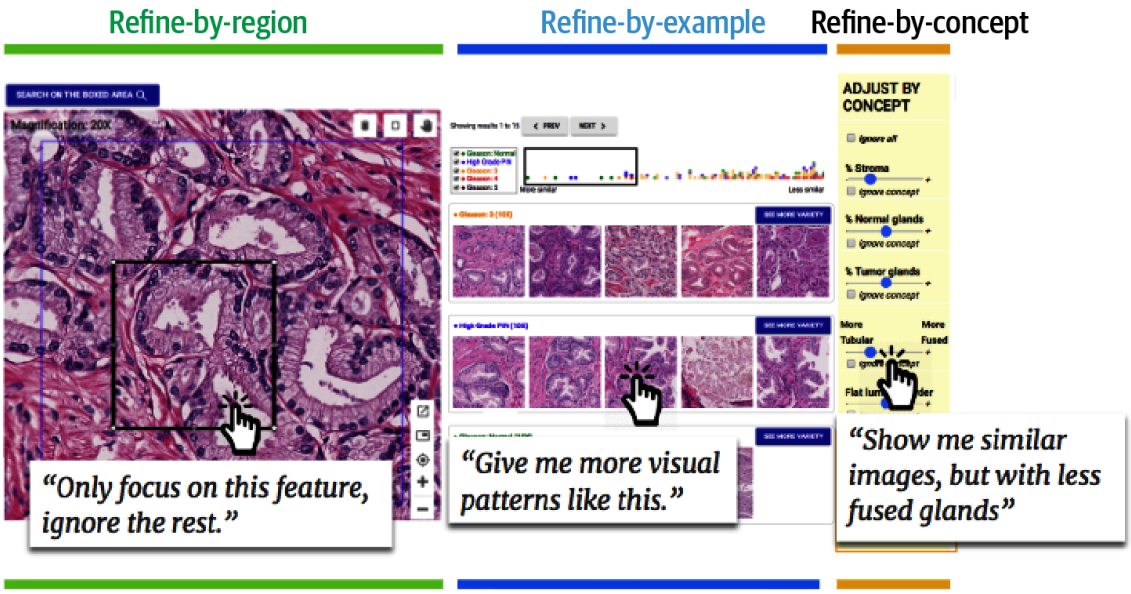

Figure 7-8. SMILY uses example-based explanations and concept explanations to help pathologists understand cell tissue slides.

Situational awareness describes how well a person understands a given scenario, and it is also a process by which people try to make decisions or arrive at conclusions when confronted with new circumstances. For ML consumers, they are often trying to improve their situational awareness when they ask the question, *Why did the ML model behave that way?*

The three steps of situational awareness are:

1. *Perceive*

   Determine what new information to gather and collect this information.

2. *Comprehend*

   Using new and existing information, build an understanding of the current situation.

3. *Project*

   With the current understanding, create assumptions about what will happen in the future. This could be which actions should be taken to change the situation, or how the situation itself will continue to evolve.

Situational awareness also heavily relies on users having a correct mental model; otherwise, it is very difficult to accurately comprehend and project, or even know what information to gather. Building situational awareness is a large field of research in its own right in human-computer interaction, so we will only briefly discuss how this pertains to ML con-

sumers of explanations. Explanations are most influential when a person is perceiving or comprehending. As the ML practitioner, your choice of explanation technique also dictates the information available to the consumer when they are trying to gather new information about the rationale behind a prediction. Similarly, useful explanation techniques are those that best help an ML consumer comprehend the ML. Unfortunately, Explainable AI currently does little, beyond setting the stage, to help users as they project. In the future, we are looking forward to XAI that is closer to an interactive dialogue to help consumers explore different scenarios as part of improving their ability to project the future behavior of an ML model.

Satisficing is a common human behavior that might be best described as "good enough for the amount of time I've got right now." More formally, people are extremely good at deriving a generally optimal solution to a problem, doing it far faster than would be expected given the time it takes them to determine the truly optimal solution, but at the cost of relying on heuristics and stereotypes. Satisficing has been observed across all professions and, counterintuitively, as someone becomes more of an expert at their job and is faster at finding the best solution, they are no less likely to satisfice. For you, this is an important consideration of how you can expect ML consumers to interact with explanations—quickly and forming conclusions that are generally correct but may miss nuances. Anecdotally, we have seen this often arise across many model modalities and explanation techniques.

For example, consumers of feature attribution charts often summarize that features with minimal contributions (even if negative) for an individual prediction are not true for the entire dataset. For example, in a classification model, it may be that most predictions are not influenced by a feature, but for those along a decision boundary, the feature is highly influential. Another example of this approach is when one quickly assigns erroneous, semantic meaning to explanation heatmaps in images, when it is clear the model does not have the capacity for semantic representation; e.g., "clearly the model learned to recognize a cat because the cat ears and eyes are highlighted" rather than the more precise "this region of pixels contains many edges that are indicative of a cat." Satisficing is useful, and with it your ML consumers will probably arrive at the right conclusion most of the time more quickly. However, it also causes failures in the

cases that may be of most interest to you and these consumers, where the model is not behaving as expected.

To counter satisficing, try to carefully curate the information presented in the explanation to the user. For example, many feature- and pixel-attribution techniques do not show negative attributions because it helps ML consumers be more focused on signals that drive the model toward the prediction (rather than away from it). It is also useful to design explanations to help answer a specific question for a consumer, rather than just being a general dashboard of the model's health.

With this discussion of mental models, situational awareness, and satisficing, it is also worth asking if explanations can be used to teach users how an ML model works. There is little research into this area of explainability, but we also expect that this would be a difficult use of explanations in their current form. By definition, the techniques we've presented in this book seek to explain a model *after* it has made a prediction, in a post hoc fashion. By asking if explainability can teach a user about how an ML model works, we are also asking if explanations could be used to create a surrogate, interpretable ML model. Whether this is feasible is still an active area of research.

Common Pitfalls in Using Explainability

In using explanations, we find there are a few common pitfalls for ML consumers. These situations occur not because there is something wrong with the explanation or ML model but come from consumers improperly understanding or using explanations. Most result from overreliance or overconfidence in the explanation technique but are also driven by how explainability results are packaged and delivered to consumers. The three most common pitfalls are assuming causality, overfitting "intent" to a model, and leveraging additional explanations in an attempt to augment the original explanation.

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 fea-

ture 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 overreach, you can help channel this positive energy in a productive direction by making it easy to retrieve additional explanations with the same technique, or proactively determining what techniques can be combined in advance and offering those to the user.

Summary

In this chapter, we discussed what happens after an explanation is generated by looking at how ML consumers interact with explanations. We introduced a framework of expertise and intents for ML consumers. These expertises, such as ML, domain, and inputs, combined with intents like improving, monitoring, understanding, and validating models, along with building trust in the AI and performing remediations, allow us to understand how to best match our audience with the right type of explanation. We then turned our attention to displaying explanations, highlighting best practices such as following information visualization guidelines, conveying the accuracy of an explanation, and building upon an ML consumer's existing understanding. Finally, we discussed common pitfalls that occur when users interact with explanations and identified some ways of avoiding these issues.

So you've now gotten to a point where your ML consumer has received the explanation, understood it, and is ready to go to the next step. But what is that next step? Throughout this book, we have occasionally referenced how explanations can be used to understand or improve a model. For ML practitioners, these are examples of an important part of model control and analysis. Understanding how a model works, we can analyze its behavior and make smarter choices about how to improve the model or dataset. For end users, though, this type of model analysis often is part of a larger decision support system, where they are synthesizing many competing sources of information and evaluations to decide on the best course of action. A sales forecaster at a large grocery chain may use your explanations to understand what time of year they can expect to sell the most strawberries. However, before deciding to place that order for a cargo ship full of berries, they are also likely to check the expected strawberry yield this season. For a regulator that is trying to establish confidence in your AI, and validate its predictions, this explanation may be one of many factors in an audit of the overall performance of a company's use of technology. To put it succinctly, generating, consuming, and interacting with an explanation is just the start.

1 This may sound analogous to the previous intent of building trust but is a distinct intent because there is no need for the model to perform well to be verified that it is working within internal and external constraints (e.g., stakeholder business requirements and industry-wide regulations).

2 The Nielsen Norman Group, one of the most respected UX research firms in the world, has an excellent article on the inverted pyramid.

3 Bicubic smoothing is by far the most popular technique, resulting in fuzzy-looking images. However, it is harder to see the flaws of interpolating in bicubic smoothing because our visual system naturally accommodates for images that are slightly blurry.

4 To the best of our knowledge, artificial perturbations of the network weights, say via retraining, and their effect on explainability techniques, has not been studied.

5 Ann-Kathrin Dombrowski et al., "Explanations Can Be Manipulated and Geometry Is to Blame," arXiv, 2019, *https://arxiv.org/abs/1906.07983*.

6 Nicholas Carlini et al., “Extracting Training Data from Large Language Models,” arXiv, 2021.

7 Juyeon Heo et al., “Fooling Neural Network Interpretations via Adversarial Model Manipulation,” arXiv, 2019.

8 An *absolute truth* explanation would be to trace an explanation throughout the entire ML’s decision process/layers and annotate each weight and variable to reference its influence. It would also be absolutely incomprehensible.