Considerations for Evaluation and Generalization in Interpretable Machine Learning

Finale Doshi-Velez* and Been Kim*

August 24, 2018

1 Introduction

From autonomous cars and adaptive email-filters to predictive policing systems, machine learning (ML) systems are increasingly commonplace; they outperform humans on specific tasks [Mnih et al., 2013, Silver et al., 2016, Hamill, 2017] and often guide processes of human understanding and decisions [Carton et al., 2016, Doshi-Velez et al., 2014]. The deployment of ML systems in complex, realworld settings has led to increasing interest in systems optimized not only for expected task performance but also other important criteria such as safety [Otte, 2013, Amodei et al., 2016, Varshney and Alemzadeh, 2016], nondiscrimination [Bostrom and Yudkowsky, 2014, Ruggieri et al., 2010, Hardt et al., 2016], avoiding technical debt [Sculley et al., 2015], or satisfying the right to explanation [Goodman and Flaxman, 2016]. For ML systems to be used robustly in realworld situations, satisfying these auxiliary criteria is critical. However, unlike measures of performance such as accuracy, these criteria often cannot be completely quantified. For example, we might not be able to enumerate all unit tests required for the safe operation of a semi-autonomous car or all confounds that might cause a credit scoring system to be discriminatory. In such cases, a popular fallback is the criterion of *interpretability*: if the system can *explain* its reasoning, we then can verify whether that reasoning is sound with respect to these auxiliary criteria.

Unfortunately, there is little consensus on what interpretability in machine learning *is*—let alone how to *evaluate* it for benchmarking or reason about how it may *generalize* to other contexts. Current interpretability evaluation typically falls into two categories. The first evaluates interpretability in the context of an application: if the interpretable system provides human-understandable explanation in either a practical application or a simplified version of it, then it must be interpretable (e.g. [Ribeiro et al., 2016, Lei et al., 2016, Kim et al., 2015a, Doshi-Velez et al., 2015, Kim et al., 2015b]). The second evaluates interpretability via a quantifiable proxy: a researcher might first claim that some model class—e.g. sparse linear models, rule lists, gradient boosted trees—are interpretable and then present algorithms to optimize within that class (e.g. [Buciluǎ et al., 2006, Wang et al., 2017, Wang and Rudin, 2015, Lou et al., 2012]).

To large extent, both evaluation approaches rely on some notion of "you'll know it when you see it." Should we be concerned about a lack of rigor? Yes and no: the notions of interpretability above appear reasonable because they *are* reasonable: they pass the first test of having face-validity on the correct test set of subjects: human beings. However, this basic notion leaves many kinds of questions unanswerable: Are all models in all defined-to-be-interpretable model classes equally interpretable? Quantifiable proxies such as sparsity may seem to allow for comparison, but how does one think about comparing a model sparse in features to a model sparse in prototypes? Moreover, if one builds and evaluates an interpretable machine learning model from a particular dataset for a particular application, does that provide insights on whether the model will be similarly interpretable with a different dataset or different application? If we are to move this field forward—to compare methods and understand when methods may generalize—we need to formalize these notions and make them evidence-based.

The objective of this chapter is to describe a set of principles for the evaluation of interpretability. The need is urgent: European Union regulation may *require* algorithms that make decisions based on user-level predictors and "significantly affect" users to provide explanation ("right to explanation") [Parliament and of the European Union, 2016]. Meanwhile, interpretable machine learning is an increasingly popular area of research, with forms of interpretability ranging from regressions with simplified functions (e.g. [Caruana et al., 2015, Kim et al., 2015a, Rüping, 2006, Buciluǎ et al., 2006, Ustun and Rudin, 2016, Doshi-Velez et al., 2015, Kim et al., 2015b, Krakovna and Doshi-Velez, 2016, Hughes et al., 2016]), various kinds of logic-based methods (e.g. [Wang and Rudin, 2015, Lakkaraju et al., 2016, Singh et al., 2016, Liu and Tsang, 2016, Safavian and Landgrebe, 1991, Wang et al., 2017]), methods of probing black box models (e.g. [Ribeiro et al., 2016, Lei et al., 2016, Adler et al., 2016, Selvaraju et al., 2016, Smilkov et al., 2017, Shrikumar et al., 2016, Kindermans et al., 2017, Ross et al., 2017, Singh et al., 2016]). International conferences regularly have workshops on interpretable machine learning, and Google Scholar finds more than 20,000 publications related to interpretability in ML in the last five years. How do we know which methods work best when? While there have been reviews of interpretable machine learning more broadly (e.g. [Lipton, 2016]), the lack of consensus on how to evaluate interpretability limits both research progress and the effectiveness of interpretability-related regulation.

In this chapter, we start with a short discussion of what interpretability is (section 2). Next we describe when interpretability is needed, including a taxonomy of use-cases (Section 3). In Section 4, we review current approaches to evaluation and propose a taxonomy for the evaluation of interpretability—application-grounded, human-grounded and functionally-grounded. Finally, we discuss considerations for generalization in Section 5. We review suggestions for researchers doing work in interpretability in section 6.

2 Defining Interpretability

According to the Merriam-Webster dictionary, the verb *interpret* means *to explain or to present in understandable terms.*[1] In the context of ML systems, we add an emphasis on providing explanation to humans, that is, *to explain or to present in understandable terms to a human.*

While explanation may be a more intuitive term than interpretability, we still must answer what then is an explanation? A formal definition of explanation remains elusive; we turn to the field of psychology for insights. [Lombrozo, 2006] argue that "explanations are more than a human preoccupation—they are central to our senses of understanding, and the currency in which we exchanged beliefs" and notes that questions such as what constitutes an explanation, what makes some explanations better than others, how explanations are generated and when explanations are sought are just beginning to be addressed. Indeed, the definition of explanation in the psychology literature ranges from the "deductive-nomological" view [Hempel and Oppenheim, 1948], where explanations are thought of as logical proofs to providing some more general sense of mechanism [Bechtel and Abrahamsen, 2005, Chater and Oaksford, 2006, Glennan, 2002]. More recently [Keil, 2006] considered a broader definition of explanations—implicit explanatory understanding. All the activities in the processes of providing and receiving explanations are considered as a part of what explanation means.

In this chapter, we propose data-driven ways to derive operational definitions and evaluations of explanations. We emphasize that the explanation needs within the context of an application may not require knowing the flow of bits through a complex neural architecture—it may be much simpler, such as being able to identify to which input the model was most sensitive, or whether a protected category was used when making a decision.

3 Defining the Interpretability Need

Interpretable Machine Learning as a Verification Tool In Section 1, we mentioned that interpretability is often used as a proxy for some other criteria. There exist many desiderata that we might want of our ML systems. Notions of *fairness* or *unbiasedness* imply that protected groups (explicit or implicit) are not somehow discriminated against. *Privacy* means the method protects sensitive information in the data. Properties such as *safety*, *reliability* and *robustness* ascertain whether algorithms reach certain levels of performance in the face of parameter or input variation. *Causality* implies that the predicted change in output due to a perturbation will occur in the real system. *Usable* methods provide information that assist users to accomplish a task—e.g. a knob to tweak image lighting—while *trusted* systems have the confidence of human users—e.g. aircraft collision avoidance systems.

There exist many ways of verifying whether an ML system meets such desiderata. In some cases, properties can be proven. For example, formalizations of fairness [Hardt et al., 2016] and privacy [Toubiana et al., 2010, Dwork et al., 2012, Hardt and Talwar, 2010] have

[1] Merriam-Webster dictionary, accessed 2017-02-07

resulted in algorithms that are guaranteed to meet those criteria. In other cases, we can track the performance of a system and validate the criteria empirically. For example, pilots trust aircraft collision avoidance systems because they knew they are based on millions of simulations [Kochenderfer et al., 2012] and these systems have an excellent track record.

However, both of these cases require us to be able to formalize our desiderata in advance, and, in the case of empirical validation, accept the cost of testing the ML system to collect data on its performance with respect to our desiderata. Unfortunately, formal definitions of auxiliary desiderata are often elusive. In such cases, explanation can be valuable to qualitatively ascertain whether desiderata such as fairness, privacy, reliability, robustness, causality, usability and trust are met. For example, one can provide a feasible explanation that fails to correspond to a causal structure, exposing a potential concern.

This observation, of interpretability as a *verification tool*, suggests that carefully thought-out work in interpretable machine learning should be able to specify *What are the downstream goals of this interpretable machine learning system?* and *Why is interpretability the right tool for achieving those goals?*

When is Interpretability the Right Tool? As noted above, there are many tools for verification. Not all ML systems require interpretability. Ad servers, postal code sorting, air craft collision avoidance systems—all can be evaluated without interpretable machine learning and perform their tasks without human intervention. In these cases, we have a formal guarantee of performance or evidence that the problem is sufficiently well-studied and validated in real applications that we trust the system's decision, even if the system is not perfect. In other cases, explanation is not necessary because there are no significant consequences for unacceptable results (e.g. an occasional poor book recommendation).

We argue that the need for interpretability stems from an *incompleteness* in the problem formalization, creating a fundamental barrier to optimization and evaluation. Indeed, in the psychology literature, [Keil et al., 2004] notes "explanations may highlight an incompleteness," that is, explanations can be one of ways to ensure that effects of gaps in problem formalization are visible to us.

Before continuing, we note that incompleteness is distinct from uncertainty: the fused estimate of a missile location may be uncertain, but such uncertainty can be rigorously quantified and formally reasoned about. In machine learning terms, we distinguish between cases where unknowns result in quantified variance—e.g. trying to learn from small data set or with limited sensors—and incompleteness that produces some kind of unquantified bias—e.g. the effect of including domain knowledge in a model selection process.

Below we provide some illustrative scenarios in which incomplete problem specifications are common:

- Scientific Understanding: The human's goal is to gain knowledge. We do not have a complete way of stating what knowledge is; thus the best we can do is ask for explanations we can convert into knowledge.

- Safety: For complex tasks, the end-to-end system is almost never completely testable;

one cannot create a complete list of scenarios in which the system may fail. Enumerating all possible outputs given all possible inputs be computationally or logistically infeasible, and we may be unable to flag all undesirable outputs.

- Ethics: The human may want to guard against certain kinds of discrimination, and their notion of fairness may be too abstract to be completely encoded into the system (e.g., one might desire a 'fair' classifier for loan approval). Even if we can encode protections for specific protected classes into the system, there might be biases that we did not consider a priori (e.g., one may not build gender-biased word embeddings on purpose, but it was a pattern in data that became apparent only after the fact).

- Mismatched objectives: The agent's algorithm may be optimizing an incomplete objective—that is, a proxy function for the ultimate goal. For example, a clinical system may be optimized for cholesterol control, without considering the likelihood of adherence; an automotive engineer may be interested in engine data not to make predictions about engine failures but to more broadly build a better car.

- Multi-objective trade-offs: Two well-defined desiderata in ML systems may compete with each other, such as privacy and prediction quality [Hardt et al., 2016] or privacy and non-discrimination [Strahilevitz, 2008]. Even if each objectives are fully-specified, the exact dynamics of the trade-off may not be fully known, and the decision may have to be case-by-case.

Additional taxonomies for situations in which explanation is needed, as well as a survey of interpretable models, are reviewed in [Lipton, 2016]. In this work, we focus on making clear that interpretability is just one tool for the verification, suited for situations in which problems are incompletely specified, and focus most of efforts on its evaluation. To expand upon our suggestion above, we suggest that research in interpretable machine learning should specify *How is the problem formulation incomplete?*

4 Evaluation

Once we know that we need an interpretable machine learning approach from Section 3, the next logical question is to determine how to evaluate it. Even in standard ML settings, there exists a taxonomy of evaluation that is considered appropriate. In particular, the evaluation should match the claimed contribution. Evaluation of applied work should demonstrate success in the application: a game-playing agent might best a human player, a classifier may correctly identify star types relevant to astronomers. In contrast, core methods work should demonstrate generalizability via careful evaluation on a variety of synthetic and standard benchmarks.

In this section we lay out an analogous taxonomy of evaluation approaches for interpretability: application-grounded, human-grounded, and functionally-grounded (see figure 1). These range from task-relevant to general, also acknowledge that while human

evaluation is essential to assessing interpretability, human-subject evaluation is not an easy task. A human experiment needs to be well-designed to minimize confounding factors, consumed time, and other resources. We discuss the trade-offs between each type of evaluation and when each would be appropriate.

Application-grounded Evaluation: Real humans, real tasks As mentioned in Sec. 3, interpretability is most often used a tool to verify some other objective, such as safety or nondiscrimination. Application-grounded evaluation involves conducting human experiments within a real application. If the researcher has a concrete application in mind—such as working with doctors on diagnosing patients with a particular disease—the best way to show that the model works is to evaluate it with respect to the task: doctors performing diagnoses. This reasoning aligns with the methods of evaluation common in the human-computer interaction and visualization communities, where there exists a strong ethos around making sure that the system delivers on its intended task [Antunes et al., 2012, Lazar et al., 2010]. For example, a visualization for correcting segmentations from microscopy data would be evaluated via user studies on segmentation on the target image task [Suissa-Peleg et al., 2016]; a homework-hint system is evaluated on whether the student achieves better post-test performance [Williams et al., 2016].

Specifically, we evaluate the quality of an explanation in the context of its end-task, such as whether it results in better identification of errors, new facts, or less discrimination. Examples of experiments include:

- Domain expert experiment with the exact application task.
- Domain expert experiment with a simpler or partial task to shorten experiment time and increase the pool of potentially-willing subjects.

In both cases, an important baseline is how well *human-produced* explanations assist in other humans trying to complete the task.

Finally, to make high impact in real world applications, it is essential that we as a community respect the time and effort involved to do such evaluations, and also demand high standards of experimental design when such evaluations are performed. As HCI community recognizes [Antunes et al., 2012], this is not an easy evaluation metric. Nonetheless, it directly tests the objective that the system is built for, and thus performance with respect to that objective gives strong evidence of success.

Human-grounded Metrics: Real humans, simplified tasks Human-grounded evaluation is about conducting simpler human-subject experiments that maintain the essence of the target application. Such an evaluation is appealing when experiments with the target community is challenging. These evaluations can be completed with lay humans, allowing for both a bigger subject pool and less expenses, since we do not have to compensate highly trained domain experts. Human-grounded evaluation is most appropriate when one wishes to test more general notions of the quality of an explanation. For example, to study what kinds

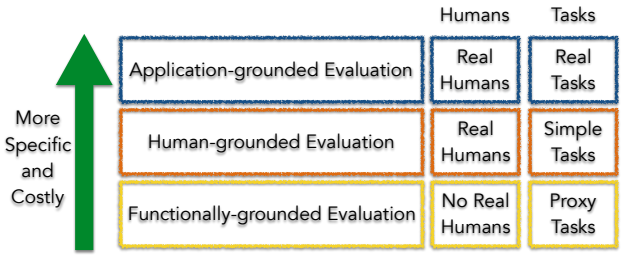

Figure 1: Taxonomy of evaluation approaches for interpretability

of explanations are best understood under severe time constraints, one might create abstract tasks in which other factors—such as the overall task complexity—can be controlled [Kim et al., 2013, 2014, Lakkaraju et al., 2016]

The key question, of course, is how we can evaluate the quality of an explanation without a specific end-goal (such as identifying errors in a safety-oriented task or identifying relevant patterns in a science-oriented task). Ideally, our evaluation approach will depend only on the quality of the explanation, regardless of whether the explanation is the model itself or a post-hoc interpretation of a black-box model, and regardless of the correctness of the associated prediction. Examples of potential experiments include:

- Binary forced choice: humans are presented with pairs of explanations, and must choose the one that they find of higher quality (basic face-validity test made quantitative).
- Forward simulation/prediction: humans are presented with an explanation and an input, and must correctly simulate the model's output (regardless of the true output).
- Counterfactual simulation: humans are presented with an explanation, an input, and an output, and are asked what must be changed to change the method's prediction to a desired output (and related variants).

As an example, the common intrusion-detection test [Chang et al., 2009] in topic models is a concrete form of the forward simulation/prediction task: we ask the human to find the difference between the model's true output and some corrupted output as a way to determine whether the human has correctly understood what the model's true output is.

Functionally-grounded Evaluation: No humans, proxy tasks Functionally-grounded evaluation requires no human experiments; instead, it uses some formal definition of interpretability as a proxy for explanation quality. Such experiments are appealing because even general human-subject experiments require time and costs both to perform and to get necessary approvals (e.g., IRBs), which may be beyond the resources of a machine learning researcher. Functionally-grounded evaluations are most appropriate once we have a class of models or regularizers that have already been validated, e.g. via human-grounded experiments. They may also be appropriate when a method is not yet mature or when human subject experiments are unethical.

The challenge, of course, is to determine what proxies to use. For example, decision trees have been considered interpretable in many situations [Freitas, 2014]. In section 5, we describe open problems in determining what proxies are reasonable. Once a proxy has been formalized, the challenge is squarely an optimization problem, as the model class or regularizer is likely to be discrete, non-convex and often non-differentiable. Examples of experiments include

- Show the improvement of prediction performance of a model that is already proven to be interpretable (assumes that someone has run human experiments to show that the model class is interpretable).
- Show that one's method performs better with respect to certain regularizers—for example, is more sparse—compared to other baselines (assumes someone has run human experiments to show that the regularizer is appropriate).

5 Considerations for Generalization

Identifying a need (Section 3) and being able to perform quantitative comparisons (Section 4) allows us to know that we are justified in our use of an interpretable machine learning approach and determine whether our approach is more interpretable than our baselines. However, we are often interested in more than just a comparison; we want insights on how our method might perform on other tasks.

For example, when it comes to the form of the explanation, [Subramanian et al., 1992] found that users prefer decision trees to tables in games, whereas [Huysmans et al., 2011] found users prefer, and are more accurate, with decision tables rather than other classifiers in a credit scoring domain. [Hayete and Bienkowska, 2004] found a preference for non-oblique splits in decision trees. When it comes to the amount of explanation, a number of human-subject studies have found that longer or more complex explanations can result in higher human accuracy and trust [Kulesza et al., 2013, Bussone et al., 2015, Allahyari and Lavesson, 2011, Elomaa, 2017], yet sparsity remains closely tied with interpretablity in the machine learning community [Mehmood et al., 2012, Chandrashekar and Sahin, 2014] (often citing the famous seven plus or minus two rule [Miller, 1956]). From this collection of results, are there ways to infer what method might perform well on a new task?

In this section, we describe a taxonomy of factors to describe contexts within interpretability is needed. These features can be used to link across experiments and the three types of evaluations, and thus being able to generalize to new problems where interpretability is needed. We also argue that a shared set of key terms for describing different interpretability contexts is essential to other researchers being able to find other methods that they should be including in their comparisons.

Task-related factors of interpretability Disparate-seeming applications may share common categories: an application involving preventing medical error at the bedside and an application involving support for identifying inappropriate language on social media might

be similar in that they involve making a decision about a specific case—a patient, a post—in a relatively short period of time. However, when it comes to time constraints, the needs in those scenarios might be different from an application involving the understanding of the main characteristics of a large omics data set, where the goal—science—is much more abstract and the scientist may have hours or days to inspect the model outputs.

Below, we list a set of factors that might make tasks similar in their explanation needs:

- *Global vs. Local.* Global interpretability implies knowing what patterns are present in general (such as key features governing galaxy formation), while local interpretability implies knowing the reasons for a specific decision (such as why a particular loan application was rejected). The former may be important for when scientific understanding or bias detection is the goal; the latter when one needs a justification for a specific decision.

- *Characterization of Incompleteness.* What part of the problem formulation is incomplete, and how incomplete is it? We hypothesize that the types of explanations needed may vary depending on whether the source of concern is due to incompletely specified inputs, constraints, domains, internal model structure, costs, or even in the need to understand the training algorithm. The severity of the incompleteness may also affect explanation needs. For example, one can imagine a spectrum of questions about the safety of self-driving cars. On one end, one may have general curiosity about how autonomous cars make decisions. At the other, one may wish to check a specific list of scenarios (e.g., sets of sensor inputs that causes the car to drive off of the road by 10cm). In between, one might want to check a general property—safe urban driving—without an exhaustive list of scenarios and safety criteria.

- *Time Constraints.* How long can the user afford to spend to understand the explanation? A decision that needs to be made at the bedside or during the operation of a plant must be understood quickly, while in scientific or anti-discrimination applications, the end-user may be willing to spend hours trying to fully understand an explanation.

- *Nature of User Expertise.* How experienced is the user in the task? The user's experience will affect what kind of *cognitive chunks* they have, that is, how they organize individual elements of information into collections [Neath and Surprenant, 2003]. For example, a clinician may have a notion that autism and ADHD are both developmental diseases. The nature of the user's expertise will also influence what level of sophistication they expect in their explanations. For example, domain experts may expect or prefer a somewhat larger and sophisticated model—which confirms facts they know—over a smaller, more opaque one. These preferences may be quite different from hospital ethicist who may be more narrowly concerned about whether decisions are being made in an ethical manner. More broadly, decison-makers, scientists, compliance and safety engineers, data scientists, and machine learning researchers all come with different background knowledge and communication styles.

Each of these factors an be isolated in human-grounded experiments in simulated tasks to determine which methods work best when they are present; more factors can be added if it turns out generalization within applications sharing these factors is poor. As mentioned above, these factors can also be used as key terms when searching for methods that might be relevant for a new problem.

Explanation-related factors of interpretability Just as disparate applications may share common categories, disparate explanations may share common qualities that correlate to their utility. As before, we provide a set of factors that may correspond to different explanation needs. Here, we define *cognitive chunks* to be the basic units of explanation.

- *Form of cognitive chunks.* What are the basic units of the explanation? Are they raw features? Derived features that have some semantic meaning to the expert (e.g. "neurological disorder" for a collection of diseases or "chair" for a collection of pixels)? Prototypes?
- *Number of cognitive chunks.* How many cognitive chunks does the explanation contain? How does the quantity interact with the type: for example, a prototype can contain a lot more information than a feature; can we handle them in similar quantities?
- *Level of compositionality.* Are the cognitive chunks organized in a structured way? Rules, hierarchies, and other abstractions can limit what a human needs to process at one time. For example, part of an explanation may involve *defining* a new unit (a chunk) that is a function of raw units, and then providing an explanation in terms of that new unit.
- *Monotonicity and other interactions between cognitive chunks.* Does it matter if the cognitive chunks are combined in linear or nonlinear ways? In monotone ways [Gupta et al., 2016]? Are some functions more natural to humans than others [Wilson et al., 2015, Schulz et al., 2016]?
- *Uncertainty and stochasticity.* How well do people understand uncertainty measures? To what extent is stochasticity understood by humans?

Identifying methods by their characteristics will also make it easier to search for general properties of high-quality explanation that span across multiple methods, and facilitate meta-analyses that study whether these factors are associated with deeper interpretability-related universals. Ultimately, we would hope to discover that certain task-related properties benefit from explanations with certain explanation-specific properties.

6 Conclusion: Recommendations for Researchers

In this work, we have laid the groundwork for a process performing rigorous science in interpretability: defining the need; careful evaluation; and defining factors for generalization.

While there are many open questions, this framework can help ensure that our research outputs in this field are evidence-based and generalizable. Below, we summarize our recommendations.

The claim of the research should match the type of the evaluation. Just as one would be critical of a reliability-oriented paper that only cites accuracy statistics, the choice of evaluation should match the specificity of the claim being made. A contribution that is focused on a particular application should be expected to be evaluated in the context of that application (application-grounded evaluation), or on a human experiment with a closely-related task (human-grounded evaluation). A contribution that is focused on better optimizing a model class for some definition of interpretability should be expected to be evaluated with functionally-grounded metrics. As a community, we must be careful in the work on interpretability, both recognizing the need for and the costs of human-subject experiments. We should also make sure that these evaluations are on problems where there is a need for interpretability.

We should categorize our applications and methods with a common taxonomy. In section 5, we hypothesized factors that may be the factors of interpretability. Creating a shared language around such factors is essential not only to evaluation, but also for the citation and comparison of related work. For example, work on creating a safe healthcare agent might be framed as focused on the need for explanation due to unknown inputs at the local scale, evaluated at the level of an application. In contrast, work on learning sparse linear models might also be framed as focused on the need for explanation due to unknown inputs, but this time evaluated at global scale. As we share each of our work with the community, we can do each other a service by describing factors such as

1. What is the ultimate verification (or other) goal? How is the problem formulation incomplete? (Section 3)

2. At what level is the evaluation being performed? (Section 4)

3. What are the task-related and explanation-related factors in the experiments? (Section 5)

These considerations should move us away from vague claims about the interpretability of a particular model and toward classifying applications by a common set of generalizable terms.

References

Philip Adler, Casey Falk, Sorelle A Friedler, Gabriel Rybeck, Carlos Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. In *Data Mining (ICDM), 2016 IEEE 16th International Conference on*, pages 1–10. IEEE, 2016.

Hiva Allahyari and Niklas Lavesson. User-oriented assessment of classification model understandability. In *11th scandinavian conference on Artificial intelligence*. IOS Press, 2011.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Pedro Antunes, Valeria Herskovic, Sergio F Ochoa, and Jose A Pino. Structuring dimensions for collaborative systems evaluation. In *ACM Computing Surveys*. ACM, 2012.

William Bechtel and Adele Abrahamsen. Explanation: A mechanist alternative. *Studies in History and Philosophy of Science Part C: Studies in History and Philosophy of Biological and Biomedical Sciences*, 2005.

Nick Bostrom and Eliezer Yudkowsky. The ethics of artificial intelligence. *The Cambridge Handbook of Artificial Intelligence*, 2014.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006.

Adrian Bussone, Simone Stumpf, and Dympna O'Sullivan. The role of explanations on trust and reliance in clinical decision support systems. In *Healthcare Informatics (ICHI), 2015 International Conference on*, pages 160–169. IEEE, 2015.

Samuel Carton, Jennifer Helsby, Kenneth Joseph, Ayesha Mahmud, Youngsoo Park, Joe Walsh, Crystal Cody, CPT Estella Patterson, Lauren Haynes, and Rayid Ghani. Identifying police officers at risk of adverse events. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1721–1730. ACM, 2015.

Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

Jonathan Chang, Jordan L Boyd-Graber, Sean Gerrish, Chong Wang, and David M Blei. Reading tea leaves: How humans interpret topic models. In *NIPS*, 2009.

Nick Chater and Mike Oaksford. Speculations on human causal learning and reasoning. *Information sampling and adaptive cognition*, 2006.

Finale Doshi-Velez, Yaorong Ge, and Isaac Kohane. Comorbidity clusters in autism spectrum disorders: an electronic health record time-series analysis. In *Pediatrics*, volume 133:1, pages e54–e63. Am Acad Pediatrics, 2014.

Finale Doshi-Velez, Byron Wallace, and Ryan Adams. Graph-sparse lda: a topic model with structured sparsity. In *Association for the Advancement of Artificial Intelligence*, 2015.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Innovations in Theoretical Computer Science Conference*. ACM, 2012.

Tapio Elomaa. In defense of c4. 5: Notes on learning one-level decision trees. *ML-94*, 254: 62, 2017.

Alex Freitas. Comprehensible classification models: a position paper. In *ACM SIGKDD Explorations*, 2014.

Stuart Glennan. Rethinking mechanistic explanation. *Philosophy of science*, 2002.

Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *arXiv preprint arXiv:1606.08813*, 2016.

Maya Gupta, Andrew Cotter, Jan Pfeifer, Konstantin Voevodski, Kevin Canini, Alexander Mangylov, Wojciech Moczydlowski, and Alexander Van Esbroeck. Monotonic calibrated interpolated look-up tables. In *Journal of Machine Learning Research*, 2016.

Sean Hamill. CMU computer won poker battle over humans by statistically significant margin. http://www.post-gazette.com/business/tech-news/2017/01/31/CMU-computer-won-poker-battle-over-humans-by-statistically-significant-margin/stories/201701310250, 2017. Accessed: 2017-02-07.

Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *ACM Symposium on Theory of Computing*. ACM, 2010.

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, 2016.

Boris Hayete and Jadwiga R Bienkowska. Gotrees: Predicting go associations from proteins. *Biocomputing 2005*, page 127, 2004.

Carl Hempel and Paul Oppenheim. Studies in the logic of explanation. *Philosophy of science*, 1948.

Michael C Hughes, Huseyin Melih Elibol, Thomas McCoy, Roy Perlis, and Finale Doshi-Velez. Supervised topic models for clinical interpretability. In *arXiv preprint arXiv:1612.01678*, 2016.

J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, and B. Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. In *DSS*. Elsevier, 2011.

Frank Keil. Explanation and understanding. *Annu. Rev. Psychol.*, 2006.

Frank Keil, Leonid Rozenblit, and Candice Mills. What lies beneath? understanding the limits of understanding. *Thinking and seeing: Visual metacognition in adults and children*, 2004.

B. Kim, C. Rudin, and J.A. Shah. The Bayesian Case Model: A generative approach for case-based reasoning and prototype classification. In *NIPS*, 2014.

Been Kim, Caleb Chacha, and Julie Shah. Inferring robot task plans from human team meetings: A generative modeling approach with logic-based prior. *Association for the Advancement of Artificial Intelligence*, 2013.

Been Kim, Elena Glassman, Brittney Johnson, and Julie Shah. iBCM: Interactive bayesian case model empowering humans via intuitive interaction. In *MIT-CSAIL-TR-2015-010*, 2015a.

Been Kim, Julie Shah, and Finale Doshi-Velez. Mind the gap: A generative approach to interpretable feature selection and extraction. In *Advances in Neural Information Processing Systems*, 2015b.

Pieter-Jan Kindermans, Kristof T Schütt, Maximilian Alber, Klaus-Robert Müller, and Sven Dähne. Patternnet and patternlrp–improving the interpretability of neural networks. *arXiv preprint arXiv:1705.05598*, 2017.

Mykel J Kochenderfer, Jessica E Holland, and James P Chryssanthacopoulos. Next-generation airborne collision avoidance system. Technical report, Massachusetts Institute of Technology-Lincoln Laboratory Lexington United States, 2012.

Viktoriya Krakovna and Finale Doshi-Velez. Increasing the interpretability of recurrent neural networks using hidden markov models. In *arXiv preprint arXiv:1606.05320*, 2016.

Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. Too much, too little, or just right? ways explanations impact end users' mental models. In *Visual Languages and Human-Centric Computing (VL/HCC), 2013 IEEE Symposium on*, pages 3–10. IEEE, 2013.

Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1675–1684. ACM, 2016.

Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. *Research methods in human-computer interaction*. John Wiley & Sons, 2010.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. Rationalizing neural predictions. *arXiv preprint arXiv:1606.04155*, 2016.

Zachary C Lipton. The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*, 2016.

Weiwei Liu and Ivor W Tsang. Sparse perceptron decision tree for millions of dimensions. In *AAAI*, pages 1881–1887, 2016.

Tania Lombrozo. The structure and function of explanations. *Trends in cognitive sciences*, 10(10):464–470, 2006.

Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012.

Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Sæbø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118:62–69, 2012.

George A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, (2):81–97, March 1956.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. In *arXiv preprint arXiv:1312.5602*, 2013.

Ian Neath and Aimee Surprenant. *Human Memory*. Wadsworth Cengage Learning, 2003.

Clemens Otte. Safe and interpretable machine learning: A methodological review. In *Computational Intelligence in Intelligent Data Analysis*. Springer, 2013.

Parliament and Council of the European Union. General data protection regulation, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *arXiv preprint arXiv:1602.04938*, 2016.

Andrew Ross, Michael C Hughes, and Finale Doshi-Velez. Right for the right reasons: Training differentiable models by constraining their explanations. In *International Joint Conference on Artificial Intelligence*, 2017.

Salvatore Ruggieri, Dino Pedreschi, and Franco Turini. Data mining for discrimination discovery. *ACM Transactions on Knowledge Discovery from Data*, 2010.

Stefan Rüping. *Thesis: Learning interpretable models*. PhD thesis, Universitat Dortmund, 2006.

S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

Eric Schulz, Joshua Tenenbaum, David Duvenaud, Maarten Speekenbrink, and Samuel Gershman. Compositional inductive biases in function learning. In *bioRxiv*. Cold Spring Harbor Labs Journals, 2016.

D Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-François Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*, 2015.

Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. Not just a black box: Interpretable deep learning by propagating activation differences. ICML, 2016.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. In *Nature*. Nature Publishing Group, 2016.

Sameer Singh, Marco Tulio Ribeiro, and Carlos Guestrin. Programs as black-box explanations. *arXiv preprint arXiv:1611.07579*, 2016.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

Lior Jacob Strahilevitz. Privacy versus antidiscrimination. *University of Chicago Law School Working Paper*, 2008.

Girish H Subramanian, John Nosek, Sankaran P Raghunathan, and Santosh S Kanitkar. A comparison of the decision table and tree. *Communications of the ACM*, 35(1):89–94, 1992.

Adi Suissa-Peleg, Daniel Haehn, Seymour Knowles-Barley, Verena Kaynig, Thouis R Jones, Alyssa Wilson, Richard Schalek, Jeffery W Lichtman, and Hanspeter Pfister. Automatic neural reconstruction from petavoxel of electron microscopy data. In *Microscopy and Microanalysis*. Cambridge Univ Press, 2016.

Vincent Toubiana, Arvind Narayanan, Dan Boneh, Helen Nissenbaum, and Solon Barocas. Adnostic: Privacy preserving targeted advertising. 2010.

Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, 102(3):349–391, 2016.

Kush Varshney and Homa Alemzadeh. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. In *CoRR*, 2016.

Fulton Wang and Cynthia Rudin. Falling rule lists. In *Artificial Intelligence and Statistics*, pages 1013–1022, 2015.

Tong Wang, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl, and Perry MacNeille. Bayesian rule sets for interpretable classification. In *International Conference on Data Mining*, 2017.

Joseph Jay Williams, Juho Kim, Anna Rafferty, Samuel Maldonado, Krzysztof Z Gajos, Walter S Lasecki, and Neil Heffernan. Axis: Generating explanations at scale with learnersourcing and machine learning. In *ACM Conference on Learning@ Scale*. ACM, 2016.

Andrew Wilson, Christoph Dann, Chris Lucas, and Eric Xing. The human kernel. In *Advances in Neural Information Processing Systems*, 2015.