Please Stop Explaining Black Box Models for High-Stakes Decisions

Cynthia Rudin
Duke University
cynthia@cs.duke.edu

Abstract

Black box machine learning models are currently being used for high stakes decision-making throughout society, causing problems throughout healthcare, criminal justice, and in other domains. People have hoped that creating methods for explaining these black box models will alleviate some of these problems, but trying to *explain* black box models, rather than creating models that are *interpretable* in the first place, is likely to perpetuate bad practices and can potentially cause catastrophic harm to society. There is a way forward – it is to design models that are inherently interpretable.

1 Introduction

There has been an increasing trend in healthcare and criminal justice to leverage machine learning (ML) for high-stakes prediction applications that deeply impact human lives. Many of the ML models are black boxes that do not explain their predictions in a way that humans can understand. The lack of transparency and accountability of predictive models can have (and has had already) severe consequences; there have been cases of people incorrectly denied parole [Wexler, 2017], poor bail decisions leading to the release of dangerous criminals, ML-based pollution models stating that dangerous situations are safe [McGough, 2018] and generally poor use of limited valuable resources in criminal justice, medicine, energy reliability, finance, and in other domains [Varshney and Alemzadeh, 2016].

Rather than trying to create models that are inherently interpretable, there has been a recent explosion of work on "Explainable ML," where a second (posthoc) model is created to explain the first black box model. This is problematic. Explanations are often not reliable, and can be misleading, as we discuss below. If we instead use models that are inherently interpretable, they provide their own explanations, which are faithful to what the model actually computes.

In what follows, we discuss the problems with Explainable ML, followed by the challenges in Interpretable ML. This document is mainly relevant to high-stakes decision making and troubleshooting models, which are the main two reasons one might require an interpretable or explainable model. Interpretability is a domain-specific notion [Freitas, 2014, Kodratoff, 1994, Huysmans et al., 2011, Rüping, 2006], so there cannot be an all-purpose definition. Usually, however, an interpretable machine learning model is *constrained in model form* so that it is either useful to someone, or obeys structural knowledge of the domain, such as monotonicity [e.g., Gupta et al., 2016], causality, structural (generative) constraints, additivity [Lou et al., 2013], or physical constraints that come from domain knowledge. Interpretable models can perform case-based reasoning for complex domains. Often for structured data, sparsity is a useful measure of interpretability, since humans can handle at most 7 ± 2 cognitive entities at once [Miller, 1956, Cowan, 2010]. Sparse models allow a view of how variables interact *jointly* rather than individually. We will discuss several forms of interpretable machine learning models for different applications below, but there can never be a single definition;

32nd Conference on Neural Information Processing Systems (NIPS 2018), Workshop on Critiquing and Correcting Trends in Machine Learning.

e.g., in some domains, sparsity is useful, and in others is it not. A vast number of papers in the field of applied statistics are interpretable predictive models. There is a spectrum between fully transparent models (where we understand how all the variables are *jointly* related to each other) and models that are lightly constrained in model form (such as models that are forced to increase as one of the variables increases, or models that, all else being equal, prefer variables that domain experts have identified as important, see [Wang et al., 2018b]).

2 Key Issues with Explainable ML

A black box model is either a function that is too complicated for any human to comprehend, or a function that is proprietary; it is a model that is difficult to troubleshoot. Deep learning models, for instance, tend to be black boxes because they are highly recursive. An explanation is a separate model that is supposed to replicate most of the behavior of a black box (e.g., "people who have been delinquent on current credit are more likely to default on a new loan").

I am concerned that the field of comprehensibility in machine learning has strayed away from the needs of real problems. This field dates back to the early 90's at least [see Freitas, 2014, Holte, 1993], and there are a huge number of papers on interpretable ML in various fields, which do not have the word "interpretable" or "explainable" in the title, as the recent papers do.[1] Recent work on explainability (rather than interpretability) contains and perpetuates critical misconceptions that have generally gone unnoticed, but that can have a lasting negative impact on the widespread use of machine learning models in society. Let us spend some time discussing this before discussing possible solutions.

(i) It is a myth that there is necessarily a trade-off between accuracy and interpretability.

There is a widespread belief that more complex models are more accurate, meaning that a complicated black box is necessary for top predictive performance. However, this is often not true, particularly when the data naturally have a good representation. When considering problems that have structured data with meaningful features, there is often no significant difference in performance between more complex classifiers (deep neural networks, boosted decision trees, random forests) and much simpler classifiers (logistic regression, decision lists) after preprocessing. In data science problems, where structured data with meaningful features are constructed as part of the data science process, there tends to be little difference between algorithms, assuming that the data scientist follows a standard process for knowledge discovery [such as KDD, CRISP-DM, or BigData, see Fayyad et al., 1996, Chapman et al., 2000, Agrawal et al., 2012].[2]

Even for applications such as computer vision, where deep learning has major performance gains, and where interpretability is much more difficult to define, some forms of interpretability can be imbued directly into the models without losing accuracy. This will be discussed more later in the Challenges section.

Figure 1, taken from the DARPA XAI BAA, exemplifies a blind belief in the myth of the accuracy-interpretability trade-off. This not a "real" figure, in that it was not generated by any data. The axes have no quantification (there is no specific meaning to the horizontal or vertical axes). The image appears to illustrate an experiment with a static dataset, where several machine learning algorithms are applied to the same dataset. However, this kind of smooth accuracy/interpretability/explainability trade-off is atypical in data science applications with meaningful features. Even if one were to quantify the interpretability/explainability axis and aim to show that such a trade-off did exist, it is not clear what algorithms would be applied to produce this figure. (Would one actually claim it is fair to compare the 1984 decision tree algorithm CART to a 2018 deep learning model and conclude that interpretable models are not as accurate?) One can always create an artificial trade-off between accuracy and interpretability/explainability by removing parts of a more complex model to reduce accuracy, but this is not representative of the analysis one would perform on a real problem. It is also not clear why the comparison should be performed on a static dataset, because any formal process

[1] Note that there is are a huge number of papers that consider interpretability. Reading only papers that have the word "interpretable" in the title would leave the reader with only a small tip of a huge iceberg of literature on this topic.

[2] Uninterpretable algorithms can still be useful in high-stakes decisions as part of the knowledge discovery process. One can use them to obtain baseline levels of performance for instance; thank you to Tom Dietterich for pointing out that this should be mentioned.

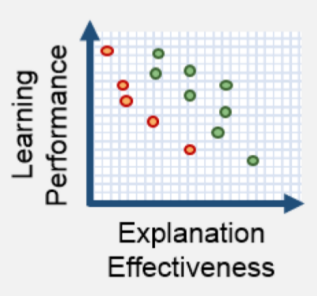

Figure 1: A fictional depiction of the “accuracy-interpretability trade-off," taken from the DARPA XAI BAA.

for defining knowledge from data [Fayyad et al., 1996, Chapman et al., 2000, Agrawal et al., 2012] would require an iterative process, where one refines the data processing after interpreting the results. Generally, in the practice of data science, the small difference between performance of machine learning algorithms can often be overwhelmed by the ability to interpret results and process the data better at the next iteration.

Efforts working within a knowledge discovery process led me to work in interpretable machine learning [Rudin et al., 2010]. This was a large-scale effort to predict electrical grid failures across New York City. The data were messy, including free text documents (trouble tickets), accounting data about electrical cables from as far back as the 1890’s, inspections data from a brand new manhole inspections program; even the structured data were not easily integrated into a database, and there were confounding issues and other problems. Algorithms on a static dataset were at most 1% different in performance, but the ability to interpret and reprocess the data led to significant improvements in performance, including correcting problems with the dataset, and revealing false assumptions about the data generation process. The most accurate predictors we found were sparse models with meaningful features that were constructed through the iterative process.

The belief that there is always a trade-off between accuracy and interpretability has led many researchers to forgo the *attempt* to produce an interpretable model. This problem is compounded by the fact that researchers are now trained in deep learning, but not in interpretable machine learning. Worse, toolkits of machine learning algorithms offer little in the way of useful interfaces for interpretable machine learning methods.

To our knowledge, all recent review and commentary articles on this topic imply (implicitly or explicitly) that the trade-off between interpretability and accuracy generally occurs. It could be possible that there are application domains where a complete black box is required for a high stakes decision. As of yet, I have not encountered such an application.

(ii) Explainable ML methods provide explanations that are not faithful to what the original model computes.

Explanations must be wrong. They cannot have perfect fidelity with respect to the original model. If the explanation was completely faithful to what the original model computes, the explanation would equal the original model, and one would not need the original model in the first place, only the explanation. (In other words, this is a case where the original model would be interpretable.) This leads to the danger that the explanation method can be an inaccurate representation of the original model in parts of the feature space.

An inaccurate (low-fidelity) explanation model limits trust in the explanation, and by extension, trust in the black box that it is trying to explain. An explainable model that has a 90% agreement with the original model indeed explains the original model most of the time. However, an explanation model that is correct 90% of the time is wrong 10% of the time. If a tenth of the explanations are incorrect, one cannot trust the explanations, and thus one cannot trust the original black box. If we cannot know for certain whether our explanation is correct, we cannot know whether to trust either the explanation or the original model.

A more important misconception about explanations stems from the terminology “explanation," which is often used in a misleading way, because explanation models do not always attempt to

Cat

Figure 2: Saliency does not explain anything except where the network is looking. We have no idea why this image is labeled as a cat when considering only saliency. Figure credit: Alexis Cook.

mimic the calculations made by the original model. Even an explanation model that performs almost identically to a black box model might use completely different features, and is thus not faithful to the computation of the black box. Consider a black box model for criminal recidivism prediction, where the goal is to predict whether someone will be arrested within a certain time after being released from jail/prison. Most recidivism prediction models depend explicitly on age and criminal history, but do not explicitly depend on race. Since criminal history and age are correlated with race in all of our datasets, a fairly accurate explanation model could construct a rule such as "This person is predicted to be arrested because they are black." This might be an accurate explanation model since it correctly mimics the predictions of the original model, but it would not be faithful to what the original model computes. This is possibly the main flaw identified by criminologists [Flores et al., 2016] in the Propublica analysis [Angwin et al., 2016, Larson et al., 2016] that accused the proprietary COMPAS recidivism model of being racially biased. Recidivism prediction will be discussed more later, as it is a key application where interpretable machine learning is necessary.

An easy fix to this problem is to change terminology. Let us stop calling approximations to black box model predictions *explanations*. For a model that does not use race, an automated explanation "This model predicts you will be arrested because you are black" is not an explanation of what the model is actually doing, and would be confusing to a judge, for instance.[3] Many of the methods that claim to produce *explanations* instead compute useful *summaries of predictions* made by the original model. Rather than producing explanations that are faithful to the original model, they show trends in how predictions are related to the features. Calling these summaries of predictions or trends rather than explanations would not be misleading, however it is possible that this terminology also would be misconstrued.[4]

(iii) Explanations often do not make sense, or are incomplete.

Even if both models are correct (the original black box is correct in its prediction and the explanation model is correct in its explanation), it is possible that the explanation leaves out so much information that it makes no sense. I will give an example from image processing, for a low-stakes decision (not a high-stakes decision where explanations are needed), but where explanation methods are often demonstrated. Saliency maps are often considered to be explanatory. Saliency maps can be useful to determine what part of the observation is irrelevant, but this leaves out all information about how relevant information *is* being used. Knowing where the network is looking within the image does not tell the user what it is doing with that part of the image, as illustrated in Figure 2. An unfortunate trend in recent work is to show only examples of explanations for *correctly* labeled observations (e.g., Figure 2 would not appear). This practice can instill a false sense of confidence in the explanation method and in the black box.

Saliency maps are only one example of where explanations are so incomplete that they might not make sense, but similar arguments can be made with other kinds of explanation methods. Sometimes it is very hard to troubleshoot a black box. For instance, only recently have researchers noticed systematic inconsistencies in random forests [Au, 2018].

[3] Thank you to Rob Holte for helping to clarify this point.

[4] Sometimes summaries (explanations) of models that are already interpretable are still useful. An example is that of Chen et al. [2018b], who work with publicly-available credit-scoring data provided by FICO. The work of Ustun et al. [2018] uses summaries to determine actionable recourse.

(iv) Black box models are often not compatible in situations where information outside the database needs to be combined with a risk assessment.

In high stakes decisions, there are often considerations outside the database that need to be combined with a risk calculation. E.g., what if the circumstances of the crime are much worse than a generic assigned charge? There are often circumstances whose knowledge could either increase or decrease someone's risk. But if the model is a black box, it is very difficult to manually calibrate how much this additional information should raise or lower the estimated risk. This issue arises constantly; for instance, the proprietary COMPAS model used in the U.S. Justice System for recidivism risk prediction does not depend on the seriousness of the current crime [Brennan et al., 2009], the judge is supposed to somehow incorporate that manually.[5]

This is a problem for any black box method, whether or not an explanation is present.

Next, we discuss reasons why black box models with separate explanation models would be preferred over inherently interpretable models, even for high-stakes decisions.

3 Key Issues with Interpretable ML

There are many cases where explanations are preferred over interpretable models, even for high-stakes decisions. However, for most applications, I am hopeful that there are ways around some of these problems, whether they are computational problems, or problems with training of researchers and availability of code. The first problem, however, is currently a major obstacle that I see no way of avoiding other than through policy, as discussed in the next section.

(i) Corporations can make profits from the intellectual property afforded to a black box.

Companies that charge for individual predictions would find their profits obliterated if an interpretable model were used instead.

Consider the COMPAS proprietary recidivism risk prediction tool discussed above that is in widespread use in the U.S. Justice System for predicting the probability that someone will be arrested after their release [Brennan et al., 2009].

The COMPAS model is equally accurate for recidivism prediction as the very simple three rule interpretable machine learning model involving only age and number of past crimes shown in Figure 3 below. However, there is no business model that would suggest profiting from the simple transparent model. The simple model in Figure 3 is an interpretable machine learning model. Even though the model looks like a rule of thumb, it is a full-blown machine learning model. A comparison of the COMPAS and CORELS models is in Table 1. Standard machine learning tools and interpretable machine learning tools seem to be approximately equally accurate for predicting recidivism, even if we define recidivism in many different ways [Zeng et al., 2017, Tollenaar and van der Heijden, 2013], based on predictions of different types of crimes using many different machine learning methods. This evidence, however, has not changed the momentum of the justice system towards proprietary models. As of this writing, California has recently eliminated its cash bail system, instead enforcing that decisions be made by algorithms; it is unclear whether COMPAS will be the algorithm used for this, despite the fact that it is not known to be any more accurate than other methods, such as the simple CORELS model.

COMPAS	CORELS
Black box	Figure 3
130+ factors	only age, priors, (optional) gender
Might include socio-economic info	no other information
Expensive (software license), within software used in US Justice System	free, transparent

Table 1: Comparison of COMPAS and CORELS models. Both models have similar true and false positive rates and true and false negative rates on data from Broward County, Florida.

[5] It is possible that many judges do not know this fact. If the model were interpretable, the judge could see directly that the seriousness of the current crime is not being considered in the risk assessment.

IF	age between 18-20 and sex is male	THEN predict arrest (within 2 years)
ELSE IF	age between 21-23 and 2-3 prior offenses	THEN predict arrest
ELSE IF	more than three priors	THEN predict arrest
ELSE	predict no arrest.	

Figure 3: This is a machine learning model from the Certifiably Optimal Rule Lists (CORELS) algorithm [Angelino et al., 2018]. This model is the minimizer of a special case of Equation 1 discussed later in the challenges section. CORELS' code is open source and publicly available at http://corels.eecs.harvard.edu/, along with the data from Florida needed to produce this model.

As a important side note, typographical errors seem to be common in computing COMPAS, which sometimes determine bail decision outcomes [Wexler, 2017, Rudin et al., 2018]. This, unfortunately, is a drawback of using black box models for recidivism prediction, and is a type of *procedural unfairness*, whereby two individuals who are identical might be randomly given different parole or bail decisions.[6]

COMPAS is not a machine learning model – it was not created by any standard machine learning algorithm. It was designed by experts based on carefully designed surveys and expertise, and it does not seem to depend heavily on past criminal history [Rudin et al., 2018]. Interestingly, if the model were not proprietary, its documentation [Brennan et al., 2009] indicates that it is actually an interpretable predictive model. Revealing this model, however, would be revealing a trade secret.

Let us switch examples to consider the proprietary machine learning model by BreezoMeter, used by Google during the California wildfires of 2018, which predicted air quality as "good – ideal air quality for outdoor activities," when air quality was dangerously bad according to multiple other models [McGough, 2018], and people reported their cars covered in ash. The Environmental Protection Agency's free, vigorously-tested air quality index would have provided a reliable result [Mannshardt and Naess, 2018]. How could BreezoMeter's machine learning method be so badly wrong and put so many in danger? We will never find out, but BreezoMeter, who has probably made a profit from making these predictions, may not have developed this new technology if its models were forced to be transparent.

In medicine, there is a trend towards blind acceptance of black box models, which will open the door for companies to sell more models to hospitals. An example of where this can go wrong is given by Zech et al. [2018], who noticed that their neural network was picking up on the word "portable" within an x-ray image, rather than the medical content of the image. If they had used an interpretable model, or even an explainable model, this issue would never have gone unnoticed. Zech et al. [2018] pointed out the issue of confounding generally; in fact, the plague of confounding haunts a vast number of datasets, and particularly medical datasets. This means that proprietary models for medicine can have serious errors.

The examples of COMPAS, Breezometer, and black box medical diagnosis all illustrate a problem with the business model for machine learning. In particular, there is a conflict of responsibility in the use of black box models for high-stakes decisions: *the companies that profit from these models are not necessarily responsible for the quality of individual predictions*. A prisoner serving an extra sentence due to a mistake entered in an overly-complicated risk score could suffer for years, whereas the company that constructed this complicated model is unaffected. On the contrary, the fact that the model was complicated and proprietary allowed the company to profit from it. In that sense, the model's designers are not incentivized to be careful in its design, performance, and ease of use. These are some of the same types of problems affecting the credit rating agencies who priced mortgages in 2008; that is, these are the same problems that contributed to the financial crisis in the U.S.

(ii) Interpretable models can entail significant effort to construct, in terms of both computation and domain expertise.

[6] In addition to the news articles on this topic, I have received multiple emails from individuals stating that these miscalculations have caused their own parole to be denied. Thank you to one of these individuals for informing me about California's recent decision to end cash bail in favor of algorithms.

As discussed above, interpretability usually translates in practice to a set of application-specific constraints on the model. Solving constrained problems is generally harder than solving unconstrained problems. Domain expertise is needed to construct the definition of interpretability for the domain, and the features for machine learning. For data that are unconfounded, complete, and clean, it is much easier to use a black box machine learning method than to troubleshoot and solve computationally hard problems. However, for high-stakes decisions, analyst time and computational time should not be as expensive relative to the quality of predictions than for other applications – it is worthwhile to put the extra effort and cost into constructing a high-quality model. But even so, many organizations do not have analysts who have the training or expertise to construct interpretable models at all.

Some companies have started to provide interpretable ML solutions using proprietary software. While this is a step in the right direction, it is not clear that the proprietary software is better than publicly available software. [7]

As discussed earlier, interpretability constraints (like sparsity) lead to optimization problems that have been proven to be computationally hard in the worst case. The theoretical hardness of these problems does not mean we cannot solve them in real cases, but these optimization problems are often difficult to solve. Major improvements have been made in the last decade [e.g., see Carrizosa et al., 2010, Su et al.], and some are discussed later in the challenges section. Explanation methods, on the other hand, are usually based on derivatives, which lead to easier gradient-based optimization.

Other reasons for preferring black boxes include the belief that they "uncover" hidden patterns in the data. However, it is not clear that a transparent model would not also uncover these same patterns, but in a transparent way.

4 A Concrete Proposal for Responsible ML Policy

Currently the Generalized Data Protection Regulation and other AI regulation plans govern "right to an explanation," where only an explanation is required, not an interpretable model [Goodman and Flaxman, 2016]. It is not clear whether the explanation is required to be accurate, complete, or faithful to the underlying model [Wachter et al., 2017].

Let us consider a concrete proposal moving forward. Consider a mandate that, for certain high-stakes decisions, *no black box should be deployed when there exists an interpretable model with the same level of performance*. Companies that produce and sell black box models could then be held accountable if an equally accurate transparent model exists. It would be considered a form of false advertising to sell a black box model if there is an equally-accurate interpretable model. The onus will then fall on companies to produce black box models only when no transparent model exists for the same task.

This suggests a change in the business model for machine learning. Opacity is viewed as essential in protecting intellectual property, but it is at odds with the requirements of many domains that involve public health or welfare. However, the combination of opacity and explainability is not the only way to incentivize machine learning experts to invest in creating such systems. Compensation for developing an interpretable model could be provided in a lump sum, and the model could be released to the public.[8] The creator of the model would not be able to profit from licensing the model over a period of time, but the fact that the models are useful for public good applications would make these problems appeal to academics and charitable foundations.

This proposal will not solve all problems, but it could at least rule out companies selling recidivism prediction models, possibly credit scoring models, and other kinds of models where we can construct accurate interpretable models. If applied too broadly, it could reduce industrial participation in cases where it might benefit society.

As mentioned earlier, I have not yet found a high-stakes application where a fully black box model is necessary, despite having worked on many applications. As long as we continue to allow for a broad definition of interpretability that is adapted to the domain, we should be able to improve decision

[7] For instance, claims made by some companies about performance of their proprietary algorithms are not impressive (e.g., Interpretable AI), and have not been compared with appropriate baselines. Some companies (e.g., Diveplane) do not make performance claims and provide no information about their products.

[8] Thank you to Ron Parr for a clear way of stating this argument.

making for serious tasks of societal importance. However, in order for people to design interpretable models, the technology must exist to do so. As discussed earlier, there is a formidable computational hurdle in designing interpretable models, even for standard structured data with already-meaningful features.

5 Algorithmic Challenges in Interpretable ML

What if every black box machine learning model could be replaced with one that was equally accurate but also interpretable? If we could do this, we would identify flaws in our models and data that we could not see before. Perhaps we could prevent some of the poor decisions in criminal justice and medicine that are caused by problems with using black box models. We could also eliminate the need for explanations that are misleading and often wrong.

Since interpretability is domain-specific, a large toolbox of possible techniques can come in handy. Below we expand on three of the challenges for interpretable machine learning that appear often. All three cases have something in common: people have been providing interpretable predictive models for these problems for decades, and the human-designed models look just like the type of model we want to create with machine learning. I also discuss some of our current work on these well-known problems.

By no means is this set of challenges close to encompassing the large number of domain-specific challenges that exist in creating interpretable models.

Challenge #1: Constructing optimal logical models

A logical model consists of statements involving "or," "and," "if-then," etc. The CORELS model in Figure 3 is a logical model, called a *rule list*. Decision trees are logical models, as well as conjunctions of disjunctions ("or's" of "and's" – for instance, IF condition A is true OR conditions B AND C are true, THEN predict yes, otherwise predict no).

Logical models have been crafted by hand as *expert systems* as far back as the 1970's. Since then, there have been many heuristics for creating logical models; for instance, one might add logical conditions one by one (greedily), and then prune conditions away that are not helpful (again, greedily). These heuristic methods tend to be either inaccurate or uninterpretable because they do not choose a globally best choice (or approximately best choice) for the logical conditions [C4.5 and CART decision trees are an example, Quinlan, 1993, Breiman et al., 1984, as well as a vast number of models from the associative classification literature]. An issue with algorithms that do not aim for optimal (or near-optimal) solutions to optimization problems is that it becomes difficult to tell whether poor performance is due to the choice of algorithm or the combination of the choice of model class and constraints (did the algorithm perform poorly because it didn't optimize its objective, or because we chose constraints that do not allow enough flexibility in the model to fit the data well?). The question of computing optimal logical models has existed since at least the mid 1990's [Auer et al., 1995].

We would like models that look like they are created by hand, but they need to be accurate, full-blown machine learning models. To this end, let us consider the following optimization problem, which asks us to find a model that minimizes a combination of the fraction of misclassified training points and the size of the model. Training observations are indexed from $i = 1, .., n$, and $\mathcal{F}$ is a family of logical models such as decision trees. The optimization problem is:

$$\min_{f \in \mathcal{F}} \left(\frac{1}{n} \sum_{i=1}^{n} 1_{[\text{training observation}_i \text{ is misclassified by } f]} + \lambda \times \text{size}(f) \right). \quad (1)$$

Here, the size of the model can be measured by the number of logical conditions in the model, such as the number of leaves in a decision tree. The parameter λ is the classification error one would sacrifice in order to have one fewer term in the model; if λ is 0.01, it means we would sacrifice 1% training accuracy in order to reduce the size of the model by one. Another way to say this is that the model would contain an additional term only if this additional term reduced the error by at least 1%.

The optimization problem in (1) is generally known to be computationally hard. Versions of this optimization problem are some of the fundamental problems of artificial intelligence. The challenge is whether we can solve (or approximately solve) problems like this in practical ways, by leveraging new theoretical techniques and advances in hardware.

1.	Prior Arrests $\geq$ 2	1 point	$\cdots$
2.	Prior Arrests $\geq$ 5	1 point	$+ \cdots$
3.	Prior Arrests for Local Ordinance	1 point	$+ \cdots$
4.	Age at Release between 18 to 24	1 point	$+ \cdots$
5.	Age at Release $\geq$ 40	-1 points	$+ \cdots$
		SCORE	$= \cdots$

SCORE	-1	0	1	2	3	4
RISK	11.9%	26.9%	50.0%	73.1%	88.1%	95.3%

Figure 4: Scoring system for risk of recidivism from Rudin and Ustun [2018] [which grew out of Ustun and Rudin, 2017, Zeng et al., 2017, Ustun and Rudin, 2015]. This is not a rule of thumb; the selection of numbers and features come from machine learning.

The model in Figure 3 is a machine learning model that comes from the CORELS algorithm [Angelino et al., 2018]. CORELS solves a special case of (1), for the special choice of $\mathcal{F}$ as the set of rule lists, and where the size of the model is measured by the number of rules in the list. Figure 3 has three "if-then" rules so its size is 3. In order to minimize (1), CORELS needs to avoid enumerating all possible models, because this would take an extremely long time (perhaps until the end of the universe on a modern laptop for a fairly small dataset). The technology underlying the CORELS algorithm was able to solve the optimization problem to optimality in under a minute for the Broward County, FL, dataset discussed above. CORELS' backbone is: (i) a set of theorems allowing massive reductions in the search space of rule lists, (ii) a custom fast bit-vector library that allows fast exploration of the search space, so that CORELS does not need to enumerate all rule lists, and (iii) specialized data structures that keep track of intermediate computations and symmetries. This set of ingredients proved to be a powerful cocktail for handling these tough computational problems.

The example of CORELS enforces two points discussed above, which are, first, that interpretable models entail hard computational problems, and second, that these computational problems can be solved by leveraging a combination of theoretical and systems-level techniques. CORELS creates one type of logical model. However, there are many more. Formally, *the first challenge is to create algorithms that solve logical modeling problems in a reasonable amount of time, for practical datasets.*

We have been extending CORELS to more complex problems, such as Falling Rule Lists [Wang and Rudin, 2015, Chen and Rudin, 2018], and optimal binary-split decision trees, but there is much work to be done on other types of logical models, with various kinds of constraints.

Challenge #2: Construct optimal sparse scoring systems

Scoring systems have been designed by hand since at least the Burgess criminological model of 1928 [Burgess, 1928]. This model was designed to predict whether a criminal would violate bail, where individuals received points for being a "ne'er do well" or a "recent immigrant" that increased their predicted probability of parole violation. (Of course, this model was not created using machine learning.) A scoring system is a sparse linear model with integer coefficients – the coefficients are the point scores. An example of a scoring system for criminal recidivism is shown in Figure 4, which predicts whether someone will be arrested within 3 years of release. Scoring systems are used pervasively throughout medicine; there are hundreds of scoring systems developed by physicians. Again, the challenge is whether scoring systems – which look like they could have been produced by a human – can be produced by a machine learning algorithm, and be as accurate as any other model from any other machine learning algorithm.

There are several ways to formulate the problem of producing a scoring system. For instance, we could use a special case of (1), where the model size is the number of terms in the model. (Figure 4 is a machine learning model with 5 terms.) Sometimes, one can round the coefficients of a logistic regression model to produce a scoring system, but that does not tend to give accurate models, and does not tend to produce models that have particularly nice coefficients (such as 1 and -1 used in

Figure 4). However, solving (1) or its variants is computationally hard, because the domain over which we solve the optimization problem is the integer lattice.[9]

The model in Figure 4 is from the solution to a very hard optimization problem. Let us discuss this optimization problem briefly. The goal is to find the coefficients b_j, $j = 1...p$ for the linear predictive model $f(\mathbf{z}) = \sum_j b_j z_j$ where z_j is the jth covariate of a test observation $\mathbf{z}$. In Figure 1, the b_j's are the point scores, which turned out to be 1, -1, and 0 as a result of optimization, where only the nonzero coefficients are displayed in the figure. In particular, we want to solve:

$$\min_{b_1,b_2,..,b_p \in \{-10,-9,...,9,10\}} \frac{1}{n} \sum_{i=1}^{n} \log \left(1 + \exp\left(-\sum_{j=1}^{p} b_j x_{i,j}\right)\right) + \lambda \sum_j 1_{[b_j \neq 0]},$$

where the point scores b_j are constrained to be integers between -10 and 10, the training observations are indexed by $i = 1, ..., n$, and p is the total number of covariates for our data. Here the model size is the number of non-zero coefficients, and again λ is the trade-off parameter. The first term is the logistic loss used in logistic regression. The problem is a hard problem, specifically a mixed-integer-nonlinear program (MINLP) whose domain is also the integer lattice.

Despite the hardness of this problem, new cutting plane algorithms have been able to solve this problem to optimality (or near-optimality) for arbitrarily large sample sizes and a moderate number of variables within a few minutes. The latest attempt at solving this problem is the RiskSLIM (Risk-Supersparse-Linear-Integer-Models) algorithm, which is a specialized cutting plane method that adds cutting planes only whenever the solution to a linear program is integer-valued, and otherwise performs branching. [Ustun and Rudin, 2017].

This optimization problem is similar to what physicians attempt to solve manually, but without writing the optimization problem out like we did above. Because physicians do not use optimization tools to do this, accurate scoring systems tend to be difficult for physicians to create themselves. One of our collaborators spent months trying to construct a scoring system himself by adding and removing variables, rounding, and using other heuristics to decide which variables to add, remove, and round. RiskSLIM was useful for helping him with this task [Ustun et al., 2017]. Formally, *the second challenge is to create algorithms for scoring systems that are computationally efficient.* Ideally we would increase the size of the optimal scoring system problems that current methods can practically handle by an order of magnitude.

Challenge #3 Define interpretability for specific domains and create methods accordingly, including computer vision

Since interpretability needs to be defined in a domain-specific way, some of the most important technical challenges for the future are tied to specific important domains. Let us start with computer vision, for classification of natural images. There is a vast and growing body of research on posthoc explainability of deep neural networks, but not as much work in designing *interpretable neural networks*. My goal in this section is to demonstrate that even for classic domains of machine learning, where latent representations of data need to be constructed, there could exist more interpretable models that are as accurate as black box models.

For computer vision in particular, there is not a clear definition of interpretability, and the sparsity-related models discussed above do not apply – sparsity in pixel space does not make sense. There can be many different ideas of what constitutes interpretability, even between different computer vision applications. However, if we can define interpretability somehow for our particular application, we can embed this definition into our algorithm.

Again, let us define what constitutes interpretability for any domain by considering *how people explain to each other* the reasoning processes behind complicated visual classification tasks. As it turns out, for classification of natural images, domain experts often direct our attention to different parts of the image and explain why these parts of the image were important in their reasoning process. The question is whether we can construct network architectures for deep learning that can also do this. The network must then make decisions by reasoning about these parts of the image so that the explanations are real, and not posthoc.

[9] To see this, consider an axis for each of $\{b_1, b_2, ..., b_p\}$, where each b_j can take on integer values. This is a lattice that defines the feasible region of the optimization problem.

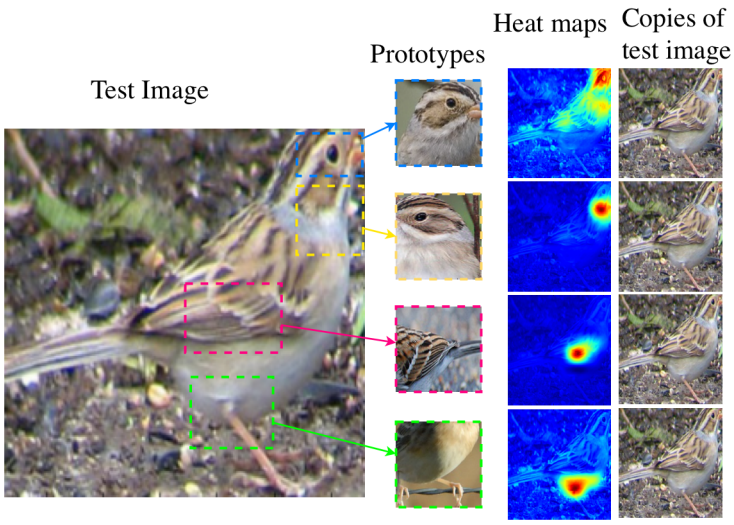

Figure 5: Image from Chen et al. [2018a], indicating that parts of the image are similar to prototypical parts of training examples. The test image to be classified is on the left, the most similar prototypes are in the middle column, and the heatmaps that show which part of the test image is similar to the prototype are on the right. We included copies of the test image on the right so that it is easier to see what part of the bird the heatmaps are referring to. The similarities of the prototypes to the test image are what determine the predicted class label of the image. Here, the image is predicted to be a clay-colored sparrow. The top prototype seems to be comparing the bird's face to a prototypical face of a clay-colored sparrow, the second prototype considers the chest of the bird, the third looks at tail feathers, and the last seems to consider the abdomen and leg.

In a recent attempt to do this, Li et al. [2018], Chen et al. [2018a] have been building architectures that append a special prototype layer to the end of the network. During training, the prototype layer finds parts of training images that act as prototypical parts of images. For instance, for bird classification, it might pick out a prototypical head of a blue jay, prototypical feathers of a blue jay, etc. The network also learns a similarity metric between parts of images. Thus, during testing, when a new image needs to be evaluated, the network finds parts of the test image that are similar to the prototypical parts it learned during training, as shown in Figure 5. The final class prediction of the network is based on the weighted sum of similarities to the prototypes; this is the sum of evidence throughout the image for a particular class. The explanations given by the network are the prototypes (and the weighted similarities to them).

Training this prototype network is not as easy as training an ordinary neural network; the tricks that have been developed for regular deep learning have not yet been developed for the prototype network. However, so far these prototype networks have been trained to be as accurate as the original black box deep neural networks before the prototype layer was added.

Discussion on Interpretability for Specific Domains

Let us finish this short discussion on challenges to interpretability for specific domains by mentioning that there are vast numbers of papers that have imbued interpretability in their methodology. Interpretability is not mentioned in the title of these papers, and often not in the body of the text. This is why it is almost impossible to create a review article on interpretability in machine learning or statistics without missing the overwhelming majority of it.[10] For instance, Gallagher et al. [2017] analyze brain-wide electrical spatiotemporal dynamics to understand depression vulnerability and find interpretable patterns in a low dimensional space. Dimension reduction to interpretable dimensions

[10] As a side point, it is not clear why reviews of interpretability and explainability make sense. We do not normally have reviews of performance/accuracy measures, despite the fact that there are many of them – accuracy, area under the ROC curve, partial AUC, sensitivity, specificity, discounted cumulative gain, F-score, G-means, and many other domain-specific measures. Interpretability is just as domain-specific as accuracy performance, so it is not clear why reviews of interpretability make any more sense than reviews of accuracy/performance.

is an important theme in interpretable machine learning. Problems residing in *applied statistics* are often interpretable because they embed the physics of the domain; e.g., Wang et al. [2018a] create models for recovery curves for prostatectomy patients whose signal and uncertainty obey specific constraints in order to be realistic. Constraints on the uncertainty of the predictions make these models interpretable.

A Technical Reason Why Accurate Interpretable Models Might Exist in Many Domains

Why is it that accurate interpretable models could possibly exist in so many different domains? Is it really possible that many aspects of nature have simple truths that are waiting to be discovered by machine learning? Although that would be intriguing, I will not make this kind of Occham's-Razor-style argument, in favor of a technical argument about function classes, and in particular, Rashomon Sets. The argument below is flushed out more formally by Semenova et al. [2018].

Here is the *Rashomon set* argument: Consider that the data permits a large set of reasonably accurate predictive models to exist. Because this set of accurate models is large, it often contains at least one model that is interpretable. This model is both interpretable and accurate.

Unpacking this argument slightly, for a given data set, we define the *Rashomon set* as the set of reasonably accurate predictive models (say within 5% of the best model accuracy of boosted decision trees). Because the data are finite, the data could admit many close-to-optimal models that predict differently from each other: a large Rashomon set. I suspect this happens often in practice because sometimes many different machine learning algorithms perform similarly on the same dataset, despite having different functional forms (e.g., random forests, neural networks). As long as the Rashomon set contains a large enough set of models with diverse predictions, it probably contains functions that can be approximated well by simpler functions, and so the Rashomon set can also contain these simpler functions. Said another way, finiteness of the data leads to a Rashomon set, a larger Rashomon set probably contains interpretable models, thus interpretable accurate models often exist.

If this theory holds, we should expect to see interpretable models exist across domains. These interpretable models may be hard to find through optimization, but at least there is a reason we might expect that such models exist.

6 Conclusion

If this whitepaper can shift the focus even slightly from the basic assumption underlying most work in Explainable ML – which is that a black box is necessary for accurate predictions – we will have considered this document a success.

If this document can encourage policy makers not to accept black box models without significant attempts at interpretable (rather than explainable) models, that would be even better.

If we can make people aware of the current challenges right now in interpretable machine learning, it will allow policy-makers the mechanism to demand that more effort should be made in ensuring safety and trust in our machine learning models for high-stakes decisions.

If we do not succeed at these efforts, it is possible that black box models will continue to be permitted when it is not safe to use them. Since the definition of what constitutes a viable explanation is unclear, even strong regulations such as "right to explanation" can be undermined with less-than-satisfactory explanations. Further, there will continue to be problems calibrating black box model predictions with outside information, and continued miscalculations of black box model inputs. This may continue to lead to poor decisions throughout our criminal justice system, incorrect safety guidance for air quality disasters, inexplicable loan decisions, and other widespread societal problems.

Acknowledgments

I would like to thank Fulton Wang, Chaofan Chen, Oscar Li, Tom Dietterich, Margo Seltzer, Elaine Angelino, Nicholas Larus-Stone, Elizabeth Mannshart, Maya Gupta, and several others who helped my thought processes in various ways, and particularly Berk Ustun, Ron Parr, and my father, Stephen Rudin, who went to considerable efforts to provide thoughtful comments and discussion.

References

D. Agrawal, P. Bernstein, E. Bertino, S. Davidson, U. Dayal, M. Franklin, ..., and J. Widom. Challenges and opportunities with big data: A white paper prepared for the computing community consortium committee of the computing research association. Technical report, 2012. URL `http://cra.org/ccc/resources/ccc-led-whitepapers/`.

Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Certifiably optimal rule lists for categorical data. *Journal of Machine Learning Research*, 19:1–79, 2018.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, 2016.

Timothy C. Au. Random forests, decision trees, and categorical predictors: The "absent levels" problem. *Journal of Machine Learning Research*, 19:1–30, 2018.

Peter Auer, Robert C. Holte, and Wolfgang Maass. Theory and applications of agnostic pac-learning with small decision trees. In *Machine Learning Proceedings 1995*, pages 21 – 29. Morgan Kaufmann, San Francisco (CA), 1995.

Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

Tim Brennan, William Dieterich, and Beate Ehret. Evaluating the predictive validity of the COMPAS risk and needs assessment system. *Criminal Justice and Behavior*, 36(1):21–40, January 2009.

Ernest W Burgess. Factors determining success or failure on parole. Illinois Committee on Indeterminate-Sentence Law and Parole Springfield, IL, 1928.

Emilio Carrizosa, Belen Martín-Barragán, and Dolores Romero Morales. Binarized support vector machines. *INFORMS Journal on Computing*, 22(1):154–167, 2010.

Pete Chapman, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rüdiger Wirth. CRISP-DM 1.0 - step-by-step data mining guide. Technical report, SPSS, 2000.

Chaofan Chen and Cynthia Rudin. An optimization approach to learning falling rule lists. In *Artificial Intelligence and Statistics (AISTATS)*, 2018.

Chaofan Chen, Oscar Li, Chaofan Tao, Alina Barnett, Jonathan Su, and Cynthia Rudin. *This* looks like *that*: Deep learning for interpretable image recognition. *ArXiv e-prints*, June 2018a.

Chaofan Chen, Kancheng Lin, Cynthia Rudin, Yaron Shaposhnik, Sijia Wang, and Tong Wang. An interpretable model with globally consistent explanations for credit risk. In *NIPS 2018 Workshop on Challenges and Opportunities for AI in Financial Services: the Impact of Fairness, Explainability, Accuracy, and Privacy*, 2018b.

Nelson Cowan. The magical mystery four how is working memory capacity limited, and why? *Current directions in psychological science*, 19(1):51–57, 2010.

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

Anthony W. Flores, Christopher T. Lowenkamp, and Kristin Bechtel. False positives, false negatives, and false analyses: A rejoinder to "Machine bias: There's software used across the country to predict future criminals". *Federal probation*, 80(2), September 2016.

Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations Newsletter*, 15(1):1–10, March 2014.

Neil Gallagher, Kyle R Ulrich, Austin Talbot, Kafui Dzirasa, Lawrence Carin, and David E Carlson. Cross-spectral factor analysis. In *Advances in Neural Information Processing Systems 30*, pages 6842–6852. Curran Associates, Inc., 2017.

Bryce Goodman and Seth Flaxman. EU regulations on algorithmic decision-making and a "right to explanation". *arXiv:1606.08813 [stat.ML]*, 2016. Presented at 2016 ICML Workshop on Human Interpretability in Machine Learning (WHI 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. *Journal of Machine Learning Research*, 17(109):1–47, 2016.

Robert C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11(1):63–91, 1993.

Johan Huysmans, Karel Dejaeger, Christophe Mues, Jan Vanthienen, and Bart Baesens. An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models. *Decision Support Systems*, 51(1):141–154, 2011.

Yves Kodratoff. The comprehensibility manifesto. *KDD Nugget Newsletter*, 94(9), 1994.

Jeff Larson, Surya Mattu, Lauren Kirchner, and Julia Angwin. How we analyzed the COMPAS recidivism algorithm. Technical report, ProPublica, 2016.

Oscar Li, Hao Liu, Chaofan Chen, and Cynthia Rudin. Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In *Proceedings of AAAI*, 2018.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Knowledge Discovery in Databases (KDD)*, August 2013.

Elizabeth Mannshardt and Liz Naess. Air quality in the USA. *Significance*, October 2018.

Michael McGough. How bad is sacramento's air, exactly? google results appear at odds with reality, some say. *Sacramento Bee*, 2018. URL `https://www.sacbee.com/news/state/california/fires/article216227775.html`.

George Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97, 1956.

John Ross Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

Cynthia Rudin and Berk Ustun. Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice. *Interfaces*, 48:399–486, 2018. Special Issue: 2017 Daniel H. Wagner Prize for Excellence in Operations Research Practice September-October 2018.

Cynthia Rudin, Rebecca Passonneau, Axinia Radeva, Haimonti Dutta, Steve Ierome, and Delfina Isaac. A process for predicting manhole events in Manhattan. *Machine Learning*, 80:1–31, 2010.

Cynthia Rudin, Caroline Wang, and Beau Coker. The age of secrecy and unfairness in recidivism prediction. *ArXiv e-prints*, November 2018.

Stefan Rüping. *Learning interpretable models*. PhD thesis, Universität Dortmund, 2006.

Lesia Semenova, Ron Parr, and Cynthia Rudin. A study in Rashomon curves and volumes: A new perspective on generalization and model simplicity in machine learning. In progress, 2018.

Guolong Su, Dennis Wei, Kush R. Varshney, and Dmitriy M. Malioutov. Interpretable two-level boolean rule learning for classification.

Nikolaj Tollenaar and P.G.M. van der Heijden. Which method predicts recidivism best?: a comparison of statistical, machine learning and data mining predictive models. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 176(2):565–584, 2013.

Berk Ustun and Cynthia Rudin. Supersparse linear integer models for optimized medical scoring systems. *Machine Learning*, pages 1–43, 2015. ISSN 0885-6125. doi: 10.1007/s10994-015-5528-6. URL `http://dx.doi.org/10.1007/s10994-015-5528-6`.

Berk Ustun and Cynthia Rudin. Optimized risk scores. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

Berk Ustun, Lenard A Adler, Cynthia Rudin, Stephen V Faraone, Thomas J Spencer, Patricia Berglund, Michael J Gruber, and Ronald C Kessler. The World Health Organization Adult Attention-Deficit/Hyperactivity Disorder Self-Report Screening Scale for DSM-5. *JAMA Psychiatry*, 74(5):520–526, 2017.

Berk Ustun, Alexander Spangher, and Yang Liu. Actionable Recourse in Linear Classification. *ArXiv e-prints 1809.06514*, September 2018.

Kush R. Varshney and Homa Alemzadeh. On the safety of machine learning: Cyber-physical systems, decision sciences, and data products. *Big data*, 5, 10 2016. doi: 10.1089/big.2016.0051.

Sandra Wachter, Brent Mittelstadt, and Luciano Floridi. Why a right to explanation of automated decision-making does not exist in the general data protection regulation. *International Data Privacy Law*, 12 2017.

Fulton Wang and Cynthia Rudin. Falling rule lists. In *Proceedings of Artificial Intelligence and Statistics (AISTATS)*, 2015.

Fulton Wang, Cynthia Rudin, Tyler H Mccormick, and John L Gore. Modeling recovery curves with application to prostatectomy. *Biostatistics*, page kxy002, 2018a. doi: 10.1093/biostatistics/kxy002. URL `http://dx.doi.org/10.1093/biostatistics/kxy002`.

Jiaxuan Wang, Jeeheh Oh, Haozhu Wang, and Jenna Wiens. Learning credible models. In *Knowledge Discovery in Databases (KDD)*, 2018b.

Rebecca Wexler. When a computer program keeps you in jail: How computers are harming criminal justice. *New York Times*, June 2017.

John R. Zech, Marcus A Badgeley, Manway Liu, Anthony B. Costa, Joseph J. Titano, and Eric K. Oermann. Confounding variables can degrade generalization performance of radiological deep models. *arXiv:1807.00431 [computer vision and pattern recognition]*, 2018.

Jiaming Zeng, Berk Ustun, and Cynthia Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 180(3):689–722, 2017.

A Algorithm Stability

A common criticism of decision trees is that they are not stable, meaning that small changes in the training data lead to completely different trees, giving no guidance as to which tree to choose.[11] In fact, the same problem can happen in linear models when there are highly correlated features. The lack of stability happens with most algorithms that are not strongly regularized.

I believe this instability in the learning algorithm is a side-effect of the Rashomon effect mentioned earlier – that there are many different almost-equally good predictive models. Adding regularization to an algorithm increases stability, but also limits flexibility of the user to choose which element of the Rashomon set would be more desirable.

For applications where the models are purely predictive and not causal (e.g., in criminal recidivism where we use age and prior criminal history to predict future crime), there is no assumption that the model represents how outcomes are actually generated. The importance of the variables in the model does not reflect a causal relationship between the variables and the outcomes. Thus, without additional guidance from the domain expert, there is no way to proceed further to choose a single "best model." As discussed above, regularization can act as this additional input.

I view the lack of algorithmic stability as an advantage rather than a disadvantage. If the lack of stability is indeed caused by a large Rashomon effect, it means that domain experts can add more constraints to the model to customize it.

[11] Thank you to Rob Holte for mentioning stability.