INTERPRETABLE APPROACHES TO OPENING UP BLACK-BOX MODELS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Hui Fen (Sarah) Tan

August 2019

INTERPRETABLE APPROACHES TO
OPENING UP BLACK-BOX MODELS

Hui Fen (Sarah) Tan, Ph.D.
Cornell University 2019

In critical domains such as healthcare, finance, and criminal justice, merely knowing what was predicted, and not why, may be insufficient to deploy a machine learning model. This dissertation proposes new methods to open up black-box models, with the goal of helping creators, as well as users, of machine learning models increase their trust and understanding of the models.

The first part of this dissertation proposes new post-hoc, global explanations for black-box models, developed using model-agnostic distillation techniques or by leveraging known structure specific to the black-box model. First, we propose a distillation approach to learn global additive explanations that describe the relationship between input features and model predictions, showing that distilled additive explanations have fidelity, accuracy, and interpretability advantages over non-additive explanations, via a user study with expert users. Second, we work specifically on tree ensembles, leveraging tree structure to construct a similarity metric for gradient boosted tree models. We use this similarity metric to select prototypical observations in each class, presenting an alternative to other tree ensemble interpretability methods such as seeking one tree that best represents the ensemble or feature importance methods.

The second part of this dissertation studies the use of interpretability approaches to probe and debug black-box models in algorithmic fairness settings. Here, black-box takes on another meaning, with many risk-scoring models for

high stakes decision such as credit scoring and judicial bail being proprietary and opaque, not lending themselves to easy inspection or validation. We propose Distill-and-Compare, an approach to probe such risk scoring models by leveraging additional information on ground-truth outcomes that the risk scoring model was intended to predict. We find that interpretability approaches can help uncover previously unknown sources of bias. Finally, we provide a concrete case study using the interpretability methods proposed in this dissertation to debug black-box models, in this case, a hybrid Human + Machine recidivism prediction model. Our methods revealed that human and COMPAS decision making anchored on the same features, and hence did not differ significantly enough to harness the promise of hybrid Human + Machine decision making, concluding this dissertation on interpretability approaches for real-world settings.

BIOGRAPHICAL SKETCH

Hui Fen Tan was born in Kuala Lumpur, Malaysia. She grew up there and came to the US in 2006 for college, where she adopted the name Sarah. After four wonderful years at the University of California, Berkeley where the first statistical model she trained was a decision tree, she moved to New York City (NYC), studying for a Masters in Statistics at Columbia University between 2010 and 2012. After that, she worked as a data scientist in public policy and healthcare in NYC before deciding to return to graduate school. She pursued a PhD in Statistics at Cornell University from 2013 to 2019, splitting her time between Cornell and the University of California, San Francisco (UCSF) from 2018 onwards. During her time in the PhD program, she worked on interpretability, fairness, causal inference, and healthcare applications. She continues to be interested in aspects of machine learning that affect people's lives.

Dedicated to my family and friends.

ACKNOWLEDGEMENTS

I cannot thank enough my advisors and committee members: Giles Hooker, Martin Wells, Rich Caruana, and Thorsten Joachims. Giles, you were endlessly patient with me. You are the kindest, most supportive advisor one could ask for. Marty, just a simple chat with you made everything better. Rich, I have learned so much about machine learning from you. Thorsten, I look up to you.

I am grateful for the opportunities Cornell has given me to work on different projects over the course of this PhD; thank you so much to all my collaborators: Sophia Day, Kevin Konty, Jina Suh, Stathis Gennatas, Gilmer Valdes, Romain Pirracchio, Julius Adebayo, Kori Inkpen, Ece Kamar, Xuezhou Zhang, Yin Lou, Paul Koch, Ursula Chajewska, James Stark, Susanna Makela, Daliah Heller, Sharon Balter, Tian Zheng, Matvey Soloviev, David Miller, James Savage, Skyler Seto, Ion Bogdan Vasi, Edward Walker, John S. Johnson, and Albert Gordo.

Thank you Charles McCulloch for hosting me at UCSF for the last leg of my PhD. Thank you also to John Kornak, Kate Rankin, Dima Lituiev, and Susan Rubin for making me feel welcome. During the course of this PhD, I was lucky enough to spend two summers at Microsoft Research, Redmond. I learned so much from everyone there. Thank you Rich, Jina, Kori, Ece, Besmira Nushi, Chris Meek, Debadeepta Dey, and Eric Horvitz. I also greatly enjoyed my summers at Xerox Research in Grenoble, France and Data Science for Social Good in Chicago. Thank you Chris Dance, Tomi Silander, and Rayid Ghani.

I benefited from the friendly and supportive environment at Cornell. I thank Thomas DiCiccio for easy conversation and being a great professor to TA for. I learned a lot in classes taught by Kilian Weinberger, David Mimno, Jacob Bien, and Tracey Brandenburg. For Diana Drake, Beatrix Johnson, Laura Burrows,

Donna Bunce, and Phillip Rusher's administrative magic, I owe a debt of gratitude. I enjoyed the Steinway in Sage Chapel and spent many caffeinated hours at Gimme Coffee.

Of the many wonderful teachers I had at Berkeley and Columbia, David Purdy, Frank Wood, Tian Zheng, and Faiza Bellounis, in their own different ways, gave me the confidence to pursue a PhD. Ronald Low at NYC Public Hospitals showed me how to wrangle medical data.

I could not thank each and everyone of my friends who supported me throughout this long PhD journey. I will attempt a non-exhaustive list, and offer my sincerest apologies to those not mentioned by name.

To all my friends who walked this PhD path with me, thank you. I was closest to PhD students in Statistics and Computer Science who started at Cornell in 2013. A shoutout to Gamma Alpha co-op mates, jamming partners, SPIC-MACAY, and Stewart House. Thank you, Khai Zhi Sim, Jun Le Goh, Arzoo Katiyar, Chaitali Joshi, Rahmtin Rotabi, Matvey Soloviev, Erin Green, Martin Ian Malgapo, Shaun Sim, Barbara Oh, Adith Swaminathan, Chern Wei Bee, Wei Min Chan, Maithra Raghu, and Tobias Schnabel. Sophie Dramé-Maigné, you defy categorization.

To all my friends before this PhD life who thought I was crazy to go back to school, thank you. You are partly right. But now I can say that it was worth it. You hosted me during my escapes from Ithaca and reminded me of life outside of PhD. Thank you, Evelyn Yung, Alyssa Li, Kaiting Zhou, Karen Lu, Ling Tan, Raghu Sudhakara, Eric Wu, and Raymond Lim.

To my summer friends who became lifelong friends, thank you. You made our short time together memorable, and I continue to learn from you. Thank you, Sabina Tomkins, Saumya Jetley, Adji Dieng, Himabindu Lakkaraju. To my

Data Science for Social Good compatriots, your efforts continue to inspire me.

To my friends in the Bay Area, German Ros and Sören Künzel, thank you. You made it less lonely. Thank you for your kindness, Bin Yu, Stefan Wager, and Sam Pimentel, in letting me attend your reading groups while in the Bay Area.

A shoutout to my fellow Women in Machine Learning (WiML) Board members and 2016 WiML Workshop organizers. We worked hard!

Last but not least, thank you to my family: my parents, brother, husband, cousin Steph and John, and a certain orange pomeranian who kept me company during the early years of my PhD. You are dearly missed.

I am grateful for funding from Microsoft Research, Engaged Cornell, the Harmony Institute, the American Statistical Association, and several small grants from Cornell that funded research and conference travel. Also thanks to ICLR, AIES, WiML Workshop, and various NeurIPS workshops for conference travel funds.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In critical domains such as healthcare, finance, and criminal justice, merely knowing what was predicted, and not why, may be insufficient to deploy a machine learning model. This dissertation proposes new methods to open up black-box models, with the goal of helping creators, as well as users, of machine learning models increase their trust and understanding of the models.

Recent research in interpretability of machine learning models has largely proceeded in two directions [94, 37]. First is the design of new model classes claimed to be interpretable, e.g. decision rules, sparse linear models, etc. [95, 96, 89, 85, 59, 6]. However, in many real-world settings, the choice of model class may be limited – a machine learning model may have already been deployed, or external factors such as hardware specifications, legal constraints, stakeholder preferences, etc. may place restrictions on the type of model that can be used [151, 56, 100, 150]. Motivated by the need still for faithful and accurate explanations in such settings, a second line of research developing "explanations" [37] for predictions made by an already-trained model has received much attention. These explanations can be "local" or "global"; the former explains the prediction made for one observation [13, 114, 99, 125, 120, 142, 26], the latter aims to explain the prediction function for an entire model [32, 66, 67, 115, 157, 87, 69].

Two chapters of this dissertation fall within this latter line of research, focusing on developing and refining post-hoc, global explanations for black-box models. In Chapter 2, we develop a method using model-agnostic distillation techniques [32, 25, 65] to learn global additive explanations that describe the relationship between input features and model predictions. Unlike other global

explanation methods such as partial dependence [45], distillation allows us to learn explanations in a discriminative manner, minimizing the fidelity error between the black-box model and the explanation while preserving the explanation's interpretability. We apply the method to fully-connected neural networks on semantically meaningful features. Developing post-hoc explanations for black-box models comes with the unique challenge of evaluation and validation, where the generated explanation has to be evaluated against ground-truth, which has to be derived from the black-box model. We designed synthetic experiments with known ground-truth prediction functions to study the proposed method. We also showed that additive explanations have interpretability advantages over non-additive explanations with a user study on expert users.

In Chapter 3, we move from model-agnostic model distillation techniques to leveraging known structure specific to the black-box model, focusing on tree ensembles such as random forests [19] and gradient boosted trees [47]. One output from training a random forest that has received less attention is the proximity matrix [19], an n-by-n matrix (n is the number of observations) describing the proportion of trees in the forest where a pair of observations end up in the same terminal node. This similarity metric between observations is locally adaptive in tree space [143] and reflects how the tree ensemble makes its predictions based on the features. We extend the formulation of proximity matrices for random forests to gradient boosted tree (GBT) models. Unlike random forests, each tree in a GBT model does not contributes equally to the prediction function, hence we propose to weigh the contribution of individual trees differently, and use the learned similarity metric to select prototypical observations in each class. This method presents an alternative to other tree ensemble interpretability methods such as seeking one tree that best represents the ensemble [14] or

feature importance methods [19, 158].

The remaining chapters in this dissertation study the use of interpretability approaches to probe and debug black-box models in algorithmic fairness settings. Here, black-box takes on another meaning, with many risk-scoring models for high stakes decision such as credit scoring and judicial bail [82] being proprietary and opaque, not lending themselves to easy inspection or validation.

In Chapter 4, we develop a methodological extension to the global additive explanation method developed in Chapter 2 to probe risk scoring models, by leveraging additional information on ground-truth outcomes that the risk scoring model was intended to predict. The proposed approach, Distill-and-Compare, was motivated by a desire to audit such models under realistic conditions, without probing the model API (since it may not be released by the model creators) or pre-defining features to audit (since bias may exist not just in features such as race or gender, but in other seemingly innocuous features). Demonstrating the method on COMPAS [8] and other data sets in the domains of credit and recidivism, we uncovered a potential misrepresentation of risk by COMPAS for younger and older individuals, the year where an online lender likely overhauled its credit scoring model, and other interesting insights that we had no prior indication of. We also proposed a statistical test to determine if a data set is missing key features used to train the black-box model, and find that the ProPublica data is likely missing key feature(s) used in COMPAS. An ancillary contribution of this chapter is a new confidence interval estimate for a class of tree-based additive models called Explainable Boosting Machine (EBM)[1]

[1]EBM is an implementation of GA^2M, a type of interpretable model introduced in [95, 96, 24]. EBM can be found at `https://github.com/microsoft/interpret`. EBM was recently renamed from iGAM. Since the paper that Chapter 4 is based on was published before the

In Chapter 5, we provide a concrete case study using interpretability approaches to debug black-box models, in this case, a hybrid Human + Machine recidivism prediction model. Previous work asked Mechanical Turk workers to evaluate a subset of defendants in the COMPAS data for risk of recidivism, and concluded that COMPAS predictions were no more accurate or fair than predictions made by humans [39]. To probe this claim further, we attempted to leverage differences between human and COMPAS decision making to create more accurate hybrid models, but these hybrid models failed to improve significantly over individual human or COMPAS decisions. Applying the methods proposed in this dissertation, we determined salient features that affected individual decision making and characterized regions where human and COMPAS decision making agreed and disagreed. The analyses revealed that human and COMPAS decisions anchored on the same features, supporting our findings that human and COMPAS decision making did not differ significantly enough to harness the promise of hybrid Human + Machine decision making in this case, concluding this dissertation on interpretability approaches for real-world settings.

renaming of iGAM to EBM, we retain the name iGAM throughout Chapter 4.

CHAPTER 2

LEARNING AND EVALUATING GLOBAL ADDITIVE EXPLANATIONS OF BLACK-BOX MODELS

2.1 Introduction

Recent research in interpretability has focused on developing *local* explanations: given an existing model and a sample, explain why the model made a particular prediction for that sample [114]. The accuracy and quality of these explanations have rapidly improved, and they are becoming important tools in interpretability. However, the human cost of examining multiple local explanations can be prohibitive, and it is unclear whether multiple local explanations can be aggregated without contradicting each other [115, 4].

In this paper, we are interested in *global* explanations: given an existing model, describe the overall behavior of the model. We operationalize this goal as describing the relationship between model inputs (features) and outputs (predictions), which is fundamental for several key tasks, such as understanding which features are important or debugging unexpected relationships learned by the model. As this task is most meaningful when each feature has semantic meaning [33], we focus on tabular data in this paper.

Given the prediction function of a black-box model, $F(\mathbf{x})$ and samples $\mathbf{x}$ consisting of features $x_i, \ldots, x_p$, we propose to use model distillation techniques

This chapter is based on material in [133].

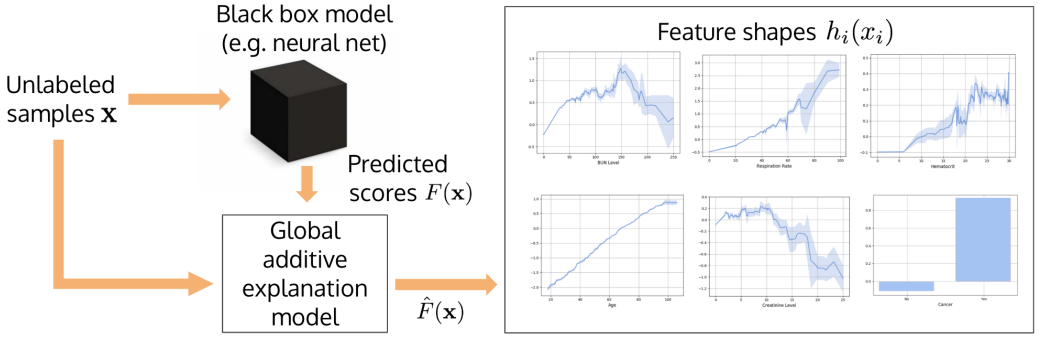

Figure 2.1: Given a black-box model and unlabeled samples (new unlabeled data or training data with labels discarded), our approach uses model distillation to learn feature shapes that describe the relationship between features and model predictions.

[22, 65] to learn post-hoc global additive explanations of the form

$$\hat{F}(\mathbf{x}) = h_0 + \sum_i h_i(x_i) + \sum_{i \neq j} h_{ij}(x_i, x_j) + \sum_{i \neq j} \sum_{j \neq k} h_{ijk}(x_i, x_j, x_k) + \cdots \tag{2.1}$$

to approximate the model's prediction function $F(\mathbf{x})$. Figure 2.1 illustrates the approach. To summarize, the black-box model is treated as a teacher and distilled into a student (an additive model) that can be visualized as a set of feature shapes $\{h_i\}$, $\{h_{ij}\}$, $\{h_{ijk}\}$.

Individual feature shapes can then be examined to determine the relationship between that feature and model predictions, the goal of our global explanations.

Feature shapes are not a new concept. Partial dependence [45], a classic post-hoc explanation method, and additive models learned directly on data [62] are also visualized in the form of feature shapes. The advantage of our approach over other additive explanations such as partial dependence (that is not learned using model distillation) is that distillation explicitly minimizes the error between the black-box model and the explanation, hence increasing the fidelity

of the learned explanation. Distilling or approximating a black-box model by a interpretable model to serve as a global explanation is also not a new concept [32, 48, 49, 118, 15, 87]. We show, with experiments on expert users (machine learning model builders) that additive explanations have interpretability advantages over decision trees for certain model understanding tasks, and hence can be a viable explanation alternative.

When learning and evaluating post-hoc explanations, some questions naturally arise: how can we tell if the explanations are telling us something real about the black-box? Our paper answers this question by designing ground-truth explanations that we then show that our approach recovers.

The main contributions of this paper are:

- We learn global additive explanations for complex, non-linear models such as neural nets by coupling model distillation with powerful additive models to learn feature shapes that directly describe the relationship between features and predictions.
- We perform a quantitative comparison of our learned explanations to other *global* explanation methods. We measure fidelity to the black-box model as a function of the complexity of the explanation model, accuracy of the explanation on independent test data, and interpretability of the explanation. The results suggest that overall, additive explanations have higher fidelity with less complexity and have interpretability advantages over decision trees and linear models for certain model understanding tasks.
- Through a user study with expert users, we quantitatively measure how interpetable different global models are and how much they help users

understand black-box models.

2.2 Related Work

Global explanations. Neural nets and other black-box models have been approximated by interpretable models such as trees, rule lists [89, 6], decision sets [85], etc. either via model distillation [32, 48] or model extraction [49, 118, 15, 87]. All of these approximated classifiers; there has been less work approximating regression models. Craven and Shavlit [32] distilled a neural net into a decision tree and evaluated the explanation in terms of fidelity, accuracy, and complexity. Frosst and Hinton [48] also distilled a neural net into a soft decision tree. Neither evaluated interpretability of their explanations. In recent work, Lakkaraju et al. [85] extracted decision set explanations customized to the features the user is interested in.

Additive explanations. Several additive explanations, not learned via distillation, have been proposed [45, 128, 67, 99]. A common theme of these methods is that they *decompose* F into $\hat{F}$ using numerical or computational methods [128, 67] (e.g. matrix inversion, quasi Monte Carlo) which can be prohibitively expensive with large n or p, or permute features and repeatedly query the black-box model with the new data [45, 99], again a computationally expensive operation we avoid by learning explanations using distillation.

Feature attribution metrics. Several metrics have been proposed for feature importance for black-box models. These include permutation-based metrics [19], gradients/saliency ([126, 127, 12, 125]; also see [104] or [5] for a review), or metrics based on variance decompositions [71] or game theory [33, 99].

These metrics provide relative rankings of features but, as they do not characterize the full relationship between features and predictions, cannot answer questions such as "when feature x_i increases by 1, how does the prediction change?", which our explanations are able to answer.

Evaluation of interpretability. There is no universal definition of interpretability [37]; many recent papers evaluate interpretability in terms of how a human uses the model to perform downstream tasks. These studies are typically performed on non-expert humans (e.g. Mechanical Turkers) [105, 111]; the exception is work mentioned above by Lakkaraju et al. [87] and concurrent work by Bastani et al. [15]; like us, they evaluate interpretability of global explanations on expert users.

2.3 Our Approach

Our goal is to learn an explanation $\hat{F}$ that (1) describes the relationship between input features $x_1, \ldots, x_p$ and the model's prediction function F; (2) approximates prediction function F well.

2.3.1 Learning Global Additive Explanations

Treating the black-box model as a teacher, we use model distillation techniques [22, 11, 65] to learn global additive explanations for the black-box model.

Black-box model: fully-connected neural nets. Our black-box models are fully-connected neural nets (FNNs) with ReLU nonlinearities (see the next sec-

tion for the training procedure). Note that our approach is not limited to neural nets, but can also be applied to learn explanations for other black-box models such as gradient boosted trees, random forests, etc. The most accurate nets we trained were FNNs with 2-hidden layers and 512 hidden units per layer (2H-512,512); nets with three or more hidden layers had lower training loss, but did not generalize as well on our data sets. In some experiments we also used a restricted-capacity model with 1 hidden layer of 8 units (1H-8). We obtain the prediction function of the black-box model, F, by having the black-box model label a set of training data.

Referring back to equation 2.1, **additive explanations** are determined by the choice of metric L between F and its approximation $\hat{F}$, degree d of highest order components (e.g. $d = 3$ in equation 2.1), and type of base learner h. Learning $\hat{F}$ using model distillation is equivalent to choosing metric L that minimizing $\|F - \hat{F}\|_L$, the empirical risk between the prediction function F and our global additive explanation $\hat{F}$ on the training data.

Our choice of two flexible, nonparametric base learners for h – splines [147] and bagged trees – gives us two global additive explanation models $\hat{F}$: **Student Bagged Additive Boosted Trees (SAT)** and **Student Additive Splines (SAS)**. In addition, we include not just main components h_i but also higher order components h_{ij} and h_{ijk} to capture any interactions between features learned by the black-box model F and increase the fidelity of the explanation $\hat{F}$ to black-box model F. Throughout this paper, we call SAT with second-order components h_{ij} **SAT+pairs** and similarly for SAS. To train SAT, SAS, and SAT+pairs, we find optimal feature shapes $\{h_i\}$ and $\{h_{ij}\}$ that minimize mean square error between

the black-box model F and the explanation $\hat{F}$, i.e.

$$L(h_0, h_1, \ldots, h_p) = \frac{1}{T} \sum_{t=1}^{T} \|F(x^t) - \hat{F}(x^t)\|_2^2$$
$$= \frac{1}{T} \sum_{t=1}^{T} \|F(x^t) - (h_0 + \sum_{i=1}^{p} h_i(x_i^t)) - \sum_{i \neq j} h_{ij}(x_i^t, x_j^t))\|_2^2, \tag{2.2}$$

where $F(x)$ is the black-box model's output (scores for regression tasks and logits for classification tasks), T is the number of training samples, x^t is the t-*th* training sample, and x_i^t is its i-*th* feature. The exact optimization details depend on the choice of h (see the next section for the training procedure).

Training procedure and implementation details: neural nets

The neural net training was done using PyTorch. We use the Adam optimizer [81] with default beta parameters, Xavier initialization [53], and early stopping based on validation loss. For each depth, we use random search to find the optimal hyperparameters (number of hidden units, learning rate, weight decay, dropout probability, batch size, enabling batch norm [70], etc) based on average validation performance on multiple train-validation splits and random initializations.

Training procedure and implementation details: student additive explanations

For student additive explanations with tree base learners (SAT), we use cyclic gradient boosting [23, 95] which learns the feature shapes in a cyclic manner. As trees are high-variance, low-bias learners [63], when used as base learners in additive models, it is standard to bag multiple trees [95, 96, 24]. We follow

that approach here. The implementation we use is called Explainable Boosting Machine (EBM)[1].

For student additive explanations with spline base learners (SAS), we use cubic regression splines trained using penalized maximum likelihood in R's `mgcv` library [148] and cross-validate the splines' smoothing parameters.

2.3.2 Visualizing Global Additive Explanations

Our global additive explanations, SAT and SAS, can be visualized as **feature shapes** (Figure 2.1). These are plots with the x-axis being the domain of input feature x_i and the y-axis being the feature's contribution to the prediction $h_i(x_i)$. Feature shapes of SAT+pairs are heatmaps of x_i and x_j, with heatmap values being the two features' interaction contribution to the prediction $h_{ij}(x_i, x_j)$. As mentioned in Section 2.1, this way of representing the relationship between features and model predictions has precedence in interpretability, with additive models learned directly on data [62] and other additive explanations (not learned using model distillation) such as partial dependence [45] and Shapley additive explanations [99] also visualized in the form of feature shapes.

Given that the **visual complexity** of additive explanations is similar – one feature shape per feature – we compare our global additive explanations to partial dependence and Shapley additive explanations in terms of fidelity (Section 2.4.2). However, an interesting question arises in terms of how to fairly compare additive explanations and non-additive explanations such as distilled decision trees, sparse linear models, etc., with each explanation having different repre-

[1] EBM is an implementation of GA^2M, a type of interpretable model introduced in [95, 96, 24]. EBM can be found at `https://github.com/microsoft/interpret`.

sentations and hence different visual complexity. We do so with a comparison of fidelity, visual complexity, and interpretability in Sections 2.4.4 and 2.5.

2.4 Experimental Results

The motivation and intended use case of global explanations described in Section 2.1 suggests the following criteria to evaluate our learned explanations:

1. **Correctness**: do learned explanations look like ground-truth explanations, if available?
2. **Fidelity**: are learned explanations faithful to the black-box model?
3. **Complexity**: how does complexity affect the fidelity and interpretability of learned explanations?
4. **Interpretability**: Can humans use the learned explanations to understand the overall behavior of the black-box model?

2.4.1 Evaluating Correctness: Synthetic Data with Ground-Truth Explanations

In this experiment, we simulate ground-truth descriptions of feature-prediction relationships to see if our explanations can correctly recover them.

Setup. Inspired by [47], we designed an additive, highly nonlinear function combining components from synthetic functions proposed by [47], [66] and [140]: $F_1(\mathbf{x}) = 3x_1 + x_2^3 - \pi^{x_3} + \exp(-2x_4^2) + \frac{1}{2+|x_5|} + x_6 \log(|x_6|) + \sqrt{2|x_7|} + \max(0, x_7) +$

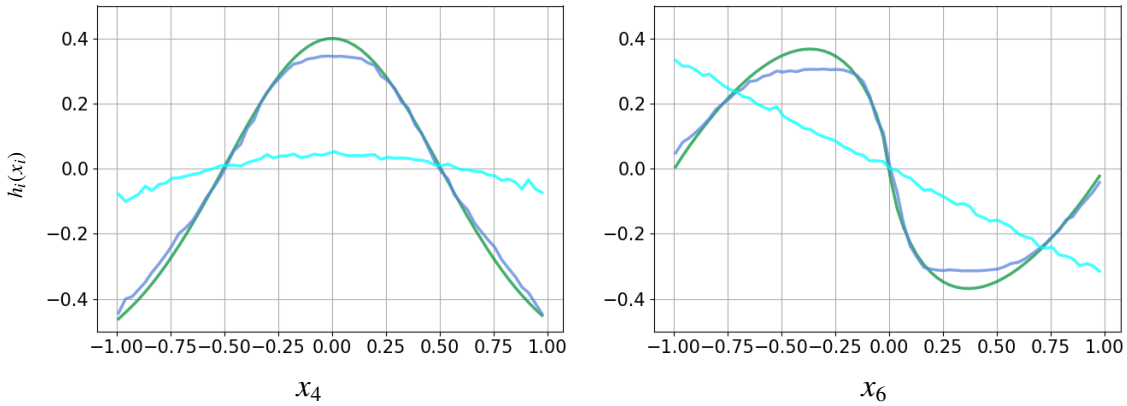

Figure 2.2: Comparison of feature shapes for ground truth, SAT of a 2H-512,512 black-box model, and SAT of a 1H-8 black-box model for two features of synthetic function F_1. Figure 2.16 at the end of the chapter contains all the feature shapes.

$x_8^4 + 2\cos(\pi x_8)$. Like [140], we set the domain of all features to be Uniform[-1,1]. Like [47], we add noise features to our samples that have no effect on $F_1(x)$ via two noise features x_9 and x_{10}. We simulate 50,000 samples, and train two neural nets, 2H-512,512 and 1H-8, to predict F_1 from the ten features.

Performance of black-box model and explanations. The high-capacity 2H neural net obtained a test accuracy RMSE of 0.14, while the low-capacity neural net obtained test accuracy RMSE of 0.48, more than 3x larger, showing that function F_1 is not trivial. We trained a SAT global additive explanation[2] for each neural net. SAT explanations are faithful, with a fidelity RMSE of 0.14 to the 1H neural net, and a fidelity RMSE of 0.08 to the 2H neural net.

Does SAT explain the black-box model, or just the original data? A first question one may have when learning post-hoc explanations of black-box models is whether the learned explanation is describing relationships encoded in the black-box model or relationships in the original data.

[2]We also experimented with SAS and obtained very similar results. For brevity and simplicity, in this section, we report only the results obtained by SAT.

Figure 2.2 compares the feature shapes of our SAT explanation to function F_1's analytic ground-truth feature shapes for two features, x_4 and x_6, of F_1 (the behavior for other features is similar). We make two observations. First, SAT's shapes for the 2H black-box model largely match the ground-truth shapes. Second, SAT's shapes for the 1H black-box model are notably different than the shapes for the 2H model, and are also less similar to the ground truth shapes. The differences in the SAT shapes for the 1H and 2H black-box models, combined with the accuracy of the black-box models and the similarity of the explanations to the ground truth, clearly indicate that the explanations explain the black-box models and not the underlying data.

Does SAT's feature shapes match the real behavior of the black-box model?

We address this question two-ways. First, we directly measure the fidelity of SAT explanations, and compare it with the accuracy of the black-box models: the 2H black-box model has an accuracy of 0.14 RMSE, and its SAT explanation has a fidelity of 0.08 RMSE; the 1H black-box model has an accuracy of 0.48 RMSE, and its SAT explanation has a fidelity of 0.14. The fidelity of the explanations is significantly better than the black box models' accuracies, indicating that the explanations are faithful to the black-box models.

Second, we measure the black box model's accuracy on samples belonging to regions where the explanations and the ground truth agree or disagree[3]. If

[3]The areas of agreement and disagreement were estimated manually by comparing the 2H-512,512 and 1H-8 feature shapes to ground truth feature shapes. Specifically, for the 2H model, where most of the feature shapes *agree* with ground truth, we define the areas of disagreement explicitly, and the areas of agreement by exclusion. For each feature shape, we defined area(s) of disagreement: $x_4 \in \{-0.1, 0.1\}, x_5 \in \{-0.1, 0.1\}, x_6 \in \{0.2, 0.4\}, x_7 \in \{-0.1, 0.1\}$. Then, we operationalized a rule to decide if a point falls in the agreement or disagreement region: it is in disagreement if it falls in areas of disagreement on at least 3 of these features, and it is in agreement if it falls in an area of disagreement in at most one feature. For the 1H model,

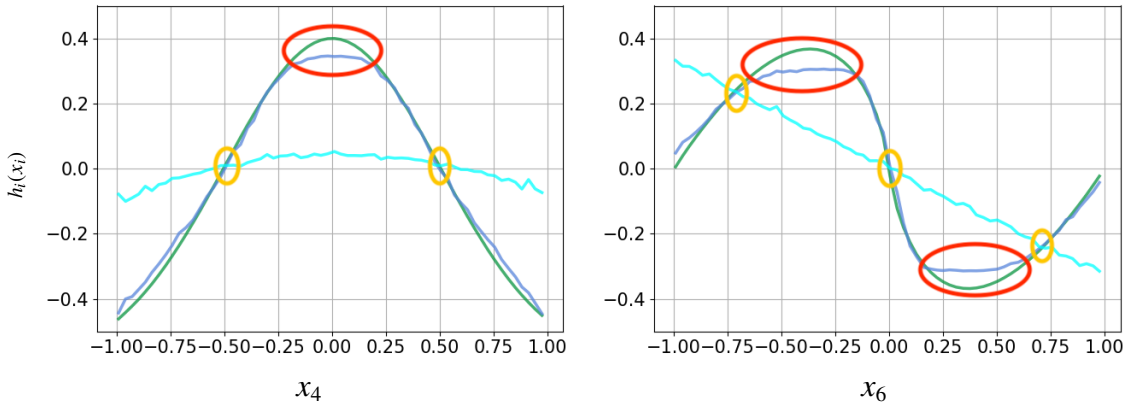

Figure 2.3: Areas of agreement and disagreement with the ground truth of SAT of a 2H-512,512 black-box model and SAT of a 1H-8 black-box model feature shapes for two features of synthetic function F_1. The areas in yellow denote the areas of agreement for the SAT 1H-8 model, and anything out of those areas is defined to be in disagreement for that model. The areas in red denote the areas of disagreement for the SAT 2H-512,512 model, and anything out of those areas is defined to be in agreement. Note that this is Figure 2.2 but with the areas of agreement and disagreement explicitly circled.

the SAT feature shapes accurately represent the black-box model, then the black-box model accuracy should be better on points sampled from areas of agreement than on points sampled from areas of disagreement. We confirm this behavior in Table 2.1: points sampled on the disagreement regions have lower accuracy than points sampled from the agreement regions[4].

How do interactions between features affect the feature shapes? We design

where most of the feature shapes *disagree* with ground truth, we define the areas of agreement explicitly, and the areas of disagreement by exclusion. For each feature shape, we defined area(s) of agreement: $x_4 \in \{-0.55, -0.45\} \cup \{0.45, 0.55\}$, $x_5 \in \{-0.5, -0.4\} \cup \{0.4, 0.5\}$, $x_6 \in \{-0.75, -0.65\} \cup \{-0.05, 0.05\} \cup \{0.65, 0.75\}$, $x_7 \in \{-0.8, 0.7\} \cup \{-0.2, -0.1\} \cup \{0.7, 0.8\}$. Then, we operationalized a similar rule to determine if a point falls in the agreement or disagreement region. Note that the areas of agreement for 1H are typically narrower than the areas of disagreement for 2H, hence the smaller area. Figure 2.3 explicitly circles the agreement and disagreement areas of features x_4 and x_6 for 1H and 2H.

[4]Note that, for 2H, the explanation matches the ground truth on most points, hence the accuracy of 'All' is similar to the accuracy of 'Agree'. For 1H, the explanation does not match the ground truth on most points, hence the accuracy of 'All' is similar to the Accuracy of 'Disagree'.

Model	**All**	**Agree**	**Disagree**
2H-512,512	0.142	0.141	0.180
1H-8	0.483	0.407	0.489

Table 2.1: RMSE error of the 2H and 1H black-box models on all samples, compared to the error on samples sampled from regions where the explanation feature shapes "agree" or "disagree" with the ground truth shape.

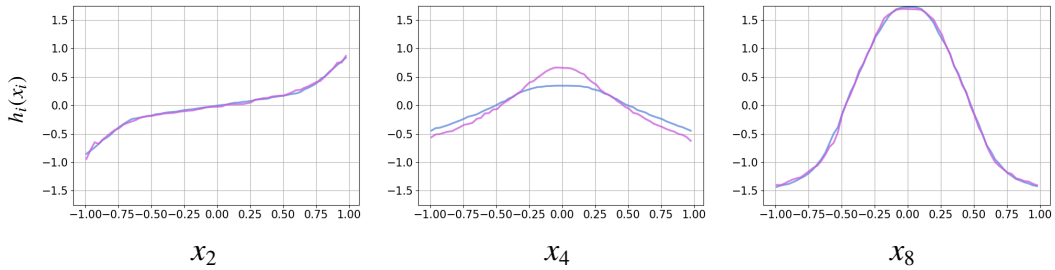

Figure 2.4: Comparison of feature shapes for SAT of a 2H-512,512 black-box model of synthetic function F_1, and SAT of a 2H-512,512 black-box model of synthetic function F_2 for three features. x_2 and x_4 participate in interactions in F_2, while x_8 does not. Figure 2.17 at the end of the chapter contains all the feature shapes.

an augmented version of F_1, $F_2(\mathbf{x}) = F_1(\mathbf{x}) + x_1x_2 + |x_3|^{2|x_4|} + \sec(x_3x_5x_6)$, which introduces interactions for features x_1 to x_6, to investigate how interactions in the black-box model's predictions are expressed by feature shapes. We simulate 50,000 samples, and train a new 2H-512,512 neural net to predict F_2 from the ten features. This function is much harder to learn (the 2H model obtained an RMSE of 0.21, compared to 0.14 of F_1) and also harder for explanation models (fidelity RMSEs of 0.35, compared to 0.08 RMSE of F_1).

Figure 2.4 displays the feature shapes of the SAT explanations from F_2 (in purple) for two features with interactions (x_4, x_2) and a feature without interactions (x_8), and compares them with the shapes from F_1 (in blue), already discussed in Figure 2.2. We first note how, for x_8 (right), the shapes from F_1 and F_2

match almost perfectly: the explanation model was not confused by the other interactions and was able to accurately match the shape of x_8. For x_4 (left), the part of the interactions that can be approximated additively by h_i has "leaked" into the h_i feature shape, slightly changing its shape as expected.

An interesting case is x_2, where, despite interacting with x_1, its feature shape has not changed and matches the feature shape from F_1. This is less surprising if we recall that feature shapes describe the *expected importance* of the feature, learned in a data-driven fashion. The interaction term is $x_1 x_2$, which, for $x_1 \sim$ Uniform[-1,1], has an expected value of zero, and therefore does not affect the feature shape. Similarly, for x_4, the expected value of the interaction $|x_3|^{2|x_4|}$ when $x_3 \sim$ Uniform[-1,1] is $1/(2|x_4| + 1)$, an upward pointing cusp, which leads to the change noticed in Figure 2.4 (left).

Finally, plots of all the feature shapes can be found in the extended result figures in Section 2.8, in Figures 2.16 and 2.17.

2.4.2 Evaluating Fidelity and Accuracy: Comparing Explanations on Real Data

In this section, we quantitatively compare our global additive explanations to other global explanations.

Setup. We selected five data sets: two UCI data sets (Bikeshare and Magic), a Loan risk scoring data set from an online lending company[5], the 2018 FICO Explainable ML Challenge's credit data set[6], and the pneumonia data set ana-

[5] https://www.lendingclub.com/info/download-data.action
[6] https://community.fico.com/s/explainable-machine-learning-challenge

				Performance		
Data	n	p	**Type**		1H	2H
Bikeshare	17,000	12	Regression	RMSE	0.60	0.38
Loan	42,506	22	Regression	RMSE	2.71	1.91
Magic	19,000	10	Classification	AUC	92.52	94.06
Pneumonia	14,199	46	Classification	AUC	81.81	82.18
FICO	9,861	24	Classification	AUC	79.08	79.37

Table 2.2: Performance of neural net black-box models. For RMSE, lower is better. For AUC, higher is better.

lyzed by [24]. We train a 2H-512,512 neural net that we will use as the main black-box model in this section (see Section 2.3 for training procedure). Table 2.2 presents the accuracy of the black-box model, as well as the accuracy of a lower-capacity 1H-8 black-box model (provided for comparison purposes) and additional details about the datasets.

Metrics. Lundberg and Lee [99] suggested viewing an explanation of a model's prediction as a model itself. With this perspective, we quantitatively evaluate explanation models as if they were models. Specifically, we evaluate not just fidelity (how well the explanation matches the black-box model's predictions) but also accuracy (how well the explanation predicts the original label). Note that [99] and [114] evaluated local fidelity (called local accuracy by [99]), but not accuracy. A similar evaluation of global accuracy was performed by [78] who used their explanations (prototypes) to classify test data. We use the feature shapes of additive explanations and distilled interpretable models to predict on independent test data.

Baselines. We compare to two types of baselines: (1) Additive explanations obtained by querying the black-box model (i.e. without distillation): partial dependence (PD) [45], Shapley additive explanations [99] and gradient-based

explanations [125]; (2) Interpretable models learned by distilling the black-box model: trees and sparse linear models.

Training procedure and implementation details: baselines

Partial dependence [45] (PD) is a classic global explanation method that estimates how predictions change as feature x_j varies over its domain: $PD(x_j = z) = \frac{1}{T}\sum_{t=1}^{T} F((x_1^t, \ldots, x_j^t = z, \ldots, x_p^t)$ where the neural net is queried with new data samples generated by setting the value of their x_j feature to z, a value in the domain of x_j. Plotting $PD(x_j = z)$ by z returns a feature shape. We implement our own version of partial dependence by repeatedly setting x_j^t for all points to a, a value in the domain of x_j, and then querying the neural net with these new data samples.

Gradient-based explanations involves constructing the additive function G through the Taylor decomposition of F, defining $G(x) = F(0) + \sum_{i=1}^{p} \frac{\partial F(x)}{\partial x_i} x_i$, and defining the attribution of feature i of value x_i as $\frac{\partial F(x)}{\partial x_i} x_i$. This formulation is related to the "gradient*input" method (e.g. [125]) used to generate saliency maps for images.

Shapley additive explanations [99] is a state-of-the-art local explanation method that satisfies several desirable local explanation properties [99]. Given a sample and its prediction, Shapley additive explanations decompose the prediction additively between features using a game-theoretic approach. We use the python package by the authors of Shapley additive explanations.

Decision trees and sparse linear models were learned using the scikit-learn

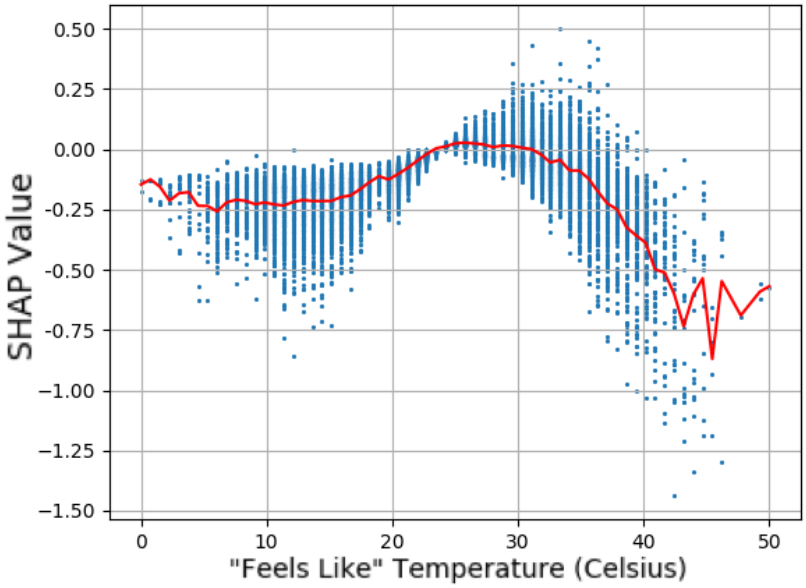

Figure 2.5: From local Shapley explanations to gSHAP, a global Shapley feature shape we create by aggregating local Shapley explanations.

Python package. **Subgroup rules** were learned using the Vikamine[7] [10] package, as we needed to learn rules for regression problems and state-of-the-art rule lists [89, 6] do not support regression. However, our results with Vikamine were unsatisfying, and we only obtained reasonable results on the Bikeshare dataset.

Constructing global explanations from local explanations. Both Shapley additive explanations and gradient-based explanations are local explanations that we adapt to a global setting by averaging the local explanations at each unique feature value. For example, the global attribution for feature "Temperature" at value 10 is the average of local attribution "Temperature" for all training samples with "Temperature=10". This is the red line passing through the points in Figure 2.5. Applying this procedure to Shapley and gradient-based local attributions, we obtain global attributions **gGRAD** and **gSHAP** that we can now plot as feature shapes.

[7] http://www.vikamine.org/

Accuracy	Global Explanation	Bikeshare RMSE	Loan RMSE	Magic AUC	Pneumonia AUC	FICO AUC
Ours	SAT	0.98 ± 0.00	2.35 ± 0.01	90.75 ± 0.06	82.24 ± 0.05	79.42 ± 0.04
	SAT+pairs	0.60 ± 0.00	2.13 ± 0.01	90.75 ± 0.06	82.23 ± 0.06	79.44 ± 0.04
	SAS	0.98 ± 0.00	2.34 ± 0.00	90.58 ± 0.02	82.12 ± 0.04	79.51 ± 0.02
Other additive methods	gGRAD	1.25 ± 0.00	6.04 ± 0.01	80.95 ± 0.13	81.88 ± 0.05	79.28 ± 0.02
	gSHAP	1.02 ± 0.00	5.10 ± 0.01	88.98 ± 0.05	82.31 ± 0.03	79.36 ± 0.01
	PD	1.00 ± 0.00	4.31 ± 0.00	82.78 ± 0.00	82.15 ± 0.00	79.47 ± 0.00
Other interpretable methods	Decision Tree	0.60 ± 0.01	2.66 ± 0.02	91.44 ± 0.29	79.38 ± 0.38	78.19 ± 0.03
	Sparse Linear	1.39 ± 0.00	3.45 ± 0.00	86.91 ± 0.01	82.06 ± 0.02	79.16 ± 0.01
Fidelity	Global Explanation	Bikeshare RMSE	Loan RMSE	Magic RMSE	Pneumonia RMSE	FICO RMSE
Ours	SAT	0.92 ± 0.00	1.74 ± 0.01	1.78 ± 0.00	0.35 ± 0.00	0.15 ± 0.00
	SAT+pairs	0.50 ± 0.00	1.47 ± 0.00	1.75 ± 0.00	0.30 ± 0.00	0.11 ± 0.00
	SAS	0.92 ± 0.00	1.71 ± 0.00	1.75 ± 0.00	0.35 ± 0.00	0.14 ± 0.00
Other additive methods	gGRAD	1.20 ± 0.00	5.93 ± 0.01	2.93 ± 0.01	0.43 ± 0.00	0.16 ± 0.00
	gSHAP	0.96 ± 0.00	4.83 ± 0.00	2.15 ± 0.00	0.46 ± 0.00	0.16 ± 0.00
	PD	0.94 ± 0.00	3.85 ± 0.00	3.17 ± 0.00	0.47 ± 0.00	0.16 ± 0.00
Other interpretable methods	Decision Tree	0.47 ± 0.01	2.12 ± 0.02	1.33 ± 0.03	0.75 ± 0.01	0.44 ± 0.01
	Sparse Linear	1.35 ± 0.00	2.87 ± 0.01	2.22 ± 0.00	0.49 ± 0.00	0.18 ± 0.00

Table 2.3: Accuracy and fidelity of global explanations for 2H black-box models. Accuracy is RMSE for regression tasks and AUROC for classification tasks; fidelity is always RMSE between the explanation model's predictions and the black-box model's scores or logits (see equation 2.2).

Results. Table 2.3 presents the fidelity and accuracy results for SAT and SAS compared to the two types of baselines: (1) other additive explanations; (2) other distilled interpretable models. We also include an augmented version of SAT that includes pairwise interactions, denoted by SAT+pairs.

We draw several conclusions. First, SAT and SAS yield similar results in all cases, both in terms of accuracy and fidelity, indicating that the particular choice of the base learner did not matter for these data sets. Capturing pairwise interactions (SAT+pairs) leads to improvements in some datasets (particularly Bikeshare and Loan, the two regression tasks), while in the remaining datasets the changes are not as remarkable. This suggests that the individual feature

shapes already provide a faithful interpretation of the model.

Compared to other additive explanations such as gSHAP and PD, SAT and SAS generally obtain better accuracy and fidelity. This is not surprising since SAT and SAS were trained specifically to mimic the black-box model. In particular, SAT and SAS are superior to PD in all tasks and metrics. Compared to other interpretable methods, SAS and SAT also obtain better results. Despite not capturing interactions, SAS and SAT are non-linear models, and hence able to model nonlinear relationships that sparse linear models cannot. Decision trees are locally adaptive smoothers [21] better able to adapt to sudden changes in input-output relationships, but that also gives them more capacity to overfit. They excel on some datasets (e.g. Bikeshare), but are not as accurate on other datasets (e.g. Pneumonia or FICO).

Figure 2.6 displays selected feature shapes for Magic and Loan. The feature shapes produced by PD tend to be much too smooth, which hurts its fidelity and accuracy. Second, in all cases, trees and splines have similar feature shapes and obtain equal or better accuracy and fidelity than the other methods. This is not surprising as the other methods are either local methods adapted to the global setting (gSHAP, gGRAD), or are global explanations that are not optimized to learn the teacher's predictions (PD). For reference, gSHAP when used as a local method (i.e. individual SHAP values, not global feature shapes) achieved a lower RMSE of 0.37 compared to 1.02 on Bikeshare, and a lower RMSE of 1.99 compared to 5.10 on Loan, which is comparable to its 2H teacher's RMSE on test data (Table 2.2). Hence, methods such as gSHAP excel at local explanations and should be used for those, but, to produce global explanations, global model distillation methods optimized to learn the teacher's predictions perform better.

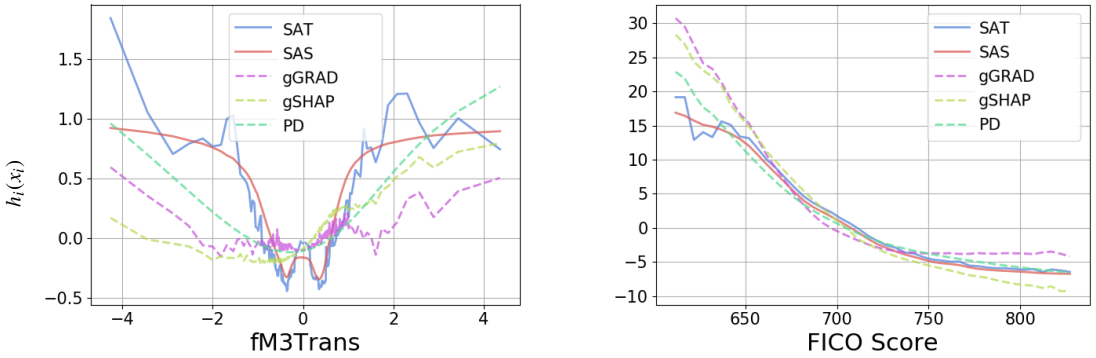

Figure 2.6: Example feature shapes for Magic data (left), and Loan data (right). SAT and SAS tend to agree.

Accuracy Teacher	Global Explanation	Bikeshare RMSE	Loan RMSE	Magic AUC	Pneumonia AUC	FICO AUC
1H-8	SAT	1.00 ± 0.00	2.82 ± 0.00	90.44 ± 0.05	82.01 ± 0.05	79.43 ± 0.02
	SAS	1.00 ± 0.00	2.82 ± 0.00	90.43 ± 0.03	81.91 ± 0.06	79.56 ± 0.02
	gGRAD	1.08 ± 0.00	2.84 ± 0.00	84.52 ± 0.67	81.63 ± 0.06	79.34 ± 0.05
	gSHAP	1.04 ± 0.00	2.87 ± 0.00	89.94 ± 0.03	82.02 ± 0.02	79.49 ± 0.02
	PD	1.00 ± 0.00	3.00 ± 0.00	85.11 ± 0.00	82.03 ± 0.00	79.46 ± 0.00
2H-512,512	SAT	0.98 ± 0.00	2.35 ± 0.01	90.75 ± 0.06	82.24 ± 0.05	79.42 ± 0.04
	SAS	0.98 ± 0.00	2.34 ± 0.00	90.58 ± 0.02	82.12 ± 0.04	79.51 ± 0.02
	gGRAD	1.25 ± 0.00	6.04 ± 0.01	80.95 ± 0.13	81.88 ± 0.05	79.28 ± 0.02
	gSHAP	1.02 ± 0.00	5.10 ± 0.00	88.98 ± 0.05	82.31 ± 0.03	79.36 ± 0.01
	PD	1.00 ± 0.00	4.31 ± 0.00	82.78 ± 0.00	82.15 ± 0.00	79.47 ± 0.00

Fidelity Teacher	Global Explanation	Bikeshare RMSE	Loan RMSE	Magic RMSE	Pneumonia RMSE	FICO RMSE
1H-8	SAT	0.64 ± 0.00	1.15 ± 0.00	1.12 ± 0.00	0.30 ± 0.00	0.21 ± 0.00
	SAS	0.64 ± 0.00	1.14 ± 0.00	1.11 ± 0.00	0.30 ± 0.00	0.21 ± 0.00
	gGRAD	0.71 ± 0.00	1.54 ± 0.00	35.40 ± 4.47*	0.36 ± 0.00	0.24 ± 0.00
	gSHAP	0.68 ± 0.00	1.28 ± 0.00	1.29 ± 0.00	0.38 ± 0.00	0.22 ± 0.00
	PD	0.65 ± 0.00	1.37 ± 0.00	1.94 ± 0.00	0.38 ± 0.00	0.25 ± 0.00
2H-512,512	SAT	0.92 ± 0.00	1.74 ± 0.01	1.78 ± 0.00	0.35 ± 0.00	0.15 ± 0.00
	SAS	0.92 ± 0.00	1.71 ± 0.00	1.75 ± 0.00	0.35 ± 0.00	0.14 ± 0.00
	gGRAD	1.20 ± 0.00	5.93 ± 0.01	2.93 ± 0.01	0.43 ± 0.00	0.16 ± 0.00
	gSHAP	0.96 ± 0.00	4.83 ± 0.01	2.15 ± 0.00	0.46 ± 0.00	0.16 ± 0.00
	PD	0.94 ± 0.00	3.85 ± 0.00	3.17 ± 0.00	0.47 ± 0.00	0.16 ± 0.00

Table 2.4: Accuracy and fidelity of global explanations for 1H and 2H black-box models. A reduced version of this table appeared in Table 2.3. This table includes results with 1H black-box models.

Finally, for completeness, we also present quantitative results using teacher of lower capacity. In particular, instead of using a 2H neural net (2H-512,512), we will use a neural net with only one hidden layer of 8 units (1H-8). In general, the lower-capacity 1H neural nets are easier to approximate (i.e. better student-teacher fidelity), but their explanations are less accurate on independent test data. Students of simpler teachers tend to be less accurate even if they are faithful to their (simple) teachers. One exception is the FICO data, where the fidelity of the 2H explanations is better. Our interpretation is that many features in the FICO data have almost linear feature shapes (see Figure 2.14 for a sample of features), and the 2H model may be able to better capture fine details while being simple enough that it can still be faithfully approximated. The accuracy of the SAT and SAS for 1H and 2H neural nets are comparable, taking into account the confidence intervals.

On the Magic data, the fidelity of the gGRAD explanation to the 1H neural net (see * in Table 2.4) is markedly worse than other explanation methods. We investigate the individual gradients of the 1H neural net with respect to each feature. 99% of them have reasonable values (between -5.6 and 6). However, 3 are larger than 1,000 (with none between 6 and 1,000) and 13 are lower than -1,000 (with none between -1,000 and -5.6), resulting in the ensuing gGRAD explanation generating extreme predictions for several samples that are not faithful to the teacher's predictions. Because AUC is a ranking loss, accuracy (AUC) is less affected than fidelity (RMSE) by the presence of these extreme values. This shows that gGRAD explanations may be problematic when individual gradients are arbitrarily large, e.g. in overfitted neural nets.

2.4.3 Evaluating Correctness: Controlled Experiments on Real Data

In this section we further validate global additive explanations on real data. Although here we do not have an analytic solution for the ground-truth feature shapes, we can still design experiments where we modify data in ways that will lead to expected known changes to the ground-truth feature shapes and then verify that these changes are captured in the learned feature shapes.

Label modification. On Bikeshare, we added 1.0 to the label (the number of rented bikes) for samples where one of the features (humidity) is between 55 and 65. We then retrained a 2H neural net on the modified data, and applied our approach to learn feature shapes from the 2H net. Ideally, the feature shapes of that new neural net should be almost identical to those of the original net except in that particular range of the humidity feature, where we should see an abrupt "bump" that increases its feature shape value by one. Figure 2.7 displays the feature shapes. Our method was able to recover the change to the label for the neural net in the new feature shape.

Data modification: expert discretization. Sometimes features are transformed before training. For example, in medical data, continuous variables such as body temperature may be discretized by domain experts into bins such as normal, mild fever, moderate fever, high fever, etc. In this experiment we test if our additive explanation models can recover these discretizations from the neural net without access to the discretized features. We train our student additive models using as input features *the original un-discretized features*, but using as labels the outputs of a neural net that was trained on discretized features. Our expectation is that if the student models are an accurate representation of

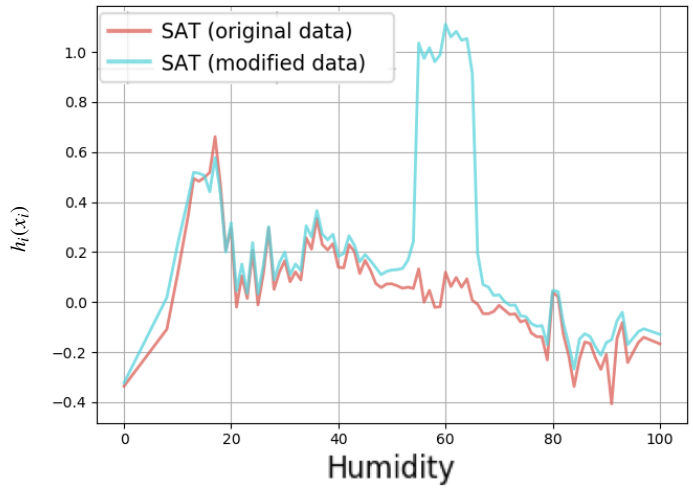

Figure 2.7: Feature shape from label modification experiment on Bikeshare data.

what the neural net learned from the discretized features, they will detect the discretizations, even if they never have access to the discretized features or to the internal structure of the neural-net teacher. We study the feature shapes of two features in the Pneumonia data (Blood pO_2 and Respiration Rate) in Figure 2.8, where we compare the feature shapes learned from teachers trained on the original continuous data (dotted lines) with those from teachers trained on discretized features (solid lines). Recall that in both cases the student models only saw non-discretized features to generate feature shapes. Our approach captures the expected discretization intervals (in yellow) as described in [30].

2.4.4 Evaluating Fidelity as a Function of Explanation Complexity

In the previous section we compared the fidelity and accuracy of SAT and SAS to other additive explanations such as gGRAD, gSHAP, and PD. Because all

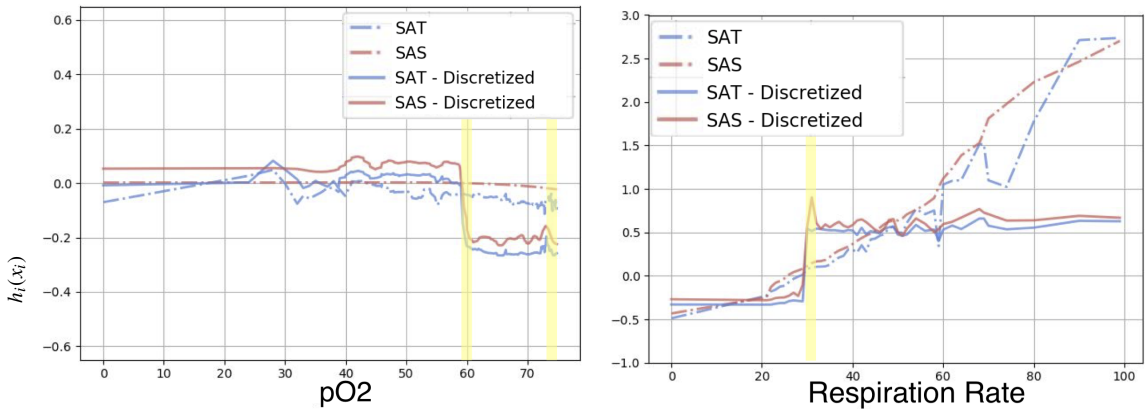

Figure 2.8: Feature shapes from data modification experiment on Pneumonia data.

these methods are additive they can be visualized in feature shapes. Models such as trees and rules, however, may be interpretable but are not additive. In this section we compare the *fidelity* of SAT explanations to sparse linear models and trees of varying complexity, showing that **the most faithful models with low complexity may be different from the most faithful models with high complexity.** In Section 2.5 we then compare the *interpretability* of SAT to trees and linear models via a user study, tying the complexity of the models with their actual interpretability.

Figure 2.9 presents the fidelity[8] of SAT and SAT+pairs compared to two other interpretable distilled models, decision trees (DT) and sparse L1-regularized linear models (SPARSE), on three of the test problems: Bikeshare, Pneumonia and Loan. The trees and linear models are trained using scikit-learn[9].

We present results as a function of a model-specific parameter K that con-

[8]The accuracy plots present very similar patterns.

[9]We also tried to compare to rule lists. However, state-of-the-art rule lists [89, 6] do not support regression, which is needed for distillation. We considered a slightly older subgroup discovery algorithm [10] that supports regression but does not generate disjoint rules, but we only achieved reasonable results on the Bikeshare dataset, hence we preferred not to report the rules results. We however use these rules for our user study on Bikeshare in Section 2.5.

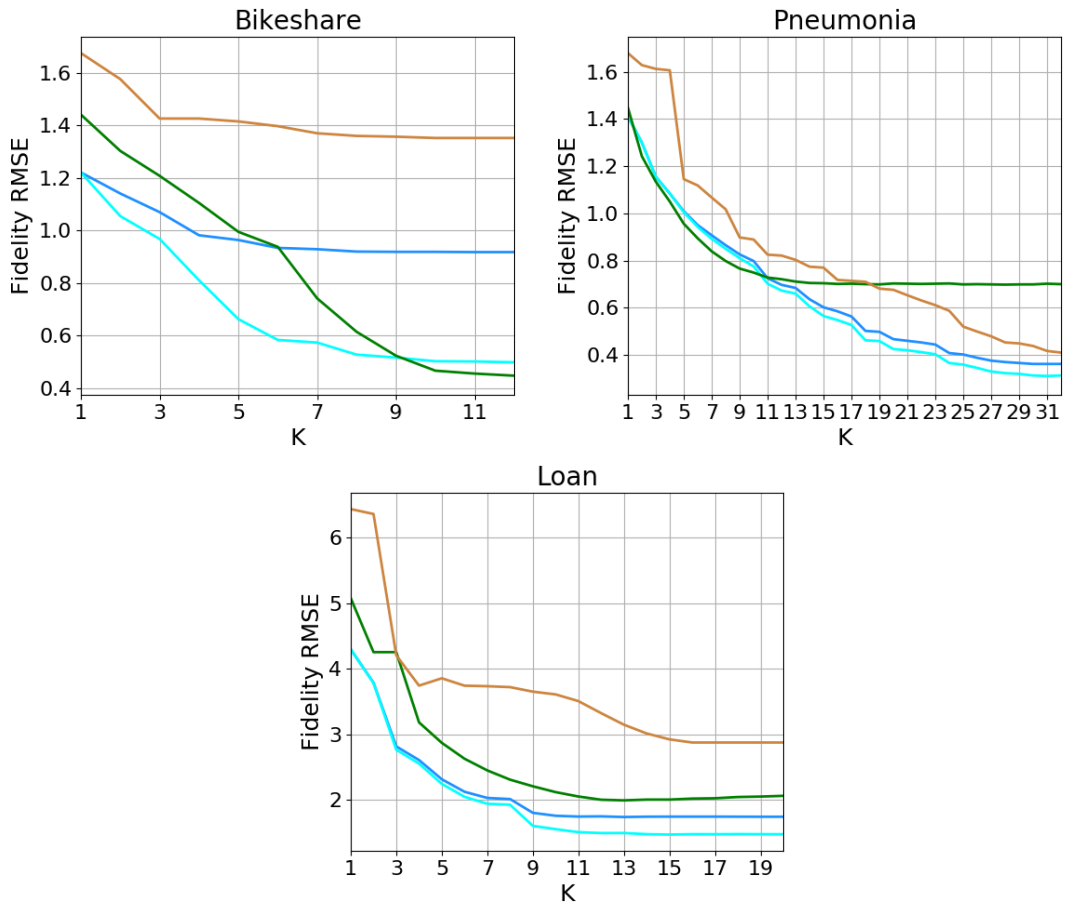

Figure 2.9: Fidelity (RMSE) of different distilled interpretable models on Bikeshare (left), Pneumonia (center) and Loan (right) data as a function of model "complexity" K. In this case, we set K as number of features for SAT and SPARSE, and tree depth for DR. Other choices are possible. The lower the fidelity RMSE, the more faithful the interpretable model to the black-box model. Key: SAT, SAT+Pairs, DT, SPARSE.

trols the complexity of the model. For SPARSE, K represents the number of features included in the model, controlled indirectly through the LASSO regularization parameter α. For DT, K is the depth of the tree. We allow a tree of depth K access to all features. Because of this, a tree of depth K might use fewer than K features (continuous or multi-valued features might be split more than once on some branches), exactly K features (e.g., if all features are Boolean), or

more than K features (by splitting different features on different branches — the most common case). For SAT and SAT+pairs, K is the number of features included in the additive model. For SAT this is also the number of shape plots. For SAT+pairs, which models pairwise interactions, the model will also include shape plots that represent stronger pairwise interactions found between the K features in the model. Note that trees of depth K can represent K-way interactions, and that the model complexity of trees falls between K and 2^K because a binary tree of depth K has 2^K leaves (2^K rules), but the complexity is somewhat less than 2^K because there is overlap in the rules resulting from the tree structure.

Overall, SPARSE has the worst fidelity. On Bikeshare and Loan, SPARSE is dominated by all other methods. On Pneumonia it is inferior to SAT and SAT+pairs for all values of K, but has better or worse fidelity than trees depending on K. Even though linear models may be interpretable, they often do not have the complexity necessary to accurately represent most black-box models. Note that two explanation methods that use sparse linear models [114] and rules [115] use them as local (not global) explanations, and only for classification (not regression).

Trees perform well given enough features and depth. On Bikeshare, trees outperform SAT by depth 7, and outperform SAT+pairs by a small amount for depth 10 and greater, although at that point one has to consider up to $2^{10} = 1024$ different paths. We suspect the deep tree is able to benefit from higher order interactions, whereas SAT+pairs is restricted to pairwise interactions to maintain intelligbility. However, the user study in Section 2.5 suggests that trees of this depth are no longer intelligible.

Overall, the best model is SAT+pairs. On Bikeshare, where interactions are important, SAT+pairs performs much better than SAT (which uses the same features but does not model interactions between features), and outperforms shallow trees of depth 8 or less. On Pneumonia and Loan, both SAT and SAT+pairs outperform SPARSE and trees of any depth. SAT+pairs consistently outperforms SAT on all three problems, by wide margin on Bikeshare, and small margins on Pneumonia and Loan.

In summary, our global additive explanations (SAT and SAT+pairs) have the highest overall fidelity to the black-box models they are trained to explain, even at low values of K. Trees sometimes exhibit high fidelity when given adequate depth, but the results from the user study in the next section suggest that depth greater than 5 or 6 hinders their intelligibility.

2.5 Evaluating Interpretability with Expert Users

We now describe the results from a user study to see if SAT additive explanations can be understood and used by humans, comparing them to other interpretable models (DT, SPARSE, RULES) distilled from the 2H-512,512 neural net. We denote the complexity of the models by model-K. For example, a tree of depth 4 would be denoted as DT-4, while a group of 5 rules would be denoted as RULES-5. Table 2.5 presents quantitative results from the user study.

Study design. 50 subjects were recruited to participate in the study. These subjects – STEM PhD students, or college-educated individuals who had taken a machine learning course – were familiar with concepts such as if-then-else structures (for trees and rule lists), reading scatterplots (for SAT), and interpret-

ing equations (for sparse linear models). Each subject only used one explanation model (between-subject design) to answer a set of questions covering common inferential and comprehension tasks on machine learning models: (1) Rank features by importance; (2) Describe relationship between a feature and the prediction; (3) Determine how the prediction changes when a feature changes value; (4) Detect an error in the data, captured by the model. The exact questions were:

1. What is the most important variable for predicting bike demand?
2. Rank all the variables from most important to least important for predicting bike demand.
3. Describe the relationship between the variable Hour and predicted bike demand.
4. What are variables for which the relationship between the variables and predicted bike demand is positive?
5. The Hour is 11. When Temperature increases from 15 to 20, how does predicted bike demand change?
6. There is one error in the data. Any idea where it might be? "Cannot find the error" is an ok answer.

In the first stage, 24 of 50 subjects were randomly assigned to see output from DT-4 or SAT-5[10]. In the second stage, we experimented with smaller versions of trees and SAT using only the two most important features, Hour and

[10] We considered DT and SAT first because they are the most accurate and faithful explanations. We used DT-4 because that is the largest tree that is readable on letter-size paper, and that does not lag too far behind the depth 6 tree in accuracy (RMSE: SAT 0.98, DT-6 1, DT-4 1.16). For reference, we show the DT-6 tree in Figure 2.12. DT-4 used five features: Hour, Temperature, Year, Working Day, Season (Figure 2.11), hence we select the corresponding five feature shapes to display for SAT-5 (Figure 2.10).

Temperature. 14 of 50 subjects were randomly assigned to see output from SAT-2 or DT-2. In the last stage, the remaining 12 subjects were randomly assigned to see output from one of the two worst performing models (in terms of accuracy and fidelity): sparse linear models (SPARSE-2) and subgroup-rules (RULES-5). The SAT-5 and DT-4 models shown to the users are in Figures 2.10 and 2.11.

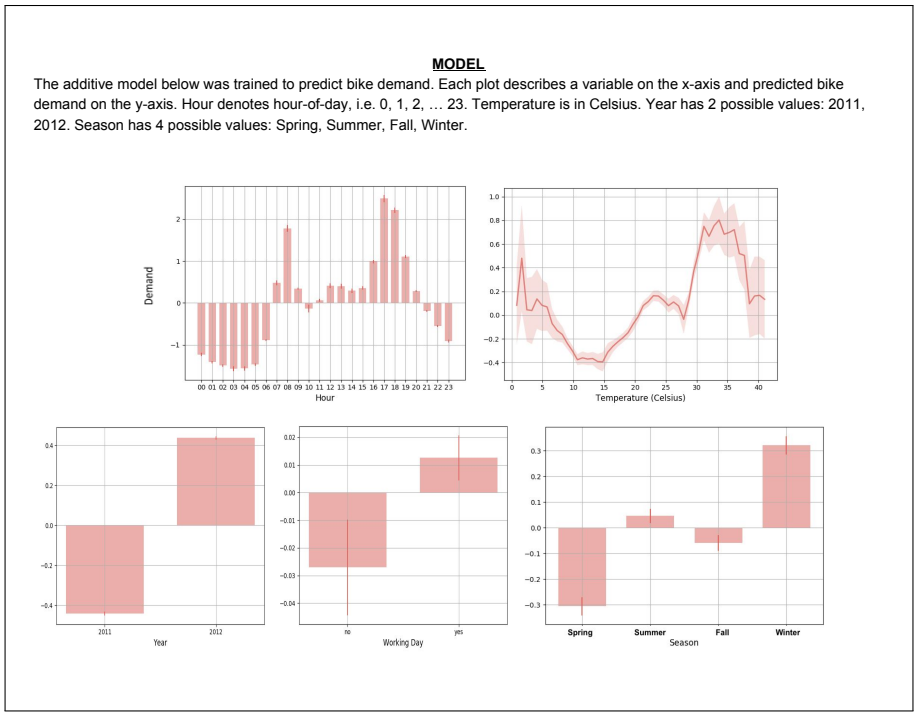

Figure 2.10: Model output shown to SAT-5 subjects in user study

Can humans understand and use feature shapes? From the absolute magnitude of the SAT feature shapes as well as Gini feature importance metrics for the tree, we determined the ground truth feature importance ranking (in decreasing order): Hour, Temperature, Year, Season, Working Day. More SAT-5 than DT-4 subjects were able to rank the top 2 and all features correctly (75% vs. 58%, see Table 2.5).

When ranking all 5 features, 0% of the DT-4 and RULES-5 subjects were able to predict the right order, while 45% of the SAT-5 subjects correctly predicted the

	First stage (n=24)		Second stage (n=14)		Third stage (n=12)	
Task	**SAT-5**	**DT-4**	**SAT-2**	**DT-2**	**SPARSE-2**	**RULES-5**
Ranked correctly top 2 features	75%	58%	100%	85.7%	83.3%	0%
Ranked correctly all (5) features	45%	0%	N/A	N/A	N/A	0%
NDCG between human ranking of top 5 features and ground-truth feature importance	0.94 ± 0.13	0.89 ± 0.11	N/A	N/A	N/A	0.27 ± 0.11
Described increased demand during rush hour	42%	0%	29%	0%	0%	33%
Described increased demand during mornings and afternoons	33%	0%	29%	0%	0%	33%
Compute change in prediction when feature changes	33%	25%	14%	100%	83%	0%
Caught data error	33%	8%	N/A	N/A	N/A	0%
Time taken (minutes)	11.7 ± 5.8	17.5 ± 14.8	7.2 ± 3.2	6.2 ± 2.2	5.2 ± 3.1	14.9 ± 8.4

Table 2.5: Quantitative results from user study. Since SAT-2, DT-2, and SPARSE-2 only had two features, the task to rank five features does not apply. Since the data error only appeared in the output of SAT-5, DT-4, and RULES-5, the other subjects could not have caught the error.

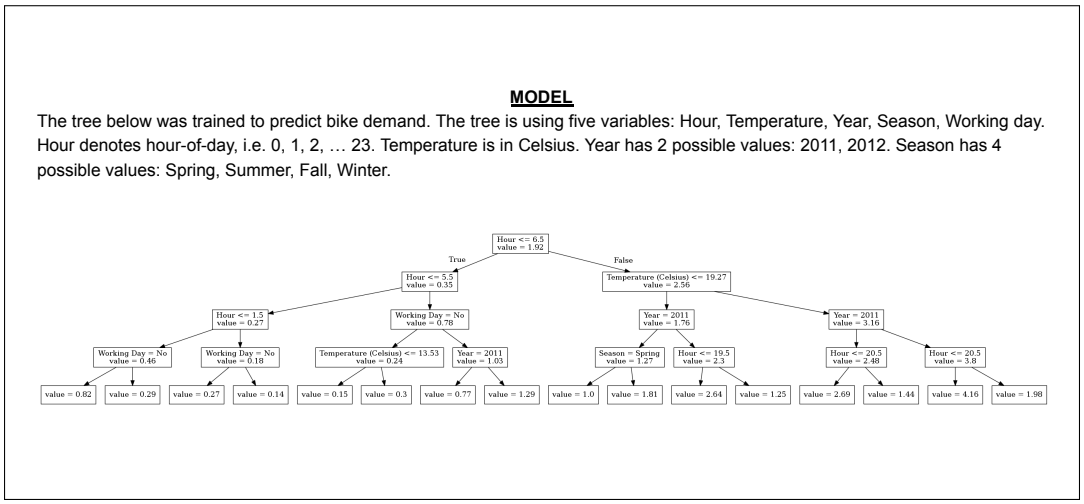

Figure 2.11: Model output shown to DT-4 subjects in user study

order of the 5 features, showing that ranking feature importance for trees is actually a very hard task. The most common mistake made by DT-4 subjects (42% of subjects) was to invert the ranking of the last two features, Season and Working Day, perhaps because Working Day's first appearance in the tree (in terms of depth) was before Season's first appearance (Figure 2.11). We also evaluate the normalized discounted cumulative gain (NDCG) between the ground truth feature importance and the user prediction, where we give relevance scores to the feature in decreasing order (i.e., for 5 features, the most important feature has a relevance score of 5, the second most important 4, etc). This gives us an idea of *how well* the features were ranked, even if the ranking is not perfect. We see how SAT-5 obtains a better score than DT-4, consistent with the previous analysis. RULES-5 obtains a significant lower score.

When asked to describe, in free text, the relationship between the variable Hour and the label, one SAT-5 subject wrote:

There are increases in demand during two periods of commuting hours: morning

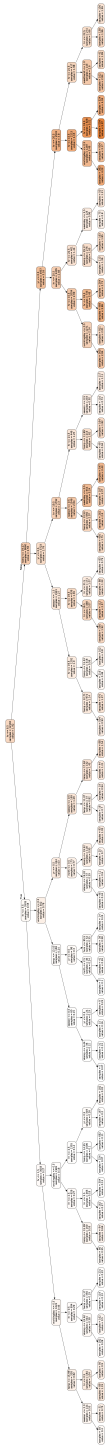

Figure 2.12: Tree of depth 6 (64 leaves), the least deep tree that matched SAT's fidelity. This uses the default tree visualizer in scikit-learn.

commute (e.g. 7-9 am) and evening commute (e.g. 4-7 pm). Demand is flat during working hours and predicted to be especially low overnight,

whereas DT-4 subjects' answers were not as expressive, e.g.:

Demand is less for early hours, then goes up until afternoon/evening, then goes down again.

75% of SAT-5 subjects detected and described the peak patterns in the mornings and late afternoons, and 42% of them explicitly mentioned commuting or rush hour in their description. On the other hand, none of the DT-4 subjects discovered this pattern on the tree: most (58%) described a concave pattern (low and increasing during the night/morning, high in the afternoon, decreasing in the evening) or a *positively correlated* relation (42%). Similarly, more SAT-5 subjects were able to precisely compute the change in prediction when temperature changed in value, and detect the error in the data – that spring had lower bike demand whereas winter had high bike demand (bottom right feature shape in Figure 2.10).

How do tree depth and number of feature shapes affect human performance? We also experimented with smaller models, SAT-2 and DT-2, that used only the two most important features, Hour and Temperature. As the models are simpler, some of the tasks become easier. For example, SAT-2 subjects predict the order of the top 2 features 100% of the time (vs 75% for SAT-5), and DT-2 subjects, 85% of the time (vs 58% for DT-4). The most interesting change is in the percentage of subjects able to compute the change in prediction after changing a feature: only 25% for DT-4, compared to 100% for DT-2. Reducing the complexity of the explanation made using it easier, *at the price of reducing the fidelity*

and accuracy of the explanation. Another important aspect is the time needed to perform the tasks: increasing the number of features from 2 to 5 increases the time needed by the subjects to finish the study by 60% for the SAT model, but increases it by 166% for the DT model, that is, interpreting a tree becomes much more costly as the tree becomes deeper (and more accurate), and, in general, subjects make more mistakes. SAT scales up more gracefully.

Remaining interpretable models: subgroup-rules and sparse linear models. These explanations were the least accurate and faithful. We found that human subjects can easily read the (few) weights of SPARSE-2, establish feature importance, and compute prediction changes, and do so quickly – at 5.1 minutes on average, this was the fastest explanation to interpret. However, the model is highly constrained and hid interesting patterns. For example, 100% of the subjects described the relation between demand and hour as increasing, and 83% predicted the exact amount of increase, but none were able to provide insights like the ones provided by SAT-5 and DT-4 subjects.

RULES-5 was the second hardest explanation to interpret based on mean time required to answer the questions: 14.9 minutes. Understanding non-disjoint rules appears to be hard: none of the subjects correctly predicted the feature importance order, even for just two features; none were able to compute exactly the change in prediction when feature value changes, and none were able to find the data error. The rules in RULES-5 are not disjoint because we could not find a regression implementation of disjoint rules. However, 66% of the subjects discovered the peak during rush hour, as that appeared explicitly in some rules, e.g. "If hour=17 and workingday=yes then bike demand is 5".

Fidelity vs. interpretability. Figure 2.13 presents detailed results for indi-

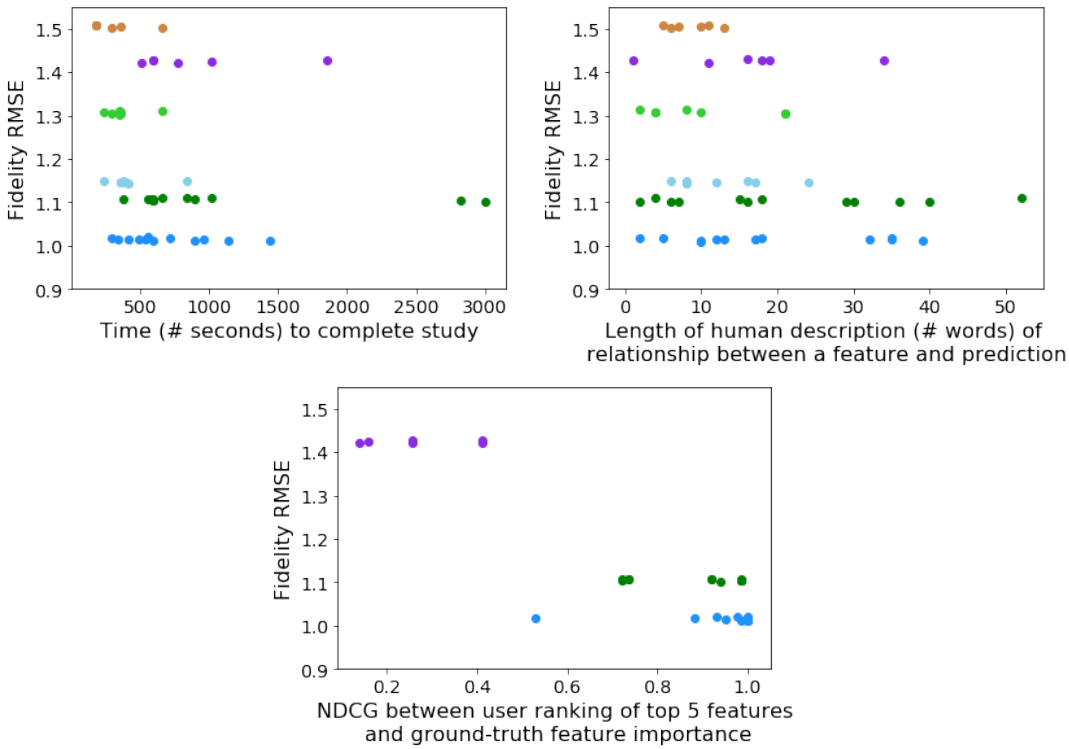

Figure 2.13: User study metrics, as proxies for interpretability, by fidelity (RMSE) for different explanations. Each point is an individual user in the user study. The metrics are time needed to finish the study (top left), length of the description (top right), and the NDCG of the ranked features (bottom). Key: SAT-5, DT-4, SAT-2, DT-2, RULES-5, SPARSE-2

vidual users by model. On the left is the time needed to finish the study (left). In the center is the length of the user's written description of the relationship between a feature and model predictions. On the right is the NDCG rank loss of user ranking of feature importance compared to ground-truth feature importance. All of these metrics can be considered interpretability metrics, when defining interpretability as grounded in human tasks [37]. On the y-axis is fidelity (RMSE).

The plots show that there is a trade-off between fidelity and interpretbility (as measured by time to complete, description length, and NDCG of feature

rankings), but not all methods behave similarly. In general, the SPARSE-2 model is easy to understand (users typically finish the study rapidly), but fidelity is poor and it leads to short descriptions. On the other hand, SAT-5 and DT-4 have much better fidelity and lead to more varied descriptions, but also took longer to understand. DT-2 was faster to complete than DT-4, but the fidelity is lower and the descriptions shorter. RULES-5 is better than SPARSE-2, but not as good as SAT-5 or DT-4. SAT-5 offers a reasonable trade-off, being both faithful and relatively easy to understand, while also leading to rich descriptions for many users.

To summarize, global additive explanations: (1) allowed humans to perform better (than decision trees, sparse linear models, and rules) at ranking feature importance, pointing out patterns between certain feature values and predictions, and catching a data error; (2) Additive explanations were also faster to understand than big decision trees; (3) Very small decision trees and sparse linear models had the edge in calculating how predictions change when feature values change, but were much less faithful and accurate.

2.6 Applications and Extensions

In this section we discuss applications of our approach and extensions to include higher-order interactions.

2.6.1 Utility of Global Additive Explanations

Checking for monotonicity

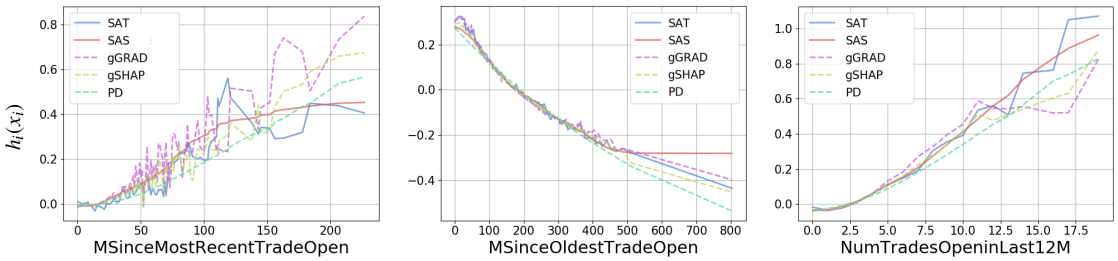

Figure 2.14: Checking for monotonicity in 3 of 16 features with expected monotonically increasing or decreasing patterns in FICO data. The feature on the left, "Months Since Most Recent Trade Open", was expected to decrease monotonically, but actually increased monotonically.

Domains such as credit scoring have regulatory requirements that prescribe monotonic relationships between predictions and some features [40]. For example, the 2018 FICO Explainable ML Challenge[11] encouraged participants to impose monotonicity on 16 features. We use feature shapes to see if the function learned by the neural net is monotone for these features. 15 of 16 features are monotonically increasing/decreasing as required. One feature, however, "Months Since Most Recent Trade Open" was expected to decrease monotonically, but actually increased monotonically. This is true not just in our explanations, but also in PD, gGRAD, and gSHAP (Figure 2.14). The two figures on the right are two related features, "Months Since *Oldest* Trade Open" and "Number of Trades Open in Last 12 Months", both of which exhibit the expected monotonically decreasing/increasing patterns. .

With the insight from the global explanations that the neural net may not be

[11] https://community.fico.com/s/explainable-machine-learning-challenge

exhibiting the expected pattern for "Months Since Most Recent Trade Open", we perform a quick experiment to verify this in the neural net. We sample values of this feature across its domain, set all data samples to this value (for this feature), and obtain the neural net's predictions for these modified samples. The majority of samples (70%) had predictions that increased as this feature increased across its domain, confirming that on average, the neural net exhibits a monotonically increasing instead of decreasing pattern for this feature. Note that we could not have checked for a monotonicity pattern (which is by definition a global behavior) without checking and aggregating multiple local explanations.

Visualizing neural net training: from underfit to overfit.

Using additive models to peek inside a neural net creates many opportunities. For example, we can see what happens in the neural net when it is underfit or overfit; when it is trained with different losses such as squared, log, or rank loss or with different activation functions such as sigmoid or ReLUs; when regularization is performed with dropout [129] or weight decay; when features are coded in different ways; etc. The video at `https://youtu.be/ATNcgurNHhc` shows what is learned by a neural net as it trains on a medical dataset. The movie shows feature shapes for five features before, at, and after the early-stopping point as the neural net progresses from underfit to optimally fit to overfit. We had expected that the main cause of overfitting would be increased non-linearity (bumpiness) in the fitting function, but a significant factor in overfitting appears to be unwarranted growth in the confidence of the model as the logits grow more positive or negative than the early-stopping shape suggests is optimal.

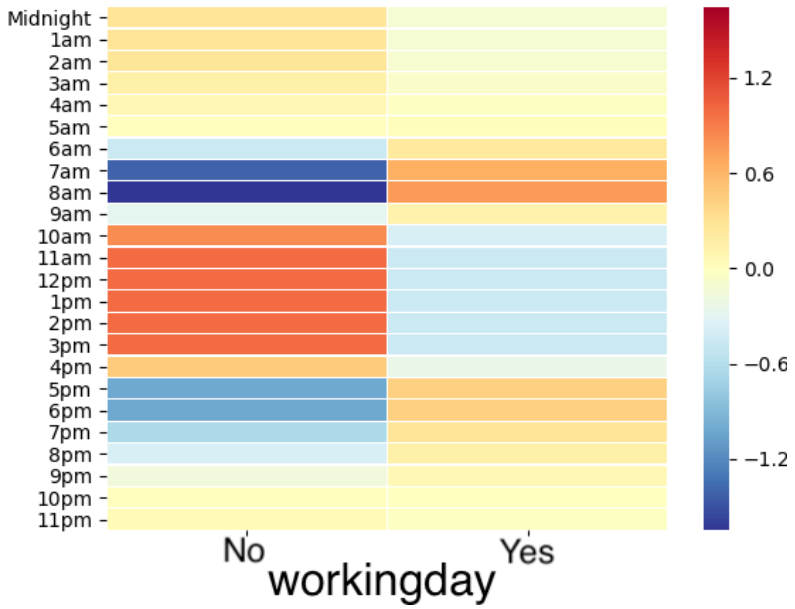

Figure 2.15: Visualizing working day, an important pairwise interaction in the Bikeshare data.

2.6.2 Extending Global Additive Explanations to Include Interactions

Functions learned by neural nets cannot always be represented with adequate fidelity by the additive function $\hat{F}$ in equation 2.1. We can improve $\hat{F}$'s expressive power by adding pairwise and higher-order components h_{ij}, h_{ijk}, and so on to account for interactions between two or more input features. In Bikeshare, RMSE decreases from 0.98 to 0.60 when we add pairwise interactions to the student model. Figure 2.15 shows an interesting interaction between two features: "Time of Day", and "Working Day". On working days, the highest bike rental demand occurs at 7-9am and 5-7pm, but on weekends there is very low demand at 7-9am (presumably because people are still sleeping) and at 5-7pm, and demand peaks during midday from 10am-4pm. These two features also form a three-way interaction with temperature. Whenever the teacher neural

net learned these (and other) interactions, a global explanation method must also incorporate interactions if it is to provide high-fidelity explanations of the teacher model. Our approach is able to do so by adding higher-order components h_{ij}, h_{ijk}, and so on to the global additive explanation $\hat{F}$.

2.7 Conclusions

We presented a method for "opening up" complex models such as neural nets trained on tabular data. The method, based on distillation with high-accuracy additive models, has clear advantages over other global explanations that learn additive explanations without distillation, and non-additive explanations such as trees that do use distillation. The method will work with any black-box classification or regression model including random forests and boosted trees, but is not designed for models such as CNNs trained on raw inputs such as images where providing a global explanation in terms of input pixels is not meaningful. Different kinds of explanations are useful for different purposes, and global additive models do not aim to replace local explanations. The results of our experiments and a user study on expert users (machine learning model builders) suggest that distillation into high-performance additive models provides explanations that have a strong combination of fidelity, low-complexity, and interpretability.

2.8 Extended Result Figures

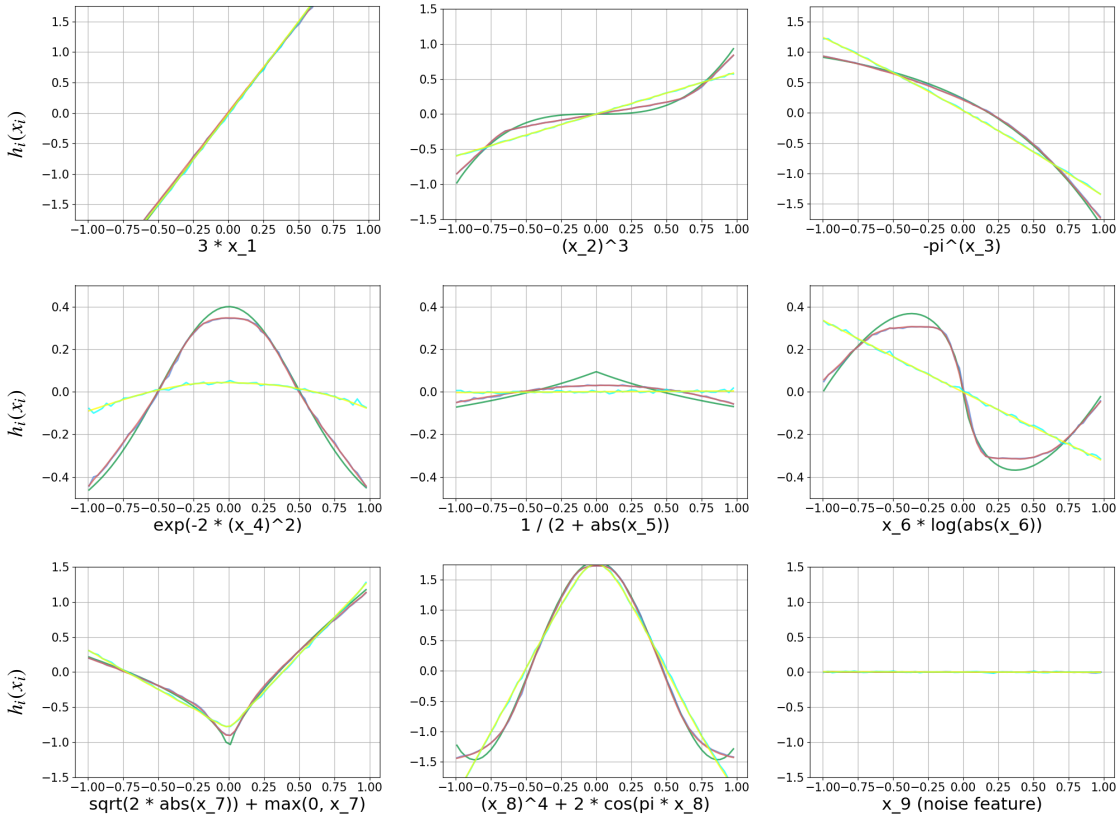

Figure 2.16: Feature shapes for features x_1 to x_9 of F_1 from Section 2.4.1. The color represents different models and configurations, as well as the ground truth shape: Ground truth, SAT-2H, SAS-2H, SAT-1H, SAS-1H, where SAT-2H represents a SAT explanation for a 2H black-box model. In a slight abuse of notation, we include the specific h function in the x axis of the plot. Notice how x_9, which is a noise feature that does not affect F_1, has been assigned an importance of approximately 0 throughout its range. The feature shape of x_{10}, another noise feature, is very similar to x_9 and hence not included here. Also, note how the scales of the plots for features x_5 and x_6 are slightly different, to allow a better visualization of the differences between models.

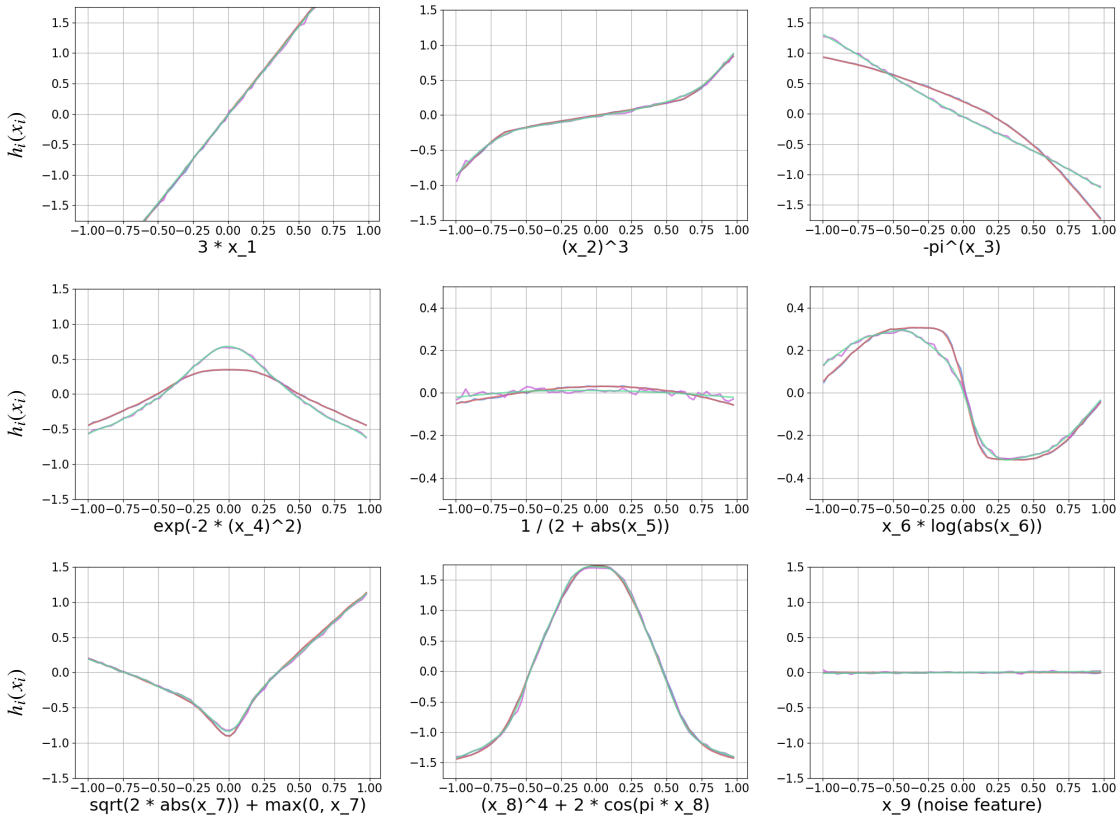

Figure 2.17: Feature shapes for features x_1 to x_9 of F_1 and F_2 from Section 2.4.1. The color represents different models and synthetic functions: SAT-F_1-2H, SAS-F_1-2H, SAT-F_2-2H, SAS-F_2-2H. In a slight abuse of notation, we include the specific h function in the x axis of the plot. Notice how x_9, which is a noise feature that does not affect F_2, has been assigned an importance of approximately 0 throughout its range. The feature shape of x_{10}, another noise feature, is very similar to x_9 and hence not included here. Also, note how the scales of the plots for features x_5 and x_6 are slightly different, to allow a better visualization of the differences between models.

CHAPTER 3

TREE SPACE PROTOTYPES: ANOTHER LOOK AT MAKING TREE ENSEMBLES INTERPRETABLE

3.1 Introduction

Ensembles of decision trees have been shown to perform well across a variety of problems [25]. These models include models such as random forests (RF) [19] and boosted trees, including gradient boosted trees (GBT) [45] and AdaBoost [44]. However, while their component models - decision trees - are typically considered interpretable [43], ensembles of hundreds or thousands of trees are no longer as interpretable and hence may be less preferred in certain settings, despite their predictive capabilities.

Current attempts to interpret tree ensembles include seeking one tree that best represents a tree ensemble according to some metric [60, 157, 119], model-agnostic (not exclusive to tree ensembles) explanations of predictions [114, 87] and feature selection in tree ensembles using, for example, variable importance measures [73, 158] or partial dependence plots [46] and variations [67, 133]. However, the interpretability of latter methods decreases as the number of features (and hence number of feature importance measures or partial dependence plots) increases.

Prototypes are representative observations that provide a condensed view

This chapter is based on material in [138].

of the data set [17]. The value of prototypes, utilized in case-based reasoning [116], has been discussed in studies of human decision making, cognition, and understanding [79]. Moreover, prototypes may be especially useful when the number of observations is too large, rendering inspection of individual observations cumbersome, or when representative observations have more meaning than some linear combination of features.

In this paper, we propose a new approach towards interpreting tree ensembles. We use an existing distance defined for RF models and extend the idea to GBT models. Then, we utilize prototype selection methods to find prototypical observations, as "seen" from the point of view of the tree ensemble. These prototypes can be utilized in multiple ways to increase the interpretability of tree ensembles - presented to a user such as a domain expert as representative observations for a class, utilized for classification in a nearest-prototype-classifier, or as warm-start points for clustering.

Note that prototype selection is different from prototype generation – the former selects observations present in the dataset; the latter generates new observations [50]. The two have distinct challenges; we focus on the former in this paper since our goal is to generate a representative set of existing observations. To the best of our knowledge, this is the first method to seek prototypes from GBT models using the naturally-learned distance from the tree ensemble.

3.2 Background

In this section, we provide an overview of relevant technical background. First, we introduce some notation we will use throughout the paper. We assume that

we are given a training set of observations S, from which we will learn k prototypes across q classes to better understand a classifier function $c : S \to [q]$ that assigns each observation to one of q classes. To learn these k prototypes, we will introduce a number of distance functions $d : S^2 \to \mathbb{R}^+$ to attempt to capture how the classifier represents differences between observations and classes.

3.2.1 Tree Ensemble Models

We broadly follow the notation of tree ensemble models in [63], adapting the notation to our needs. Let t denote the total number of trees in the RF model. The ith tree ($i \in [t]$) has some number τ_i of terminal nodes, each of which represents some region $R_{j,i}$ ($j \in [\tau_i]$) of the feature space. Each individual tree induces a classifier

$$c_i^{\text{Tree}}(s) = \sum_{j=1}^{\tau_i} \alpha_{j,i} \mathbb{I}(s \in R_{j,i}),$$

where $\alpha_{j,i}$ is the predicted value in the jth terminal node of the ith tree (for binary classification, this is just the proportion of observations in that terminal node with label 1) and $\mathbb{I}$ is the indicator function. The RF classifier is the average of this, taken over all trees:

$$c^{\text{RF}}(s) = \frac{1}{t} \sum_{i=1}^{t} c_i^{\text{Tree}}.$$

Next, we consider the GBT classifier, which is constructed iteratively:

$$c_i^{\text{GBT}}(s) = c_{i-1}^{\text{GBT}}(s) + \gamma_i c_i^{\text{Tree}}(s)$$

where the initial value c_0^{GBT} is initialized, depending on implementation, as zero, the fraction of elements of S with label 1 in the case of binary classification, etc.

γ_i is a step size, typically found using line-search. The GBT classifier then is the one that incorporates all t trees:

$$c^{\mathrm{GBT}}(s) = c_t^{\mathrm{GBT}}(s).$$

RF Distance

Tree structure lends itself to a natural definition of proximity and distance between observations as "seen" by a tree, if we consider a pair of observations that travel down the same path in a tree and end up in the same terminal node closer than another pair of observations that do not end up in the same terminal node.

Definition 1. *[20] The RF proximity of a pair of observations is an unweighted average of the number of trees in the RF model in which the observations end up in the same terminal node:*

$$\begin{aligned} & \mathsf{proximity}^{\mathrm{RF}}(s, s') \\ = \; & \frac{1}{t} \sum_{i=1}^{t} \sum_{j=1}^{\tau_i} \mathbb{I}(s \in R_{j,i}) \mathbb{I}(s' \in R_{j,i}). \end{aligned}$$

The RF distance between a pair of observations is then:

$$d^{\mathrm{RF}}(s, s') = 1 - \mathsf{proximity}^{\mathrm{RF}}(s, s').$$

Since the regions $\{R_{j,i}\}_{j=1}^{\tau_i}$ partition the feature space, each point $s \in S$ can be in at most one region, and so the inner sum takes on value 0 or 1 for each tree. Thus the proximity, as a convex combination of these, lies between 0 and 1, and so does the distance function. It is easily confirmed that the proximity of a point to itself is 1, and hence $d(s, s) = 0$, but it should be noted that d is not in general a metric, but a pseudosemimetric as it does not satisfy the triangle inequality –

as noted by [149], this it not uncommon in the metric learning literature, and in fact, no locally adaptive distance (distance that varies across feature space [92]) can satisfy the triangle inequality [149].

Later, we will adapt the RF distance to construct a distance function for GBTs.

3.2.2 The k-Medoids Problem

The goal of the *k-medoids clustering* problem is to find a subset $M \subseteq S$ of k *medoids*, such that the sum distance from each object to the nearest medoid is minimized. In the prototype selection literature, medoids have been taken as prototypes [17], hence our interest in it. Formally, the k-medoids algorithm aims to find the set $M \subseteq S$ that minimizes the objective function

$$f(M) = \sum_{s \in S} \min_{m \in M} d(s, m). \tag{3.1}$$

This problem is known to be NP-hard [109]. However, [55] present a greedy algorithm that starts with an empty set and repeatedly adds the single object $s \in S \setminus M$ that increases the value of a related function by the most, which they show produces a reasonable approximation in polynomial time.

If the objects are labelled by a classifier, it is natural to only consider for each object the medoids that belong to the same class. Thus, we define the *q-classwise k-medoids problem* as finding the subset $M \subseteq S$ of k medoids such that the sum distance from each object to the nearest medoid *belonging to the same class* is minimized, i.e. that minimizes

$$f(M) = \sum_{s \in S} \min_{m \in M : c(m) = c(s)} d(s, m). \tag{3.2}$$

Even in the presence of multiple classes, it is possible to use the single-class algorithm of [55] by applying it separately to every class in turn to generate k_1, $\ldots$, k_q prototypes for each class ($\sum_i k_i = k$). However, it is not clear what the right choice of k_i for each class is, and one could easily wind up losing accuracy by overprovisioning one compact class that would be adequately covered by a small number of prototypes while not having sufficiently many prototypes for another class whose points are spread into many clusters. With the naive choice that $k_1 = \ldots = k_q = k/q$, we call this the *uniform* greedy prototype selection algorithm, and use it as one of our candidate methods.

However, it turns out that an analysis similar to that for the single-class case can also be applied directly to the q-classwise objective function. Based on this, we will introduce a greedy algorithm that operates on all classes in the q-classwise k-medoids problem simultaneously. Since this algorithm in effect chooses the class where adding another prototype yields the largest improvement, we will call it *adaptive* in contrast with the uniform algorithm.

3.2.3 Submodular optimization

Optimization problems such as (3.2) are often approached using approximation algorithms that are guaranteed to find solutions within some factor of the optimum. One approach to the k-medoids problem [55, 103] is to identify a related positive monotone submodular function and use a greedy search for a good element of its domain, which is then guaranteed to be within a factor of $(1 - 1/e)$ of the optimum for that function, where e is the Euler constant. We quickly review the relevant result.

Definition 2. *A function $f : \mathcal{P}(S) \to \mathbb{R}$ that maps subsets of S to reals is* monotone *if $f(X) \leq f(Y)$ whenever $X \subseteq Y$. It is* submodular *if whenever $X \subseteq Y$, adding a particular element $s \in S$ to Y will not be more useful than adding it to X:*

$$f(Y \cup \{s\}) - f(Y) \leq f(X \cup \{s\}) - f(X).$$

Proposition 1. *[106] Suppose $f : \mathcal{P}(S) \to \mathbb{R}^+$ is a non-negative monotone submodular function. Let $T_0 = \emptyset$ and*

$$T_i = T_{i-1} \cup \arg\max_{s \in S} f(T_{i-1} \cup \{s\})$$

be the result of greedily maximizing f for i steps. Also, let

$$T_i^* = \arg\max_{T \subset S : |T| = k} f(T)$$

be the set of size i that maximizes f. Then

$$f(T_i) \geq (1 - 1/e) f(T_i^*).$$

We will use this proposition to define a greedy algorithm on an appropriate submodular function to find an approximate solution to the q-classwise k-medoids problem (3.2).

3.3 Method

Our goal is to find prototypes for tree ensemble models such as RF and GBT, as an alternative approach to interpreting tree ensembles. In this section, we describe three methodological contributions of this paper: defining a distance

function for GBT models, and two prototype selection algorithm that choose a variable number of prototypes based on which class could benefit the most from another prototype: one that exploits the submodular approximation guarantees of Proposition 1, and one that tries to directly optimize for accuracy on the training set.

3.3.1 Constructing a Distance Function for GBT

Unlike the RF distance function in Definition 1, in GBT each tree is no longer generated by an identical process. Hence, each tree can no longer be weighted equally. We propose to weigh the contribution of each tree to the proximity function by the size of its contribution to the overall prediction. Here, we measure size by the variance among the predictions made by $c_i^{\text{Tree}}(s)$. This can be seen as a measure of the L^2 norm of c_i^{Tree} on the distribution of the training set. *gamma* provides a correction to account for the quadratic approximation to the loss that is used by gradient boosting. We thus arrive at the following definition:

Definition 3. *The GBT proximity of a pair of observations is a weighted average of the number of trees in the GBT model in which the observations end up in the same terminal node:*

$$\begin{aligned} &\mathsf{proximity}^{\text{GBT}}(s, s') \\ = \quad & \sum_{i=1}^{t} \sum_{j=1}^{\tau_i} \frac{w_i}{\sum_{i=1}^{t} w_i} \mathbb{I}(s \in R_{j,i}) \mathbb{I}(s' \in R_{j,i}), \end{aligned}$$

where the ith tree's weight w_i is

$$w_i = \gamma_i^2 \cdot \text{Var}\{c_i^{\text{Tree}}(s) : s \in S\}$$

The GBT distance between a pair of observations is then:

$$d^{\mathrm{GBT}}(s, s') = 1 - \mathsf{proximity}^{\mathrm{GBT}}(s, s').$$

While other choices of measures of the size of predictions in each tree can be made, e.g. using the L^1 norm, we do not pursue them here and focus instead on demonstrating the properties of the chosen distance function.

3.3.2 Adaptive Greedy Submodular Prototype Selection

Algorithm 1: Adaptive greedy submodular prototypes

Input: Set of points S, distance function $d : S^2 \to [0, 1]$, class assignment $c : S \to [q]$

Output: Set of prototypes M, $|M| = k$

1 Create set of phantom examples $P = \{p_1, \ldots, p_q\}$ and set $d(p_i, s) = d(s, p_i) = 1$ for all s

2 $M \leftarrow \varnothing$

3 **for** *i=1* **to** k **do**

4 $\quad s^* \leftarrow \arg\max_{s \in S} \left[f(P) - f(P \cup M \cup \{s\}) \right]$

5 $\quad M \leftarrow M \cup \{s^*\}$

Our goal is to find a good approximately optimal solution for the q-classwise k-medoids problem (3.2). We will achieve this by using a greedy algorithm on an appropriate non-negative, monotone, submodular function (Prop. 1). However, the function (3.2) itself is not monotone submodular: in fact, adding more prototypes to M *decreases* the value of $f(M)$. This can be avoided by negating f,

but then the function will take non-positive values. Therefore, adapting an idea of [55], we will define a related function g as

$$g(M) = f(P) - f(P \cup M), \tag{3.3}$$

where P is an appropriately chosen set of *phantom exemplars*, one from each class. The resulting algorithm is listed as Algorithm 1. We want to derive a guarantee on the approximation of this algorithm using Proposition 1; to that end, we first need to show that g satisfies the necessary conditions.

Lemma 1. *The objective function (3.3) is non-negative, monotone and submodular.*

Proof. Observe that whenever $X \subseteq Y$, we have $f(X) \geq f(Y)$, since adding more points to a set can only make the closest point to a given point closer. From this, monotonicity and non-negativity is immediate, since $f(P) \geq f(P \cup M)$.

To establish submodularity, we will show that the function f of (3.2) satisfies

$$f(Y) - f(Y \cup \{t\}) \leq f(X) - f(X \cup \{t\})$$

whenever $X \subseteq Y \subseteq S$. The inequality of definition 2 then follows for g by plugging into its definition (3.3).

For any point $s \in S$, define $p_M(s)$ to be the closest point to s in M of the same class, that is,

$$p_M(s) = \underset{m \in M : c(m) = c(s)}{\arg\min} \; d(s, m).$$

Then we can rewrite $f(M)$ as

$$\sum_{s \in S} d(s, p_M(s)),$$

and it suffices to show that

$$\begin{aligned} & d(s, p_Y(s)) - d(s, p_{Y \cup \{t\}}(s)) \\ \leq \; & d(s, p_X(s)) - d(s, p_{X \cup \{t\}}(s)). \end{aligned}$$

for all $s \in S$. Both sides of this inequality are non-negative (+), since adding points can only shorten the distance to the closest point. Suppose $p_{Y \cup \{t\}}(s) \in Y$. Then it must be equal to $p_Y(s)$, since the closest point is present in Y, and so the first line is 0, and the inequality follows from (+).

Suppose instead $p_{Y \cup \{t\}}(s) \notin Y$. Then it must be t. So $p_{X \cup \{t\}}(s) = t$ as well (as $X \subseteq Y$), and the inequality reduces to $d(s, p_Y(s)) \leq d(s, p_X(s))$. But this is immediate, since $Y \supseteq X$ and adding more points can only shorten the distance to the closest point. □

By selecting the set of phantom exemplars P in such a fashion that $d(p, s) \geq d(s', s)$ for all $p \in P$ and $s, s' \in S$, we ensure that $f(T \cup P) = f(T)$ for all nonempty sets $T \subseteq S$. Hence, the set T_i^* that maximizes g among all sets of size i also minimizes f among all such sets.

Let T_i be the result of running the greedy maximization algorithm on (3.3) for i steps, and f be the original objective function (3.2). Then by Prop. 1 and choice of P,

$$f(T_i) \leq f(P) + (1 - 1/e)(f(T_i^*) - f(P)),$$

i.e. the approximation T_i takes us $1 - 1/e$ of the way from $f(P)$ to the optimum. Crucially, this means that the approximation guarantee depends on $f(P)$, i.e. how good the phantom exemplars alone would be as a solution to the k-medoids problem.

3.3.3 Supervised Greedy Prototype Selection

Instead of optimizing the k-medoids value function f of equation (3.2), we can instead directly pick prototypes, in a greedy fashion, that yield the best (train or validation set) improvement in a classification accuracy metric. From Table 3.1, the resulting algorithm beats the unsupervised k-medoids-derived approach in terms of accuracy in several cases, but we do not know of any theoretical guarantees that it satisfies, as these accuracy metrics are not submodular. A listing of this algorithm is given as Algorithm 2.

Algorithm 2: Supervised Greedy Prototypes

Input: Set of points S, distance function $d : S^2 \to [0, 1]$, class assignment $c : S \to [q]$

Output: Set of prototypes M, $|M| = k$

1 $M \leftarrow \varnothing$

2 **for** i=1 **to** k **do**

3 $\quad s^* \leftarrow \arg\max_{s \in S} [\text{balancedAccuracy}(S, M \cup \{s\})]$

4 $\quad M \leftarrow M \cup \{s^*\}$

3.4 Related Work

Tree Ensemble Distance. While not mentioned in the original random forest paper [19], Breiman defined the proximity matrix of a RF model in the documentation accompanying his implementation of random forests [20]. RF proximity has found a variety of applications, including clustering [124], outlier detection [156], and multiple imputation to handle missing data [123, 130]. Less is known

about the theoretical properties of the RF proximity matrix. The proximity between two observations can be expressed as a kernel [52, 97], and it is common to take a function of 1 - proximity as RF distance [155, 149, 124], as we do in this paper.

Prototype Selection. There is a long line of literature proposing prototype selection methods, also known as instance reduction, data summarization, exemplar extraction, etc. We point the reader to the taxonomy and review by Garcia et al. [50], who suggest that prototype selection methods can be grouped into three categories: condensation [61] , edition [146], or hybrid methods that remove both noisy and redundant points from the prototype selection set. We briefly mention a few methods: a classic prototype selection method is k-medoids clustering, for which different algorithms have been proposed, such as the PAM algorithm [75] and submodular approaches [91, 55, 103] like the one used in this paper. Prototype selection has also been cast as a set cover problem, where a minimum number of prototypes are selected to maximally cover the remaining observations [112, 17]. Recently, Kim et al. proposed a method based on maximum mean discrepancy between observations to select prototypes [78], a method later generalized by Khanna et al. to explain model predictions [77].

Implementations. While not many RF implementations provide the proximity matrix or prototypes, the exception is the R `randomForest` package [90] and RAFT, a random forest visualization tool by Cutler and Breiman [20]. In their documentation, they described a prototype-finding procedure that is partially implemented in the R `randomForest` package. For each class, the procedure selects the observation with most of its l neighbors being of the same class, then generates a prototype from the median feature values of its l neighbors.

The procedure can be sensitive to the choice of l, a distinct parameter from k, the number of prototypes desired. Since in this paper, we focus on selecting prototypes from existing observations not generating new observations, plus the implementation currently only generates one prototype[1], we do not include this method in our comparison but mention it for completeness.

3.5 Experiments and Analysis

In this section, we report the results of experiments to analyze the tree ensemble distances and proposed prototype selection methods.

3.5.1 Experimental Setup

Datasets

We selected five classification datasets, four with tabular data and one consisting of images – MNIST, for which we use digits 3 and 5, two commonly confused classes[2]. The datasets had different levels of label imbalance.

Training Procedure

For all datasets, RF models with 1000 trees were trained using Python's `scikit-learn` package. The maximum tree depth was not restricted. Using

[1]The package documentation suggests that the method may be updated to generate more prototypes. See https://cran.r-project.org/web/packages/randomForest/randomForest.pdf

[2]https://ml4a.github.io/demos/confusion_mnist/

the validation set, we cross-validate the number of features to consider when looking for the best split, considering "sqrt", "auto", "0.33", "0.5", "0.7", and "7" as options, and take the option that minimizes validation loss. We similarly train GBT models with `scikit-learn`. We modified the code to train GBTs with one gamma multiplier per tree, as opposed to one gamma multiplier per terminal node[3]. We set the number of trees to 200, and cross-validated the maximum depth (between 3 and 5) and the learning rate (0.1 or 0.01). We kept only the first t trees, where t was cross-validated to minimize validation loss.

Metric

Because the datasets used in this paper are imbalanced, we use balanced accuracy as our primary metric of classification performance. Note that we do not use ranking metrics such as AUC because nearest-neighbor classifiers do not output scores. Balanced accuracy is defined as:

$$\begin{aligned} &\mathsf{balancedAccuracy}(S, M) \\ &= \frac{1}{q} \sum_{j \in [q]} \frac{\#\left\{ s \in S_j : c\left(\arg\min_{m \in M} d(s, m) \right) = j \right\}}{|S_j|}, \end{aligned}$$

where S_j is the set of those points $s \in S$ in class j.

3.5.2 Evaluating Prototypes: Nearest-Prototype Classifier

One way to validate that the selected prototypes are reasonable is to use them in in a nearest-*prototype* classifier [17]. This evaluation is in line with recent ideas

[3]Despite one gamma per tree being a standard formulation for GBTs, it is not available in `scikit-learn`.

on evaluating explanations by checking their accuracy or fidelity on independent test-data [114, 99, 133]. We use a 1-nearest-prototype classifier to classify test-set data, and ask the following questions:

- Does a supervised distance derived from the tree ensemble models, improve over an unsupervised distance on feature values?
- Does a prototype selection method improve over using every possible point as a prototype?

Figure 3.1 illustrates test-set balanced accuracy as a function of the number of prototypes, for different data sets. To answer the first question, we examine the RF and GBT plots (left and center) compared to EUCLIDEAN plots (right) in Figure 3.1. A fair comparison would be to hold the prototype selection method constant (i.e. look at the same colored line across left to right). RF distance performed better than the unsupervised Euclidean distance; GBT distance had more variable performance, sometimes performing better than RF distance, sometimes performing worse than Euclidean distance. We defer an analysis of the difference between the RF and GBT distances to the next section.

To answer the second question, we compare the performance of the three prototype selection methods to the 1-NN method that does not select prototypes, and instead uses all (training set) points as prototypes. A fair comparison would be to hold the distance constant (i.e. look at the different colored lines in the same plot). We found that the accuracy of the prototype selection method varies according to the number of prototypes desired, k. This was particularly noticeable for the k-medoids based methods, which are not supervised, compared to greedily optimizing for balanced accuracy. This suggests that k should

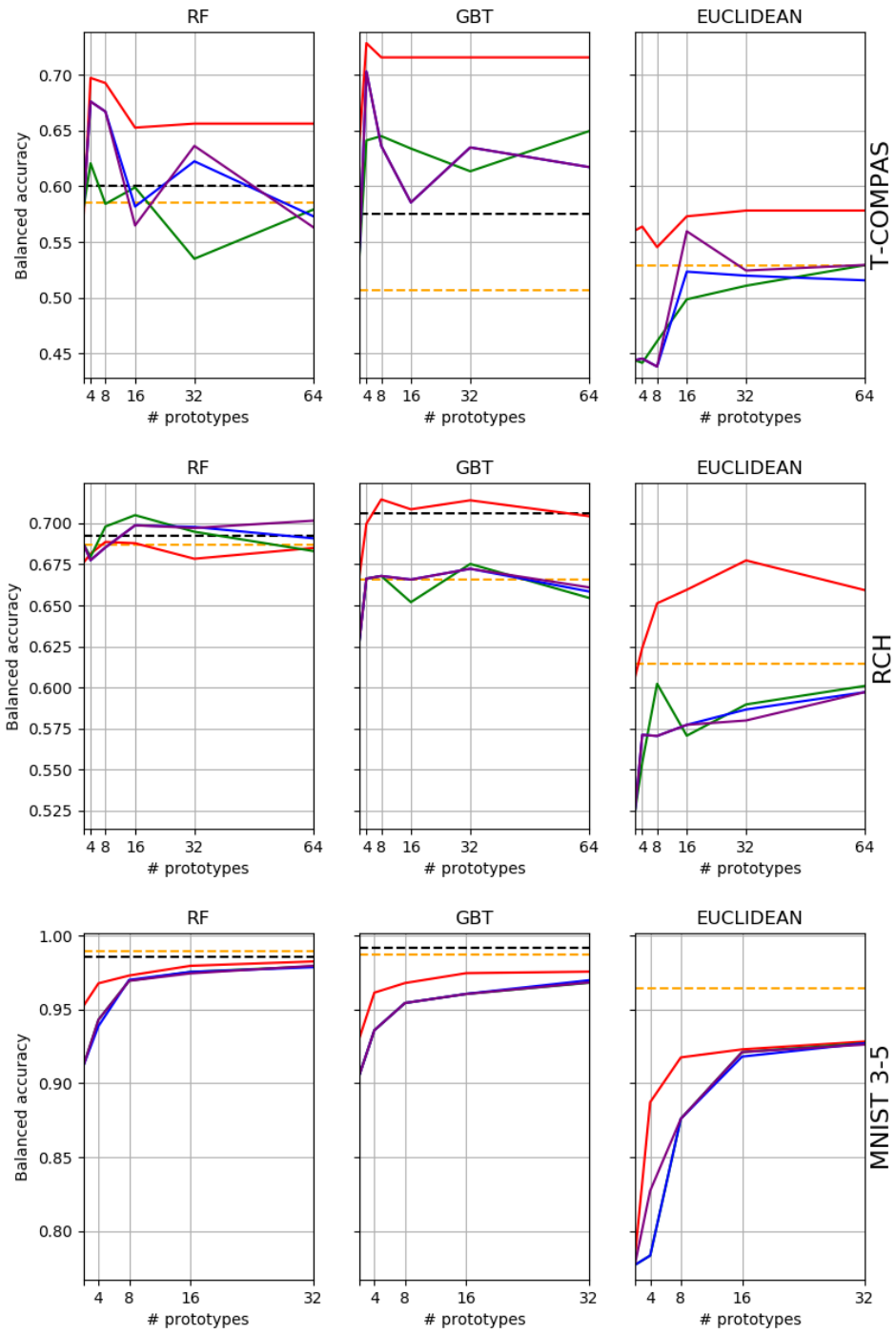

Figure 3.1: Test-set balanced accuracy as a function of k, the number of prototypes. The black dashed line represents the original model, and the dashed orange line represents the 1-NN baseline using all training points (in other words, treating all training points as prototypes). Different prototype selection methods were used: supervised greedy (SG), uniform greedy submodular (SM-U), adaptive greedy submodular (SM-A) (ours), weighted adaptive greedy submodular (SM-WA) (ours).

be tuned separately for each prototype selection method and is in line with the results from [17] where the optimal number of prototypes was found to vary significantly by selection method (from example, from 5 to 194 prototypes on the Diabetes data).

After we validated the optimal k separately for each prototype selection method, the differences in performance of different prototype selection methods became more evident. Table 3.1 presents the results. For all datasets, at least one prototype selection method outperformed 1-NN, suggesting the value of prototype selection for accuracy, besides interpretability (reducing the number of observations that need to be shown to a user). SM-WA is competitive against SM-U. Despite the lack of theoretical guarantees, SG had clear advantages on a number of datasets, demonstrating the value of supervision.

3.5.3 Analysis: Tree Ensemble Distance by Tree Depth

We now study the learned tree ensemble distances to gain some intuition into their behavior. Figure 3.2 illustrates the distribution of distances for RF and GBT distances compared to Euclidean distance on one of the datasets, Breastcancer. Since, unlike RF and GBT distances, it is not supervised distance, Euclidean distance has no preference for the extremes of the distribution.

The distance derived from RF models is more "granular", with less "clumping", than the distance derived from GBT models. Most default implementations of RF algorithms allow trees to grow to unrestricted depth [19], hence on the same data set, trees in RF models tend to be deeper than trees in GBT models. This can be confirmed in Table 3.2, which presents statistics about tree depth

Model		Breastcancer	Diabetes	T-COMPAS	RHC	MNIST 3-5	CALTECH256 G-M
RF	Original	0.90	0.74	0.60	0.69	0.99	0.95
	1-NN	0.90 (341)	0.70 (460)	0.59 (600)	0.69 (3441)	0.99 (8072)	0.97 (129)
	SG	0.87 (4)	0.74 (16)	0.64 (32)	0.68 (64)	0.98 (32)	0.95 (4)
	SM-U	0.91 (64)	0.74 (64)	0.57 (64)	0.69 (64)	0.98 (32)	0.97 (64)
	SM-A	0.92 (32)	0.73 (64)	0.58 (64)	0.69 (64)	0.98 (32)	0.97 (64)
	SM-WA	0.92 (64)	0.77 (64)	0.56 (64)	0.70 (64)	0.98 (32)	0.97 (64)
GBT	Original	0.90	0.71	0.58	0.71	0.99	0.91
	1-NN	0.89 (341)	0.69 (460)	0.51 (600)	0.67 (3441)	0.99 (8072)	0.91 (129)
	SG	0.91 (8)	0.74 (32)	0.73 (4)	0.70 (64)	0.98 (32)	0.91 (4)
	SM-U	0.89 (64)	0.63 (64)	0.68 (4)	0.67 (8)	0.97 (32)	0.91 (64)
	SM-A	0.89 (64)	0.65 (64)	0.54 (64)	0.67 (4)	0.97 (32)	0.88 (32)
	SM-WA	0.89 (64)	0.63 (64)	0.68 (4)	0.67 (8)	0.97 (32)	0.91 (64)
EUCLIDEAN	1-NN	0.93 (341)	0.63 (460)	0.53 (600)	0.61 (3441)	0.96 (8072)	0.90 (129)
	SG	0.92 (16)	0.71 (32)	0.58 (32)	0.66 (64)	0.93 (32)	0.86 (16)
	SM-U	0.92 (8)	0.64 (64)	0.52 (64)	0.59 (32)	0.93 (32)	0.88 (64)
	SM-A	0.92 (8)	0.64 (64)	0.53 (64)	0.60 (64)	0.93 (32)	0.88 (64)
	SM-WA	0.93 (8)	0.64 (64)	0.53 (64)	0.58 (32)	0.93 (32)	0.82 (64)

Table 3.1: Best number of prototypes (according to train loss / accuracy) and test balanced accuracy for different prototype selection methods: supervised greedy (SG), uniform greedy submodular (SM-U), adaptive greedy submodular (SM-A) (ours), weighted adaptive greedy submodular (SM-WA) (ours), against the 1-NN baseline treating all training points as prototypes.

Dataset	n	GBT Depth	RF Depth Min	Mean	Max	Var
Breastcancer	569	4	2	3.4	5	0.59
Diabetes	768	3	5	7.0	10	0.9
T-COMPAS	1000	3	6	8.5	12	1.12
RHC	5736	3	11	14.7	21	1.41
MNIST 3-5	13454	4	12	16.3	23	1.98

Table 3.2: Statistics of RF and GBT models tree depth across different datasets. n is the number of observations in the dataset. All RF models had 1000 trees. All GBT trees had an optimal number of trees (based on validation set loss) less than or equal to 200.

in RF and GBT models. Deeper trees have more terminal nodes. Hence, the deeper the tree, the lower the probability of a pair of observations ending up in the same terminal node, a possible explanation for the more granular behavior of RF distance compared to GBT.

From Table 3.2, we also see that not so surprisingly, the larger the data set, the deeper the RF trees, whereas the GBT trees have been restricted to depth 3 to 5.

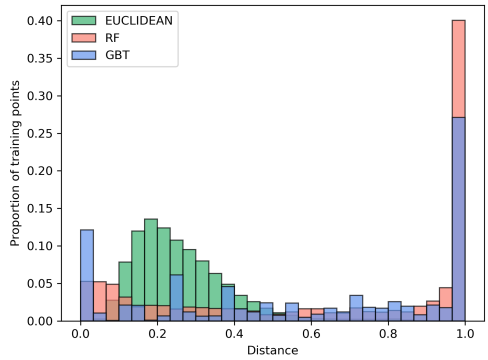

Figure 3.2: Distribution of RF and GBT distance compared to Euclidean distance on one of the datasets, Breastcancer.

3.5.4 Visualizing Tree Ensemble Distance

As pointed out in the previous section, GBT distance can be quite different from RF distance. Figure 3.3 uses the t-sne method [101] to visualize distances derived from RF and GBT classifiers trained on MNIST. While the two classes – the digits 3 and 5 – are clearly separable from Figure 3.3, agreeing with the good performance of the RF and GBT classifiers at more than 98% balanced accuracy (cf. Figure 3.1), both tree ensemble models appear to be learning different representations, with the GBT model grouping points together in smaller and more compact clusters than the RF model. Moreover, on this problem, the prototypes selected for RF compared to GBT are different.

When the classifier's performance is not as good, the separation between classes is less clear, as can be seen in Figure 3.4, which visualizes the distances derived from RF and GBT classifiers trained on the Turker-COMPAS dataset. Here, RF has 0.60 balanced accuracy and GBT has 0.58 balanced accuracy (cf. Table 3.1). Similar to before, the GBT distance groups points together in smaller and more compact clusters than the RF model. For this dataset, there is more overlap in prototypes selected by RF and GBT distances.

A natural next question may be the following: to what degree are differences between the GBT and RF distances caused by different tree depth, different weights used in constructing the distance, or that different patterns in the data are being learned by RF compared to GBT models? While the top right corner of Figures 3.3 and 3.4 visualizes distances from GBT models trained to default settings (short), and the bottom right corner of the same figures visualizes distances from RF models trained to default settings (unrestricted depth), the bottom left corner visualizes RF models trained *to the same depth* as the cor-

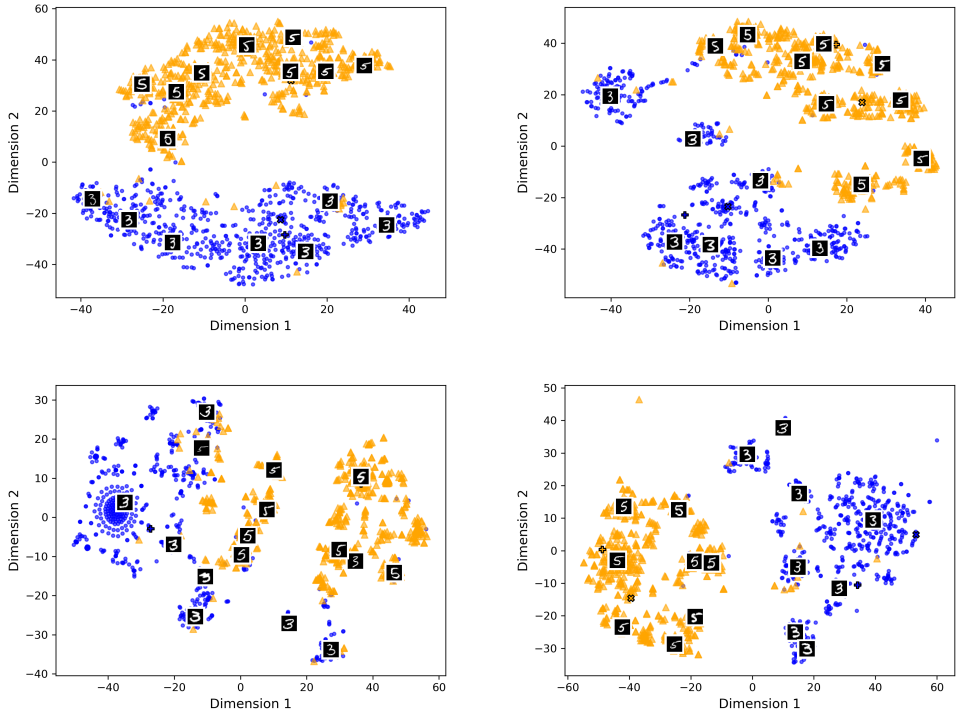

Figure 3.3: Visualization of dissimilarities using t-sne [101] for optimal GBT with unweighted trees (top left), trees weighted by vg^2 (top right), RF with short trees matching GBT depth of 3 (bottom left), and optimal RF (bottom right) with mean depth 16 on the MNIST (3 vs 5) dataset, using the adaptive greedy submodular (SM-A) method. Red represents the digit 5, blue represents the digit 3.

responding default GBT model on that dataset. While the short RF model has smaller and more compact clusters than the default RF model, the RF and GBT models of same depth can still be told apart.

The top left corner visualizes an unweighted distance function derived from the same GBT model as in the top right corner. Note that the visualization in the top right corner is of a weighted distance function. While the top left corner in Figure 3.3 for MNIST looks more similar to the bottom right corner (default RF model), in Figure 3.4 for the Turker-COMPAS data this is not the case.

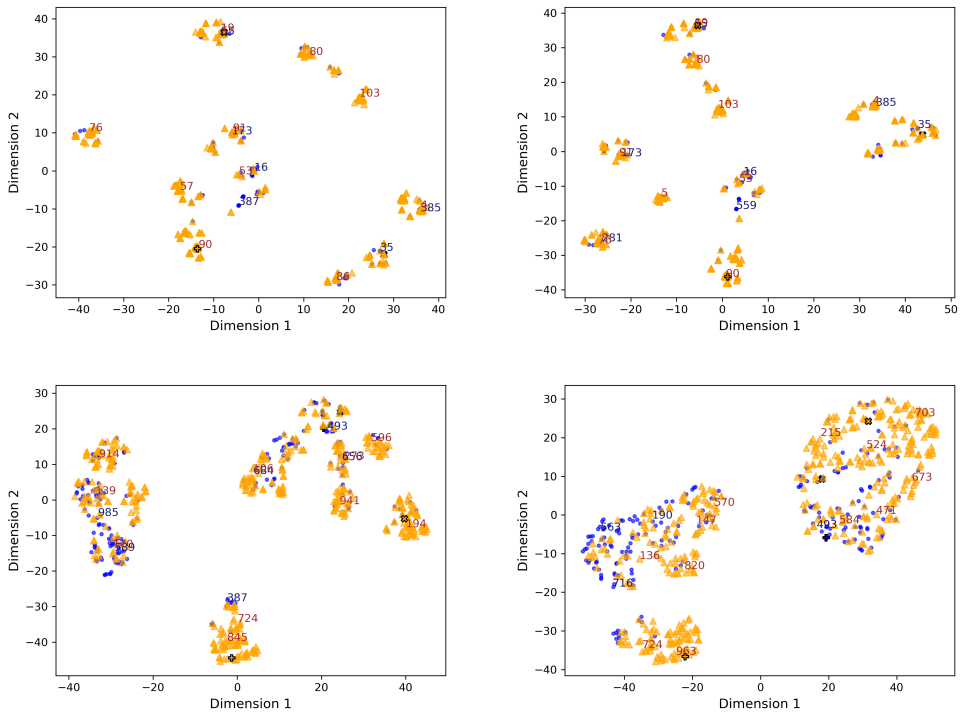

Figure 3.4: Visualization of dissimilarities using t-sne [101] for optimal GBT with unweighted trees (top left), trees weighted by vg^2 (top right), RF with short trees matching GBT depth of 3 (bottom left), and optimal RF (bottom right) with depth 8. Different colors represent different classes, and prototypes are marked by their indices.

3.5.5 Comparing Prototype Selection Methods

Figure 3.5 displays the prototypes found by the adaptive greedy submodular k-medoids prototype selection method (SM-A) compared to the prototypes found by the supervised greedy method (SG) on MNIST. We see that the prototypes selected by SM-A (left part of Figure 3.5) have good coverage of the space of observations, which is not the case for SG (right part of Figure 3.5).

Consider the case of a single class 0 point deep in class 1 territory, with all other class 0 points are already well-covered. As k-medoids (both uniform and

adaptive flavors) does not consider distance to other classes or labels, k-medoids would still want to turn that class 0 point into a prototype, even though that would mean misclassifying the class 1 points near to it. In addition, even after such a class 0 prototype has been placed, hence misclassifying the class 1 points near to it, k-medoids, not being supervised, would not be aware of the sudden dip in classification accuracy, to correct for the error in subsequent prototypes.

Figure 3.6 presents the prototypes found by the different methods when using RF distances on the CALTECH256 G-M data. First, even if we requested 16 prototypes, the supervised greedy algorithm found that only 3 prototypes were necessary. These include canonical views of popular guitar models and a frontal picture of a mandolin. These do not cover all representative guitars or mandolins, but are good enough to correctly classify most images. The submodular methods select prototypes that ensure coverage, surfacing different poses and shapes that are in many cases different from the archetypal ones. Interestingly, both adaptive methods tend to select more mandolins than guitars, including close-ups and people, not found in the guitar prototypes. Hence, the different prototype methods selects different types of prototypes depending on whether the objective function is optimizing for coverage or accuracy, which brings us back to the question of what is a prototypical observation – we suspect this depends on the setting in which the prototypes are to be used.

3.6 Discussion

Of the four prototype selection methods used in this paper, three – supervised greedy, adaptive greedy submodular, weighted adaptive greedy submodular –

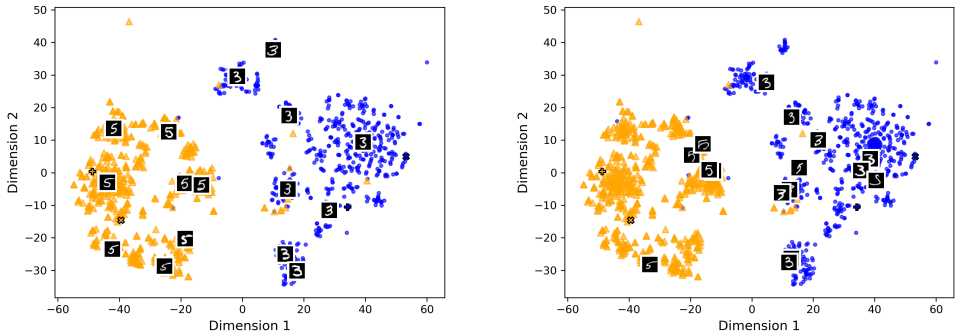

Figure 3.5: Visualization of dissimilarities using t-sne [101] for optimal RF with mean depth 16 on the MNIST (3 vs 5) dataset, using the adaptive greedy submodular (SM-A) prototype selection method (left) and supervised greedy (SG) (right) method. The SM-A figure on the left already appeared in the bottom right of Figure 3.3.

are able to select different number of prototypes for each class. Unlike uniform greedy submodular which returns $\mathsf{round}(k/q)$ prototypes per class regardless of class imbalance or distance behavior, these three methods pick the next prototype from the class that needs it the most, as measured by the objective function.

The ability to pick different number of prototypes by class could be advantageous for interpretability purposes. We posit the following question:

> A physician is trying to take aid from a black-box RF or GBT model to diagnose a disease. The disease is a rare disease, hence there is class imbalance. The majority of patients will not have this disease, however the patients who do have this disease are not easily characterized. If the physician only has a budget of k prototypes that she can inspect before making a decision for this patient (as many other patients are waiting to be seen), might surfacing several prototypical (and varied) patients with this disease be a better use of the

Figure 3.6: 16 CALTECH256 G-M prototypes found using RF distances for the studied prototype methods. First row: supervised greedy (SG) (first sub-row: guitar, second sub-row: mandolin). Second row: uniform greedy submodular (SM-U). Third row: adaptive greedy submodular (SM-A). Last row: weighted adaptive greedy submodular (SM-WA).

physicians' "attention" budget, rather than seeing an equal number of prototypical patients with and without the disease?

In conclusion, in this paper, we proposed a new approach for interpreting tree ensembles, using an existing distance defined for RF models and extending the idea to GBT models. We also proposed two new prototype selection methods to find prototypical observations, as "seen" from the point of view of the tree ensemble. Our ultimate goal is to rank observations and surface prototypical ones in a meaningful order to domain experts, stopping when their "attention" budget is exhausted. Hence, we pursued submodular approaches where adding the next prototype has diminishing returns and does not hurt the existing selected prototypes. An alternative to unsupervised submodular prototype selection methods was presented by greedy supervised prototype selection methods that performed well in several settings.

CHAPTER 4

DISTILL-AND-COMPARE: AUDITING BLACK-BOX MODELS USING TRANSPARENT MODEL DISTILLATION

4.1 Introduction

Risk scoring models have a long history of usage in criminal justice, finance, hiring, and other critical domains [31, 98]. They are designed to predict a future outcome, for example defaulting on a loan. Worryingly, risk scoring models are increasingly used for high-stakes decisions, yet are typically proprietary or opaque.

Several approaches have been proposed [64, 41, 2, 1, 33, 142] to audit black-box risk scoring models: remove, permute, or obscure a protected feature, then see how the the model's predictions change after retraining the model or probing the model API with the transformed data. However, creators of proprietary risk scoring models often do not provide unrestricted access to model APIs, much less release the model form or training data. Moreover, approaches that focus on one or two protected features defined in advance are less likely to detect biases that are not *a priori* known.

In this paper, we study a more realistic setting where we only have a data set labeled with the risk score (as produced by the risk scoring model), the ground-truth outcome, and some or all features; we are not able to probe the model API

This chapter is based on material in [134].

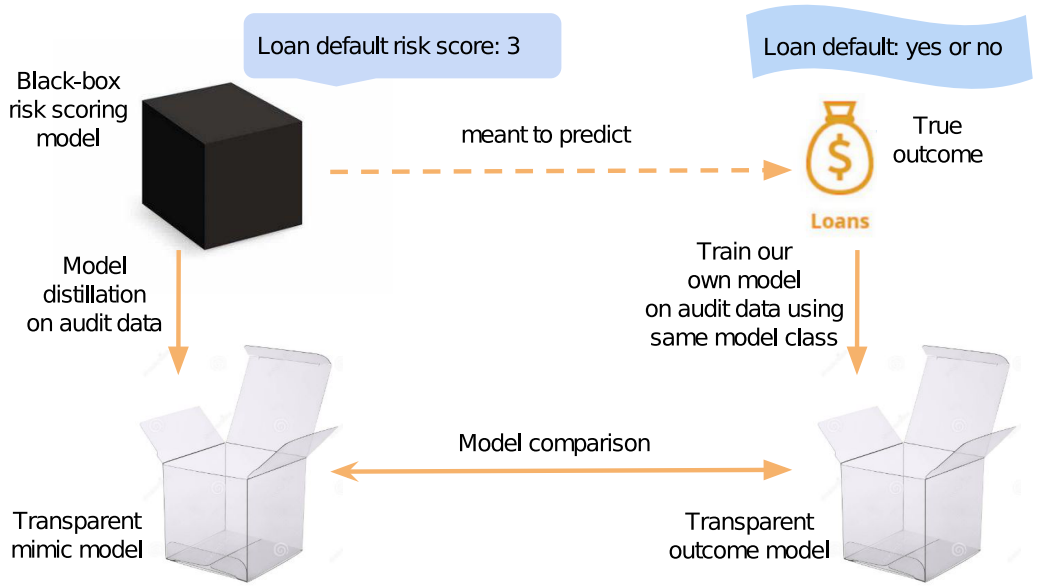

Figure 4.1: Distill-and-Compare audit approach on a loan risk scoring model.

with new data. We call this data set the *audit data*. We add two potential complications: the audit data may not be the original training data, and the audit data may not have all features used to train the risk scoring model. For example, ProPublica obtained data for their COMPAS study [8] not from the company that created COMPAS, but through a public records request to Broward County (BC), a US jurisdiction that used COMPAS in their criminal justice system [7]. ProPublica may not have had access to all the features BC used for COMPAS.

We propose Distill-and-Compare, an approach to audit black-box risk scoring models using audit data with both black-box risk scores and ground-truth outcomes, without pre-defining feature regions to audit. First, we train a model on the audit data to mimic the black-box model. Then we train another model to predict outcomes (Section 4.2.1). To gain insight into the black-box model, we uncover feature regions where the two models are significantly different (Section 4.2.3), and ask "what could be happening in the black-box model, that

could explain the differences we are seeing between the mimic and outcome models?". Finally, we use a statistical test (Section 4.2.2) to determine if the black-box model used additional features we do not have access to (i.e. features not in the audit data).

The contributions of this paper are: 1) We propose an approach to audit black-box risk scoring models under realistic conditions. 2) We show the importance of calibrating risk scores to remove audit data shift or scale post-processing that may been introduced by creators of risk scoring models. 3) We propose a statistical test to determine if the audit data is missing key features used to train the black-box model. 4) We apply the approach to audit four risk scoring models. 5) An ancillary contribution of this paper is a new confidence interval estimate for iGAM[1], a type of transparent model.

4.2 Audit Approach

Our goal is to gain insight into a black-box risk scoring model. We draw from model distillation and comparison technique to develop our approach. Section 4.2.1 discusses related work.

[1]iGAM was an implementation of GA^2M, a type of interpretable model introduced in [95, 96, 24]. iGAM was recently renamed Explainable Boosting Machine (EBM) and can be found at `https://github.com/microsoft/interpret`. Since the paper that this chapter is based on was published after the renaming of iGAM to EBM, we retain the name iGAM throughout this chapter.

4.2.1 Distill-and-Compare

Model distillation was first introduced to transfer knowledge from a large, complex model (teacher) to a faster, simpler model (student) [22, 65, 11]. This was done by running unlabeled samples (either new unlabeled data or training data with labels discarded) through the teacher model to obtain the teacher's outputs, then training the student model to mimic the teacher's outputs. We draw parallels to our setting, taking the risk scoring model to be the teacher and the audit data to be unlabeled samples ran through the teacher (risk scoring model) to obtain the teacher's output (risk scores). We train the mimic model to minimize mean squared error between the teacher and student, i.e.,

$$L(S, \hat{S}) = \frac{1}{T} \sum_{t=1}^{T} \left(S(x^t) - \hat{S}(x^t) \right)^2, \tag{4.1}$$

where x^t is the t-th sample in the audit data, $S(x^t)$ is the output of the teacher model (risk scores) for sample x^t, $\hat{S}(x^t)$ is the output of the mimic model for sample x^t, and T is the number of samples. Throughout this paper, we will call the teacher model the *black-box model* and the student model the *mimic model*.

Next, we leverage the ground-truth outcome information. We train *our own risk scoring model* on the audit data to predict the ground-truth outcome, i.e.,

$$\begin{aligned} L(O, \hat{O}) = \frac{1}{T} \sum_{t=1}^{T} \Big\{ & O(x^t) \log \left(P(\hat{O}(x^t) = 1) \right) + \\ & (1 - O(x^t)) \log \left(P(\hat{O}(x^t) = 0) \right) \Big\}, \end{aligned} \tag{4.2}$$

where $O(x^t) \in \{0, 1\}$ is the ground-truth outcome for sample x^t and $\hat{O}(x^t) \in \{0, 1\}$ is the output of the model for sample x^t. Throughout this paper, we call this

model the *outcome model*. Note that the outcome model is not a mimic model.

It is critical that both the mimic model and outcome model are trained using the same model class that allows for interpretation and comparison. Not all model classes have the property that two models of that class can be compared. For example, it is not obvious how to compare two decision trees, random forests or neural nets. We want a model class that is as rich and complex as possible so that the mimic model can be faithful to the black-box model and the outcome model can accurately predict ground-truth outcomes. However, this model class should still be transparent [37] so that we can examine its predictions across different feature regions. In this paper, we use a particular transparent model class, iGAM (Section 4.2.3); other choices are possible.

The risk score and the ground-truth outcome are closely related—the ground-truth outcome is what the black-box risk scoring model was meant to predict. If the black-box model is accurate *and* generalizes to the audit data, it would predict the ground-truth outcomes in the audit data correctly; the converse is true if the black-box model is not accurate *or* does not generalize to the audit data.

Because both the mimic and outcome models are trained with the same model class on the same audit data using the same features, the more faithful the mimic model, and the more accurate the outcome model, the more likely it is that observed differences between the mimic and outcome models stem from differences between the black-box model and ground-truth outcomes. This allows us to ask, "what could be happening in the black-box model, that could explain the differences we are seeing between the mimic and outcome models?". In addition, similarities between the mimic and outcome models (e.g., on

COMPAS in Section 4.3.2, the Number of Priors feature is modeled very similarly by the two models) increases confidence that the mimic model is a faithful representation of the black-box model, and that any differences observed on other features are meaningful.

Related Work

Several auditing approaches also use model distillation techniques to distill black-box models when they cannot be queried or to understand them [1, 2]. Other approaches also train their own outcome models, then uncover feature regions where the model is not accurate [154, 80, 3, 76]. Kim et al.'s iterative procedure [80] not only uncovers such regions but also modifies the model to improve accuracy in these regions. However, they require repeated calls to the model; Agarwal et al. [3] and Kearns et al. [76] similarly require repeated calls or knowledge of the model. Tramer et al. uncovered unexplained associations between black-box outputs and protected features on audit data [139].

Our approach is different from the above, as we avoid repeated calls to the black-box model API (that may not realistically be available), and instead utilize information on both risk scores and outcomes already available in some data sets in this domain (e.g. ProPublica COMPAS data). Some other approaches also compare two models, but not risk scores and outcomes at the same time. Wang et al. trained a model to predict outcomes and another to predict membership in a protected feature region [145]. Chouldechova and G'Sell trained two different outcome models then identified feature regions where the two models differed [28].

4.2.2 Testing for Missing Features

If the audit data is missing features used by the black-box model, the audit data alone may be insufficient to audit the black-box model. We propose a statistical test to check the likelihood of the audit data missing important features based on the following observation:

> *If the black-box model used features that are missing from the audit data but are useful for predicting the ground-truth outcome, the error between the mimic model (learned on the audit data) and the risk score, $||\hat{S} - S||_E$, should be positively correlated with the error between the outcome model (learned on the audit data) and ground-truth outcome, $||\hat{O} - O||_E$.*

where E is an error metric. Since the test uses predictions from both the mimic and outcome models, the test is performed after both models are trained. In Section 4.3.4, we perform the test on all risk scoring models we audit in this paper, to check if the audit results are significantly affected by missing features. Note that this test does not require the mimic and outcome models to be transparent.

4.2.3 Comparing Mimic and Outcome Models

In this section, we provide technical details on how to train the mimic and outcome models so that they are comparable.

Choice of model class

As noted in Section 4.2.1, we train the mimic model and outcome model using the same transparent model class—in this paper, iGAM [95, 96, 24]. We point the reader to [95, 96, 24] to learn more about iGAMs and to [133] for a distillation example where it was used as a student. Briefly, iGAM has the form

$$E[g(y)] = h_0 + \sum_i h_i(x_i) + \sum_{i \neq j} h_{ij}(x_i, x_j), \tag{4.3}$$

where g is the logistic function for classification and identity function for regression, h_0 is the intercept, and the contribution of any one feature x_i or pair of features x_i and x_j to the prediction can be visualized in graphs such as Figure 4.4 (with $h_i(x_i)$ on the y-axis) and Figure 4.7 (with regions colored by $h_{ij}(x_i, x_j)$). For classical GAMs [62], $h(\cdot)$ are fitted using splines; for iGAM, they are fitted using ensembles of shallow trees and centered for identifiability. Crucially, since iGAM is an additive model, two iGAM models can be compared by simply taking a difference of their feature contributions $h(\cdot)$, which we exploit in Section 4.2.3 to detect differences between the mimic and outcome models.

Calibrating model inputs

Calibration is the process of matching predicted and empirical probabilities [34, 107]. If a risk score is well-calibrated, the relationship between the risk score and empirical probabilities will be linear (e.g., COMPAS and Stop-and-Frisk in the top row of Figure 4.2). While developing the method, we discovered that not all risk scores exhibit the desired linear relationship with outcomes in the audit data. For example, the Chicago Police risk score (third column of Figure 4.2) is rather flat for risk scores less than 350, then exhibits a sharp kink upwards.

One possible explanation for any nonlinear relationship is that the risk score was well-calibrated on its original training data, but the audit data has a different distribution (data shift) [117]. Another possible explanation is post-processing by model creators to reduce sensitivity in less important parts of the risk score scale and enhance separation in more important parts of the scale [93].

We make the reasonable assumption that risk scores should be monotonic and well-calibrated [93] and use this assumption to undo scale post-processing or audit data shift before training the mimic and outcome models. Specifically, we learn a nonlinear transformation of the risk score (the blue line in Figure 4.2), similar to isotonic regression [107], to make the risk scores and outcomes linearly related on a scale of choice. The mimic model is then trained with the transformed risk scores as labels; the outcome model is trained with outcomes, unchanged.

This pre-training calibration step is necessary to compare the mimic and outcome models, as it makes their labels linearly related on a scale that their predicted labels will later be compared on. We select this scale to be logit probability (since the predicted outcomes are already on this scale), and perform this calibration step for Chicago Police and Lending Club but not COMPAS and Stop-and-Frisk, since the latter two already exhibit the desired linear relationships.

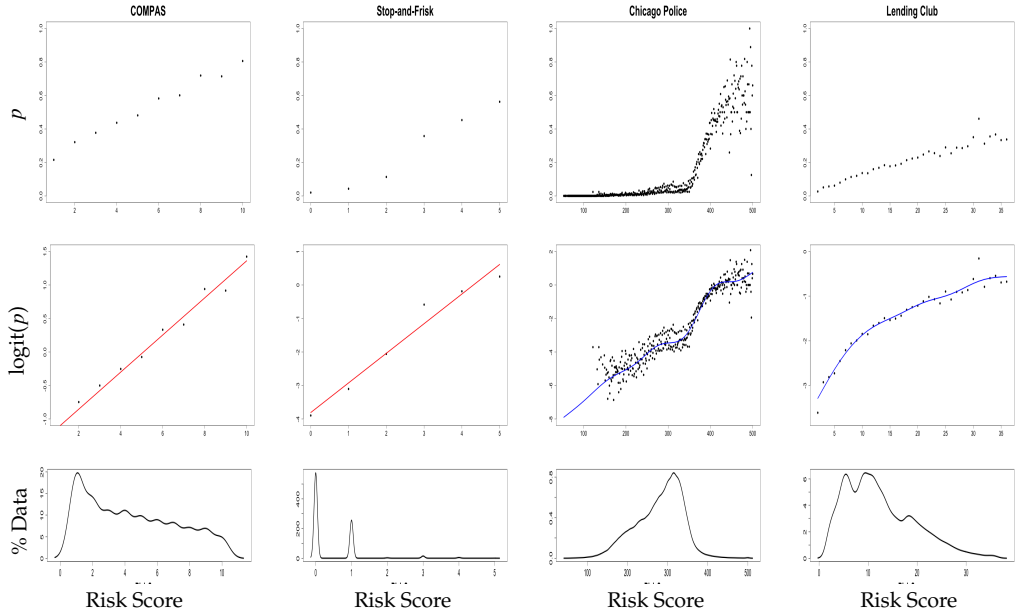

Figure 4.2: Empirical probability of positive outcomes (y-axis) vs. risk score (x-axis) for COMPAS, Stop-and-Frisk, Chicago Police, and Lending Club on probability scale (top row) and logit probability scale (middle row). The risk score distribution is in the bottom row. The red lines on the logit probability scale (middle row) are best-fit straight lines. A good fit (COMPAS and Stop-and-Frisk) suggests that the risk score and logit probability of outcomes (middle row) have a linear relationship. In this case, the mimic model can be trained directly on the raw risk score. When the relationship is not linear (Chicago Police and Lending Club), the risk score must be calibrated. The blue monotonic curves (middle row) are the nonlinear transformations learned during the calibration step. This transformation is applied to the raw risk score to yield the transformed risk score in Figure 4.3.

Detecting differences

To not mistake random noise for real differences between the mimic and outcome models, we control potential sources of noise during the training process. To avoid data sample-specific effects, we train the mimic and outcome models

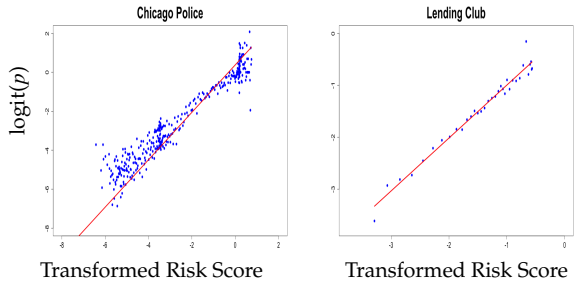

Figure 4.3: Logit empirical probability (y-axis) vs. transformed risk score (x-axis). The red lines are best-fit straight lines. A good fit suggests that the transformed risk score and logit probability of outcomes now have a linear relationship. The mimic model can now be trained on the transformed risk score.

on the same data sample. Let $sh_i(x_i)$ be feature x_i's contribution to the mimic model, and similarly $oh_i(x_i)$ for the outcome model. We calculate the difference in feature x_i's contribution to the two models, $sh_i(x_i) - oh_i(x_i)$, and construct a confidence interval for this difference to tell if it is statistically significant. One ancillary contribution of this paper is a new method to estimate confidence intervals for the iGAM model class, by employing a *bootstrap-of-little-bags* approach [122] to estimate the variance of $h_i(x_i)$ and $sh_i(x_i) - oh_i(x_i)$. See the next section for details. The resulting confidence intervals are the dotted lines in Figures 4.4–4.6.

A new confidence interval estimate for iGAM

It is not trivial to estimate confidence intervals for nonparametric learners such as trees [102]; iGAM's base learners are shallow trees. We employ a *bootstrap-of-little-bags* approach originally developed for bagged models in [122] to estimate the variance of feature x_i's contribution to the model, $h_i(x_i)$, and difference in feature x_i's contribution to the mimic and outcome models, $sh_i(x_i) - oh_i(x_i)$.

Bootstrap-of-little-bags exploits two-level structured cross-validation (e.g. 15% of data points are selected for the test set, with the remaining 85% split into training (70%) and validation (15%) sets). Repeating this inner splitting L times and outer splitting K times gives a total of KL bags on which we train the model. Let $h_i^{lk}(x_i)$ be feature x_i's contribution to the model in the lth inner and kth outer fold. The variance of $h_i(x_i)$ can then be estimated as

$$\widehat{\mathrm{Var}}(h_i(x_i)) = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{1}{L}\sum_{l=1}^{L} h_i^{kl}(x_i) - \frac{1}{KL}\sum_{l=1}^{l}\sum_{k=1}^{K} h_i^{kl}(x_i)\right)^2,$$

and its mean $\overline{h_i(x_i)}$ can be estimated by averaging $h_i^{lk}(x_i)$ over KL bags.

We can now construct pointwise confidence intervals (CI) for feature contributions to iGAM models. The 95% CI for feature x_i's contribution to the model, $h_i(x_i)$, is $\overline{h_i(x_i)} \pm 1.96\sqrt{\widehat{\mathrm{Var}}(h_i(x_i))}$ and the 95% CI for the difference in feature x_i's contribution to the mimic and outcome models, $sh_i(x_i) - oh_i(x_i)$, is $\overline{sh_i(x_i)} - \overline{oh_i(x_i)} \pm 1.96\sqrt{\widehat{\mathrm{Var}}(sh_i(x_i)) + \widehat{\mathrm{Var}}(oh_i(x_i)) - 2\widehat{\mathrm{Cov}}(sh_i(x_i), oh_i(x_i))}$, with $\widehat{\mathrm{Cov}}(sh_i(x_i), oh_i(x_i))$ also estimated using bootstrap-of-little-bags.

This variance estimate is conservative (meaning it overestimates true variability), however, given that we are trying to detect differences between the mimic and outcome models, overestimating means we are less likely to mistake random noise for real differences. For large K and L, consistency of this estimate was established in [9].

Note that are pointwise, not uniform, CIs. That is, using the feature Age as an example, these CIs capture the variability of the effect of Age at Age=50, not the entire effect of Age. Uniform CIs can be constructed by adjusting the critical value of the CI.

4.3 Results

4.3.1 Validating the Audit Approach

In this section, we demonstrate Distill-and-Compare on risk scoring models where we have some information on how they were trained, and check that the approach can recover this information.

Stop-and-Frisk.

Using the New York Police Department's Stop-and-Frisk[2] data, Goel et al. [54] proposed a simple risk scoring model for weapon possession: $S = 3 \times \mathbf{1}_{PS} + 1 \times \mathbf{1}_{AS} + 1 \times \mathbf{1}_{Bulge}$, where S is the risk score, PS denotes primary stop circumstance being presence of suspicious object, AS denotes secondary stop circumstance being sight of criminal activity, and $Bulge$ denotes bulge in clothing [54]. Since we know the risk scoring model's functional form, we can verify that the mimic model correctly recovers these coefficients. We apply the risk scoring model to label 2012 data (T=126,457, p=40) after following Goel et al.'s data preprocessing steps [54].

Result. The mimic model recovers the coefficients (3, 1, 1) for the three features used in the risk scoring model (PS, AS, $Bulge$) and 0 for the remaining features.

[2] http://www1.nyc.gov/site/nypd/stats/reports-analysis/stopfrisk.page

Chicago Police "Strategic Subject" List.

The Chicago Police Department released arrest data[3] from 2012 to 2016 that was used to create a risk score for an individual being involved in a shooting incident as a victim or offender. This data set contains 16 features, but only 8 are used by the black-box model, which gives us an opportunity to test if Distill-and-Compare can accurately detect which features are and are not used by a black-box model.

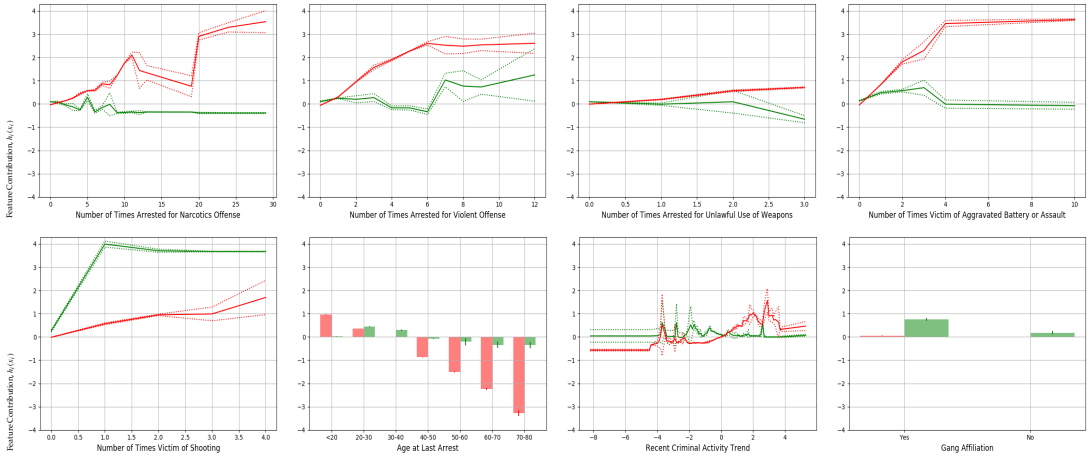

Figure 4.4: Eight features the Chicago Police says are used in their risk scoring model. The COMPAS mimic model is in red, the outcome model is in green.

We trained a mimic model, intentionally including all 16 features. Figure 4.4 shows the feature contributions of the mimic model (in red) and outcome model (in green) for the 8 features the Chicago Police says were used by the black-box model; Figure 4.5 shows the 8 features the Chicago Police says were *not used* in their model.

Result. There is a striking difference between Figures 4.4 and 4.5: the mimic

[3] https://data.cityofchicago.org/Public-Safety/Strategic-Subject-List/4aki-r3np

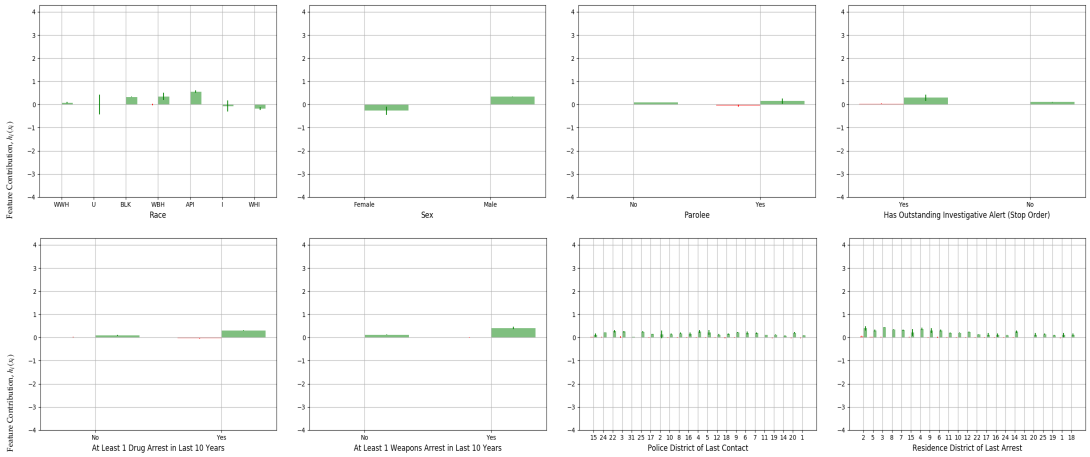

Figure 4.5: Eight features the Chicago Police says are *not used* in their risk scoring model.

model (in red) assigns importance to the features in Figure 4.4, but does not assign any importance to the features in Figure 4.5. This agrees with Chicago Police's statement about which features were and were not used in the black-box model. We also note that the outcome model (in green) does assign importance to the unused features (Figure 4.5), suggesting that there is signal available in the 8 unused features that the Chicago Police risk scoring model could have used, but chose not to use. Race and sex are 2 of these 8 features, which the Chicago Police especially emphasized are not used. These experiments show that mimic models can provide insights into black-box models, and demonstrate the advantages of using outcome information.

4.3.2 Auditing COMPAS

COMPAS, a proprietary score developed to predict recidivism risk, has been the subject of scrutiny for racial bias [8, 83, 27, 31, 18, 36]. We do not know what model class, input features or data were used to train COMPAS. As described

in Section 4.1, the COMPAS audit data[4] was collected by ProPublica; it is likely different from the original COMPAS training data.

Figure 4.6 compares the COMPAS mimic model (in red) and outcome model (in green) for four features: Age, Race, Number of Priors, and Gender. The dotted lines are 95% pointwise confidence intervals. We observe the following:

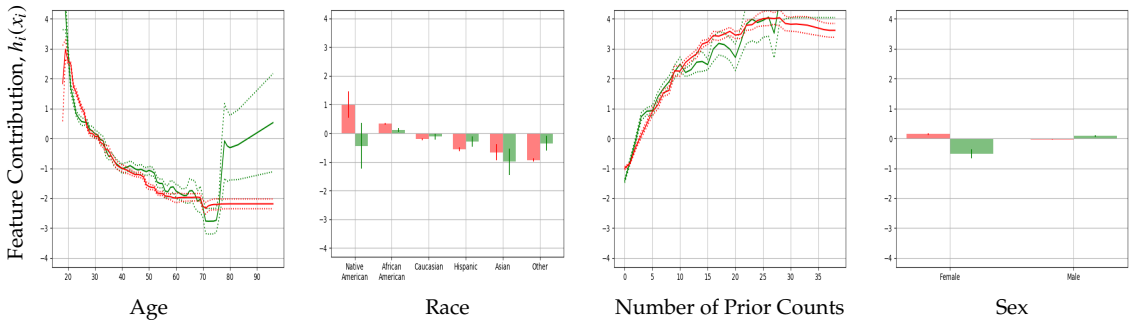

Figure 4.6: Feature contributions of four features to the COMPAS mimic model (in red) and outcome model (in green).

COMPAS agrees with ground-truth outcomes regarding the number of priors. In the 3rd plot in Figure 4.6, the mimic model and outcome model agree on the impact of Number of Priors on risk; their confidence intervals overlap through most of its range.

COMPAS disagrees with ground-truth outcomes for some age and race groups. The 1st and 2nd plots in Figure 4.6 show the effect of Age and Race on the mimic and outcome models. The mimic model (red) and the outcome model (green) are very similar between ages 20 to 70, and their confidence intervals overlap. For ages greater than 70, there is evidence that the models disagree as the confidence intervals do not overlap.

[4] https://github.com/propublica/COMPAS-analysis

The mimic and outcome models are also different for ages 18 and 19: the mimic model predicts low risk for young individuals, but we see no evidence to support this in the outcome model, with risk appearing to be highest for young individuals.

The mimic model predicts that African Americans are even higher risk, and Caucasians lower risk, than the ground-truth outcomes suggest is warranted. Note that the ground-truth outcomes might themselves be biased due to historical discrimination against African Americans.

Gender has opposite effects on COMPAS compared to ground-truth outcome. In the 4th plot in Figure 4.6, we see a discrepancy between the mimic model and outcome model on Gender. The mimic model predicts higher risk than warranted by ground-truth outcomes for females, and conversely for males.

Using differences to gain insight into COMPAS. We now ask "what could be happening in COMPAS, that could explain the differences we are seeing between the mimic and outcome models?":

1. Some feature regions may be underrepresented in the black-box model's training data and/or the audit data. In this audit data, only 3% of samples are between 18 and 20 years old, only 0.5% are older than 70 years old, and only 19% are female, which makes learning accurate models in these regions harder.
2. The black-box model may be deliberately simple for some feature regions. For ages greater than 70, the outcome model has much wider confidence intervals than the mimic model. The ground-truth outcomes are potentially high-variance in this region, yet the black-box model's scoring func-

tion may have been kept deliberately simple for extreme feature values like this.

3. The black-box model may have a very different form than the transparent model class. The mimic model predicts low risk for young individuals, but there is no evidence to support this in the outcome model. We trained an iGAM model with interactions between pairs of features, and observed strong interactions between very young age and other variables such as Gender, Charge Degree, and Length of Stay. If COMPAS has a more simple form and does not model interactions well, this may explain why COMPAS needs to predict low risk for very young individuals (because it cannot otherwise predict a reduced risk via interactions of age with other variables).

4. The black-box model may have used features missing from the audit data, that interact with the non-missing features. We investigate this in Section 4.3.4.

While we cannot tell (without further investigation) the definitive reason that explains a particular difference between the mimic and outcome models, this has surfaced ideas about the black-box model and uncovered potentially problematic feature regions that we did not *a priori* know, but can now proceed to investigate further.

4.3.3 Auditing Lending Club

Lending Club, an online peer-to-peer lending company, rates loans it finances on an A1-G5 scale. We use a subset of five years (2007-2011) of loans[5] that have matured, so that we have ground-truth on whether the loan defaulted. We do not know what model class or input features Lending Club used to train their risk scoring model. We believe the data sample we have is similar to the data they would have used to train their models. According to Lending Club, their models are refreshed periodically.

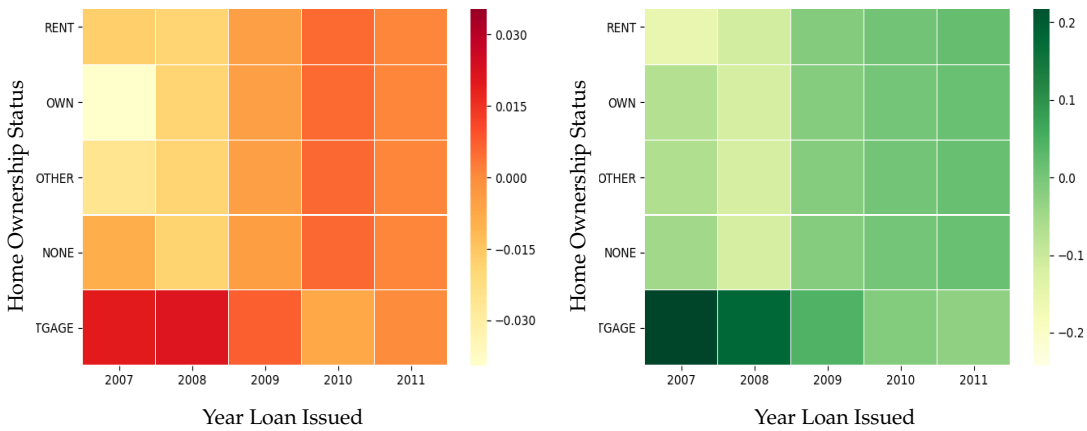

Figure 4.7: Interaction between loan issue year and home ownership in Lending Club mimic model (in red shades) and outcome model (in green shades). Regions colored by $h_{ij}(x_i, x_j)$.

We use this Lending Club example to discuss an insight gained into the black-box model from inspecting feature interactions in the transparent models. Figure 4.7 shows the interaction of loan issue year and home ownership in the Lending Club mimic model (in red) and ground-truth outcome model (in green). Having a home mortgage in 2007-2008 increases the loan default risk more than having a home mortgage in 2009 and beyond. Recall that 2007-2008 is around the time of the subprime housing crisis. Note the difference in

[5] https://www.lendingclub.com/info/download-data.action

ranges between the two plots—the range goes up to 0.2 for the outcome model (in green) whereas the range is much lower for the mimic model (in red). One possible explanation for this difference is that the Lending Club risk scoring model is updated conservatively (with some lag time), instead of being rapidly updated as economic conditions and behavior change.

4.3.4 Which Audit Data Are Missing Features?

As black-box models may use additional features we do not have access to, we developed a test (Section 4.2.2) to assess the impact missing features could have on the audit. Table 4.1 provides 95% confidence intervals for three correlation measures (linear and nonlinear) used in the test. If zero is in the confidence interval, the error of the mimic model (trained on the audit data) is not correlated with the error of the outcome model (also trained on the audit data). Then, it is unlikely that the audit data is missing key feature(s) that are a) predictive of outcomes (and hence will negatively affect the error of the outcome model if missing); and b) used in the black-box model (and hence will negatively affect the error of the mimic model if missing).

In Lending Club and Stop-and-Frisk we cannot distinguish these correlations from zero, suggesting that no key features are missing from the audit data. For Chicago Police, the confidence intervals contain 0 or are very close to 0 (lower limit 0.01), hence there is little evidence of missing key features. For COMPAS, there is evidence of positive correlation, indicating that the ProPublica data may be missing key features used in the COMPAS model. This is supported by the findings in Section 4.3.5 that no mimic models trained on the

Table 4.1: Statistical test for likelihood of audit data missing key features used by black-box model.

Risk Score	**Pearson ρ**	**Spearman ρ**	**Kendall τ**
COMPAS	[0.10, 0.13]	[0.10, 0.14]	[0.08, 0.10]
Lending Club	[0.00, 0.03]	[-0.01, 0.01]	[-0.01, 0.01]
Stop-and-Frisk	[0.00, 0.01]	[-0.03, 0.01]	[-0.02, 0.01]
Chicago Police	[0.00, 0.01]	[0.01, 0.03]	[0.01, 0.02]

ProPublica data, however powerful (e.g., random forests), could mimic COMPAS well.

4.3.5 Fidelity and Accuracy

To quantitatively evaluate the audit approach, we report fidelity (how well the mimic model predicts the black-box model's risk scores, measured in RMSE) and accuracy (how well the outcome model predicts the ground-truth outcomes, measured in AUC) for all the risk scoring models we audit in Table 4.2. For comparison, we also train linear models (a simpler model class than iGAM) and random forests (more complex, but less interpretable).

For COMPAS, all model classes (linear model, iGAM, random forest) have roughly equal fidelity and accuracy. Interestingly, none obtained RMSE lower than 2 on a 1-10 scale. Comparing outcome model AUCs across different model classes, iGAM's results are generally comparable to (or slightly better than) more complex random forests (Table 4.2). For the risk score mimic models, random forests are competitive for Lending Club and Chicago Police. Linear mimic

Table 4.2: Fidelity of mimic model and accuracy of outcome model. Lower RMSE is better, higher AUC is better.

	Risk Score	**Metric**	**Linear model**	**iGAM**	**iGAM w/ interactions**	**Random Forest**
Fidelity of mimic model	COMPAS	RMSE (1-10)	2.11 ± 0.057	2.01 ± 0.045	$\mathbf{2.00 \pm 0.047}$	2.02 ± 0.053
	Lending Club	RMSE (2-36)	3.27 ± 0.037	2.60 ± 0.049	2.52 ± 0.051	$\mathbf{2.48 \pm 0.033}$
	Chicago Police	RMSE (0-500)	17.4 ± 0.102	17.2 ± 0.125	16.5 ± 0.130	$\mathbf{14.0 \pm 0.280}$
	Stop-and-Frisk	RMSE (0-5)	$\mathbf{0.00 \pm 2 \times 10^{-15}}$	$\mathbf{0.00 \pm 1 \times 10^{-5}}$	$\mathbf{0.00 \pm 2 \times 10^{-5}}$	$0.01 \pm 2 \times 10^{-3}$
Accuracy of outcome model	COMPAS	AUC	0.73 ± 0.029	0.74 ± 0.027	$\mathbf{0.75 \pm 0.029}$	0.73 ± 0.026
	Lending Club	AUC	0.69 ± 0.006	$\mathbf{0.69 \pm 0.016}$	$\mathbf{0.69 \pm 0.014}$	0.68 ± 0.020
	Chicago Police	AUC	$\mathbf{0.95 \pm 0.007}$	$\mathbf{0.95 \pm 0.007}$	$\mathbf{0.95 \pm 0.007}$	0.93 ± 0.009
	Stop-and-Frisk	AUC	0.84 ± 0.020	0.85 ± 0.020	0.85 ± 0.020	$\mathbf{0.87 \pm 0.024}$

models are not far behind iGAMs for several risk scoring models (COMPAS, Chicago Police, Stop-and-Frisk), suggesting that the black-box model's functional form might be very simple. We know this to be true for Stop-and-Frisk from Section 4.3.1 where the model was a simple linear model.

4.3.6 Using Additional Data for Distillation

One possible reason why COMPAS is challenging to mimic may be that the ProPublica data is missing key features. This agrees with the results of the statistical test in Section 4.3.4. Another possible reason is the small sample size (less than 7,000 samples).

One advantage of using a model distillation approach to inspect black-box models is that the approach may be able to benefit from additional unlabeled data if the black-box model can be queried to label the additional data [22]. We found an additional 3,000 individuals in the ProPublica data with COMPAS risk scores *but no ground-truth outcomes*. Adding them to the training (not testing) data for the mimic model and retraining the mimic model, we find marginal improvement in the mimic model's fidelity (from RMSE 2.0 to 1.98). Doing the opposite—removing individuals from the training data in 1,000 increments—decreased the mimic model's fidelity only marginally (to RMSE 2.1, training on only 1,000 individuals). These analyses suggest that for COMPAS, missing key features is a more pressing issue than insufficient data.

4.4 Discussion

Sometimes we are interested in detecting bias on features intentionally excluded from the black-box model. For example, a credit risk scoring model is probably not allowed to use race as an input. Unfortunately, not using race does not prevent the model from learning to be biased. Racial bias in a data set is likely to be in the outcomes — the labels used for learning; not using race as an input *feature* does not remove the bias from the *labels*.

If race were uncorrelated with all other features (and combinations of features) provided to the model, then removing race would prevent the model from learning to be racially biased because it would not have any input features on which to model this bias. Unfortunately, in any real-world, high-dimensional data set, there is massive correlation among the features, and a model trained to predict credit risk will learn to be biased from correlation of the *excluded* race feature with other features that likely remain in the model (e.g., income or education).

Hence, removing a protected feature like race or gender does not prevent a model from learning to be biased. Instead, removing protected features make it harder to detect how the model is biased, or correct the bias, because the bias is now spread in a complex way among all the correlated features throughout the model instead of being localized to the protected features. The main benefit of excluding protected features like race or gender from the inputs of a machine learning model is that it allows the group deploying the model to claim (incorrectly) that their model is not biased because it did not use these features.

When training a transparent model to mimic a black-box model, we inten-

tionally include all features—even protected features like race and gender—specifically because we are interested in seeing what the mimic model *could* learn from them. If, when the mimic model mimics the black-box model, it does not see any signal on the race or gender features and learns to model them as flat (zero) functions, this suggests whether the black-box model did or did not use these features, but also if the black-box model exhibits race or gender bias even if race or gender were not used as inputs.

4.5 Conclusion

The Distill-and-Compare approach to auditing black-box models was motivated by a realistic setting where access to the black-box model API is not available. Instead, only a data set labeled with the risk score (as produced by the risk scoring model) and the ground-truth outcome is available. The efficacy of Distill-and-Compare increases when a model class that can be highly faithful to the black-box model and highly accurate at predicting the ground-truth outcomes is used, and when the audit data is not missing key features used in the black-box model.

A key advantage of using transparent models to audit black-box models is that we do not need to know in advance what to look for. Many current auditing approaches focus on one or two protected features defined in advance, and thus are less likely to detect biases that are not *a priori* known. The Distill-and-Compare audit approach using transparent models can hence be most useful for real-world, high-dimensional data with multiple, unknown sources of bias.

CHAPTER 5

INVESTIGATING HUMAN + MACHINE COMPLEMENTARITY: A CASE STUDY ON RECIDIVISM

5.1 Introduction

The criminal justice field has used forecasting tools to perform risk assessment since the 1920s [51]. There is an ongoing debate whether AI systems, such as risk assessment models, are superior to human judgment. Grove et al. found that decisions made by expert humans such as judges can sometimes be highly variable and biased by unobserved, irrelevant features not predictive of recidivism [58].

In a recent study related to Human vs. Machine predictions of recidivism, Dressel and Farid [39] asked Mechanical Turk workers to predict whether a defendant would recidivate within two years (the same label predicted by COMPAS). They also ran a second variant of their study where defendants' race was revealed. They did not find Human and COMPAS accuracies to be significantly different (COMPAS: 65.2%, Humans without defendant race information: 67.0%, and Humans with defendant race information: 66.5%).

Although the Dressel and Farid study demonstrated that the accuracy of COMPAS and Human predictions were comparable [39], it was unclear whether COMPAS and Humans were accurate on the same or disjoint sets of defendants. Significant overlap would suggest that the Humans and COMPAS make similar

This chapter is based on material in [132].

assessments; less overlap suggests that human reasoning differed from machine analysis. Humans may have access to additional information or context not available to algorithmic systems; machines may not be influenced by the same biases that plague human judgment or may be better at using statistical signals learned from large amounts of data.

While the Dressel and Farid study focused on an analysis of machine vs. human performance [39], the goal of several real-world implementations of recidivism models is for such models to completement decision making by humans such as judges, parole officers, etc. [16, 42].

In this paper, we study complementarity between human and machine decision making for recidivism prediction. Instead of focusing on the superiority (or lack thereof) of algorithmic systems compared to human judgment, we explore the similarities and differences between Human and COMPAS decisions, and construct hybrid models that combine the strengths and weaknesses of human and machine decision making.

Our contributions in this paper are:

- An understanding of how human and machine decisions differ, and how and when they make errors.
- A characterization of agreement and disagreement between human and machine decision making to better understand their complementarity.
- An investigation of hybrid models to leverage differences in human and machine decision making.

Based on our findings, we discuss the potential of hybrid models and short-

comings of existing data sets. We make recommendations for data collection best practices for future study of hybrid decision making in the fairness domain.

5.2 Related Work

Humans and decision making. Dressel and Farid [39] were the first to compare decisions made by COMPAS to non-expert humans (Mechanical Turkers); consequently, our analysis is based on their data. Before that, Kleinberg et al. compared decisions made by machine learning models to expert humans (judges) [82]. Lakkaraju et al. [88] showed that analyses of recidivism based on human decisions are further complicated by the "selective labels" problem, where observability of outcomes are affected by judges' release decisions. Other work incorporating human input or feedback in the fairness domain include investigating human perception of which features are fair or otherwise [57].

Hybrid models. Investigations across different domains identify that humans and machines have weaknesses and complementary abilities, thus suggesting benefits from hybrid models. In medicine, recent research showed that existing machine learning models with lower accuracy rates than human experts can decrease expert error rates by 85% [144]. On challenging face recognition tasks, combining multiple expert opinions does not improve task accuracy, however complementing an expert with a inferior face recognition system can [110]. On the other hand, research on complementary computing demonstrated how humans and machines can be more effective together in problem solving [68] and image classification tasks [74].

Diagnosing errors. The key to aggregating machine and human analyses for

Table 5.1: Characterizing agreement and disagreement between COMPAS decisions, Human decisions, and ground truth. The number of defendants and characteristics for each of the eight cases are described.

Case	COMPAS Score	Human Score	Ground Truth	Agreement	Correctness	% Defendants	Feature Characteristics*
1	High	High	Yes	Agree	Both correct	49.0%	$1.5 < \text{Priors} \leq 12.5$
2	Low	Low	No	Agree	Both correct		$23.5 < \text{Age} \leq 48.5$ & $\text{Priors} < 1.5$
3	High	Low	Yes	Disagree	COMPAS correct	16.2%	$23.5 < \text{Age} \leq 48.5$ & $\text{Priors} < 0.5$
4	Low	High	No	Disagree	COMPAS correct		$1.5 < \text{Priors} \leq 5.5$ & $\text{Age} > 32$
5	Low	High	Yes	Disagree	Human correct	15.9%	Similar to Case 4
6	High	Low	No	Disagree	Human correct		Similar to Case 3
7	High	High	No	Agree	Both incorrect	18.9%	No pattern, similar to Cases 1-6
8	Low	Low	Yes	Agree	Both incorrect		

* Characteristics determined by decision tree (Figure 5.6) and clustering analysis. See more details in Section 5.4.3.

improved performance is understanding where and how machines and humans fail [84]. Various approaches have been proposed for understanding where machine errors come from. Lakkaraju et al. [86] defined *unknown unknowns* as cases where the model is highly confident of its prediction but is wrong. We adopt this definition and take agreement and disagreement as a measure of confidence, hence our unknown unknowns are cases where COMPAS and Human scores agree, yet are wrong. Ramakrishnan et al. learned models to predict blind spots in reinforcement learning settings [113]. Nushi et al. used distillation of black box model decisions to interpretable model classes to explain failures of AI systems [108]. Similarly, Tan et al. used distillation to interpret black box risk scoring models such as COMPAS [134]. We follow a similar approach of utilizing interpretable machine learning models such as decision trees to analyze how machines and humans reason about recidivism, when and how their decisions differ and how they can be aggregated.

5.3 Approach

5.3.1 Constructing Human Risk Score

Our goal in this paper was to compare algorithmic and human decision making for complex decisions using recidivism predictions as our initial domain. [39] provide data on both human predictions (from Mechanical Turk workers), and algorithmic predictions (from COMPAS). One question is whether decisions made by Mechanical Turk workers on this data are *internally* consistent, or, in other words, whether different Turk workers assess risk similarly for the same

defendant. Large agreement among Turk workers increases confidence that our subsequent findings based on generating Human scores from Turk worker predictions generalize to Human decision making.

We found that on average, 80% of the Turk workers that assessed the same defendant agreed with each other. This was a high level of agreement, particularly for Mechanical Turk, where spam labeling is commonly observed [72]. Hence, we perform a majority aggregation of Turk workers' predictions to assemble a Human risk score for recidivism risk, h_j. Specifically, we construct h_j by taking the mean prediction across the Turk workers for each defendant: let h_{ij} be Turk worker i's prediction for defendant j where $h_{ij} \in \{0, 1\}$, $i = 1, \ldots, 20$, $j = 1, \ldots, 1000$, we take $h_j = \sum_i h_{ij}/2$, dividing by two to scale h_j to 1-10, which is COMPAS' scale. We constructed scores for both conditions mentioned in the Introduction - a with-race Human score (HWR) for when Turk workers were told the defendants' race, and a no-race version (HNR).

For each score, we find the optimal cutoff point to binarize the score by computing calibration, false positive, and false negative rates at various cutoff points from 1 to 10. COMPAS, HNR, and HWR scores have approximately equal accuracy, false positive, and false negative rates at the cutoff point of >= 5 (Figure 5.1). Hence, we chose this cutoff point for all three scores. Note that Northpointe, the creator of COMPAS, also uses a >= 5 cut-off [18], and >= 5 is implied by [39]'s use of a "wisdom-of-the-crowd" based majority rules criterion.

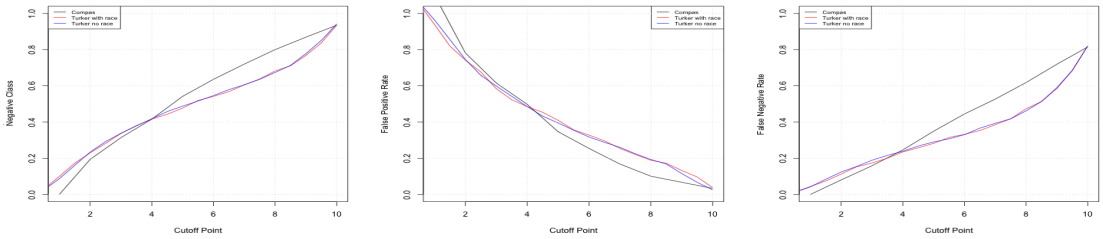

Figure 5.1: Accuracies (left), false positive rates (center), and false negative rates (right) for COMPAS and Human scores at different cutoff points for binarizing the scores.

5.3.2 Partitioning by Agreement and Correctness

We now sketch our approach towards studying how COMPAS and Human scores agree or disagree, and interact with ground truth. Table 5.1 describes eight possible combinations of two binary risk scores and ground truth. These eight combinations can be grouped into the four partitions illustrated in Table 5.1: Both correct, Both incorrect, Human correct, and COMPAS correct.

Comparing the level of agreement and correctness between the Human and COMPAS scores, we found that almost 50% of the time, Humans and COMPAS agree and are correct (Table 5.1). However, for the remaining 50% of defendants, either one, or both scores were incorrect. This suggests that if error regions of COMPAS and Humans do not perfectly overlap **and can be characterized**, then decision-making processes can potentially be improved through utilizing the complementary views of humans and machines.

When both risk scores agree and are correct, either score will return the same prediction, hence it does not matter which is used (in terms of accuracy). The space where both scores agree but are incorrect according to ground truth is a

blind spot for COMPAS and Humans, also called *unknown unknowns* [86]. To characterize the space of agreement or disagreement between COMPAS and Human scores, we use clustering and decision trees. Table 5.1 summarizes our findings of the features that characterize each case. Finally, when COMPAS and Human scores disagree (Cases 3-6 in Table 5.1) we train hybrid risk scoring models to see if they can leverage disagreement between the two scores to improve on the accuracy of single scores.

5.3.3 Designing Hybrid Models

The simplest hybrid model is an average of two risk scores. We train a slightly more sophisticated model - a **weighted average** hybrid model that learns the optimal linear combination of two risk scores to predict ground truth.

If we had access to an oracle that can be queried to obtain ground truth recidivism for any new observation, we can determine which of COMPAS or Human scores better predicts ground truth. However, test-time access to a ground truth oracle is not realistic. Hence, we relax this assumption of oracle access at test-time to only training-time, and train a binary classification model **only on observations where the two risk scores disagree** to predict which risk score to pick. In other words, this model predicts which score – COMPAS or Human – to use for Cases 3-6 in Table 5.1 using features available at training time such as defendant features, Turk worker features, COMPAS score, and Human score. We call this an **indirect** hybrid model – indirect because the hybrid model takes as input the prediction of which risk score is better, and outputs the desired ground truth recidivism prediction. Figure 5.2 shows this model.

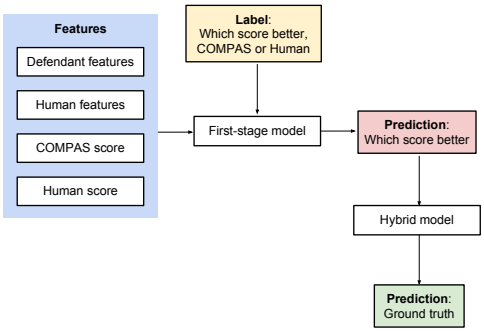

Figure 5.2: Schematic of indirect hybrid model that predicts whether to use COMPAS or Human scores to predict ground truth recidivism].

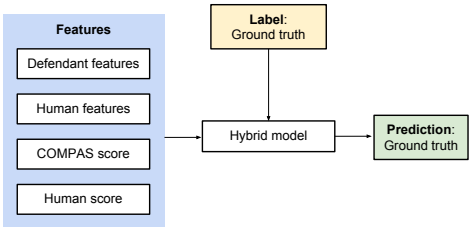

Figure 5.3: Schematic of direct hybrid model that directly predicts ground truth recidivism using COMPAS and Human scores as features.

We also **directly** predict ground truth recidivism as a function of not just features but also the two risk scores. Figure 5.3 shows this model.

We test the hybrid models against **random** and **single** score baselines. We use two types of random baselines: random ground truth labels, and random risk score. Single score baselines are COMPAS or Human scores on their own (1-10 scale, or binarized at >=5), or models trained with defendant and Turk worker features and the single score to predict ground truth.

All hybrid and single models in this paper were trained using the random forest model class, a model class shown to perform well on many problems [25].

We use area under the ROC curve (AUC) as our main accuracy measure, in line with several papers measuring recidivism [29]. Besides AUC, we also report balanced accuracy (Bal Acc), i.e., the mean classification accuracy across classes. For error rates, we track false positives (FPR), false negatives (FNR), false discovery (FDR), and false omission (FOR). Equations for these error rates are below. Note that Kleinberg [83] and Choudechouva [27] showed the impossibility of satisfying several of these metrics simultaneously. These results are reported in Section 5.7. All metrics are reported over ten 80%-20% train-test splits to account for variability between test sets.

Definitions of Metrics

Given a binary label and a binary prediction, let FP denote the number of false positives, FN denote the number of false negatives, TP denote the number of true positives, and TN denote the number of true negatives.

Balanced accuracy

$$BalAcc = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)$$

False positive rate (FPR)

$$FPR = \frac{FP}{FP + TN}$$

False negative rate (FPR)

$$FPR = \frac{FN}{FN + TP}$$

False discovery rate (FDR)

$$FDR = \frac{FP}{FP + TP}$$

False omission rate (FOR)

$$FOR = \frac{FN}{FN + TN}$$

5.4 Analysis and Results

In this section, we report our findings of COMPAS and Human complementarity in terms of predictive performance and decision making, characterize the space of COMPAS and Human agreement and disagreement, and discuss results from our hybrid models.

5.4.1 COMPAS vs. Humans: Predictive Performance

Across 1,000 defendants, Human scores have slightly higher means than COMPAS (mean HNR 5.1, HWR 5.2, COMPAS 4.6, all on 1-10 scale), and the Human scores are more correlated with each other than with COMPAS (COMPAS and HNR correlation 0.52, COMPAS and HWR 0.53, HNR and HWR 0.93).

Table 5.2 displays the predictive performance of COMPAS and Human scores, on all defendants and by race (this is similar to Table 1 in [39] 's findings when evaluating accuracies of the three scores.

Table 5.3 presents predictive performance separated by recidivism status: whether the defendant recidivated or not. Here, Humans were better at pre-

Table 5.2: COMPAS and Human performance for ground truth recidivism prediction.

	Accuracy			AUC		
Race	C	HNR	HWR	C	HNR	HWR
All	0.65	0.66	0.66	0.70	0.71	0.71
Black	0.68	0.66	0.65	0.70	0.69	0.69
White	0.66	0.66	0.64	0.71	0.69	0.70
Other	0.65	0.60	0.66	0.64	0.65	0.65

dicting defendants who recidivate, while COMPAS was better at predicting defendants who do not recidivate. In other words, on this data, Humans tended to have higher true positive rates (and hence lower false negative rates) and COMPAS tended to have higher true negative rates (and hence lower false positive rates).

We see similar effects for the level of agreement between risk scores, race, and ground truth. COMPAS and Humans demonstrate higher levels of agreement for correctly predicting that black defendants will recidivate, but their level of agreement drops significantly for white or other race defendants who recidivate. The opposite is true for defendants who do not recidivate. COMPAS and Humans have higher levels of agreement for correctly predicting that white and other race defendants will not recidivate, but this level of agreement drops for black defendants who do not recidivate.

Table 5.3: COMPAS and Human performance for ground truth recidivism prediction, by recidivism status (this table is a refinement of Table 5.2 by recidivism status). Left: defendants who **do** recidivate. Right: defendants who **do not** recidivate. Only accuracies are displayed because AUC cannot be calculated when ground truth only has one value ("yes" for do recidivate, "no" for do not recidivate).

	Accuracy					
	Do recidivate			Do not recidivate		
Race	C	HNR	HWR	C	HNR	HWR
All	0.62	0.68	0.69	0.69	0.64	0.63
Black	0.74	0.74	0.70	0.61	0.55	0.59
White	0.60	0.50	0.59	0.69	0.75	0.68
Other	0.38	0.59	0.65	0.80	0.61	0.66

5.4.2 COMPAS vs. Humans: Decision Making

Which features are most important in COMPAS and Human decision making? It is known that COMPAS scores can be predicted from only a few features, in particular the "number of priors" and age [28, 6, 134]. In fact, Equivant (formerly Northpointe, Inc.), the creator of COMPAS, clarified that the COMPAS risk assessment has only six inputs[1] (exactly which features are used is not known). To determine if Human decision making places more importance on other features, we trained interpretable models to predict each of the three scores. All three models saw the same set of features – age, race, sex, number of juvenile misdemeanors, number of juvenile felonies, number of (non-juvenile) priors, crime charge degree (misdemeanor or felony), and crime charge. First,

[1] http://www.equivant.com/blog/official-response-to-science-advances

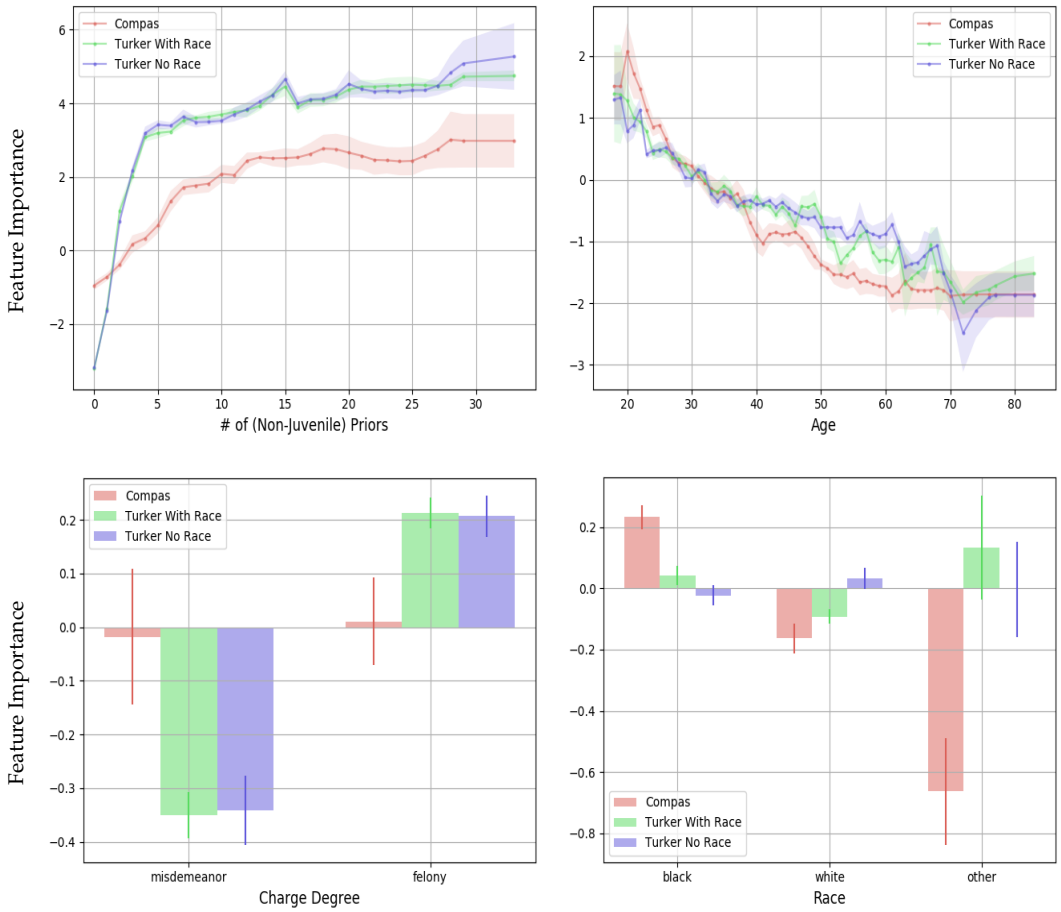

Figure 5.4: Explaining relationships between COMPAS and scores derived from judgements of Human Turkers shown defendants' race (HWR) or not shown defendants' race (HNR) and various features. The larger the magnitude on the y-axis, the more important the feature. "Number of priors", with y-axis scale -3 to 6, is the most important feature for all three scores, followed by "age".

we trained iGAM models, a type of additive model based on nonparametric base learners [24]. Figure 5.4 illustrates the importance of four of these features for predicting each score. **Like COMPAS, the two most important features in Human decision making are the "number of priors" and "age".** However, Human scores place more weight on the "number of priors" and "charge de-

gree" features than COMPAS, whereas age's impact is similar for COMPAS and Human scores. Decision trees trained to predict each of the three risk scores confirm that "number of priors" is the most important feature, with every tree's root node splitting on this feature.

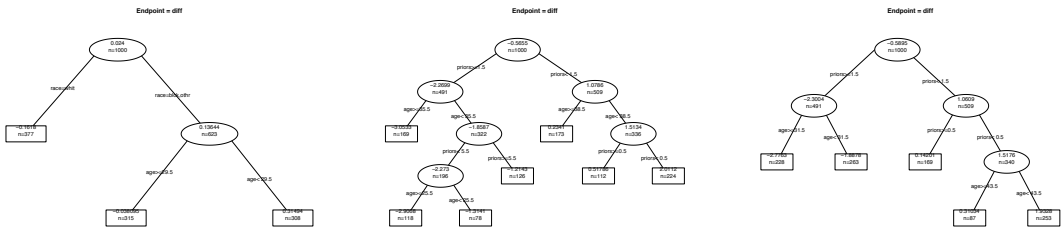

Figure 5.5: Decision tree to explain differences between COMPAS and Human scores. Left: difference in scores derived from judgements of Human Turk workers when and when not told of defendants' race (HWR - HNR). Center: difference in scores given by COMPAS and Human Turk workers not told of defendants' race (C - HNR). Right: difference in scores given by COMPAS and Human Turk workers told of defendants' race (C - HWR).

Including race when predicting these scores, even when the scores may not have seen race, returns some interesting findings. Recall that HNR scores were generated from Turk workers who were not told the defendants' race. We considered the impact of race on Human recidivism predictions, by comparing the importance of the race feature on HWR (green) and HNR (purple) scores in Figure 5.4. We find that black defendants were assessed to have slightly higher recidivism risk by Turk workers when told of their race. The decision tree predicting the difference of HWR and HNR scores in Figure 5.5 also agreed with this finding, returning a first split on race where white defendants were assigned slightly lower risk (-0.16) in the Human with-race condition, and black and other race individuals were assigned slightly higher risk (+0.14). In con-

trast, both decision trees predicting the difference between COMPAS and HWR scores, as well as COMPAS and HNR scores, split on "number of priors" and age but not race.

Hence, even though revealing race did not significantly affect the predictive performance of Humans for ground truth, as found by [39], including race appeared to slightly effect Humans' perception of recidivism risk (magnitude around +/- 0.15 on a 1-10 score scale). Note, however, that the set of Turk workers in the no-race and with-race conditions were different; this effect may diminish or exacerbate if the experiment is re-run with the same set of Turk workers.

5.4.3 COMPAS + Humans: Characterizing Agreement and Disagreement

We now determine the features that drive agreement or disagreement between COMPAS and Human scores. To do so, we use two techniques – clustering and decision trees. Specifically, we performed mean-shift clustering [35], a robust-clustering method that avoids the need to specify an arbitrary number of clusters, to cluster defendants in each of Cases 1-8 from Table 5.1. We also built a multiclass decision tree to classify individuals into each of the eight cases. Finally, we assessed the distribution of features across the found clusters and tree partitions. Figure 5.6 presents the decision tree. We elaborate on our findings below. A summary of the feature characteristics is in Table 5.1.

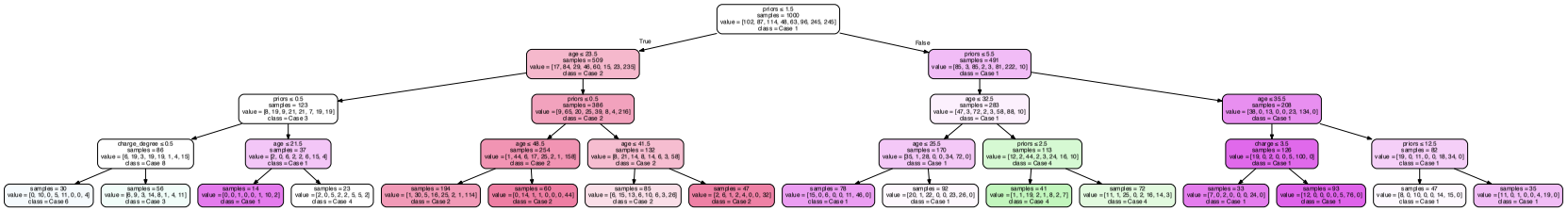

Figure 5.6: Decision tree to explain the three-way interaction between COMPAS, Human scores, and ground truth recidivism. The label for the prediction task corresponds to the 8 different cases from Table 5.1. The five values in each node are (1) the partition of features defining that node (2) the number of samples in that node (3) the number of samples in each of the 8 cases (4) the predicted case (of the 8 cases).

Characterizing the space of agreement

Easy calls: COMPAS and Humans agree, both correct. When we cluster defendants in this region of correct agreement, two clusters emerge that map to the two cases. The key separation between Cases 1 (COMPAS high, human high, both correct) and 2 (COMPAS low, Human low, both correct) is the number of priors, and to a lesser extent age. The average number of priors for defendants in Case 1 is 7.9, and 0.34 for Case 2. The average age for defendants in Case 1 is 30.3, and 40.56 for Case 2. Consequently, these cases correspond to what one might consider *easy calls*, i.e., defendants for whom the number of priors and age alone provide sufficient information to predict recidivism accurately.

Unknown unknowns: COMPAS and Humans agree, but both incorrect. Now we turn our attention to the region of wrong agreement - defendants whose COMPAS and Human scores agree, yet fail to predict ground truth (Cases 7 & 8). These defendants are very similar to defendants in other cases – they are truly *unknown unknowns*. Effectively, defendants in Cases 7 & 8 are exactly defendants for whom the number of priors and age alone are not different enough to distinguish them from defendants in other cases, despite these defendants having fundamentally different ground truth labels. Because both COMPAS and Human scores are over reliant on the number of priors and age, both scores fail for defendants for whom these two features alone are not sufficient to predict recidivism.

Characterizing the space of disagreement

Our key finding for defendants for whom COMPAS and Human scores disagree (Cases 3-6) mirrors our findings for the unknown unknowns. These defendants had similar number of priors and age as defendants in other cases. In general, the four cases in the space of disagreement could not be cleanly separated from each other – Cases 3 and 6 were similar; Cases 4 and 5 were similar – and also overlapped with the space of agreement. For example, defendants with low COMPAS scores, high Human scores but did not recidivate (Case 4) tended to have 1.5 to 5.5 priors and are younger than 32.5 years old. However, these defendants significantly overlap with defendants in several other cases (Cases 1, 7, and 5 as seen in Table 5.1).

COMPAS score high, Human score low (Cases 3 & 6). The difference between Cases 3 and 6 is their ground truth label - defendants in Case 3 recidivated, whereas defendants in Case 6 did not. According to the decision tree's partitions, defendants in Cases 3 and 6 tend to have < 0.5 priors. In fact, the key distinguishing feature between Cases 3 and 6 is the type of crime that the defendant was charged with. In addition, we found that some of the multiclass trees we built to predict classification into the eight cases did not always have terminal nodes with Case 6. Sometimes, Case 6 is combined with Case 3, indicating that the features do not have sufficient signal to adequately distinguish these two cases.

COMPAS score low, Human score high (Cases 4 & 5). The difference between Cases 4 and 5 is also their ground truth label - defendants in Case 4 did not recidivate, whereas defendants in Case 5 did. Case 4 defendants tended to have 1.5 to 5.5 priors and be older than 32.5 years old. Case 5 was not always

present as a terminal node in our trees, and are very similar to defendants in Case 4 and also Cases 1 and 7 (in the space of agreement).

5.4.4 COMPAS + Humans: Leveraging Disagreement to Build Hybrid Models

Since defendants for whom COMPAS and Human scores disagree have the highest possibility of benefiting from hybrid models, we build two separate sets of hybrid models: (1) models on only the space of disagreement – 32% of defendants in this data (cf. Table 5.4); (2) models on all defendants (cf. Table 5.5).

Hybrid methods tended to outperform single scores (or models trained on features and single scores) by a small margin. In Table 5.4, the best performing model (AUC 0.60) is a hybrid random forest predicting ground truth using features, COMPAS, and Human (no-race condition) scores. This was better than single risk scores (HNR 0.56, HWR 0.54, Compas 0.49), but comparable to a random forest model trained on the original features plus the HNR scores (but not with COMPAS), which obtained an AUC of 0.59.

Interestingly, despite the low AUC of COMPAS (0.49), combining it with HNR did not degrade the hybrid model's performance and in fact led to a small AUC improvement of 0.01. However, this improvement is within the margin of error.

Next, we examine these results by race. Table 5.7 presents these results for blacks, Table 5.8 for whites, and Table 5.9 for other races. The trend is again similar, where hybrid models tended to obtain slightly better results than their

Table 5.4: Test-set performance of hybrid models trained on defendants whose COMPAS and Human scores disagree. Best results in cyan and bolded. See Table 5.6 for an extended version of this table.

Type	Model	AUC
Hybrid	Best hybrid of C and HNR	**0.60 ± 0.07**
	Best hybrid of C and HWR	0.58 ± 0.08
	Best hybrid of C, HWR, HNR	0.58 ± 0.07
Single	Predict GT from features and HNR	0.59 ± 0.07
	HNR (1-10 scale)	0.56 ± 0.05
	Predict GT from features and HWR	0.54 ± 0.06
	HWR (1-10 scale)	0.54 ± 0.04
	Predict GT from features and C	0.51 ± 0.07
	C (1-10 scale)	0.49 ± 0.06
None	Predict GT from features	0.52 ± 0.07
Random	Randomly pick between C and HNR	0.55 ± 0.08
	Randomly pick between C and HWR	0.54 ± 0.07
	Randomly pick between C, HWR, HNR	0.54 ± 0.06

single-model counterparts, but improvements are typically within the margin of error. Hybrid models for blacks had the best accuracy and error rates; single models for other races (only 31 defendants) had the best accuracy and error rates.

In general, the best hybrid models tended to leverage defendant and Human worker features, plus both risk scores, to either directly or indirectly predict ground truth recidivism. For the space of disagreement, the best hybrid models

Table 5.5: Test-set performance of hybrid models trained on *all* defendants, not just defendants for whom COMPAS and Human scores disagree (see Table 5.4 for those results). Best results in cyan and bolded. The benevolent oracle is the risk score best at predicting ground truth, to provide an upper bound on the accuracy reachable on this data set of any hybrid COMPAS-Human model built on the two risk scores. The adversarial oracle is the risk score **worse** at predicting ground truth, to provide a lower bound. See Table 5.10 for an extended version of this table.

Type	**Model**	**AUC**
Oracle	Benevolent oracle	0.85 ± 0.03
	Adversarial oracle	0.57 ± 0.03
Hybrid	Best hybrid of C and HNR	**0.74 ± 0.03**
	Best hybrid of C and HWR	**0.74 ± 0.04**
	Best hybrid of C, HWR, HNR	0.73 ± 0.03
Single	HNR (1-10 scale)	0.72 ± 0.03
	HWR (1-10 scale)	0.72 ± 0.03
	C (1-10 scale)	0.71 ± 0.03
	Predict GT from features and C	0.71 ± 0.03
	Predict GT from features and HWR	0.71 ± 0.03
	Predict GT from features and HNR	0.70± 0.03
None	Predict GT from features	0.69 ± 0.02
Random	Randomly pick between C and HWR	0.73 ± 0.04
	Randomly pick between C and HNR	0.72 ± 0.04
	Randomly pick between C, HWR, HNR	0.71 ± 0.03

also tended to prefer HNR over HWR, particularly when evaluating races other than whites. On the other hand, for the space of disagreement, hybrid models

based on (weighted) averages of the COMPAS and human scores tend to underperform models that incorporated defendant and Human worker features. Notably, this is not the case for all defendants as the best performing hybrid models for all defendants were the optimally weighted average models (Table 5.10).

We have shown that for defendants for whom COMPAS and Human scores disagree, hybrid models can be more beneficial than single risk scores (even when one of the scores is not as high performing as the other, as is the case for COMPAS compared to Humans for this set of individuals), but, in general, the improvements are marginal and, in many cases, within the margin of error.

5.5 Discussion

Our key finding is that Human and Machine decision making for recidivism predictions does differ and we were able to characterize the space of how these decisions relate to each other. Our exploration of a hybrid Human+Machine model showed slight improvements in accuracy, but further iteration is required to enhance this approach. From our analysis, the number of priors is a key feature in both COMPAS and Human decision making. We saw that COMPAS and Humans tended to agree (and were right) on defendants with a very high or very low number of priors. We saw that the defendants that COMPAS and humans agreed on (but were wrong) were truly *unknown unknowns* – there was no discernable pattern in these cases. Unfortunately, they make up 19% of the data, which bounds the maximal possible improvement from a hybrid model on this data.

When we focused on the 32% of defendants where COMPAS and Human decisions disagree, our hybrid models started to exhibit some improvement, though still within the margin of error. The cases in this region were also the most uncertain, with single risk scores achieving between 0.49 and 0.56 AUC. We saw that for this region of uncertainty, single risk scores could be further improved by allowing them to see some amount of ground truth labels, alongside defendant features. We saw that number of priors, once again, and age were the two most important features to determine whether a defendant would fall in Case 3-6, although separation between these four cases was often not clear.

Several reasons could explain why we were not getting better accuracy from the hybrid models: 1) Ground truth labels are noisy. 2) Turk workers are not experts. 3) Ground truth is inherently unpredictable or the features we have do not present enough information to predict ground truth accurately. 4) Small sample size.

5.5.1 Noisy Ground Truth Labels

One limitation of our hybrid models is possible noise (or bias) in the ground truth labels in the ProPublica COMPAS data. The "primary" definition of recidivism from the US Sentencing Commission . As we continue to develop machine learning models for recidivism, we need to reevaluate the ground truth labels we are collecting to ensure they are unbiased.

5.5.2 Criminal Justice Expertise

It is important to note that the Human risk scores in our analyses were obtained from Mechanical Turk workers. The ecological validity of using Turk workers may be low, as they have no criminal justice experience, and the decisions they are asked to make (whether a defendant recidivates or not) may have little relation to the types of decisions they make in their day-to-day lives. Gathering human data from judges, in actual legal settings, will help us further investigate the potential of hybrid models in fairness domains. We need to gather more quantitative and qualitative data on when judges and algorithmic systems agree and disagree, and what additional information the judge may be using to inform her decision. This could help hybrid models better discern when to choose human judgment over algorithmic prediction to achieve better performance overall.

5.5.3 Lacking Evidence About the World

We have two other hypotheses why our hybrid models only marginally improve over the accuracy of COMPAS or Human scores alone despite the presence of differences in COMPAS and Human reasoning. First, perhaps recidivism is an unpredictable event with a lot of inherent uncertainty, and as such, the accuracy of any model is limited. This is consistent with prior work that found similar AUCs for commercial recidivism prediction systems [38]. Second, it could be that the seven features included in this data are not sufficient to properly evaluate recidivism risk. This second explanation is likely, since the Turk worker ratings are only based on those seven features (besides race). In a real world

court setting, a judge has access to additional information that could be used to inform their reasoning. This "private information" may be helpful, however it remains to be seen if private information may also be detrimental to human reasoning, as seen in [58]), many defendants may appear similar when viewed through the lens of only two features.

5.5.4 Small Sample Size

In our hybrid models trained on only the 340 defendants for which COMPAS and Human scores disagreed, the improvements demonstrated were subsumed by large margin of errors. This was also the case for further subgroups of races (169 blacks, 114 whites, 31 other races). Repeating the Mechanical Turk experiment and hybrid models on a larger sample of the original ProPublica COMPAS data will provide more evidence as to whether human judgment can help machines in making recidivism predictions.

5.6 Conclusion

In complex settings, like a courtroom or hospital, it is unlikely that algorithmic systems will be making all decisions without input from human experts. Our approach focused efforts on cases where humans and machines disagree as a potential area to enhance decision making. Ultimately, we want to leverage the best of both worlds: humans that glean subtle, interpersonal insights from rich context, and machine algorithms that provide rigor and consistency. However, on this data set, our hybrid models only showed minor improvements in

ground truth prediction of recidivism. An important next step will be to further our investigation to include predictions made by judges in real-world settings or explore our hybrid Human + Machine model approach on other domains or datasets. We hypothesize that the richness of the real-world may provide better context for enhanced hybrid Human + Machine models.

A key debate in recidivism predictions involves issues of bias and fairness, particularly for false positive and false negative judgments. Although our work uncovered a few aspects where race had an impact, it was not the primary focus of our work. We intend to look more closely at issues of bias and fairness in future work, especially as we gather more real-world data. Although both humans and algorithms can have inherent biases, if these biases are different, a hybrid model has the potential to help overcome them.

5.7 Extended Result Tables

Table 5.6: Extended result table for test-set performance of hybrid models trained on defendants whose COMPAS and Human scores disagree. Best results in cyan and bolded. See Table 5.4 for a reduced version of this table. Rows marked with * are the rows labeled as *best* in Table 5.4.

Type	Model	AUC	Bal Acc	FPR	FNR	FDR	FOR
Hybrid	Direct C HNR*	**0.60 ± 0.07**	**0.56 ± 0.07**	0.44 ± 0.13	0.45 ± 0.10	0.50 ± 0.10	**0.39 ± 0.08**
Hybrid	Composed indirect C HWR*	0.58 ± 0.08	**0.56 ± 0.08**	0.37 ± 0.10	0.50 ± 0.10	**0.47 ± 0.15**	0.40 ± 0.10
Hybrid	Direct C HWR HNR*	0.58 ± 0.07	0.55 ± 0.08	0.47 ± 0.14	0.43 ± 0.09	0.50 ± 0.09	0.40 ± 0.10
Hybrid	Indirect C HWR*	0.58 ± 0.08	**0.56 ± 0.08**	0.37 ± 0.10	0.50 ± 0.10	**0.47 ± 0.15**	0.40 ± 0.10
Hybrid	Composed indirect C HNR	0.56 ± 0.09	0.54 ± 0.06	0.45 ± 0.07	0.47 ± 0.09	0.52 ± 0.07	0.40 ± 0.08
Hybrid	Indirect C HNR	0.56 ± 0.09	0.54 ± 0.06	0.45 ± 0.07	0.47 ± 0.09	0.52 ± 0.07	0.40 ± 0.08
Hybrid	Direct C HWR	0.53 ± 0.06	0.52 ± 0.04	0.37 ± 0.09	0.58 ± 0.09	0.52 ± 0.14	0.44 ± 0.08
Hybrid	Weighted average of C HNR	0.51 ± 0.05	0.50 ± 0.04	0.38 ± 0.25	0.63 ± 0.3	0.56 ± 0.22	0.43 ± 0.07
Hybrid	Weighted average of C HWR HNR	0.50 ± 0.04	0.50 ± 0.05	**0.23 ± 0.07**	0.77 ± 0.13	0.58 ± 0.09	0.45 ± 0.11
Hybrid	Weighted average of C HWR	0.47 ± 0.04	0.49 ± 0.03	0.39 ± 0.26	0.63 ± 0.26	0.56 ± 0.12	0.46 ± 0.11
Single	Predict GT from features and HNR	0.59 ± 0.07	0.55 ± 0.06	0.44 ± 0.09	0.46 ± 0.10	0.51 ± 0.09	**0.39 ± 0.07**
Single	HNR (1-10 scale)	0.56 ± 0.05	0.52 ± 0.02	0.55 ± 0.08	0.40 ± 0.08	0.54 ± 0.04	0.41 ± 0.07
Single	Predict GT from features and HWR	0.54 ± 0.06	0.54 ± 0.05	0.35 ± 0.10	0.57 ± 0.08	0.49 ± 0.14	0.42 ± 0.09
Single	HWR (1-10 scale)	0.54 ± 0.04	0.52 ± 0.03	0.54 ± 0.05	0.41 ± 0.04	0.53 ± 0.09	0.43 ± 0.10
Single	Predict GT from features and C	0.51 ± 0.07	0.52 ± 0.05	0.41 ± 0.07	0.55 ± 0.08	0.52 ± 0.11	0.44 ± 0.10
Single	C (1-10 scale)	0.49 ± 0.06	0.48 ± 0.01	0.40 ± 0.07	0.65 ± 0.08	0.59 ± 0.06	0.46 ± 0.04
Single	C (binarized >=5)	-	0.48 ± 0.01	0.40 ± 0.07	0.65 ± 0.08	0.59 ± 0.06	0.46 ± 0.04
Single	HNR (binarized >=5)	-	0.52 ± 0.01	0.60 ± 0.07	**0.35 ± 0.08**	0.54 ± 0.04	0.41 ± 0.06
Single	HWR (binarized >=5)	-	0.51 ± 0.03	0.63 ± 0.05	0.36 ± 0.05	0.54 ± 0.07	0.44 ± 0.12
None	Predict GT from features	0.52 ± 0.07	0.51 ± 0.06	0.37 ± 0.07	0.61 ± 0.09	0.54 ± 0.13	0.45 ± 0.09
Random	Randomly pick between C HNR	0.55 ± 0.08	0.52 ± 0.05	0.46 ± 0.06	0.50 ± 0.08	0.54 ± 0.07	0.42 ± 0.07
Random	Randomly pick between C HWR	0.54 ± 0.07	0.52 ± 0.06	0.46 ± 0.06	0.49 ± 0.11	0.53 ± 0.14	0.43 ± 0.08
Random	Randomly pick between C HWR HNR	0.54 ± 0.06	0.52 ± 0.06	0.50 ± 0.07	0.46 ± 0.08	0.53 ± 0.10	0.43 ± 0.11

Table 5.7: Performance by subgroup (African-Americans) of hybrid models presented in Table 5.6. Best results in cyan and bolded.

Type	Model	AUC	Bal Acc	FPR	FNR	FDR	FOR
Hybrid	Direct C HNR	**0.65 ± 0.06**	0.58 ± 0.07	0.48 ± 0.15	0.35 ± 0.13	0.46 ± 0.11	0.37 ± 0.10
Hybrid	Direct C HWR HNR	0.63 ± 0.07	**0.59 ± 0.07**	0.50 ± 0.15	**0.32 ± 0.11**	0.45 ± 0.09	**0.36 ± 0.13**
Hybrid	Composed indirect C HWR	0.57 ± 0.12	0.58 ± 0.10	0.41 ± 0.15	0.43 ± 0.14	0.43 ± 0.17	0.41 ± 0.12
Hybrid	Indirect C HWR	0.57 ± 0.12	0.58 ± 0.10	0.41 ± 0.15	0.43 ± 0.14	0.43 ± 0.17	0.41 ± 0.12
Hybrid	Weighted average of C HNR	0.55 ± 0.06	0.51 ± 0.07	0.33 ± 0.19	0.64 ± 0.31	0.65 ± 0.21	0.42 ± 0.12
Hybrid	Weighted average of C HWR HNR	0.55 ± 0.08	0.55 ± 0.08	**0.12 ± 0.09**	0.78 ± 0.15	**0.35 ± 0.21**	0.45 ± 0.15
Hybrid	Composed indirect C HNR	0.53 ± 0.09	0.52 ± 0.07	0.54 ± 0.10	0.41 ± 0.07	0.52 ± 0.07	0.44 ± 0.13
Hybrid	Indirect C HNR	0.53 ± 0.09	0.52 ± 0.07	0.54 ± 0.10	0.41 ± 0.07	0.52 ± 0.07	0.44 ± 0.13
Hybrid	Direct C HWR	0.51 ± 0.07	0.51 ± 0.07	0.40 ± 0.14	0.57 ± 0.12	0.50 ± 0.17	0.48 ± 0.10
Hybrid	Weighted average of C HWR	0.48 ± 0.09	0.49 ± 0.07	0.37 ± 0.27	0.65 ± 0.26	0.50 ± 0.16	0.51 ± 0.15
Single	Predict GT from features and HNR	0.64 ± 0.06	0.56 ± 0.06	0.48 ± 0.13	0.39 ± 0.13	0.48 ± 0.10	0.39 ± 0.09
Single	HNR (1-10 scale)	0.55 ± 0.07	0.56 ± 0.05	0.46 ± 0.09	0.42 ± 0.15	0.49 ± 0.08	0.39 ± 0.09
Single	HWR (1-10 scale)	0.53 ± 0.08	0.53 ± 0.06	0.47 ± 0.08	0.47 ± 0.11	0.49 ± 0.13	0.46 ± 0.13
Single	Predict GT from features and HWR	0.52 ± 0.08	0.53 ± 0.08	0.36 ± 0.15	0.57 ± 0.13	0.46 ± 0.17	0.46 ± 0.12
Single	Predict GT from features and C	0.49 ± 0.11	0.50 ± 0.06	0.48 ± 0.12	0.52 ± 0.13	0.52 ± 0.13	0.49 ± 0.12
Single	C (1-10 scale)	0.46 ± 0.06	0.44 ± 0.05	0.51 ± 0.10	0.60 ± 0.16	0.61 ± 0.09	0.51 ± 0.08
Single	C (binarized >=5)	-	0.44 ± 0.05	0.51 ± 0.10	0.60 ± 0.16	0.61 ± 0.09	0.51 ± 0.08
Single	HNR (binarized >=5)	-	0.56 ± 0.05	0.49 ± 0.10	0.40 ± 0.16	0.49 ± 0.08	0.39 ± 0.09
Single	HWR (binarized >=5)	-	0.51 ± 0.06	0.57 ± 0.11	0.42 ± 0.12	0.51 ± 0.11	0.48 ± 0.16
None	Predict GT from features	0.49 ± 0.10	0.48 ± 0.10	0.42 ± 0.13	0.62 ± 0.12	0.54 ± 0.19	0.50 ± 0.11
Random	Randomly pick between C HWR	0.59 ± 0.06	0.57 ± 0.07	0.47 ± 0.08	0.40 ± 0.13	0.46 ± 0.14	0.41 ± 0.11
Random	Randomly pick between C HWR HNR	0.54 ± 0.07	0.53 ± 0.06	0.48 ± 0.09	0.47 ± 0.11	0.49 ± 0.11	0.46 ± 0.13
Random	Randomly pick between C HNR	0.53 ± 0.13	0.50 ± 0.10	0.53 ± 0.11	0.47 ± 0.11	0.54 ± 0.07	0.46 ± 0.15

Table 5.8: Performance by subgroup (whites) of hybrid models presented in Table 5.6. Best results in cyan and bolded.

Type	Model	AUC	Bal Acc	FPR	FNR	FDR	FOR
Hybrid	Composed indirect C HWR	0.59 ± 0.09	0.55 ± 0.08	**0.30 ± 0.14**	0.61 ± 0.11	0.49 ± 0.21	0.39 ± 0.10
Hybrid	Indirect C HWR	0.59 ± 0.09	0.55 ± 0.08	**0.30 ± 0.14**	0.61 ± 0.11	0.49 ± 0.21	0.39 ± 0.10
Hybrid	Composed indirect C HNR	0.56 ± 0.2	0.52 ± 0.17	0.41 ± 0.16	0.56 ± 0.2	0.55 ± 0.19	0.42 ± 0.18
Hybrid	Direct C HWR	0.56 ± 0.08	**0.58 ± 0.09**	0.31 ± 0.18	0.53 ± 0.17	**0.45 ± 0.24**	**0.36 ± 0.12**
Hybrid	Indirect C HNR	0.56 ± 0.2	0.52 ± 0.17	0.41 ± 0.16	0.56 ± 0.2	0.55 ± 0.19	0.42 ± 0.18
Hybrid	Direct C HNR	0.53 ± 0.17	0.52 ± 0.13	0.39 ± 0.11	0.57 ± 0.21	0.56 ± 0.17	0.42 ± 0.15
Hybrid	Direct C HWR HNR	0.52 ± 0.19	0.50 ± 0.13	0.43 ± 0.14	0.56 ± 0.2	0.57 ± 0.17	0.44 ± 0.14
Hybrid	Weighted average of C HWR	0.48 ± 0.11	0.48 ± 0.06	0.42 ± 0.28	0.61 ± 0.25	0.59 ± 0.19	0.45 ± 0.14
Hybrid	Weighted average of C HWR HNR	0.46 ± 0.08	0.44 ± 0.05	0.36 ± 0.10	0.76 ± 0.19	0.72 ± 0.13	0.46 ± 0.11
Hybrid	Weighted average of C HNR	0.44 ± 0.08	0.46 ± 0.06	0.47 ± 0.33	0.60 ± 0.3	0.55 ± 0.21	0.51 ± 0.21
Single	Predict GT from features and HWR	0.59 ± 0.09	**0.58 ± 0.08**	0.32 ± 0.15	0.52 ± 0.12	0.46 ± 0.21	**0.36 ± 0.1**
Single	Predict GT from features and C	0.56 ± 0.14	0.55 ± 0.12	0.34 ± 0.10	0.56 ± 0.23	0.53 ± 0.16	0.39 ± 0.14
Single	HWR (1-10 scale)	0.56 ± 0.12	0.53 ± 0.11	0.60 ± 0.18	0.34 ± 0.18	0.55 ± 0.12	0.39 ± 0.22
Single	Predict GT from features and HNR	0.53 ± 0.19	0.53 ± 0.15	0.41 ± 0.12	0.54 ± 0.22	0.55 ± 0.17	0.41 ± 0.17
Single	HNR (1-10 scale)	0.53 ± 0.15	0.46 ± 0.10	0.67 ± 0.15	0.42 ± 0.16	0.59 ± 0.09	0.51 ± 0.25
Single	C (1-10 scale)	0.52 ± 0.11	0.48 ± 0.09	0.35 ± 0.15	0.69 ± 0.17	0.60 ± 0.21	0.44 ± 0.10
Single	C (binarized >=5)	-	0.48 ± 0.09	0.35 ± 0.15	0.69 ± 0.17	0.60 ± 0.21	0.44 ± 0.10
Single	HNR (binarized >=5)	-	0.47 ± 0.07	0.72 ± 0.13	0.34 ± 0.14	0.58 ± 0.08	0.50 ± 0.24
Single	HWR (binarized >=5)	-	0.52 ± 0.09	0.65 ± 0.15	**0.31 ± 0.17**	0.56 ± 0.10	0.40 ± 0.21
None	Predict GT from features	0.55 ± 0.14	0.55 ± 0.12	0.37 ± 0.11	0.53 ± 0.2	0.51 ± 0.16	0.39 ± 0.14
Random	Randomly pick between C HNR	**0.61 ± 0.14**	0.57 ± 0.13	0.37 ± 0.11	0.49 ± 0.18	0.49 ± 0.14	0.38 ± 0.17
Random	Randomly pick between C HWR HNR	0.54 ± 0.12	0.50 ± 0.14	0.54 ± 0.13	0.45 ± 0.18	0.57 ± 0.14	0.42 ± 0.17
Random	Randomly pick between C HWR	0.49 ± 0.08	0.47 ± 0.08	0.47 ± 0.12	0.58 ± 0.13	0.61 ± 0.17	0.45 ± 0.09

Table 5.9: Performance by subgroup (other races) of hybrid models presented in Table 5.6. Best results in cyan and bolded.

Type	Model	AUC	Bal Acc	FPR	FNR	FDR	FOR
Hybrid	Weighted average of C HNR	0.64 ± 0.33	0.58 ± 0.23	0.47 ± 0.33	0.38 ± 0.44	0.63 ± 0.34	0.22 ± 0.25
Hybrid	Composed indirect C HNR	0.62 ± 0.32	0.59 ± 0.24	0.29 ± 0.26	0.53 ± 0.39	0.52 ± 0.38	0.29 ± 0.21
Hybrid	Indirect C HNR	0.62 ± 0.32	0.59 ± 0.24	0.29 ± 0.26	0.53 ± 0.39	0.52 ± 0.38	0.29 ± 0.21
Hybrid	Weighted average of C HWR HNR	0.50 ± 0.25	0.57 ± 0.21	0.58 ± 0.37	0.29 ± 0.3	0.52 ± 0.27	0.33 ± 0.37
Hybrid	Weighted average of C HWR	0.49 ± 0.24	0.52 ± 0.14	0.64 ± 0.26	0.32 ± 0.39	0.59 ± 0.23	0.29 ± 0.37
Hybrid	Direct C HWR HNR	0.46 ± 0.22	0.44 ± 0.22	0.47 ± 0.31	0.66 ± 0.35	0.74 ± 0.33	0.48 ± 0.29
Hybrid	Direct C HNR	0.43 ± 0.22	0.48 ± 0.18	0.32 ± 0.28	0.72 ± 0.25	0.61 ± 0.42	0.39 ± 0.13
Hybrid	Direct C HWR	0.43 ± 0.17	0.39 ± 0.16	0.37 ± 0.23	0.85 ± 0.16	0.72 ± 0.37	0.52 ± 0.15
Hybrid	Composed indirect C HWR	0.39 ± 0.24	0.44 ± 0.18	0.41 ± 0.31	0.70 ± 0.31	0.68 ± 0.37	0.51 ± 0.25
Hybrid	Indirect C HWR	0.39 ± 0.24	0.44 ± 0.18	0.41 ± 0.31	0.70 ± 0.31	0.68 ± 0.37	0.51 ± 0.25
Single	HNR (1-10 scale)	**0.65 ± 0.22**	0.59 ± 0.16	0.60 ± 0.17	0.21 ± 0.25	0.58 ± 0.2	0.25 ± 0.27
Single	Predict GT from features and HWR	0.50 ± 0.18	0.47 ± 0.16	0.3 ± 0.22	0.75 ± 0.19	0.57 ± 0.36	0.47 ± 0.18
Single	Predict GT from features and HNR	0.47 ± 0.26	0.47 ± 0.19	0.31 ± 0.25	0.75 ± 0.27	0.67 ± 0.44	0.38 ± 0.11
Single	HWR (1-10 scale)	0.44 ± 0.26	0.43 ± 0.22	0.71 ± 0.23	0.44 ± 0.28	0.59 ± 0.24	0.52 ± 0.34
Single	Predict GT from features and C	0.36 ± 0.31	0.49 ± 0.2	0.3 ± 0.18	0.71 ± 0.37	0.67 ± 0.37	0.35 ± 0.2
Single	C (1-10 scale)	0.33 ± 0.2	0.47 ± 0.14	**0.21 ± 0.16**	0.85 ± 0.17	0.67 ± 0.41	0.50 ± 0.21
Single	C (binarized >=5)	-	0.39 ± 0.17	0.35 ± 0.22	0.88 ± 0.25	0.88 ± 0.25	0.35 ± 0.14
Single	HNR (binarized >=5)	-	**0.61 ± 0.17**	0.65 ± 0.22	**0.12 ± 0.25**	0.65 ± 0.14	**0.12 ± 0.25**
Single	HWR (binarized >=5)	-	0.53 ± 0.14	0.79 ± 0.16	0.15 ± 0.17	**0.50 ± 0.21**	0.33 ± 0.41
None	Predict GT from features	0.48 ± 0.39	0.54 ± 0.26	0.3 ± 0.2	0.62 ± 0.42	0.61 ± 0.42	0.32 ± 0.23
Random	Randomly pick between C HWR HNR	0.36 ± 0.28	0.44 ± 0.2	0.53 ± 0.22	0.59 ± 0.29	0.64 ± 0.25	0.46 ± 0.25
Random	Randomly pick between C HWR	0.32 ± 0.2	0.34 ± 0.14	0.35 ± 0.21	0.96 ± 0.09	0.86 ± 0.38	0.55 ± 0.16
Random	Randomly pick between C HNR	0.3 ± 0.33	0.33 ± 0.24	0.51 ± 0.32	0.83 ± 0.22	0.75 ± 0.35	0.53 ± 0.22

Table 5.10: Extended result table for test-set performance of hybrid models trained on *all* defendants, not just defendants whose COMPAS and Human scores disagree. Best results in cyan and bolded. See Table 5.5 for a reduced version of this table. Rows marked with * are the rows labeled as *best* in Table 5.5.

Type	Model	AUC	Bal Acc	FPR	FNR	FDR	FOR
Oracle	Benevolent oracle	0.85 ± 0.03	0.81 ± 0.02	0.19 ± 0.04	0.19 ± 0.03	0.20 ± 0.04	0.18 ± 0.03
Oracle	Adversarial oracle	0.57 ± 0.03	0.51 ± 0.02	0.50± 0.03	0.49 ± 0.05	0.53 ± 0.03	0.46 ± 0.04
Hybrid	Weighted average of C HNR*	**0.74 ± 0.03**	0.65 ± 0.06	0.41 ± 0.21	0.29 ± 0.1	0.38 ± 0.06	**0.30 ± 0.04**
Hybrid	Weighted average of C HWR*	**0.74 ± 0.04**	0.65 ± 0.06	0.40 ± 0.2	0.29 ± 0.11	0.36 ± 0.06	0.31 ± 0.05
Hybrid	Direct C HWR HNR*	0.73 ± 0.03	0.66 ± 0.03	0.32 ± 0.05	0.36 ± 0.05	0.35 ± 0.05	0.34 ± 0.05
Hybrid	Direct C HNR	0.72 ± 0.04	0.65 ± 0.03	0.34 ± 0.05	0.36 ± 0.06	0.38 ± 0.04	0.32 ± 0.04
Hybrid	Direct C HWR	0.72 ± 0.03	0.65 ± 0.03	0.32 ± 0.06	0.38 ± 0.06	0.35 ± 0.06	0.35 ± 0.05
Single	HNR (1-10 scale)	0.72 ± 0.03	0.66 ± 0.03	0.35 ± 0.04	0.32 ± 0.04	0.37 ± 0.03	**0.30 ± 0.03**
Single	HWR (1-10 scale)	0.72 ± 0.03	0.66 ± 0.02	0.36 ± 0.04	0.31 ± 0.03	0.36 ± 0.04	0.32 ± 0.04
Single	C (1-10 scale)	0.71 ± 0.03	0.65 ± 0.03	0.32 ± 0.03	0.38 ± 0.06	0.37 ± 0.03	0.33 ± 0.04
Single	Predict GT from features and C	0.71 ± 0.03	0.64 ± 0.04	0.35 ± 0.04	0.36 ± 0.06	0.38 ± 0.04	0.33 ± 0.05
Single	Predict GT from features and HWR	0.71 ± 0.03	**0.67 ± 0.03**	**0.31 ± 0.05**	0.36 ± 0.06	**0.34 ± 0.05**	0.33 ± 0.06
Single	Predict GT from features and HNR	0.70± 0.03	0.64 ± 0.02	0.35 ± 0.04	0.37 ± 0.05	0.39 ± 0.03	0.33 ± 0.04
Single	C (binarized >=5)	-	0.65 ± 0.03	0.32 ± 0.03	0.38 ± 0.06	0.37 ± 0.03	0.33 ± 0.04
Single	HNR (binarized >=5)	-	0.66 ± 0.03	0.38 ± 0.04	0.30 ± 0.04	0.38 ± 0.03	**0.30 ± 0.04**
Single	HWR (binarized >=5)	-	0.66 ± 0.03	0.40 ± 0.04	0.28 ± 0.04	0.37 ± 0.04	0.31 ± 0.05
None	Predict GT from features	0.69 ± 0.02	0.63 ± 0.03	0.37 ± 0.05	0.37 ± 0.06	0.40 ± 0.04	0.34 ± 0.04
Random	Randomly pick between C HWR	0.73 ± 0.04	**0.67 ± 0.03**	0.34 ± 0.03	0.32 ± 0.03	0.35 ± 0.05	0.32 ± 0.04
Random	Randomly pick between C HNR	0.72 ± 0.04	0.66 ± 0.04	0.33 ± 0.03	0.34 ± 0.05	0.36 ± 0.04	0.31 ± 0.04
Random	Randomly pick between C HWR HNR	0.71 ± 0.03	**0.67 ± 0.03**	0.35 ± 0.03	0.31 ± 0.05	0.37 ± 0.03	**0.30 ± 0.04**

Table 5.11: Performance by subgroup (African-Americans) of hybrid models presented in Table 5.10. Best results in cyan and bolded.

Type	Model	AUC	Bal Acc	FPR	FNR	FDR	FOR
Oracle	Benevolent oracle	0.85 ± 0.03	0.81 ± 0.03	0.23 ± 0.03	0.15 ± 0.04	0.18 ± 0.03	0.19 ± 0.05
Oracle	Adversarial oracle	0.54 ± 0.07	0.49 ± 0.05	0.60± 0.08	0.42 ± 0.04	0.46 ± 0.05	0.56 ± 0.08
Hybrid	Direct C HNR	**0.73 ± 0.06**	0.65 ± 0.05	0.47 ± 0.09	0.23 ± 0.08	0.34 ± 0.05	0.34 ± 0.1
Hybrid	Weighted average of C HNR	**0.73 ± 0.05**	0.65 ± 0.07	0.48 ± 0.18	0.22 ± 0.09	0.33 ± 0.05	0.30 ± 0.14
Hybrid	Direct C HWR HNR	0.72 ± 0.03	0.65 ± 0.03	0.44 ± 0.05	0.27 ± 0.06	0.30 ± 0.04	0.41 ± 0.07
Hybrid	Weighted average of C HWR	0.72 ± 0.04	0.64 ± 0.06	0.47 ± 0.19	0.25 ± 0.1	0.30 ± 0.05	0.41 ± 0.07
Hybrid	Weighted average of C HWR HNR	0.71 ± 0.05	0.62 ± 0.07	0.59 ± 0.24	**0.17 ± 0.13**	0.36 ± 0.07	**0.23 ± 0.18**
Hybrid	Direct C HWR	0.70± 0.04	0.62 ± 0.02	0.47 ± 0.07	0.29 ± 0.06	0.32 ± 0.05	0.43 ± 0.05
Single	Predict GT from features and HNR	0.71 ± 0.05	0.63 ± 0.04	0.49 ± 0.08	0.24 ± 0.08	0.35 ± 0.05	0.36 ± 0.1
Single	HNR (1-10 scale)	0.71 ± 0.04	**0.68 ± 0.04**	**0.39 ± 0.06**	0.26 ± 0.05	0.30 ± 0.04	0.34 ± 0.07
Single	Predict GT from features and C	0.70± 0.04	0.62 ± 0.04	0.49 ± 0.08	0.27 ± 0.06	0.32 ± 0.05	0.43 ± 0.08
Single	HWR (1-10 scale)	0.70± 0.03	0.65 ± 0.03	0.43 ± 0.05	0.27 ± 0.04	**0.29 ± 0.04**	0.41 ± 0.05
Single	C (1-10 scale)	0.69 ± 0.05	0.63 ± 0.04	0.43 ± 0.06	0.31 ± 0.07	0.34 ± 0.05	0.39 ± 0.07
Single	Predict GT from features and HWR	0.69 ± 0.04	0.64 ± 0.04	0.45 ± 0.07	0.26 ± 0.06	0.30 ± 0.05	0.40 ± 0.07
Single	C (binarized >=5)	-	0.63 ± 0.04	0.43 ± 0.06	0.31 ± 0.07	0.34 ± 0.05	0.39 ± 0.07
Single	HNR (binarized >=5)	-	0.67 ± 0.04	0.41 ± 0.07	0.25 ± 0.05	0.32 ± 0.05	0.34 ± 0.07
Single	HWR (binarized >=5)	-	0.64 ± 0.03	0.48 ± 0.05	0.24 ± 0.05	0.31 ± 0.04	0.39 ± 0.07
None	Predict GT from features	0.69 ± 0.04	0.62 ± 0.04	0.52 ± 0.06	0.23 ± 0.08	0.36 ± 0.05	0.36 ± 0.11
Random	Randomly pick between C HWR	0.72 ± 0.04	0.66 ± 0.03	0.44 ± 0.04	0.24 ± 0.04	**0.29 ± 0.04**	0.38 ± 0.05
Random	Randomly pick between C HNR	0.70± 0.04	0.65 ± 0.04	0.42 ± 0.05	0.27 ± 0.05	0.32 ± 0.04	0.36 ± 0.07
Random	Randomly pick between C HWR HNR	0.70± 0.05	0.66 ± 0.04	0.42 ± 0.07	0.25 ± 0.05	0.32 ± 0.05	0.35 ± 0.08

Table 5.12: Performance by subgroup (whites) of hybrid models presented in Table 5.10. Best results in cyan and bolded.

Type	Model	AUC	Bal Acc	FPR	FNR	FDR	FOR
Oracle	Benevolent oracle	0.84 ± 0.04	0.8 ± 0.03	0.13 ± 0.06	0.28 ± 0.04	0.23 ± 0.07	0.16 ± 0.05
Oracle	Adversarial oracle	0.55 ± 0.05	0.48 ± 0.05	0.42 ± 0.06	0.61 ± 0.09	0.63 ± 0.08	0.4 ± 0.07
Hybrid	Weighted average of C HWR	**0.75 ± 0.05**	0.65 ± 0.06	0.34 ± 0.22	0.37 ± 0.14	0.46 ± 0.1	**0.23 ± 0.1**
Hybrid	Weighted average of C HWR HNR	0.74 ± 0.05	0.57 ± 0.08	0.64 ± 0.36	**0.21 ± 0.22**	0.55 ± 0.09	**0.23 ± 0.31**
Hybrid	Weighted average of C HNR	0.72 ± 0.03	0.62 ± 0.07	0.35 ± 0.23	0.4 ± 0.16	0.46 ± 0.1	0.26 ± 0.11
Hybrid	Direct C HWR	0.70± 0.04	0.62 ± 0.05	0.21 ± 0.07	0.54 ± 0.1	0.43 ± 0.12	0.29 ± 0.06
Hybrid	Direct C HWR HNR	0.70± 0.04	0.63 ± 0.04	0.21 ± 0.05	0.53 ± 0.08	0.44 ± 0.1	0.29 ± 0.06
Hybrid	Direct C HNR	0.67 ± 0.05	0.62 ± 0.04	0.22 ± 0.04	0.55 ± 0.07	0.43 ± 0.1	0.31 ± 0.05
Single	HWR (1-10 scale)	0.74 ± 0.05	**0.67 ± 0.05**	0.29 ± 0.07	0.38 ± 0.06	0.44 ± 0.08	0.24 ± 0.07
Single	HNR (1-10 scale)	0.70± 0.04	0.62 ± 0.05	0.33 ± 0.04	0.43 ± 0.06	0.47 ± 0.05	0.29 ± 0.07
Single	C (1-10 scale)	0.69 ± 0.06	0.63 ± 0.05	0.24 ± 0.05	0.50± 0.12	0.44 ± 0.1	0.29 ± 0.05
Single	Predict GT from features and HWR	0.69 ± 0.05	0.63 ± 0.06	0.2 ± 0.06	0.54 ± 0.09	0.43 ± 0.12	0.29 ± 0.06
Single	Predict GT from features and C	0.67 ± 0.05	0.63 ± 0.03	**0.19 ± 0.05**	0.55 ± 0.05	**0.4 ± 0.09**	0.3 ± 0.05
Single	Predict GT from features and HNR	0.66 ± 0.06	0.61 ± 0.05	0.23 ± 0.07	0.56 ± 0.08	0.45 ± 0.11	0.32 ± 0.05
Single	C (binarized >=5)	-	0.63 ± 0.05	0.24 ± 0.05	0.50± 0.12	0.44 ± 0.1	0.29 ± 0.05
Single	HNR (binarized >=5)	-	0.63 ± 0.05	0.36 ± 0.04	0.38 ± 0.06	0.47 ± 0.05	0.27 ± 0.07
Single	HWR (binarized >=5)	-	0.66 ± 0.04	0.31 ± 0.07	0.36 ± 0.05	0.45 ± 0.07	0.24 ± 0.07
None	Predict GT from features	0.65 ± 0.05	0.60± 0.05	0.22 ± 0.08	0.57 ± 0.08	0.44 ± 0.13	0.32 ± 0.05
Random	Randomly pick between C HWR	0.73 ± 0.04	0.64 ± 0.03	0.26 ± 0.07	0.46 ± 0.05	0.45 ± 0.08	0.27 ± 0.06
Random	Randomly pick between C HNR	0.72 ± 0.06	0.65 ± 0.06	0.25 ± 0.03	0.45 ± 0.09	0.42 ± 0.07	0.28 ± 0.06
Random	Randomly pick between C HWR HNR	0.70± 0.03	0.65 ± 0.05	0.3 ± 0.05	0.4 ± 0.08	0.44 ± 0.07	0.27 ± 0.06

Table 5.13: Performance by subgroup (other races) of hybrid models presented in Table 5.10. Best results in cyan and bolded.

Type	Model	AUC	Bal Acc	FPR	FNR	FDR	FOR
Oracle	Benevolent oracle	0.84 ± 0.12	0.82 ± 0.09	0.09 ± 0.1	0.27 ± 0.17	0.16 ± 0.17	0.14 ± 0.12
Oracle	Adversarial oracle	0.48 ± 0.1	0.46 ± 0.1	0.45 ± 0.14	0.62 ± 0.17	0.70± 0.1	0.38 ± 0.17
Hybrid	Weighted average of C HWR HNR	**0.76 ± 0.15**	**0.74 ± 0.18**	**0.18 ± 0.12**	0.33 ± 0.37	**0.41 ± 0.32**	**0.14 ± 0.13**
Hybrid	Weighted average of C HNR	0.75 ± 0.15	0.68 ± 0.17	0.30 ± 0.27	0.34 ± 0.29	0.44 ± 0.27	0.25 ± 0.28
Hybrid	Direct C HNR	0.69 ± 0.12	0.56 ± 0.17	0.30 ± 0.1	0.58 ± 0.28	0.62 ± 0.24	0.30 ± 0.16
Hybrid	Direct C HWR	0.69 ± 0.15	0.59 ± 0.12	0.21 ± 0.14	0.61 ± 0.24	0.55 ± 0.3	0.29 ± 0.14
Hybrid	Direct C HWR HNR	0.68 ± 0.08	0.58 ± 0.08	0.27 ± 0.09	0.58 ± 0.2	0.58 ± 0.21	0.28 ± 0.15
Hybrid	Weighted average of C HWR	0.66 ± 0.07	0.62 ± 0.07	0.31 ± 0.12	0.44 ± 0.13	0.52 ± 0.17	0.25 ± 0.11
Single	HNR (1-10 scale)	0.73 ± 0.16	0.69 ± 0.14	0.32 ± 0.17	0.31 ± 0.2	0.44 ± 0.19	0.21 ± 0.18
Single	Predict GT from features and C	0.69 ± 0.12	0.56 ± 0.17	0.29 ± 0.08	0.59 ± 0.27	0.60 ± 0.21	0.30 ± 0.15
Single	Predict GT from features and HNR	0.69 ± 0.14	0.60± 0.21	0.26 ± 0.13	0.54 ± 0.37	0.55 ± 0.3	0.29 ± 0.19
Single	Predict GT from features and HWR	0.67 ± 0.14	0.65 ± 0.1	0.22 ± 0.12	0.48 ± 0.17	0.45 ± 0.19	0.25 ± 0.13
Single	HWR (1-10 scale)	0.66 ± 0.07	0.61 ± 0.05	0.37 ± 0.1	0.41 ± 0.12	0.54 ± 0.14	0.26 ± 0.1
Single	C (1-10 scale)	0.64 ± 0.11	0.61 ± 0.09	0.20 ± 0.1	0.57 ± 0.14	0.46 ± 0.13	0.28 ± 0.14
Single	C (binarized >=5)	-	0.61 ± 0.09	0.20 ± 0.1	0.57 ± 0.14	0.46 ± 0.13	0.28 ± 0.14
Single	HNR (binarized >=5)	-	0.71 ± 0.14	0.34 ± 0.17	**0.24 ± 0.2**	0.43 ± 0.19	0.18 ± 0.17
Single	HWR (binarized >=5)	-	0.64 ± 0.08	0.43 ± 0.11	0.30 ± 0.14	0.54 ± 0.11	0.23 ± 0.12
None	Predict GT from features	0.68 ± 0.16	0.57 ± 0.21	0.32 ± 0.14	0.53 ± 0.35	0.59 ± 0.28	0.30 ± 0.19
Random	Randomly pick between C HWR HNR	0.66 ± 0.13	0.60± 0.09	0.30 ± 0.11	0.51 ± 0.14	0.53 ± 0.14	0.28 ± 0.15
Random	Randomly pick between C HNR	0.62 ± 0.24	0.63 ± 0.18	0.26 ± 0.17	0.48 ± 0.28	0.48 ± 0.28	0.26 ± 0.18
Random	Randomly pick between C HWR	0.58 ± 0.14	0.55 ± 0.08	0.28 ± 0.12	0.61 ± 0.15	0.57 ± 0.12	0.32 ± 0.14

CHAPTER 6

CONCLUSION

With the goal of helping creators as well as users of machine learning models increase their trust and understanding of the models, this dissertation developed new interpretability approaches to open up black-box machine learning models.

The first part of this dissertation proposed new post-hoc, global explanations for black-box models, developed using model-agnostic distillation techniques or by leveraging known structure specific to the black-box model. In Chapter 2, we proposed a distillation approach to learn global additive explanations that describe the relationship between input features and model predictions, showing that distilled additive explanations have fidelity, accuracy, and interpretability advantages over non-additive explanations, via a user study with expert users. In Chapter 3, we worked specifically on tree ensembles, leveraging tree structure to construct a similarity metric for gradient boosted tree models. We used this similarity metric to select prototypical observations in each class, presenting an alternative to other tree ensemble interpretability methods such as seeking one tree that best represents the ensemble or feature importance methods.

The second part of this dissertation studied the use of interpretability approaches to probe and debug black-box models in algorithmic fairness settings. In Chapter 4, we proposed Distill-and-Compare, an approach to probe such risk scoring models by leveraging additional information on ground-truth outcomes that the risk scoring model was intended to predict. We found that interpretability approaches can help uncover previously unknown sources of bias. Finally, in

Chapter 5, we provided a concrete case study using the interpretability methods proposed in this dissertation to debug black-box models, in this case, a hybrid Human + Machine recidivism prediction model. Our methods revealed that human and COMPAS decision making anchored on the same features, and hence did not differ significantly enough to harness the promise of hybrid Human + Machine decision making, concluding this dissertation on interpretability approaches for real-world settings.

We highlight one compelling reason to investigate the use of interpretable models in algorithmic fairness settings. In settings where specific biases may not be *a priori* known, interpretability approaches that do not require pre-defining features to audit (since bias may exist not just in features such as race or gender, but in other seemingly innocuous features) may be useful to suggest areas of potential bias that did not previously come to mind but warrant more investigation. For example, Distill-and-Compare suggested that COMPAS predicted recidivism risk for younger and older age groups (feature regions that we had not suspected of bias) to be significantly different than that for true recidivism outcomes. This then allowed us to go back to the data and attempt to generate possible explanations for this discrepancy that we could then further investigate. When deploying this approach initially on the UCI German credit data[1], after training a transparent student model on the true outcome, we found our error bars for the effect for native Germans much larger than that for foreign nationals. A quick examination of the data revealed that the data comprises mostly foreign nationals, with only a handful of German nationals, suggesting that this data is drawn from a very specific population that likely is not representative of the population one wishes to study when investigating possible bias in issuing

[1] https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data

loans. Hence, interpretable methods for bias detection could be useful when there are likely many sources of biases that may be *a priori* not known.

Several research directions remain to be explored in the field of interpretability. To date, the definition of interpretability in machine learning is still the subject of debate. One such definition is axiomatic, with certain classes of machine learning models considered to be interpretable, while others are not. Another definition grounds interpretability in specific human tasks, where if the human performs better at the task, then the explanation the human used to perform the task must have been more interpretable. Yet another definition is couched in terms of proxies or characteristics (e.g. sparsity, simplicity, etc.) that are easy to evaluate without user studies. Recently, some research groups have started recruiting Mechanical Turkers for user studies to probe the definitions of interpretability, by varying interpretability proxies (e.g. number of features in a linear model) and observing how Turkers performance on simplistic prediction tasks change [111, 105]. However, there are fewer user studies with doctors or judges who actually interact with machine learning models to make decisions, perhaps due to the cost and effort of recruiting domain experts. It would be interesting to conduct a large-scale user study on domain experts, performing prediction tasks that they would naturally perform in the setting in which they use machine learning models.

Finally, we close this dissertation by putting forward the viewpoint that interpretability approaches should not be viewed as a panacea to all the problems arising from training machine learning models on real-world data, with all of its limitations and biases, but rather yet another useful tool in a responsible machine learning pipeline.

APPENDIX A

PUBLICATIONS

Besides the papers on which the chapters of this dissertation are based on, the following papers were produced during the course of this PhD:

- Xuezhou Zhang, **Sarah Tan**, Paul Koch, Yin Lou, Urszula Chajewska, Rich Caruana. Axiomatic Interpretability for Multiclass Additive Models. *In KDD*. 2019. [152]
- Yujia Zhang, Kuangyan Song, Yiming Sun, **Sarah Tan**, Madeleine Udell. "Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations. *In ICML Workshop on AI for Social Good*. 2019. [153]
- **Sarah Tan**, Susanna Makela, Daliah Heller, Kevin Konty, Sharon Balter, Tian Zheng, James H. Stark. A Bayesian Evidence Synthesis Approach to Estimate Disease Prevalence in Hard-To-Reach Populations: Hepatitis C in New York City. *Epidemics 23*. 2018. [136]
- **Sarah Tan**. Interpretable Approaches to Detect Bias in Black-Box Models. *AAAI/ACM AIES Doctoral Consortium*. 2018. [131]
- Skyler Seto, **Sarah Tan**, Giles Hooker, Martin T. Wells. A Double Parametric Bootstrap Test for Topic Models. *In NIPS Symposium on Interpretable Machine Learning*. 2017. [121]
- **Sarah Tan**, Giles Hooker, Martin T. Wells. Probabilistic Matching: Incorporating Uncertainty to Correct for Selection Bias. *In NIPS Workshop on Inference and Learning of Hypothetical and Counterfactual Interventions in Complex Systems*. 2016. [135]

- **Sarah Tan**, David I. Miller, James Savage. Proximity Score Matching: Using the Random Forest Proximity Matrix for Matching in Causal Inference. *In NIPS Healthcare Workshop*. 2015. **[1 of 3 Student Paper Awards from American Statistical Association's SSPA section]** [137]
- Ion B. Vasi, Edward T. Walker, John S. Johnson, **Sarah Tan**. "No Fracking Way!" Documentary Film, Discursive Opportunity, and Local Opposition against Hydraulic Fracturing in the United States, 2010 to 2013. *American Sociological Review 80 (5)*. 2015. **[2 Best Paper Awards from American Sociological Association's CITAMS and CBSM sections]** [141]

BIBLIOGRAPHY

[1] Julius Adebayo and Lalana Kagal. Iterative orthogonal feature projection for diagnosing bias in black-box models. In *FAT/ML Workshop*, 2016.

[2] Philip Adler, Casey Falk, Sorelle A. Friedler, Gabriel Rybeck, Carlos Eduardo Scheidegger, Brandon Smith, and Suresh Venkatasubramanian. Auditing black-box models for indirect influence. In *ICDM*, 2016.

[3] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudik, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *ICML*, 2018.

[4] David Alvarez-Melis and Tommi S Jaakkola. Towards robust interpretability with self-explaining neural networks. In *NeurIPS*, 2018.

[5] Marco Ancona, Enea Ceolini, Cengiz Oztireli, and Markus Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *ICLR*, 2018.

[6] Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. Learning certifiably optimal rule lists. In *KDD*, 2017.

[7] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. How we analyzed the COMPAS recidivism algorithm. https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm, 2016. Accessed May 26, 2017.

[8] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine Bias: There's software used across the country to predict future criminals. And it's biased against blacks. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing, 2016. Accessed May 26, 2017.

[9] Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *The Annals of Statistics*, 47(2), 2019.

[10] Martin Atzmueller and Florian Lemmerich. VIKAMINE - Open-Source Subgroup Discovery, Pattern Mining, and Analytics. In *ECML PKDD*, 2012.

[11] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *NIPS*, 2014.

[12] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Muller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS ONE*, 10(7), 2015.

[13] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Muller. How to explain individual classification decisions. *Journal of Machine Learning Research*, 10, 2010.

[14] Mousumi Banerjee, Ying Ding, and Anne-Michelle Noone. Identifying representative trees from ensembles. *Statistics in Medicine*, 31(15), 2012.

[15] Osbert Bastani, Carolyn Kim, and Hamsa Bastani. Interpreting blackbox models via model extraction. In *FAT/ML Workshop*, 2017.

[16] Richard A. Berk. *Criminal Justice Forecasts of Risk*. Springer, 2012.

[17] Jacob Bien and Robert Tibshirani. Prototype selection for interpretable classification. *The Annals of Applied Statistics*, 5(4), 2011.

[18] Thomas Blomberg, William Bales, Karen Mann, Ryan Meldrum, and Joe Nedelec. Validation of the COMPAS risk assessment classification instrument. Technical report, Florida State University, 2010.

[19] Leo Breiman. Random forests. *Machine Learning*, 45(1), 2001.

[20] Leo Breiman and Adele Cutler. Random forests manual. `https://www.stat.berkeley.edu/~breiman/RandomForests`, 2002. Accessed July 6, 2019. Year 2002 based on copyright year indicated in the authors' Fortran code.

[21] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees (CART)*. Wadsworth International Group, 1984.

[22] Cristian Bucilua, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, 2006.

[23] Peter Buhlmann and Bin Yu. Boosting with the L2 loss: Regression and Classification. *Journal of the American Statistical Association*, 98(462), 2003.

[24] 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 *KDD*, 2015.

[25] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *ICML*, 2006.

[26] Jianbo Chen, Le Song, Martin J. Wainwright, and Michael I. Jordan. Learning to explain: An information-theoretic perspective on model interpretation. In *ICML*, 2018.

[27] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2), 2017.

[28] Alexandra Chouldechova and Max G'Sell. Fairer and more accurate, but for whom? In *FAT/ML Workshop*, 2017.

[29] US Sentencing Commission. Measuring Recidivism: the Criminal History Computation of the Federal Sentencing Guidelines. https://www.ussc.gov/sites/default/files/pdf/research-and-publications/research-publications/2004/200405_Recidivism_Criminal_History.pdf, 2004. Accessed July 21, 2019.

[30] Gregory F. Cooper, Constantin F. Aliferis, Richard Ambrosino, John M. Aronis, Bruce G. Buchanan, Rich Caruana, Michael J. Fine, Clark Glymour, Geoffrey J. Gordon, Barbara H. Hanusa, Janine E. Janosky, Christopher Meek, Tom M. Mitchell, Thomas S. Richardson, and Peter Spirtes. An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine*, 9, 1997.

[31] Sam Corbett-Davies, Emma Pierson, Avi Feller, Sharad Goel, and Aziz Huq. Algorithmic decision making and the cost of fairness. In *KDD*, 2017.

[32] Mark W. Craven and Jude W. Shavlik. Extracting tree-structured representations of trained networks. In *NIPS*, 1995.

[33] Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via

quantitative input influence: Theory and experiments with learning systems. In *IEEE Symposium on Security and Privacy*, 2016.

[34] Morris H. DeGroot and Stephen E. Fienberg. The comparison and evaluation of forecasters. *Journal of the Royal Statistical Society: Series D*, 32(1/2), 1983.

[35] Konstantinos G. Derpanis. Mean shift clustering. http://www.cse.yorku.ca/~kosta/CompVis_Notes/mean_shift.pdf, 2005. Accessed July 21, 2019.

[36] William Dieterich, Christina Mendoza, and Tim Brennan. COMPAS risk scales: Demonstrating accuracy equity and predictive parity. Technical report, Northpointe Inc., 2016.

[37] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Springer, 2018.

[38] Elizabeth Drake. Predicting Criminal Recidivism: A Systematic Review of Offender Risk Assessments in Washington State. Technical report, Washington State Institute for Public Policy, 2014.

[39] Julia Dressel and Hany Farid. The accuracy, fairness, and limits of predicting recidivism. *Science Advances*, 4(1), 2018. Data at www.cs.dartmouth.edu/farid/downloads/publications/scienceadvances17.

[40] Federal Reserve Governors. Report to the Congress on Credit Scoring and Its Effects on the Availability and Affordability of Credit. https://www.federalreserve.gov/boarddocs/rptcongress/creditscore/creditscore.pdf, 2007. Accessed July 21, 2019.

[41] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *KDD*, 2015.

[42] Anthony W. Flores, Kristin Bechtel, and Christopher Lowenkamp. False Positives, False Negatives, and False Analyses: A Rejoinder to "Machine Bias". *Federal Probation Journal*, 80, 2016.

[43] Alex A Freitas. Comprehensible classification models: a position paper. *ACM SIGKDD Explorations*, 15(1), 2014.

[44] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *ICML*, 1996.

[45] Jerome H. Friedman. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, 29(5), 2001.

[46] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5), 2001.

[47] Jerome H. Friedman and Bogdan E. Popescu. Predictive learning via rule ensembles. *The Annals of Applied Statistics*, 2(3), 2008.

[48] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

[49] LiMin Fu. Rule generation from neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(8), 1994.

[50] Salvador Garcia, Joaquín Derrac, José Ramón Cano, and Francisco Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 34(3), 2011.

[51] Paul Gendreau, Tracy Freeze, and Claire Goggin. A meta-analysis of the predictors of adult offender recidivism: What works! *Criminology*, 34(4), 1996.

[52] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1), 2006.

[53] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[54] Sharad Goel, Justin M. Rao, and Ravi Shroff. Precinct or Prejudice? Understanding Racial Disparities in New York City's Stop-and-Frisk Policy. *The Annals of Applied Statistics*, 10(1), 2016.

[55] Ryan Gomes and Andreas Krause. Budgeted nonparametric learning from data streams. In *ICML*, 2010.

[56] Bryce Goodman and Seth Flaxman. European Union regulations on algorithmic decision-making and a "right to explanation". *AI Magazine*, 38(3), 2017.

[57] Nina Grgic-Hlaca, Elissa M. Redmiles, Krishna P. Gummadi, and Adrian Weller. Human perceptions of fairness in algorithmic decision making: A case study of criminal risk prediction. In *WWW*, 2018.

[58] William M. Grove, David H. Zald, Boyd S. Lebow, and Beth E. Snitzand Chad Nelson. Clinical versus mechanical prediction: A meta-analysis. *Psychological Assessment*, 12(1), 2000.

[59] 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), 2016.

[60] Satoshi Hara and Kohei Hayashi. Making tree ensembles interpretable: A bayesian model selection approach. In *AISTATS*, 2018.

[61] Peter Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3), 1968.

[62] Trevor Hastie and Rob Tibshirani. *Generalized Additive Models*. Chapman and Hall/CRC, 1990.

[63] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[64] Andreas Henelius, Kai Puolamki, Henrik Bostrm, Lars Asker, and Panagiotis Papapetrou. A peek into the black box: exploring classifiers by randomization. *Data Mining and Knowledge Discovery*, 28(5-6), 2014.

[65] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2014.

[66] Giles Hooker. Discovering additive structure in black box functions. In *KDD*, 2004.

[67] Giles Hooker. Generalized functional anova diagnostics for high-

dimensional functions of dependent variables. *Journal of Computational and Graphical Statistics*, 16(3), 2007.

[68] Eric Horvitz and Tim Paek. Complementary computing: policies for transferring callers from dialog systems to human receptionists. *User Modeling and User-Adapted Interaction*, 17(1-2), 2007.

[69] Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. Global explanations of neural networks: Mapping the landscape of predictions. In *AAAI/ACM AIES*, 2019.

[70] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[71] Bertrand Iooss and Paul Lemaitre. A review on global sensitivity analysis methods. In *Uncertainty Management in Simulation-Optimization of Complex Systems*. Springer, 2015.

[72] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on Amazon Mechanical Turk. In *KDD Workshop on Human Computation*, 2010.

[73] Hemant Ishwaran. Variable importance in binary regression trees and forests. *Electronic Journal of Statistics*, 1, 2007.

[74] Ece Kamar, Severin Hacker, and Eric Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *AAMAS*, 2012.

[75] Leonard Kaufman and Peter J Rousseeuw. Clustering by means of medoids. In *Statistical Data Analysis Based on the L1 Norm*. Birkhuser Basel, 1987.

[76] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *ICML*, 2018.

[77] Rajiv Khanna, Been Kim, Joydeep Ghosh, and Oluwasanmi Koyejo. Interpreting black box predictions using fisher kernels. In *AISTATS*, 2019.

[78] Been Kim, Rajiv Khanna, and Oluwasanmi Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *NIPS*, 2016.

[79] Been Kim, Cynthia Rudin, and Julie A Shah. The bayesian case model: A generative approach for case-based reasoning and prototype classification. In *NIPS*, 2014.

[80] Michael P. Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *AAAI/ACM AIES*, 2019.

[81] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[82] Jon Kleinberg, Himabindu Lakkaraju, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. Human decisions and machine predictions. *The Quarterly Journal of Economics*, 133(1), 2017.

[83] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *Innovations in Theoretical Computer Science*, 2017.

[84] Todd Kulesza, Margaret Burnett, Weng-Keen Wong, and Simone Stumpf. Principles of explanatory debugging to personalize interactive machine learning. In *IUI*, 2015.

[85] Himabindu Lakkaraju, Stephen H Bach, and Jure Leskovec. Interpretable decision sets: A joint framework for description and prediction. In *KDD*, 2016.

[86] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. In *AAAI*, 2017.

[87] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. Faithful and customizable explanations of black box models. In *AAAI/ACM AIES*, 2019.

[88] Himabindu Lakkaraju, Jon Kleinberg, Jure Leskovec, Jens Ludwig, and Sendhil Mullainathan. The selective labels problem: Evaluating algorithmic predictions in the presence of unobservables. In *KDD*, 2017.

[89] Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3), 2015.

[90] Andy Liaw and Matthew Wiener. Classification and regression by randomForest. *R News*, 2(3), 2002.

[91] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *ACL*, 2011.

[92] Yi Lin and Yongho Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101(474), 2006.

[93] Manuel Lingo and Gerhard Winkler. Discriminatory power-an obsolete validation criterion? *Journal of Risk Model Validation*, 2(1), 2008.

[94] Zachary C. Lipton. The Mythos of Model Interpretability. In *ICML Workshop on Human Interpretability of Machine Learning*, 2016.

[95] Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *KDD*, 2012.

[96] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *KDD*, 2013.

[97] Gilles Louppe. Understanding random forests: From theory to practice. *PhD dissertation, Universite de Liege*, 2014.

[98] Francisco Louzada, Anderson Ara, and Guilherme B. Fernandes. Classification methods applied to credit scoring: Systematic review and overall comparison. *Surveys in Operations Research and Management Science*, 21(2), 2016.

[99] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NIPS*, 2017.

[100] Scott M. Lundberg, Bala Nair, Monica S Vavilala, Mayumi Horibe, Michael J. Eisses, Trevor Adams, David E. Liston, Daniel King-Wai Low, Shu-Fang Newman, Jerry Kim, et al. Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10), 2018.

[101] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov), 2008.

[102] Lucas Mentch and Giles Hooker. Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *The Journal of Machine Learning Research*, 17(26), 2016.

[103] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization: Identifying representative elements in massive data. In *NIPS*, 2013.

[104] Gregoire Montavon, Wojciech Samek, and Klaus-Robert Muller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing*, 73, 2018.

[105] Menaka Narayanan, Emily Chen, Jeffrey He, Been Kim, Sam Gershman, and Finale Doshi-Velez. How do Humans Understand Explanations from Machine Learning Systems? An Evaluation of the Human-Interpretability of Explanation. *arXiv preprint arXiv:1802.00682*, 2018.

[106] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming*, 14(1), 1978.

[107] Alexandru Niculescu-Mizil and Rich Caruana. Predicting good probabilities with supervised learning. In *ICML*, 2005.

[108] Besmira Nushi, Ece Kamar, and Eric Horvitz. Towards Accountable AI: Hybrid human-machine analyses for characterizing system failure. In *HCOMP*, 2018.

[109] Christos H. Papadimitriou. Worst-case and probabilistic analysis of a geometric location problem. *SIAM Journal on Computing*, 10(3), 1981.

[110] P. Jonathon Phillips, Amy N. Yates, Ying Hu, Carina A. Hahn, Eilidh Noyes, Kelsey Jackson, Jacqueline G. Cavazos, Geraldine Jeckeln, Rajeev Ranjan, Swami Sankaranarayanan, Jun-Cheng Chen, Carlos D. Castillo, Rama Chellappa, David White, and Alice J. O'Toole. Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms. *PNAS*, 115(24), 2018.

[111] Forough Poursabzi-Sangdeh, Daniel G. Goldstein, Jake M. Hofman, Jennifer Wortman Vaughan, and Hanna Wallach. Manipulating and measuring model interpretability. *arXiv preprint arXiv:1802.07810*, 2018.

[112] Carey E. Priebe, David J. Marchette, Jason G. DeVinney, and Diego A. Socolinsky. Classification using class cover catch digraphs. *Journal of classification*, 20(1), 2003.

[113] Ramya Ramakrishnan, Ece Kamar, Debadeepta Dey, Julie Shah, and Eric Horvitz. Discovering blind spots in reinforcement learning. In *AAMAS*, 2018.

[114] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why Should I Trust You?": Explaining the predictions of any classifier. In *KDD*, 2016.

[115] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, 2018.

[116] Michael M. Richter and Agnar Aamodt. Case-based reasoning foundations. *The Knowledge Engineering Review*, 20(3), 2005.

[117] Richard D. Riley, Joie Ensor, Kym I.E. Snell, Thomas P.A. Debray, Doug G. Altman, Karel G.M. Moons, and Gary S. Collins. External validation of clinical prediction models using big datasets from e-health records or ipd meta-analysis: opportunities and challenges. *The BMJ*, 353, 2016.

[118] Ivan Sanchez, Tim Rocktaschel, Sebastian Riedel, and Sameer Singh. Towards extracting faithful and descriptive representations of latent variable models. *AAAI Spring Syposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches*, 2015.

[119] Vitaly Schetinin, Jonathan E. Fieldsend, Derek Partridge, Timothy J. Coats, Wojtek J. Krzanowski, Richard M. Everson, Trevor C. Bailey, and Adolfo Hernandez. Confident interpretation of bayesian decision tree ensembles for clinical applications. *IEEE Transactions on Information Technology in Biomedicine*, 11(3), 2007.

[120] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[121] Skyler Seto, Sarah Tan, Giles Hooker, and Martin T Wells. A double parametric bootstrap test for topic models. *arXiv preprint arXiv:1711.07104*, 2017.

[122] Joseph Sexton and Petter Laake. Standard errors for bagged and random forest estimators. *Computational Statistics and Data Analysis*, 53(3), 2009.

[123] Anoop D. Shah, Jonathan W. Bartlett, James Carpenter, Owen Nicholas, and Harry Hemingway. Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study. *American Journal of Epidemiology*, 179(6), 2014.

[124] Tao Shi and Steve Horvath. Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics*, 15(1), 2006.

[125] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017.

[126] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR Workshop*, 2014.

[127] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[128] Ilya M. Sobol. Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Mathematics and Computers in Simulation*, 55(1-3), 2001.

[129] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 2014.

[130] Daniel J. Stekhoven. missForest: Nonparametric missing value imputation using random forest. *Astrophysics Source Code Library*, 2015.

[131] Sarah Tan. Interpretable approaches to detect bias in black-box models. In *AAAI/ACM AIES Doctoral Consortium*, 2018.

[132] Sarah Tan, Julius Adebayo, Kori Inkpen, and Ece Kamar. Investigating Human + Machine Complementarity: A Case Study on Recidivism. *arXiv preprint arXiv:1808.09123*, 2018.

[133] Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo.

Learning and evaluating global additive explanations of black-box models. *arXiv preprint arXiv:1801.08640*, 2018.

[134] Sarah Tan, Rich Caruana, Giles Hooker, and Yin Lou. Distill-and-compare: Auditing black-box models using transparent model distillation. In *AAAI/ACM AIES*, 2018.

[135] Sarah Tan, Giles Hooker, and Martin T. Wells. Probabilistic Matching: Incorporating Uncertainty to Improve Propensity Score Matching. In *NIPS Causal Inference Workshop*, 2016.

[136] Sarah Tan, Susanna Makela, Daliah Heller, Kevin Konty, Sharon Balter, Tian Zheng, and James H Stark. A Bayesian evidence synthesis approach to estimate disease prevalence in hard-to-reach populations: hepatitis C in New York City. *Epidemics*, 23, 2018.

[137] Sarah Tan, David I. Miller, and Jim Savage. Proximity Score Matching: Using the Random Forest Proximity Matrix for Matching in Causal Inference. In *NIPS Healthcare Workshop*, 2015.

[138] Sarah Tan, Matvey Soloviev, Giles Hooker, and Martin T. Wells. Tree space prototypes: Another look at making tree ensembles interpretable. *arXiv preprint arXiv:1611.07115*, 2016.

[139] Florian Tramer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, Jean-Pierre Hubaux, Mathias Humbert, Ari Juels, and Huang Lin. Fairtest: Discovering unwarranted associations in data-driven applications. In *IEEE European Symposium on Security and Privacy*, 2017.

[140] Michael Tsang, Dehua Cheng, and Yan Liu. Detecting statistical interactions from neural network weights. In *ICLR*, 2018.

[141] Ion Bogdan Vasi, Edward T. Walker, John S. Johnson, and Sarah Tan. No Fracking Way! Documentary Film, Discursive Opportunity, and Local Opposition against Hydraulic Fracturing in the United States, 2010 to 2013. *American Sociological Review*, 80(5), 2015.

[142] Sandra Wachter, Brent Mittelstadt, and Chris Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harvard Journal of Law & Technology*, 31(2), 2018.

[143] Stefan Wager and Susan Athey. Estimation and inference of heteroge-

neous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523), 2017.

[144] Dayong Wang, Aditya Khosla, Rishab Gargeya, Humayun Irshad, and Andrew H. Beck. Deep learning for identifying metastatic breast cancer. *arXiv preprint arXiv:1606.05718*, 2016.

[145] Hao Wang, Berk Ustun, and Flavio P. Calmon. On the direction of discrimination: An information-theoretic analysis of disparate impact in machine learning. In *International Symposium on Information Theory*, 2018.

[146] D.L. Wilson. Asymptotic properties of nearest neighbor rules using edited data sets. *Transactions on Systems, Man and Cybernetics*, 2(3), 1972.

[147] Simon N. Wood. *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC, 2006.

[148] Simon N. Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B*, 73(1), 2011.

[149] Caiming Xiong, David Johnson, Ran Xu, and Jason J. Corso. Random forests for metric learning with implicit pairwise position dependence. In *KDD*, 2012.

[150] Kun-Hsing Yu, Andrew L. Beam, and Isaac S. Kohane. Artificial intelligence in healthcare. *Nature Biomedical Engineering*, 2(10), 2018.

[151] Jiaming Zeng, Berk Ustun, and Cynthia Rudin. Interpretable classification models for recidivism prediction. *Journal of the Royal Statistical Society: Series A*, 180(3), 2016.

[152] Xuezhou Zhang, Sarah Tan, Paul Koch, Yin Lou, Urszula Chajewska, and Rich Caruana. Axiomatic interpretability for multiclass additive models. In *KDD*, 2019.

[153] Yujia Zhang, Kuangyan Song, Yiming Sun, Sarah Tan, and Madeleine Udell. "Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations. *arXiv preprint arXiv:1904.12991*, 2019.

[154] Zhe Zhang and Daniel B. Neill. Identifying significant predictive bias in classifiers. In *FAT/ML Workshop*, 2017.

[155] Peng Zhao, Xiaogang Su, Tingting Ge, and Juanjuan Fan. Propensity score and proximity matching using random forest. *Contemporary Clinical Trials*, 47, 2016.

[156] Qi-Feng Zhou, Hao Zhou, Yong-Peng Ning, Fan Yang, and Tao Li. Two approaches for novelty detection using random forest. *Expert Systems with Applications*, 42(10), 2015.

[157] Yichen Zhou, Zhengze Zhou, and Giles Hooker. Approximation trees: Statistical stability in model distillation. *arXiv preprint arXiv:1808.07573*, 2018.

[158] Zhengze Zhou and Giles Hooker. Unbiased measurement of feature importance in tree-based methods. *arXiv preprint arXiv:1903.05179*, 2019.