FinRegLab

Machine Learning Explainability & Fairness: Insights from Consumer Lending

Empirical White Paper

FinRegLab in collaboration with Laura Blattner and Jann Spiess

UPDATED
JULY 2023

About FinRegLab

FinRegLab is a nonprofit, nonpartisan innovation center that tests new technologies and data to inform public policy and drive the financial sector toward a responsible and inclusive financial marketplace. With our research insights, we facilitate discourse across the financial ecosystem to inform public policy and market practices.

This white paper is part of a broader research project on the explainability and fairness of machine learning in credit underwriting. The empirical research described herein was conducted in collaboration with Professors Laura Blattner and Jann Spiess at the Stanford Graduate School of Business. Other reports in this series, including an overview of our policy and empirical findings, are available at https://finreglab.org/ai-machine-learning/#aipublications.

Principal Investigators and Other Contributors:

Laura Blattner, co-principal investigator, is a former Assistant Professor of Finance at the Stanford Graduate School of Business. Laura earned her Ph.D. at Harvard University. She also holds a B.A. in Philosophy, Politics, and Economics and an M.Phil. in Economics from the University of Oxford. At Stanford, Laura taught an MBA class on Financial Technology (FinTech) and researched the governance and regulation of algorithmic credit underwriting. She is currently head of climate risk at Watershed.

P-R Stark, co-principal investigator, served as FinRegLab's first Director of Machine Learning Research. In that role, she led the organization's efforts to develop and execute policy-relevant research on the use of AI in financial services. She is an experienced advisor to financial institutions, helping firms respond to regulatory inquiries and manage adoption of new technologies, among other things. P-R holds an A.B. in Classics from Princeton University, an M.A. (Oxon) in Philosophy, Politics, and Economics from the University of Oxford, and a J.D. from Harvard University.

Jann Spiess, co-investigator, is an Assistant Professor of Operations, Information & Technology at Stanford University's Graduate School of Business. He is an econometrician working on machine learning and causal inference. Jann is particularly interested in developing methods for transparent, robust, and replicable inferences from complex data.

Contributing Authors
Sarah Davies
Duncan McElfresh
Sormeh Yazdi
Zishun Zhao

Data Science Team
Mario Curiki
Georgy Kalashnov
Mashrur Khan
Fiona Sequeira
Sormeh Yazdi

Support for this publication and other aspects of FinRegLab's Machine Learning in Credit Underwriting project was provided by the Mastercard Center for Inclusive Growth, JPMorgan Chase, and Flourish Ventures. Detailed information on our funders and additional acknowledgments can be found on the inside back cover.

Contents

1 Introduction **5**

2 Background **12**
2.1 Explainability and Machine Learning in Consumer Lending . . . 12
2.2 Related Literature . . . 13

3 Research Design & Methodology **15**
3.1 Research Questions . . . 15
3.2 Evaluation Framework . . . 15
3.2.1 Fidelity . . . 15
3.2.2 Consistency . . . 15
3.2.3 Usability . . . 16
3.3 Data . . . 17
3.3.1 Data Description . . . 17
3.3.2 Protected Class Data . . . 18
3.4 Model Development . . . 18
3.4.1 Baseline Models . . . 19
3.4.2 Company Models . . . 20
3.4.3 Model Performance . . . 21
3.5 Evaluation Participants . . . 22
3.6 Open-Source Tools . . . 24

4 Results: Adverse Action Notices **26**
4.1 Background . . . 26
4.1.1 Legal and Regulatory Overview . . . 26
4.1.2 Operational Considerations . . . 27
4.2 Empirical Evaluation: Summary . . . 28
4.3 Participation Details and Diagnostic Tools . . . 30
4.3.1 Description of Tasks . . . 30
4.3.2 Participation Details . . . 31
4.3.3 Description of Tools . . . 31
4.4 Evaluation: Fidelity . . . 33
4.4.1 Evaluation Description . . . 33
4.4.2 Fidelity Results . . . 35
4.5 Evaluation: Consistency . . . 39
4.5.1 Evaluation Description . . . 40

4.5.2 Consistency Results . . . 42
4.6 Statistical Association Analysis . . . 43
4.7 Evaluation: Usability . . . 47
4.7.1 Evaluation Description . . . 47
4.7.2 Usability Results . . . 48

5 Results: Fair Lending and Disparate Impact 52
5.1 Background . . . 52
5.1.1 Legal and Regulatory Overview . . . 52
5.1.2 Operational Considerations . . . 53
5.2 Empirical Evaluation: Summary . . . 54
5.3 Fairness and Disparate Impact Properties . . . 57
5.3.1 Description of Fairness Metrics . . . 58
5.3.2 Results: Fairness Properties . . . 60
5.4 Participation Details and Diagnostic Tools . . . 67
5.4.1 Description of Tasks . . . 68
5.4.2 Participation Details . . . 68
5.4.3 Description of Tools . . . 69
5.5 Evaluation: Fidelity . . . 72
5.5.1 Evaluation Description . . . 72
5.5.2 Fidelity Results . . . 74
5.6 Evaluation: Consistency . . . 79
5.6.1 Evaluation Description . . . 80
5.6.2 Consistency: Results . . . 82
5.7 Evaluation: Usability . . . 83
5.7.1 Evaluation Description . . . 86
5.7.2 Usability: Results . . . 87

6 Results: Model Risk Management 102
6.1 Background . . . 102
6.1.1 Legal and Regulatory Overview . . . 102
6.1.2 Operational Considerations . . . 103
6.2 Empirical Evaluation: Summary . . . 103
6.3 Participation Details and Diagnostic Tools . . . 104
6.3.1 Description of Tasks . . . 104
6.3.2 Participation Details . . . 104
6.3.3 Description of Tools . . . 104
6.4 Evaluation: Fidelity . . . 107
6.4.1 Evaluation Description . . . 107
6.4.2 Fidelity Results . . . 108
6.5 Evaluation: Consistency . . . 113
6.5.1 Evaluation Description . . . 113

6.5.2 Consistency: Results 114
6.6 Evaluation: Usability 118
6.6.1 Evaluation Description 118
6.6.2 Usability: Results 120

7 Conclusion **127**

8 Literature **129**

A Common Terms and Acronyms **132**

B Additional Results **136**

C Perturbation **146**
C.1 Partial Perturbation 146
C.2 Full Perturbation 147

D Inferring Minority Status **148**

E Model Risk Management - Fidelity: Nearest Neighbor Test **149**

1 INTRODUCTION

Lending decisions informed by credit underwriting models affect the lives of hundreds of millions of Americans. Advanced prediction technology based on machine learning techniques has the potential to increase credit access by more accurately identifying applicants who are likely to repay loans and to reduce the number of people given loans that they are unlikely to be able to afford. However, the black-box nature of machine learning models has focused attention on model transparency as a critical threshold question – both for lenders considering whether and how to use these models and for regulators working to adapt expectations and oversight processes to promote fair, responsible, and inclusive use. The need to describe the behavior of black-box machine-learning models has spurred the development of a range of model diagnostic tools and techniques. These model diagnostic tools can help lenders answer questions such as: what is driving the model's prediction/behavior for *a specific loan applicant*; what is driving the model's predictions for a *group of applicants*, *e.g.*, applicants of a certain race; and what is driving the *overall behavior of the model*.

Approach To evaluate whether model diagnostic tools can successfully answer questions about model behavior, it is necessary to first understand the purpose these questions serve. In the context of consumer lending, model transparency serves to further widely shared goals regarding anti-discrimination, consumer empowerment, and responsible risk-taking. For lenders, model transparency is a key instrument to help them evaluate whether a model can be responsibly used in an intended application; to enable the day-in, day-out work of managing relevant prudential and consumer protection risks; and to document efforts to comply with law and regulation. For consumers, model transparency helps ensure that they receive basic information about how certain kinds of adverse credit decisions are made and enable recourse where appropriate. For regulators and policymakers, model transparency is an instrument to enable oversight and detect shortcomings in compliance with laws and regulations. In this context, we believe that evaluating model transparency as a means to an end is a productive structure for analysis, particularly given that there is not currently a universal standard – or even definition – of what makes a predictive model "transparent."

Our evaluation of model diagnostic tools is grounded in this perspective of model transparency as an instrument to further certain goals of consumer protection and prudential risk management. In this report, we focus on three areas of regulation – consumer protection regulations regarding adverse action notices and disparate impact, and prudential regulations regarding model risk management. In turn, these areas of regulation present lenders with specific model transparency challenges because they have to: (1) provide loan applicants who are denied credit or charged higher prices with the principal reasons for those decisions; (2) investigate whether their underwriting models have disproportionately adverse effects on the basis of race or other protected characteristics, and if so to search for alternative models; and (3) investigate how changing economic conditions affect model behavior and performance.

Research Questions We address the following two research questions. First, we ask whether and how model transparency is important to achieving the goals expressed in certain regulations applicable to consumer lending. Second, in those instances where it is, we ask to what degree available feature-based model diagnostic tools further those goals – in particular, how well do these tools help users of machine learning models address specific transparency challenges, develop and manage machine learning underwriting models in compliance with applicable requirements, and facilitate necessary oversight by lenders and their regulators. Specifically, we evaluate the tools for identifying

drivers of key model behaviors based on three dimensions: (1) fidelity, that is, the ability to reliably identify features that relate to particular model predictions or decisions; (2) consistency, that is, the degree to which tools identify the same drivers; and (3) usability, that is, the ability to identify information that enables users to perform certain tasks (such as helping consumers improve their likelihood of approval or lenders to comply with particular regulatory requirements).

Research Team To answer these research questions, FinRegLab partnered with Stanford Business School professors Laura Blattner and Jann Spiess to conduct an empirical analysis of the ability of model diagnostic tools to help lenders understand and manage machine learning credit underwriting models. FinRegLab is a non-profit research organization that was founded in 2018 based on the premise that independent, rigorous research is a primary ingredient in helping develop market norms and policy solutions that enable responsible innovation in financial services. Professors Blattner and Spiess, experts on credit underwriting and machine learning, worked with FinRegLab to design the empirical methodology used in this research and executed this research with assistance from a data science team at Stanford University and FinRegLab. This evaluation is also informed by two consultative processes: (1) wide-ranging input from an advisory body composed of subject matter experts from bank and non-bank lenders, civil society organizations, technology companies, and academic institutions and (2) interviews with lenders, advocates, and other stakeholders to assess market practices with respect to the use of machine learning underwriting models. Further, FinRegLab previously explored many of the motivations and concerns about the use of machine learning underwriting models in its Market Context and Data Science Report, which provides extensive information about many of the topics and themes presented herein.[1]

Research Participants Our analysis evaluates model diagnostic tools offered by seven technology companies – Arthur, H2O.ai, Fiddler, RelationalAI, SolasAI, Stratyfy, and Zest AI – that offer various AI and machine learning services. Working with tools being offered in the market leverages these companies' expertise in generating information about model behavior and their decisions about how best to serve their clients. Accordingly, the tools evaluated in this study reflect each company's individual judgments about a range of important technological, computational, strategic, and regulatory questions rather than those of the research team.

Our analysis also includes a set of open-source model diagnostic tools implemented by the research team. Our intent is not to identify the "best" or "worst" products or companies. Instead, we aim to (1) conduct an analysis that helps all stakeholders understand better the capabilities, limitations, and performance of a variety of model diagnostic tools in the context of existing consumer credit regulations and broader policy considerations and (2) offer a framework that lenders, regulators, and researchers can use as a starting point for evaluating the quality and usefulness of information about the behavior of machine learning underwriting models.

Underwriting Models To conduct our analysis, each company completed a set of model diagnostic tasks on a set of identical credit underwriting models. The types of models used as a baseline include a logistic regression model, a simple neural network trained on a small subset of available credit report features, an XGBoost model, and a (complex) neural network model trained on the full set of hundreds of credit report features (all four models are collectively the "Baseline Models"). Four of the participating companies opted to provide models that they trained using the same data as the Baseline Models. These models expand the number of model types included in this evaluation:

[1] FinRegLab (2021).

a random forest classifier, an ensemble of generalized linear models, an ensemble of gradient boosted machines, a monotonicity-constrained XGBoost, an unconstrained XGBoost, and an unspecified proprietary model (collectively, the "Company Models").

Across the Baseline and Company Models, our intent was to consider types of machine learning models that broadly correspond to those in use by lenders offering various consumer lending products and to establish a spectrum of simple and complex machine learning models against which we can understand the capabilities and limitations of the model diagnostic tools under evaluation. However, we note that our use of the terms "simple" and "complex" in this context is relative and captures both aspects of a model's architecture and the number of features it uses. Our usage of the terms "simple" and "complex" does not conform to any external standard or definition. Given this usage of "simple" and "complex," a lender who is using what they consider to be a relatively simple model may fall somewhere in the middle of the range defined by the set of models in this research. Finally, we note that the term "complex" as used in this evaluation is *not* used to describe all machine learning models or compare machine learning models to models developed with more conventional means.

The Baseline and Company Models were developed using a panel of actual consumer credit data provided by a leading credit bureau. The data is a representative sample of consumers from across the United States, covering the period between 2009 and 2017. Our data provider required the masking of certain feature descriptions, which meant that the research companies did not have access to those feature labels in the original data set although they had some basic descriptive information about the nature of the variables. This limited their ability to apply domain knowledge through performing qualitative feature reviews or creating features manually in the course of developing their models.

Key Results Our research focuses on the capabilities, limitations, and performance of model diagnostic tools in the context of (a) consumer disclosure regulations that require adverse action notices, (b) fair lending requirements regarding disparate impact, and (c) prudential model risk management governance that requires that firms account for the robustness of model performance in changing conditions, among other things. Below, we present the following main findings:

1. **Our research suggests that there are diagnostic tools which can help lenders address transparency challenges associated with machine learning underwriting models**. Across all three regulatory areas, there were tools that were able to identify relevant information about the model's behavior. However, there was no single tool or approach that performed best across all three types of regulatory requirements or across all tasks that we evaluated. This points to the importance of a range of decisions that users of these tools – whether they be lenders, technology vendors, or others – make when designing, implementing, and monitoring the effectiveness of model diagnostic tools. These decisions are, in turn, critical to the responsible use of this kind of diagnostic tool as well as of the underlying machine learning underwriting models.

   In particular, we find the following results:

   (a) **Among the model diagnostic tools we evaluated, some tools can reliably identify features in the model that are related to adverse credit decisions for individual loan applicants.** These tools are able to identify features of rejected applicants such that other applicants who have similar credit characteristics are also likely to be rejected. These tools are also able to identify features that, when changed in a favorable direction, reduce predicted default probabilities by more than randomly chosen or even closely correlated

features. However, there were also tools that did not perform better than correlated or even randomly chosen features. Although our study could not assess how well the tools map the identified features to the types of more holistic "reason codes" given to applicants on the required disclosures, the identification of model-level drivers of adverse credit decisions is the critical first input to the process of producing those disclosures, making our findings relevant to ongoing debates about whether and in what circumstances lenders can produce compliant adverse action notices for credit decisions informed by machine learning underwriting models.

(b) **Among the model diagnostic tools we evaluated, some tools can identify features that describe a significant part of the disparities produced by underwriting models.** These tools are able to reliably identify features that are related to the disparities in the model such that equalizing the distribution of these features across groups sizably reduces disparities on the basis of protected characteristics. These tools are also able to identify features that, when changed in a favorable direction, reduce predicted disparities by more than randomly chosen or even closely correlated features. However, there were also tools that did not perform better than correlated or even randomly chosen features.

(c) **Among the model diagnostic tools we evaluated, the ability to produce less discriminatory model alternatives – a critical component of fair lending capabilities – is not solely a function of their ability to address model transparency challenges associated with machine learning underwriting models.** In the context of disparate impact requirements, lenders have historically relied on strategies that involve dropping or transforming individual features that have been identified as making the largest contributions to disparities based on race, ethnicity, or other protected characteristics. While we did not test the full spectrum of potential mitigation approaches, implementation of specific company recommendations to drop a few individual features identified as most important in creating disparities does not significantly improve fairness and indeed imposes a cost in the form of significantly reduced predictive performance. In contrast, methods that rely on more automated approaches – such as joint optimization and adversarial debiasing – to identify less discriminatory alternative models were able to produce a menu of model specifications that efficiently trade off fairness and predictive performance. These more automated approaches likely perform better in this regulatory task because they assess a broader range of features and incorporate fairness considerations into the model's development from the start.

(d) **Among the model diagnostic tools we evaluated, some tools can identify features that describe a significant part of the overall model behavior as well as the changes in model behavior in a deployment dataset drawn from a different time period.** These tools are also able to identify features that, when changed in a favorable direction, reduce predicted default probabilities by more than randomly chosen or even closely correlated features. Moreover, these tools can also account for a significant portion of the change in the distribution of model predictions and model performance in an out-of-time context. However, there were also tools that did not perform better than correlated or even randomly chosen features. Our findings help to illustrate how certain model diagnostic tools can support lenders' capabilities to manage certain kinds of model risks as required for banks.

2. **Fidelity, defined as a tool's ability to provide reliable and accurate information about a model's behavior, is crucial when evaluating the performance of diagnostic tools in various risk areas.** Diagnostic tools with high fidelity tend to perform well for both simple and more complex machine-learning models. In contrast, tools with poor fidelity struggle to accurately describe the behavior of both simple and complex models. However, the

fidelity gap between the best and worst diagnostic tools is more pronounced for complex models, suggesting that the quality of implementation choices for diagnostic tools becomes even more important when lenders employ more complex machine learning underwriting models.

3. **Our study assesses whether various diagnostic tools, when applied to the same underwriting model, can be expected to yield consistent results.** We find that high-fidelity tools demonstrate a high degree of consistency in the information they produce to describe a model's behavior, particularly when factoring in feature correlations. Conversely, low-fidelity tools are more likely to generate inconsistent information about a model's behavior. In the context of machine learning models, the ability to identify and utilize granular predictive relationships becomes increasingly relevant as model complexity rises. This, in turn, affects the information that diagnostic tools provide. Higher granularity can lead to disagreements among diagnostic tools, even within the subset of high-fidelity tools. For instance, a traditional logistic model may attribute past mortgage delinquency as the primary driver of an adverse credit decision, while a machine learning underwriting model must consider past 30-day, 60-day, and 90-day mortgage delinquencies as individual factors influencing such decisions. Encouragingly, the majority of disagreements among high-fidelity tools for more complex models are resolved when accounting for broader feature families and feature correlations. This suggests that much of the disagreement arises from the increased granularity in our initial consistency tests. By aggregating granular inputs related to delinquency into a broader "mortgage delinquency" category, we observe agreement patterns similar to those of simple models. The significance of disagreements stemming from heightened granularity in terms of compliance with specific regulatory requirements is highly context-specific and depends on policy judgments beyond the scope of this paper.

Limitations Our analysis comes with several important limitations. First, our analysis considers machine learning models developed using readily available algorithms and tools. Our models do not represent all nuances of production-grade underwriting models and have not gone through nearly the same level of iterative testing and revision. Accordingly, some of the specific metrics reported herein – including measures of model predictiveness, disparities, and performance/fairness tradeoffs – return different values than what practitioners are accustomed to seeing, which means that some results presented herein may not generalize well. Second, we do not consider the universe of available model diagnostic tools or relevant techniques, although we believe that focusing on the tools employed by the participating companies as well as a small set of open-source tools includes the dominant approaches that are available to lenders today. Third, the masking of feature descriptions – required by our data provider – limited manual feature creation as well as the introduction of domain knowledge into the diagnostic process. For example, the feature masking did not allow the translation of drivers of an adverse credit decision into the more holistic reasons that lenders typically state on adverse action notices. Fourth, our evaluation is performed on real-world data rather than on a highly curated, fully controlled experiment where ground truth is known. As a result, our evaluation of diagnostic tools can only make statements about (a) the relative performance of tools and (b) the performance of tools relative to a benchmark based on either randomly chosen features or closely correlated features in the model. However, we cannot make absolute statements about the performance of the model diagnostic tools we evaluate. For example, while we can discuss the importance and consistency of the identified features relative to other subsets of features, we cannot confirm definitively that the identified features are the most important features in actually causing defaults or disparities. Finally, we do not consider the role of transparency in the model-building process itself but instead focus on *post hoc* feature-based diagnostics on the resulting underwriting

models. Relatedly, we do not evaluate the intrinsic value of simple, easy-to-describe, or inherently interpretable models, but instead focus on implications for *post hoc* analysis in the specific use cases considered in this report.

Broader implications Our analysis has broader implications for the use of governance of machine learning underwriting models in consumer credit.

1. **Our findings underscore the critical importance of carefully selecting the right diagnostic tools and implementing them for use in the context of specific regulatory compliance tasks**. While encouraging, our research to date suggests that there are no universal or "one size fits all" model diagnostic tools that lenders can use to help them explain, understand, and manage all aspects of machine learning underwriting models or all regulatory compliance expectations related thereto. Lenders must make well-considered judgments when they select and implement such tools as well as when they act on the information that these tools produce. The responsible use of model diagnostic tools adds another dimension to the many consequential decisions that lenders must make – and will be responsible for – when designing, implementing, and operating machine learning underwriting models. These choices entail both which tool to use for a specific task – that is, using SHAP, LIME or a permutation importance tool – and how to implement that tool in more granular ways.
2. **Our framework provides a path forward for the rigorous empirical testing and evaluation of model diagnostic tools.** Given rapidly evolving technologies and the likely expansion of the use of machine learning for extending consumer credit, continued testing and evaluation of model diagnostic tools will be important to defining expectations for the responsible use of such tools and monitoring risks related to their use. Notably, even when ground truth is not known, we show that we can design tests that allow us to compare not only the relative performance of diagnostic tools but also the performance of a tool relative to objective benchmarks that select random or closely correlated features. This contribution addresses a gap in current oversight structures, which do not clearly guide lenders, technology vendors, or other stakeholders in making the array of implementation choices that account for important performance variations among the tools in the tasks in our evaluation.

Implications for Model Transparency Model transparency in credit underwriting is a complex issue that requires a multifaceted approach. Our analysis demonstrates the importance of considering context, purpose, and the limitations of different diagnostic tools when aiming to provide actionable insights for both regulators and consumers. The following key points outline the broader implications of our analysis for the debate about the transparency of machine learning models in credit underwriting:

1. **With high confidence:**

    1. **Contextual Interpretation** Interpretation of identified model behavior drivers necessitates a comprehensive understanding of the relationships between features. Features that a particular diagnostic tool identifies as "important" serve as an approximation of variations in model behavior that are linked both to the reported features and their correlated counterparts. Because multiple features and correlations may be involved, other features may also be "important." This consideration is particularly vital as models become more complex and feature-rich as attributing prediction variations to a single feature within a correlated cluster is likely incomplete.

2. **Complementary Roles of Explanation and Optimization** Ex-post explanations complement but do not replace targeted ex-ante optimization towards a specific goal. For instance, optimizing to avoid disparate impact is generally more effective than making ex-post adjustments.

⇒ **With medium to high confidence:**

1. **Simplified Representations of Complex Models** Certain diagnostic tools can effectively identify crucial drivers of application-relevant model behaviors, even when models are intricate and feature-rich. These tools pinpoint a select set of features that, collectively, represent a substantial portion of the model's behavior.
2. **Purpose-driven Descriptions and Adjustments** Tools for describing and modifying model behavior should be tailored to their specific use, as there is no universal or absolute answer to what determines model behavior or how models should be adjusted to achieve a goal. Instead, the appropriate response depends on a deep understanding of the problem at hand, the context in which the model will be applied, and the unique requirements and constraints of the specific use case.
3. **Grouped Feature Explanations and Aggregation Benefits** Combining feature-based explanations with additional structure, specifically feature groupings, can yield significant advantages. Features might be grouped based on their correlations or logical relationships. While our evaluation did not explicitly focus on this aspect, it points toward the potential benefits of assessing the importance of meaningful feature groups rather than individual features.

⇒ **With medium confidence:**

1. **Limitations of Simple Feature-based Explanations** Providing a concise list of key drivers may not be sufficient for all tasks requiring understanding of model behavior or generating usable information about it. If there is considerable uncertainty about important model properties, reducing a model to a few key drivers could result in a loss of valuable information about behavior in atypical situations not well represented by average model behavior.

Contribution This report is the only publicly available, independent evaluation of commercial and open-source model diagnostic tools embedded in the specific context of consumer lending regulation of which we are aware. In the near term, we hope that this report can contribute to the reconsideration of current expectations and practices and the articulation of policy and market practices regarding the development, implementation, and monitoring of machine learning technology in consumer lending. Our research offers a framework for evaluating the quality and usability of information produced about machine learning models' behavior in cases where model transparency is an important threshold question for fair and responsible use of such models. We hope that our evaluation provides a compelling case study for those shaping market practice and policy regarding the fair, responsible, and inclusive use of machine learning models in extending consumer credit and other financial services applications. We also hope that it helps advance academic and policy debates about how to foster the responsible use of machine learning models in medicine, criminal justice, employment, and other sectors, where these prediction tools increasingly affect highly consequential decisions.

2 BACKGROUND

2.1 EXPLAINABILITY AND MACHINE LEARNING IN CONSUMER LENDING

Many lenders have hesitated to adopt machine learning models – or adopted them in forms that limit much of their value – due to uncertainty about an important threshold question: Given that machine learning models can be more complex and less transparent than the models they would replace, can lenders determine whether particular models can be trusted and comply with applicable regulatory requirements?

Machine learning models do not inherently need to be transparent to make accurate predictions. To date U.S. law and regulation have not generally required users of machine learning models to meet defined thresholds for model transparency. But without appropriate transparency, internal and external model stakeholders – model developers, model users, risk managers, regulators, and investors – cannot be confident that a model is fit for purposes or can be or is being used responsibly and fairly. Emerging initiatives from U.S. and international policymakers[2], as well as in academic discourse across several disciplines,[3] have begun to articulate frameworks that identify transparency as a critical component of establishing when AI or ML systems can be trusted.

Highly regulated sectors like financial services have pre-existing legal and regulatory frameworks that force potential users of machine learning to resolve questions about model transparency earlier and more holistically than elsewhere. For example, implementing machine learning in the context of extending consumers credit brings into play regulatory requirements focused on promoting responsible risk-taking and providing consumers broad, non-discriminatory access. Efforts to ensure that emerging uses of technology satisfy these requirements gives technologies that gain acceptance in consumer credit and financial services disproportionate potential to shape how other sectors answer the same questions.

Given these regulatory frameworks, concerns about model transparency shape lenders' decisions at every stage of the process of developing, implementing, and managing machine learning underwriting models. Model developers may in effect work backwards from the transparency requirements of their use case – by designing and planning their modeling approach based on the level and type of transparency required. In practice, the developer of an underwriting model needs to be able to establish that each relationship in the model has an intuitive, defensible relationship to an applicant's likelihood of default.[4] Confronted with the need for model transparency, developers might build inherently interpretable models – ones that promise to be explained and understood without additional analysis. Alternatively, they might build explainable models – ones that use more complex or black box predictive models alongside supplemental models, analyses, and techniques designed to improve the predictive model's transparency. Once they have made that decision, model developers have to decide how much complexity in the model they can or wish to manage.

Lenders' efforts to implement fair, responsible, and inclusive machine learning underwriting models have been shaped by the technical demands of identifying what drives the predictions of machine learning models. These

[2] Emerging approaches to regulating the use of AI in other jurisdictions, such as the EU, have recognized the sensitivity of credit scoring and underwriting among AI applications and called for them to be treated as "high-risk" for risk management purposes. See European Commission, Proposal for a regulation of the European Parliament and of the Council laying down harmonised rules on artificial intelligence (Artificial Intelligence Act) and amending certain Union legislative acts (2021). Similarly, the National Institute for Standards and Technology has proposed a framework for AI risk management, and the Monetary Authority of Singapore released a series of white papers detailing approaches to identifying and measuring Fairness, Ethics, Accountability and Transparency principles in AI systems to guide the responsible use of AI by regulated entities. See National Institute for Standards and Technology, et al., AI Risk Management Framework: Initial Draft; Monetary Authority of Singapore, FEAT Principles Assessment Methodology (2022).

[3] See generally Dwork et al. (2012); Barocas et al. (2017); Hutchinson and Mitchell (2019); Rodolfa et al. (2021).

[4] See Office of the Comptroller of the Currency, Bulletin 1997-24: Credit Scoring Models: Examination Guidance (May 20, 1997); Evans (2017).

demands can be especially challenging where the machine learning algorithms develop and rely on relationships that are inherently complex, non-intuitive, difficult to assess, large in number, or dependent on a large number of input features or relationships. The ability of machine learning models to use non-monotonic[5] and/or non-linear features[6] poses a particular challenge for lenders that must be responsive to the transparency needs of specific laws and regulations that apply to the extension of consumer credit. Some may opt to impose monotonicity or linearity constraints on the learning algorithm to make it easier to describe the behavior of the resulting machine learning models. Others may rely on an array of developments in the data science of explainability that have led to the emergence of varied model diagnostic tools designed to support users of machine learning models.

Firms, policymakers, advocates, and other stakeholders are invested in understanding to what degree these model diagnostic tools can help foster fair, responsible, and inclusive use of AI and machine learning across all sectors. In consumer credit and other use cases, there is much work to be done to assess the capabilities and performance of these explainability tools, techniques, and analyses. This work involves not only assessing the technical reliability of the explanatory information produced by emerging diagnostic tools, but more importantly assessing the usability of such information in specific contexts. This includes understanding and considering the needs of specific stakeholders when developing context-appropriate and usable explanations of a model's behavior, as explainability is a distinct psychological process that depends on the user of the explanation (Broniatowski et al., 2021). Accurate information about a model's behavior may not be sufficient if it does not also enable actions contemplated by public policy, such as the mitigation of discrimination risks. To achieve this, the explanatory information about machine learning models must be usable to support and inform strategic, governance, and risk management decisions involving diverse users of model explanations, each of whom need to be able to understand and act on that information in different ways.

This evaluation is designed to address this gap in the context of consumer lending. Our research evaluates whether and in what circumstances diagnostic tools using *post hoc* explainability methods can support lenders' compliance with a set of specific laws and regulations applicable to consumer credit that in one form or another all depend on the ability to explain underwriting models. Our contribution to understanding the capabilities and performance of these explainability tools, techniques, and analyses in the context of consumer credit takes two forms: articulation of a structured framework for assessing when we can trust information about machine learning underwriting models and applying that framework in a case study that uses real lending data.

2.2 RELATED LITERATURE

Our approach and findings directly relate to current debates in the academic and policy literature on the fair, inclusive, and responsible deployment of machine learning models. We hope that our findings can add to the empirical evidence that helps move these debates forward.

First, we relate to work on the importance and challenges of explainability, interpretability, and transparency of complex machine learning models in critical applications (*e.g.*, Lundberg and Lee, 2017; Slack et al., 2019). Our work aims to add an economic notion of model transparency in the context of specific use cases to existing mathematical notions of complexity and explainability. Specifically, we argue that when used in critical applications, model descriptions should be related to specific policy goals and be interpreted in their specific context. We document that

[5] Adding salt to a savory dish presents an intuitive example of a non-monotonic relationship. A small amount of salt will generally make the dish taste better. However, after a certain point, adding salt will not improve the taste of the dish and, in fact, will make the dish taste worse. This is an example of a non-monotonic relationship, as the relationship is positive in some cases and negative in others, which means the relationship is not one-directional.

[6] A non-linear relationship is one in which increases or decreases in an input feature do not always produce proportionally consistent changes in the target or output feature.

reasonable model descriptions may disagree (related to recent work about disagreement by Krishna et al., 2022).

Second, we speak to a debate in computer science, economics, and law on different ways of restricting models to ensure their fairness and avoid discrimination (Barocas and Selbst, 2016). Our work adds to a growing list of theoretical, simulation-based, legal, and empirical findings that discuss the limits of input restrictions and consider alternatives (Kleinberg et al., 2018a; Gillis and Spiess, 2019; Gillis, 2022). Specifically, we show the promise of an approach that directly optimizes for specific policy targets, such as lowering disparities across groups. Furthermore, we show that input restrictions can be costly for performance with limited gain in reducing disparities.

Third, while not the main focus of our study, we also contribute empirical evidence to discussions around different notions of disparity and fairness metrics (Hellman, 2020). Specifically, we show that different natural measures of disparities lead to different rank-orderings between models, and thus confirm existing theoretical and empirical findings that suggest that different notions of fairness cannot all be fulfilled at the same time but represent inherent trade-offs (Kleinberg et al., 2016; Chouldechova, 2017). At the same time, in our study there are also groups of (typically more complex) models that perform well across measures and dominate other (typically simple) models, suggesting that the choice of measures matters, as does the choice of model class, and a combination of model and optimization can generally improve properties across the board (related *e.g.* to Coston et al., 2021).

Finally, we also relate to a discussion about trade-offs between model complexity, performance, and fairness in the case of consumer finance (Fuster et al., 2022; Bartlett et al., 2022). In our study, notwithstanding the limitations in the development of our underwriting models noted in Section 1 and Section 3.4, the more complex models in this study generally outperform simpler models across measures of both disparity and model fit criteria, confirming the general potential of modern machine learning methods. At the same time, we observe fairness–performance trade-offs within complex models, tracing out a Pareto frontier that joint optimization can achieve.

While our evaluation studies the ability of feature-based model diagnostic tools to respond to specific policy and regulatory needs, we note that there are two other aspects of the transparency of automated machine learning systems that hold promise for their safe and fair use, which we do not deal with in depth in our report. First, we consider descriptions of models themselves, and not of the algorithms that produced these models. But one advantage of increasingly automated model-building pipelines is that they can often be described more completely and more accurately than the hand-curation of features and models by human model-builders, thus offering an opportunity for procedural scrutiny and transparency (*e.g.* Kleinberg et al., 2018b). Second, even complex models can be evaluated by simulating model behavior across hypothetical distributions of applicants or validating on actual segments of particular interest, thus potentially making their performance and critical properties available to regulators even before deployment. This form of "discrimination stress testing" (Gillis and Spiess, 2019) offers an alternative way towards transparency that holds promise even as models become more complex.

3 RESEARCH DESIGN & METHODOLOGY

3.1 RESEARCH QUESTIONS

In this report, (1) we consider when and how model transparency is needed to help lenders serve the goals expressed in certain regulations applicable to consumer lending, and, in those instances where it is, (2) we evaluate the degree to which available model diagnostic tools further those goals – in particular, we evaluate how well those tools help users of machine learning models overcome specific transparency challenges to enable appropriate oversight. We view model transparency as a critical means to an end, rather than an end in itself, where the goal is to enable fair and responsible use of machine learning models and to document compliance with applicable laws and regulations. We view this as a sensible and productive approach given the lack of a widely accepted definition of model transparency or explainability and the absence of a universal standard of what makes a model sufficiently transparent or explainable.

The process of defining our research questions began with extensive outreach to understand what issues lenders, technology companies, policymakers, consumer advocates, and researchers see as critical to promoting fair, responsible, and inclusive use of machine learning underwriting models in consumer credit. Our contribution lies in providing a rigorous and independent assessment of the threshold questions that all stakeholders have identified as critical to defining responsible use of machine learning underwriting models.

3.2 EVALUATION FRAMEWORK

This section introduces our empirical evaluation framework. Our evaluation is based on the idea that model transparency is best viewed as a necessary means to an end rather than as an end in and of itself. We offer a framework that lenders, regulators, and researchers can use as a starting point for evaluating whether and in what circumstances information about how machine learning credit underwriting models work can be used to manage models in accordance with applicable requirements and in the context of specific regulatory compliance tasks. While the existing literature has proposed desirable criteria for explanations produced by diagnostic tools, the strength of our approach is to highlight which of these technical properties are desirable when viewed through the lens of certain legal and regulatory requirements that apply to consumer credit.[7]

We evaluate model diagnostic tools with respect to the following properties: fidelity, consistency, and usability.

3.2.1 FIDELITY

We define fidelity as the ability to reliably identify features that are relevant to a model's prediction. Intuitively, fidelity asks how well an explanation approximates the prediction of the black box model. If the diagnostic tool is not able to reflect the mechanics in the model, the explanation it produces will not be an answer with high fidelity.

3.2.2 CONSISTENCY

We define consistency as the degree to which different tools identify the same drivers either for the same model or across models.

[7] Further context on the relevant legal and regulatory background for adverse action reporting, fair lending, and model risk management is included below in Section 4.1, Section 5.1, and Section 6.1.

We investigate two notions of consistency:

⇒ **Consistency across diagnostic tools.** This notion of consistency assesses whether the same explanation is produced by different diagnostic tools when applied to underwriting models developed with and applied to the same data. Although intuitively attractive in some respects, this type of consistency is not necessarily to be expected or even desirable. If the model diagnostic tools analyzed in this study all exhibit high fidelity, then consistency is likely a favorable property – we obtain similar answers regardless of the precise tool used, although the degree of similarity will depend on the granularity of the results being compared. If, however, some tools perform worse than others, it is not clear that we would expect – *or want* consistency in the information returned about the model. As an extreme example, consider the case where one tool simply randomly draws features. Clearly, we would not want consistency with this random draw. For this reason, we present results by the type of diagnostic tool to reflect the differences in fidelity we presented in the preceding section. To the extent that there is lack of consistency, this evaluation asks if this is driven by the use of different approaches to generating the explanation or if it reflects the fact that differing explanations reflect variables that are highly correlated and playing a similar role in the model. Further, it is important to note that the regulations used to structure our evaluation do not generally require consistency and in some cases such as the generation of adverse action notices explicitly recognize variations in acceptable methodologies that might lead to inconsistent results across tools.

⇒ **Consistency across models.** This portion of the framework asks how similar a given explanation is across distinct models that have been trained on the same task and data. Consistency across models is helpful to gain insights into how diagnostic tools work but is not necessarily a desirable property for users of machine learning underwriting models or these diagnostic tools. If we believe that different models learn similar fundamental (causal) relationships about the world and our goal is to identify the most important of those relationships for disparities, then consistency is desirable. In other words, if we are hoping to learn about relationships in the world, we would expect a good model diagnostic tool to consistently identify these facts about the world. If, however, we believe that different models learn different correlation patterns in the data and that consumer protection regulations are interested in why a particular model exhibits adverse impact, then it is not clear that we would want (or expect) consistency across models. If models are learning different patterns, we would prefer the model diagnostic tool to correctly identify the pattern that drives behavior for that particular model.

Further, we recognize a potential third form of consistency relevant to lending: consistency across similarly situated applicants.[8] Such applicants have similar credit characteristics and would receive nearly identical scores, but the specific inquiry to identify drivers of model behavior relevant to adverse action notices and disparate impact require different inquiries as detailed in subsequent sections.

3.2.3 USABILITY

We define usability as the ability to identify information that helps the recipient act in ways intended by the particular regulation at hand. For example, information about options for reducing disparities in the model can help lenders manage particular compliance risks in accordance with the purposes of fair lending requirements. In the context of

[8] Some of the analysis presented herein uses this aspect of consistency to validate the quality of model explanations from different perspectives. For example, our fidelity tests in the section on adverse action notices uses a nearest neighbor test to identify similarly situated applicants to evaluate fidelity of information provided by the diagnostic tools in the context of adverse action reporting. See subsection 4.4.1.

adverse action notices, this could direct attention to whether the tools can produce information that is advisory, rather than merely descriptive, so that an individual recipient of an adverse credit decision might be empowered to improve their prospects of approval in the future. Unlike fidelity and consistency, the exact definition of usability depends on the specific policy or regulatory goals in question.

3.3 DATA

Our work is based on data from one of the three main nationwide credit reporting agencies for a representative sample of 50 million individuals from across the U.S., covering the period between 2009–2017 (semiannually – May and November). We built prediction models for credit card default and select a random sample of individuals who applied for a new credit card between November 2011 and May 2012. We identify credit card applications based on a credit card inquiry in the data. We identify an opening based on whether an inquiry is followed by a new credit card account up to six months after the inquiry. Since we do not have tradeline-level data we cannot match an inquiry at a particular credit card issuer to an account opening with that same issuer.

3.3.1 DATA DESCRIPTION

Our credit report data is not at the tradeline level but instead provides information aggregated by account type. For example, we have information on the payment histories for all of a consumer's credit card accounts combined, but this information is not broken out separately for each credit card account. The following information is available in our data:

- ⇒ Derogatory public records;
- ⇒ Historical payment delinquencies (at any point in time);
- ⇒ Recent payment delinquencies (1-2 years);
- ⇒ Evolution of debt balance;
- ⇒ Credit availability/utilization;
- ⇒ File thickness (length of history/number of accounts);
- ⇒ Credit inquiries broken down by account type and recency of inquiry;
- ⇒ Mortgage-related features; and
- ⇒ Specific credit card features (limits, payment patterns, utilization, balances).

Several features express changes in these characteristics over time. For example, we might have features expressing how the monthly credit card balances have been changing for the past two years.

We did not use credit score, geography, or income estimates in building the Baseline Models.

In addition to these baseline features, we created outlier indicators and missing value categorical features, yielding a total of 652 features. We replaced the different missing value codes (MVC) in the original data with a single value (-1 or 0). We chose a missing value code of zero for balances and counts where missingness naturally implies a value of zero. We created an indicator for outlier observations for numeric features with thicker-tailed distributions (more than 5% above two standard deviations from the mean). The feature masking still allowed us to identify a family of features consisting of the base feature and its associated outlier and missing value categorical feature. Finally, we transformed certain numeric features to $\ln(1 + \text{feature})$ to account for their skewness. No additional feature selection or pre-processing was performed.

Our data provider required that certain features be masked in the research. For these features, no feature name or description was provided to the participating companies. We did provide background information on each feature (masked and unmasked) to the participating companies. This background information included the type of feature (*e.g.*, ratio or balance), whether we applied a transformation (*e.g.*, taking the logarithm), basic summary statistics, the time horizon of mutability, and the direction of possible change (*e.g.*, positive change only).

All features are obtained from the semester immediately before the application such that the features reflect credit reports before the respective application and inquiry take place. Our outcome of interest (label for supervised learning) is an indicator of severe delinquency (a MOP rating of 4 or 5) for any credit card, any bankruptcy, or any credit card charge-off within 24 months after the opening of the credit card of reference (thus until May 2014, in our case). Note that since we do not have tradeline data available, our labels refer to "any credit card" default as opposed to default on the particular credit card opened up to 24 months ago. This limitation of our data might lead us to observe relatively high default rates when compared to default rates observed in tradeline data. That said, all models in our analysis were trained using the same default outcome. For that reason, our results should not be confounded by this property of our data.

Given resource constraints, we split our sample into a training and a test set for the estimation of all models. The total sample size is just over 600,000 evenly split between training data and testing data. The equal split is motivated by the fact that we conduct our evaluation analysis on the test data set and require a sufficient sample size to have confidence in the results of our evaluation. Table 1 provides statistics on the default labels in the dataset. An out-of-time sample is also discussed and used for various analyses in sections 4, 5, and 6.

3.3.2 PROTECTED CLASS DATA

The protected class indicator considered in this research consists of a flag that divides applicants into two groups based on race/ethnicity. The majority group (or non-minority group) consists of non-Hispanic White applicants. The minority group is defined as either Black or Hispanic and excludes the category Asian. We infer race/ethnicity status using a Bayesian Improved Surname Geocoding (BISG) approach, which uses name and geographic information to predict race and ethnicity (see Appendix D for details).

Table 1: SUMMARY STATISTICS: DEFAULT LABELS

	# Observations	# Default	# Default Missing	Default Rate
Training Data	312,715	21,746	142,033	14.60%
Test Data	337,737	34,537	108,015	15.03%

Note: This table summarizes the sample training and test set sizes and default label counts. Default refers to credit card default up to 24 months after opening a new credit card account. Default Missing are driven by rejected applicants in the data.

3.4 MODEL DEVELOPMENT

We consider two types of credit underwriting models in this study. The first set of models (each a "Baseline Model" or collectively, the "Baseline Models") were built by the research team. A second set of models were built by the participating companies on an identical training data set (each a "Company Model" or collectively, the "Company Models"). The development process for the Baseline Models and the Company Models was less intensive than

lenders typically conduct. For example, we did not go through iterative fair lending compliance reviews to mitigate bias in Baseline Models, in part because that kind of analysis is part of the evaluation (see Section 5.4). We describe each set of models in turn.

3.4.1 BASELINE MODELS

To support evaluation of the identified proprietary and open-source diagnostic tools, the research team developed machine learning models using commonly available algorithms and software tools. Throughout this process, we consulted with an array of stakeholders with expertise in credit modelling and consumer lending to help ensure that – while "out-of-the-box" – our models still approximate industry practice wherever feasible given available data and resources. Executives at established bank and non-bank lenders reviewed and provided feedback on qualitative and statistical information we provided about our models. We believe that the Baseline Models fit the purpose of this evaluation – that is, they sufficiently emulate industry practice to support evaluation of the capabilities, limitations, and performance of the kinds of model diagnostic tools offered by the participating companies. One area of divergence between our models and production-grade underwriting models – and the sense in which our models are "out-of-the-box" – is the time spent on manual feature pre-processing and selection.

We fit four types of models to the training data: two "simple" models and two "complex" models. Our aim was to set up within this set of models comparisons between simple and complex machine learning models based loosely on differences in model architecture and the number of features used in the model. However, our use of the terms "simple" and "complex" is relative and not meant to conform to any external standard, as we are not aware of any one dominant usage of these terms in the context of debates about responsible machine learning, model transparency, and/or consumer lending. The term "complex," in particular, is *not* used to describe all machine learning models or compare machine learning models to models developed with more conventional means. Accordingly, a lender that is using what it considers to be a relatively simple model may fall somewhere in the middle of the range defined by our four models.

The following four algorithms were chosen to create a set of underwriting models that would be used to evaluate diagnostic tools from the participating research companies (each a "Baseline Model" or, collectively, the "Baseline Models"):

⇒ Logistic Regression

⇒ Simple Neural Network

⇒ XGBoost (Classifier)

⇒ Neural Network (NN)

Complex models The complex models (XGBoost and neural network) are built on the full set of 652 features that we obtained after the feature pre-processing steps described in Section 3.3. Hyperparameter tuning for the complex models is based on a random search algorithm over a predetermined search space, together with the use of cross validation for performance evaluation and choice of best model from the ones searched. Based on the library documentation and available materials on consumer credit risk prediction models, we gather a list of tunable hyperparameters and define initial search ranges. After conducting the random search over this initial search space and analyzing the ranking of searched combinations by the respective performance score, we observe whether the top

performing combinations (average score over validation sets) include values at the margin of the different hyperparameter search ranges. In such cases, we extend the respective search range in the appropriate direction. Table B.1 in the appendix reports the search space and chosen hyperparameters for each model. Across all machine learning models, we use 100 iterations in the random search and use 5-fold cross-validation. Our Python implementation relies primarily on the `scikit-learn` library.

Simple models Both sets of simple models (logit model and simple neural network) use a subset of features that are selected using LASSO, a common feature selection method. LASSO identified 44 out of the full set of 652 features, which are then used in the training of the logistic regression (logit) and simple neural net models. Neither of the simple models use a search space for tuning the hyperparameters, and the parameters used are based on a manual trial-and-error method. In particular, the simple neural network model, built only on 44 features, uses only 2 hidden layers with 8 nodes (compared to 2 hidden layers and 30 nodes for the complex neural network model).

It is important to note that our logit model is not meant to approximate incumbent underwriting models prior to adoption of machine learning and is likely more complex than those models. But it does share with those incumbent models a high degree of transparency in the sense that the model's coefficients can in principle be directly inspected.

3.4.2 COMPANY MODELS

Four out of the seven participating research companies opted to provide one or more custom models (each a "Company Model" or, collectively, the "Company Models"). These custom models were trained on the same data set as the Baseline Models. We provided the full set of 652 features to all companies (see Section 3.3 for a broad description of these features). None of the companies chose to conduct reject inference, that is, they dropped applicants with missing default labels from the training data. Each of these companies made its own decisions about what type of model to deliver and how to balance various concerns, such as level of the complexity, transparency, and predictive accuracy appropriate for a prediction model for use in the context of extending consumer credit. Below we provide a brief description of the types of models represented in the set of Company Models.

Unspecified Proprietary Model One company chose to select the 20 most important features and build a model using a proprietary modeling approach designed to provide greater intepretability and transparency. The missing value indicators and the outliers indicator variables were disregarded. All the categorical variables underwent frequency encoding based on James-Stein methodology. The variables that had more than 80% of their values missing were dropped. Multicollinearity analysis was conducted and the variables that had more than 98% correlation among each other were omitted.

Random Forest Classifier One company submitted a random forest classifier. A grid search was performed to tune for the best hyperparameters. No reject inference was conducted.

Ensemble of Generalized Linear Models One company submitted a model that is a stacked ensemble of four generalized linear models. This model was trained on the original features only.

Ensemble of Gradient Boosted Machines One company submitted a stacked ensemble of gradient boosting machines trained on the original features, manually engineered features, and automatically engineered features. The

models used 1306 features, including 344 from the set of original and manually created features and 962 automatically generated features. The final stacked ensemble included five gradient boosters – four XGBoost models and one LightGBM model.

Monotonicity-Constrained XGBoost Model One company provided a monotonicity-constrained XGBoost model that contained 192 features. The company detected and replaced 260 features that were missing values with NaN. 260 binary flags to indicate a missing value corresponding to those features were added to aid with interpretability. From this larger set, 255 duplicate features were detected and removed. 73 highly correlated features were also detected and removed. A greedy feature selection algorithm identified and removed 392 features that contributed little to predictive performance of the model. Monotonicity constraints were automatically applied to 187 non-categorical features based on analysis of SHAP feature importance. Bayesian hyperparameter search was conducted based on a validation dataset and XGBoost hyperparameters that maximized predictive performance as measured by AUC on the validation dataset were selected.

XGBoost model One company submitted an XGBoost model with 500 input variables after the removal of 152 duplicate features. Bayesian hyperparameter search was conducted based on a validation dataset and XGBoost hyperparameters that maximized predictive performance as measured by AUC on the validation dataset were selected. No further tuning or preprocessing was conducted.

3.4.3 MODEL PERFORMANCE

Table 2 shows model performance on the test set for the Baseline and Company Models. The performance of the Company Models is similar to our Complex Baseline Models. Complex models perform better than simple models. We show the following performance metrics.

A common way of quantifying the magnitude of classification errors is in terms of the receiver operating characteristic (ROC) curve. The ROC curve plots the true positive rate against the false positive rate at each credit score threshold. A standard summary measure of classification accuracy is the area under the ROC curve (AUC). An AUC value of 0.5 corresponds to classification no better than a random pick whereas an AUC value of 1 corresponds to perfect classification.

Log Loss is a common evaluation metric that determines how close the predicted probabilities are to the true outcomes. Mathematically, Log Loss is -1 times the logarithm of the likelihood function. Intuitively, the likelihood function answers the question of how likely a prediction model thought the observed set of true outcomes is. Smaller values correspond to more predictive accuracy. Unlike the AUC metric, Log Loss takes into consideration not just whether the model predicted the wrong outcome (*e.g.*, a default when the applicant in fact did not default) but also by how much the prediction was off.

Mean Squared Error (MSE) computes the difference between the model's predictions and the true outcome. The MSE metric is obtained by first squaring these differences and then reporting the average across the test set. Smaller values correspond to more predictive accuracy. Unlike the AUC metric, MSE takes into consideration not just whether the model predicted the wrong outcome (*e.g.* a default when the applicant in fact did not default) but also by how much the prediction was off.

Similar to the AUC metric, the Gini coefficient is used to evaluate the performance of a prediction model for a binary outcome, that is, default or no default. The value of the coefficient ranges between 0 and 1, with higher

numbers corresponding to better performance.

The Kolmogorov-Smirnov (KS) metric is also used to evaluate the performance of a prediction model for a binary outcome, that is, default or no default. The KS metric measures the degree of separation between two statistical distributions. The KS statistic is 1 if the prediction model perfectly separates loan applicants into true default and true non-defaults (*i.e.*, it perfectly predicts outcomes). In contrast, the KS statistic is 0 if the model made a random guess. Higher values of the KS statistic therefore correspond to better performance.

Table 2: MODEL PERFORMANCE

Company Models	ROC AUC	Log loss	MSE	KS Stat	Gini coefficient
Alpha model	0.866	0.298	0.093	0.585	0.619
Beta model	0.871	0.294	0.091	0.591	0.623
Gamma model	0.870	0.295	0.092	0.590	0.625
Delta model	0.868	0.297	0.092	0.586	0.626
Epsilon model	0.844	0.317	0.098	0.554	0.549
Zeta model	0.863	0.302	0.094	0.578	0.591
Simple models	ROC AUC	Log loss	MSE	KS Stat	Gini coefficient
Logit	0.820	0.335	0.105	0.516	0.554
Simple NN	0.802	0.340	0.105	0.523	0.530
Complex models	ROC AUC	Log loss	MSE	KS Stat	Gini coefficient
XGBoost	0.871	0.294	0.091	0.591	0.626
Neural net	0.860	0.317	0.097	0.574	0.690

Note: This table shows predictive performance metrics for the underwriting models in this study. The top panel shows models built by participating companies. The middle panel shows the two simple Baseline Models and the bottom panel the two Complex Baseline Models.

3.5 EVALUATION PARTICIPANTS

This evaluation assesses the performance and capabilities of a set of proprietary and open-source model diagnostic tools. The intent is not to identify winners or losers among those tools, but rather to understand in more nuanced and context-specific ways whether and how currently available explainability techniques can support fair and responsible use of machine learning underwriting models.

All of the techniques and tools in this evaluation are plausible for use in conjunction with models that evaluate consumer credit applications, in part because they can meet the speed requirements of desktop underwriting and digital lending.

The proprietary tools in this evaluation come from a set of technology companies that differ in several respects: business models and strategies; depth of experience in financial services and consumer credit; corporate development and resources; and client bases. Given these differences, the capabilities of the tools vary, which in some cases meant that individual companies opted not to participate in some parts of the evaluation. In many cases, specific tasks in the evaluation required the companies to do something or produce information in ways that varied from their normal interactions with clients, and several companies participated in aspects of the evaluation that were outside the capabilities currently offered to clients. Further, the companies vary in how they engage with their clients. Some provide software tools that enable model development and model monitoring with little interaction with clients beyond training and technical support. Others offer clients a full range of advisory services to support use of their

software tools.

Considered together, these factors mean that the individual tools in this evaluation reflect a variety of specific design choices made to solve technological, operational, and strategic challenges relevant to each company's clients. By working with companies actively offering the model diagnostic tools in this study, the research team leverages the expertise and resources each company has brought to bear on these challenges. However, this may mean the tools reflect very different judgments with respect to aspects of this evaluation, given uncertainty about market and regulatory factors. For instance, as discussed more below in Section 5.1, the use of protected class information in model development processes is one area where the companies' judgments about regulatory interpretation and strategy result in the creation of tools that reflect different approaches.

The technology companies represented in the evaluation do not represent every known approach to explaining and managing machine learning models. Still, we believe that collectively they reflect many of the dominant approaches and meaningful variations in methods for producing model explanations that are commercially available.

The proprietary tools in this evaluation have been provided by the following companies:[9]

⇒ **ArthurAI's** platform measures and improves machine learning models to help data scientists, product owners, and business leaders accelerate model operations and optimize for accuracy, explainability, and fairness. Arthur's research-led approach supports a range of capabilities in computer vision, natural language processing, bias mitigation, and other critical areas.

⇒ **FiddlerAI** offers an enterprise solution for teams to monitor, explain, analyze, and improve their models and build trust into AI. The unified environment provides a common language, centralized controls, and actionable insights to operationalize machine learning/AI with trust. Fiddler integrates deep explainable AI and analytics to help clients grow into advanced capabilities over time and build a framework for responsible AI practices. Fiddler's platform can be used across training and production models to accelerate AI time-to-value and scale and increase revenue by connecting predictions to business context.

⇒ **H2O.ai** partners with organizations across sectors and around the world, accelerating capabilities in automated machine learning (autoML), time series forecasting, and responsible AI. Its platform, the H2O.ai Cloud, enables businesses, government entities, nonprofits and academic institutions to make, operate and innovate with AI.

⇒ **RelationalAI** (RAI) provides the first knowledge graph built for developing intelligent data apps. RAI stores graphs and logic together as executable models. This knowledge-centric approach connects domain expertise to systems for advanced analytics and composite AI from fraud prevention and recommendations to network optimization. As a cloud-native system, RAI fits into existing architectures to leverage any data type and accelerate data-centric development.

⇒ **SolasAI** provides software that affords modelers, compliance stakeholders, and executives the ability to minimize discrimination in their predictive decisioning models without hurting their business. SolasAI builds on the thought leadership and expertise of BLDS, LLC – an industry-leading consultancy with decades of experience advising on the topic of algorithmic fairness across areas such as banking, insurance, healthcare, and employment. In recent years, BLDS has focused on the development and implementation of techniques that provide a clearer understanding of AI decision-making and evaluate the fairness of such models which are now available through the SolasAI application.

[9] These extracts were prepared by each company participating in this evaluation.

⇒ **Stratyfy** is an ethical AI company that offers predictive analytics and decision optimization software for credit and risk teams, helping lenders provide more people with access to fair and transparent credit. Stratyfy's unique solutions provide the level of understanding and control that regulated institutions require to proactively identify and mitigate bias and make better credit decisions. With Stratyfy's solutions, users can seamlessly combine the precision of data and the wisdom of domain expertise, optimizing risk-based decisions without introducing regulatory or operational risk.

⇒ **Zest AI's** software helps lenders of all sizes make more fair and accurate credit underwriting decisions. Zest's Model Management System allows lenders to build, adopt, and operate models that use hundreds of FCRA-compliant attributes and advanced machine learning techniques to improve the accuracy of risk assessment and to successfully underwrite borrowers from under-represented groups. Since 2009, Zest has provided credit scores for hundreds of millions of prospective borrowers worldwide, including those with little to no credit history.

Anonymity To protect the anonymity of these companies and the confidentiality of their proprietary methodologies, results for individual tests are presented herein with masked identifiers. The identifiers are varied across different sections of the report. Finally, we have embedded in the discussion of results an overview of methodologies used to produce information responsive to the evaluation protocol. These descriptions will help readers understand the range of methodologies used by the companies in this study but will not identify which company or companies have used particular approaches.

3.6 OPEN-SOURCE TOOLS

The research team has also included its own implementation of open-source tools to present a benchmark for the performance of the responses provided by the participating companies, many of which build on open-source techniques in developing their proprietary offerings. The open-source techniques used in various parts of this evaluation include:[10]

⇒ **LIME**: Local Interpretable Model-Agnostic Explanations (LIME) is an explainability technique for complex models that uses local linear surrogate models around a particular data point to approximate the complex model's output.[11] The resulting local surrogate models are used to both explain the model's behavior around individual data points and to quantify feature importance for the overall model. This surrogate model does not altogether explain how the model arrived at the result but focuses instead on how slight changes may affect the ultimate prediction of the model. LIME includes a fidelity measure, giving the user insight into how well the explanation from the surrogate model approximates the underlying model.

⇒ **SHAP**: Shapley Additive Explanations – known widely as "SHAP" – is an increasingly common approach to explaining complex model outputs. SHAP uses mathematical methods derived from a significant body of cooperative game theory research[12] to analyze and explain the contributions of particular features to a model's

[10] For a more detailed description of how various explainability techniques work, see Section 3.4 of FinRegLab (2021).

[11] In general terms, LIME develops surrogate models by sampling several data points and labeling them using the complex model. LIME then assigns weights based on how far away the sample points are from the particular point being explained, giving a larger weight to the sampled points closest to the point of interest. Finally, LIME trains an interpretable model – typically a linear model – on the weighted points to produce the surrogate model (Dieber and Kirrane, 2020).

[12] See Shapley (1951); Aumann and Shapley (2015).

prediction.[13] Similar to LIME, SHAP explains how a model behaves locally. SHAP measures feature importance by conditionally averaging over features from a data point and quantifying how much the removed features impact the model output.[14] Since an exact computation of SHAP values is computationally infeasible, software tools provide efficient approximations. The most popular of these tools is a software package called SHAP.

⇒ **Permutation Importance**: This method measures how important a feature is to a model by calculating how the feature impacts the model's accuracy. Permutation Importance values are calculated by randomly shuffling (or "permuting") the values of the feature in the test dataset – so that every data point has a new value for the feature, and that value comes from a different data point. Permutation is considered more realistic than other methods for modifying the dataset (such as adding random noise to each feature), since all values are taken from the original dataset.

[13] The concept behind Shapley values is as follows: in a cooperative game with N players and a function that values how much total output is generated if all the players contribute together, the Shapley value is a method that attempts to measure the individual contribution of player i to the output generated by the cooperation of all players. If the features are the players in a given complex model, from an economic standpoint, the Shapley value can be interpreted as a weighted average of a feature's marginal contribution to every possible subset of grouped features(Kumar et al., 2020).

[14] Mathematically, this process works as follows: if the point to be explained has three associated features, x_1, x_2, and x_3, binary features are assigned to each one representing whether the feature is known or unknown (so $z_1 = 0$ if x_1 is unknown or missing, and $z_2 = 1$ if x_2 is known). Next, SHAP values (feature importance values) are generated (a_1, a_2, a_3), assigning a score to each of the features that present a score for each of the features. The higher the score, the more important the feature.

4 RESULTS: ADVERSE ACTION NOTICES

This section presents our evaluation of the participating model diagnostic tools with respect to a particular consumer disclosure requirement: adverse action notices. It consists of two main sections: (1) background and (2) empirical evaluation. The former section provides an overview of relevant policy considerations, legal and regulatory expectations, and operational considerations. The latter section presents our analysis that evaluates how well diagnostic tools identify drivers of an adverse credit decision for purposes of producing required disclosures and also considers broader policy considerations regarding improving the actionability of such disclosures.

4.1 BACKGROUND

This section considers in turn the legal and regulatory requirements regarding adverse action notices and operational considerations relevant to this evaluation.

4.1.1 LEGAL AND REGULATORY OVERVIEW

Adverse action notices are among the most direct model transparency requirements in consumer lending.[15] The Equal Credit Opportunity Act and the Fair Credit Reporting Act require lenders to disclose to consumers their principal reasons for denying credit applications or taking other "adverse actions," including offering less favorable terms based on information in applicants' credit reports.[16] These disclosures must describe the facts that were "relevant to a decision, but [need not provide] a description of the decision-making rules themselves." (Selbst and Barocas, 2018). These regulations do not mandate a specific methodology for determining the principal bases for a decision or require lenders to state whether the applicant's assessment as to those bases was too high or too low.[17] The regulations discourage presenting to recipients more than four principal bases for adverse decisions.[18] These requirements reflect broader efforts to prohibit discrimination, enable the correction of errors in credit reports, and educate consumers about managing their finances. Contemporary discussion of these provisions has increasingly focused on how we serve broader policy goals by helping applicants who receive adverse credit decisions understand their financial position and potentially adjust their financial behavior to improve the probability of a favorable credit decision in the future.[19]

[15] This overview and its counterpart in the fair lending and disparate impact section exclusively considers requirements for lenders operating in the United States and focuses on requirements applicable to the distinct step of estimating the probability of default associated with an application for credit. That means this evaluation does not consider activities like developing pre-screened offers of credit, fraud reviews, stress testing, and other areas of regulatory obligation.

[16] The laws define "adverse action" to include denials of credit applications on substantially the same terms and in substantially the same amount as requested, unless the lender makes a counteroffer. Adverse actions also include unfavorable decisions on existing credit arrangements, such as negative changes in terms, denials of line increases, and reductions or cancellations of credit lines. 15 U.S.C. §§1681a(k)(1), 1691(d)(6). In 2011, an FCRA amendment took effect to require similar risk-based pricing notices where credit terms are "materially less favorable" than the terms granted to a "substantial proportion" of other consumers. 15 U.S.C. §1681m(h); 12 C.F.R. §§222.70-.75. ECOA's disclosure requirements apply to both consumer and commercial credit, although some details are different for business applicants. Federal agencies have excluded business credit from FCRA's disclosure requirements. 15 U.S.C. §1681a(c); 12 C.F.R. §§222.70(a)(2), 1002.9(a).

[17] 12 C.F.R. pt. 1002, supp. I, cmt. 9(b)(2)-3, 4, 5. For example, when determining the principle reasons for an adverse decision, regulatory guidance allows lenders to benchmark the recipient of an adverse credit decision against applicants whose total score was at or slightly above the minimum passing score, against the average for all applicants, or to rely on other methodologies that produce substantially similar results. 12 C.F.R. pt. 1002, supp. I, cmt. 9(b)(2)-5.

[18] In the event that the number of inquiries is a key factor, the adverse action notice may state up to five principal bases for the decision. See 12 CFR pt. 202, Supp. I, cmt. 9(b)(2)-9.

[19] See Ficklin et al. (2020), Board of Governors of the Federal Reserve System, Supervisory & Regulation Letter 11-7: Supervisory Guidance on Model Risk Management (Apr. 4, 2011); Office of the Comptroller of the Currency, Bulletin 2011-12: Sound Practices for Model Risk Management: Supervisory Guidance on Model Risk Management (Apr. 4, 2011); Federal Deposit Insurance Corporation, Financial Institution Letter 22-2017: Adoption of Supervisory Guidance on Model Risk Management (Jun. 7, 2017).

In the context of AI and machine learning systems, the CFPB recently acknowledged both the use of machine learning underwriting models and secondary explainability techniques in consumer credit. Its recent compliance circular reminded supervised entities that they must "validate the accuracy" of their chosen methodology for producing adverse action notices, especially where those techniques "approximate models" to explain "credit decisions based on complex algorithms."[20] The issuance did not specify appropriate methodologies for assessing the accuracy of various secondary explainability techniques to generate adverse action notices.

Lenders rely on "reason codes" to help fill the gap between technical explanations of the principal reasons driving predicted default risk and explanations that communicate more effectively the consumer's positioning in relation to their credit standards. Lenders can populate consumer disclosures with the set of reason codes suggested in implementing regulations[21] or similarly broad categories to reduce regulatory uncertainty and manage concerns about revealing competitively sensitive information about their underwriting processes and strategies. This approach may also facilitate provision of information about the credit decision that is likely to be more understandable to consumers – both because the reason codes often use less technical language than the names of individual model features and because the reason codes can serve to aggregate very specific findings into explanations that provide a more holistic sense of why the adverse decision occurred.

4.1.2 OPERATIONAL CONSIDERATIONS

In order to produce adverse action notices, lenders must be able to do two things: (1) identify drivers of the model's prediction for individual applicants who are subject to adverse decisions and (2) map those drivers to descriptions or reason codes that will be given to the consumer. Generating information about drivers of adverse credit decisions is an almost entirely technical challenge, and most of the firms participating in this evaluation are equipped to provide this information for consumer lending clients. Mapping those drivers to reason codes is frequently automated but reflects individual judgments that lenders make about what information to provide on the required disclosures (including whether to use the illustrative reason codes in Regulation B or use bespoke reason codes). Even with incumbent logistic regression models, lenders often group the identified features by correlations or logic so that the mapping step will produce more meaningful and holistic statements about the applicant's financial positioning. Viewed in this light, one issue related to the use of more complex models is whether and how much additional information about the model's behavior is lost when information about a 100, 1000, or 10000 features is compressed in the mapping process into four reason codes.

Few of the companies participating in this evaluation provide these mapping services to their clients, and the restrictions on our ability to share the names of certain features in our data set with the participating companies would have further complicated this process. Accordingly, this evaluation is limited to only the first activity – identifying drivers of the model's prediction for applicants who were subject to adverse decisions.

Production of adverse action notices requires that lenders be able to describe the reasons behind individual model predictions. Technically, this requires that models have sufficient transparency so that two things can be discerned: which features affect the model's prediction and how those features affect estimated relationships in the model. Producing the features that affected a specific prediction by a model – a local explanation[22] – can be challenging

[20] CFPB Compliance Circular 2022-03, fn 1.

[21] 12 C.F.R.pt.1002, App.C. The list is based on historically common underwriting factors and actually does provide some explanation for many of the items listed, such as "Income insufficient for amount of credit requested," "Insufficient number of credit references provided," and "Unacceptable type of credit references provided." Others such as "Length of employment" and "Length of residence" are more general. *Id.*

[22] A local explanation identifies the basis for a specific prediction made by a model. By contrast, a global explanation refers to the identification of a model's high-level decision-making processes and such explanations are frequently used to evaluate a model's overall behavior and fitness-

with machine learning models. But arguably providing key technical drivers of an underwriting model's predicted default risk or score does not fully address the adverse action requirements. Adverse action notices need to include principal reasons for the adverse decision, which may include both model-level factors and decision rules that do not derive from the prediction of an underwriting model.[23] To determine which model-level factors are "principal," lenders must assess those factors for an individual applicant relative to some baseline, such as the average for all applicants or the average for a group of accepted applicants. This analysis helps to determine which factors are driving that particular applicant's predicted default or score the most in the wrong direction.

Lenders report that they use complex analytical and judgmental processes to generate adverse action reason codes based on identified drivers of an adverse decision related to the model and those related to non-model decision rules. One way to identify model-level reasons is to consider for each recipient of an adverse decision a counterpart whose application was accepted. The reference counterpart can be either the same for all adverse action recipients or can be chosen based on specific attributes of the applicant in question. Various analytical methods can then be used to understand the distance between the adverse action recipient and the chosen counterpart in order to identify key drivers of the differences between them. Another related approach is to consider the contribution of individual features to the difference in predicted default or score between a particular recipient of an adverse decision and a collection of approved borrowers. The reason identified by either of these approaches – key model-level drivers of a specific adverse decision – can then be mapped to the more holistic, consumer-facing reason codes that will populate the required disclosure.

4.2 EMPIRICAL EVALUATION: SUMMARY

This section describes key results from our evaluation of model diagnostic tools in the context of consumer disclosure and adverse action notice requirements.

We evaluate diagnostic tools designed to describe features related to adverse credit decisions on three dimensions: (1) fidelity, that is, the ability to reliably identify features that are related to adverse credit decisions; (2) consistency, that is, the degree to which tools identify the same drivers for the same applicant; and (3) usability, that is, the ability to identify drivers that provide a rejected applicant with a feasible path to acceptance within one year.

KEY FINDINGS

We evaluate diagnostic tools designed to identify key features related to an adverse credit decision. There are a set of diagnostic tools that are able to identify features of rejected applicants such that other applicants who are similar on those features are also likely to be rejected. These tools are also able to identify features that, when changed in a favorable direction, reduce predicted default probabilities by more than randomly chosen or even closely correlated features. However, not all models perform equally well, and lender choices about which diagnostic tools to use and how to deploy them is important to achieving their consumer disclosure goals, particularly for complex models. Careful interpretation of the output of model diagnostic tools is central to their effective use in consumer disclosure. Our usability tests show changing only a few features in isolation is unlikely to overcome a rejection, even for the best-performing tools. However, an approach that considers broader groups of related features can provide a plausible path to loan acceptance.

for-use.

[23] A decision rule that categorically prohibits extending credit to consumers with bankruptcies listed in their credit records is an example of a non-model-based driver of an adverse credit decision.

Our main results are as follows:

1. There are a set of diagnostic tools that exhibit high fidelity across both simple and complex models in a specific sense. These tools are able to identify features of rejected applicants such that other applicants who have similar credit characteristics are also likely to be rejected. These tools are also able to identify features that, when changed in a favorable direction, reduce predicted default probabilities by more than randomly chosen or even closely correlated features. Although our study could not assess the process for mapping outputs of the tools to the types of more holistic "reason codes" given to applicants on adverse action notices, the identification of feature-level information is the critical first input to the process of producing those consumer disclosures, making our findings relevant to adverse action notices for credit decisions informed by machine learning underwriting models.

2. Lender choices about which diagnostic tools to use and how to deploy them can be important to achieving their consumer disclosure goals, particularly for complex models. The high-fidelity tools all use a version of Shapley Additive Explanation ("SHAP") feature importance measures and identify drivers as those with the largest positive values (contributing most to a high default prediction for a particular applicant).[24] The remaining tools, which select drivers either based on absolute SHAP feature importance values or use a version of Local Interpretable Model-Agnostic Explanations ("LIME"),[25] perform well for simple models but struggle to provide fidelity for complex models. The tools that perform well on fidelity often identify somewhat different sets of features to describe the behavior of underwriting models, even for the same regulatory purpose. In other words, these tools identify many of the same drivers of an adverse decision for the same applicant. In general, if two tools have a similar fidelity performance – either good or bad – the chances are high that they also exhibit a similar level of consistency with each other, suggesting that successful tools are successful in the same way, while tools with limited fidelity also struggle in the same way. In contrast, high-fidelity and low-fidelity tools tend to exhibit little overlap in the drivers they identify.

3. Careful interpretation of the output of model diagnostic tools is central to their effective use in consumer disclosure. Our two fidelity tests suggest that analyses of which features drive an adverse action decision are more powerful if they account for feature correlations. Our fidelity test based on inspecting approval decisions for similar – nearest neighbor – applicants implicitly accounts for these correlations. Correspondingly, we find that most nearest neighbors would have also been rejected by the model. In contrast, our perturbation test manipulates drivers of an adverse credit decision *in isolation* and exhibits muted reductions in default probabilities. Since complex models put small weights on individual features, looking at the effect of individual features in isolation has inherent limits in producing sizable changes in predicted default probabilities.

4. While model diagnostic tools are able to identify plausible drivers of model behavior, these descriptions do not automatically translate into actionable paths for rejected applicants. This finding is based on our usability tests that asked companies to provide information that might plausibly let rejected applicants obtain a more favorable outcome within one year. Specifically, we analyze whether paths to loan acceptance suggested by the participating companies overcome the adverse decision and provide a feasible path to an approval. While model diagnostic tools do better at this task for simple models, changing the proposed drivers is often not sufficient to overturn the adverse credit decision for complex models – unless we either allow for a large number

[24] For an explanation of SHAP, see Section 3.4.2.2.1 of our Market Context & Data Science Report FinRegLab (2021).
[25] For an explanation of LIME, see Section 3.4.2.1.1 of our Market Context & Data Science Report FinRegLab (2021).

of changes or for impractically large magnitudes of changes (relative to what we observe in the data). This finding reflects that in models with more features, and especially in those that constitute complex models in this evaluation, the marginal contribution of any one feature is small. This points to increased potential for greater information loss for users of complex rather than simple models in that production of adverse action reason codes in conventional mapping processes may compress more information for complex models in order to provide up to four reasons to the applicant. However, we want to caution that our findings should not be interpreted as an argument against model complexity. Rather, we see the challenge of providing actionable sets of features as an inherent limitation of a complex world with non-trivial feature relationships and inter-dependencies. Even for simple models, this process can be challenging because many of the feasible paths to acceptance that exist in the world are not directly captured by the model of relatively few features. In the end, describing the effect that a change in applicant behavior would have on the approval decision likely requires an understanding of the causal relationship between features, which cannot be learned from the data alone and goes beyond the scope of this study.

4.3 PARTICIPATION DETAILS AND DIAGNOSTIC TOOLS

4.3.1 DESCRIPTION OF TASKS

Participating companies were asked to complete two tasks pertaining to required adverse action disclosures. Both tasks were performed on a set of 3000 rejected applicants who had applied for a credit card based on a credit inquiry in their credit report. This set is a random sample drawn from the set of applicants in our training data for whom our XGBoost Baseline Model predicted a probability of default higher than 10%. We use an approval threshold of 10% throughout the analysis. This threshold leads to an average predicted default rate of roughly 3% across the approved applicants. We refer to applicants whose predicted probability of default exceeds the 10% approval threshold as rejected applicants – or 'rejects'. Our analysis could easily be replicated for other approval thresholds or other populations, including samples focused on applicants who were relatively close to obtaining approvals. The set of rejects is identical across all tests in this section. Note that because the rejects were sampled based on the prediction of a particular prediction model, it is possible for a 'rejected' applicant to receive a default prediction below the 10% approval threshold by another prediction model.

1. The first task asked each company to generate four drivers of adverse credit decisions for the set of 3000 rejected applicants. Companies provided a set of drivers for each Baseline Model as well as for the company's underwriting model(s) where applicable. The choice of four drivers was motivated by the number of drivers that are typically provided by adverse action notices. Companies did not map those local explanations into reason codes as commonly provided to consumers under current disclosure regulation. This choice partly reflects the required feature masking by our data provider as well as the fact that few of the companies participating in this evaluation provide these mapping services to their clients. All of our tests therefore pertain to the local explanations of the model's prediction, as opposed to the coarser information provided on adverse action notices – the reasons that individual consumers typically see in practice.

2. The second task asked each company to identify a feasible path toward acceptance within 12 months for each of the 3000 rejected applicants. Since this exercise goes beyond current law, our choice of a period of one year for the feasibility analysis is arbitrary – it simply reflects a period long enough for the identified behavioral changes to affect a person's credit criteria and short enough to be fully captured for most individuals in our

data sample. We again asked for a small number of changes (ideally up to four) though some approaches required that we relax the constraint on the number of changes. To enable companies to incorporate domain-specific constraints, we provided the following additional information for the usability tasks. First, we provided information on whether (and in which direction) a feature was mutable over 12 months. This information was based on a manual assessment of each feature description by the research team. For example, the length of the credit history can increase but not decrease over a 12-month window. Second, we provided summary statistics on changes typically observed in the data over a 12-month window. To compute these statistics, we pulled data on the same individuals 12 months and 24 months prior to their application date and computed two sets of changes over a 12-month window each. Third, for categorical features, we provided transition matrices that summarized the most likely transitions to another level of the categorical features in the data.

These summary statistics have two important limitations. First, we did not provide summary statistics for different starting values of a feature. For example, a 10% change in debt balances is quite a different change when we start from a debt balance of $100 versus a debt balance of $100,000. Second, we did not provide information on which features tend to change together. There may be mechanical interdependencies in the data, for example, if the *maximum* credit utilization over the past 12 months changes, the *average* credit utilization over the past 12 month would also change. There may also be interdependencies that are not mechanical but frequently observed in the data. For example, if an applicant's credit utilization increases, the applicant's total debt balance is also likely to increase. Both logical and effective interdependencies were not provided as part of the summary statistics given the complexity of documenting these changes. These limitations imply that we might over- *or* underestimate the degree to which feasible paths to acceptance exist. If we ignore that some other features are likely to mechanically improve along with the suggested changes, we will underestimate the potential for feasible paths to acceptance. However, it is also possible that other features would deteriorate as a result of the suggested change – leading us to overestimate the size of the suggested improvement. Consider the case where an applicant is told to reduce the outstanding balances on credit cards. The applicant might instead turn to other sources of credit such as personal loans and increase loan balances on personal loans. The net effect on the credit score of these two opposing changes might be much smaller than the estimated effect of looking solely at the reduction in credit card balances.

4.3.2 PARTICIPATION DETAILS

Out of seven companies participating in this study, six companies participated in the adverse action notice part of the research. Out of these six companies, one company completed only the second task of finding feasible paths to acceptance. Some companies only provided results for selected models. We also included results for four open-source tools that were generated by the research team. Due to processing constraints, these open-source tools were only included in the first task.

4.3.3 DESCRIPTION OF TOOLS

We describe the approaches participating companies took in solving the two tasks.

Drivers of an adverse credit decision We first describe the tools used to generate the four drivers of adverse credit decisions. The following types of model diagnostic tools were used in the analysis. Three companies used

some version of the SHAP feature importance package and based their drivers of adverse credit decisions on raw SHAP importance values. That is, these approaches select the four features with the largest positive SHAP values. Among these three companies, one company considered only continuous and globally monotonic features.

One company used the SHAP feature importance package but ranked features by the absolute sign of the SHAP feature importance (as opposed to focusing only on the features that were assigned a positive feature importance by the SHAP software package). We computed an open-source version of the SHAP feature importance package and created drivers both based on absolute and raw SHAP feature importance values. One company used the LIME feature importance package and based the drivers on absolute LIME feature importance values. We also included an open-source implementation of the LIME feature importance package and included both a version based on raw and absolute importance values. See Section 3.5 for more background on these model diagnostic tools.

Path to acceptance The second task asked participating companies to find a feasible path to acceptance for each rejected applicant within one year. Generating a feasible path to acceptance is a challenging problem to solve because we are looking for a minimal number of small changes to get an applicant over the approval boundary. There are three key challenges. First, there is a computational challenge. We not only have to consider possible combinations of features but also possible ways of perturbing these features. Second, there is the challenge of ensuring that explanations are feasible (*e.g.*, an applicant's number of total past defaults generally does not decrease unless a particular default is more than seven years old) and plausible (*e.g.* an applicant's income is unlikely to double in a 12-month window). This second challenge calls for incorporating domain-specific constraints in the search for a path to approval. Finally, there are questions about how to handle features moving together in response to changes in applicant behavior.

Broadly speaking, companies took two types of approaches to this task. The first set of approaches involved predominantly manual analyses that proceeded in two steps. The first step selects a small set of features that both meet some basic feasibility/plausibility constraints and have high feature importance values. The second step is then a trial-and-error approach to finding feature perturbations that are sufficient to bring the applicant over the approval threshold. The second approach relies on more formalized algorithmic methods that solve the optimization problem for the minimal number of smallest changes under a set of feasibility/plausibility constraints.

We now describe individual approaches in more detail. The first approach considers all continuous (non-categorical), globally monotonic features that are mutable over a 12-month horizon (according to the classification the research team provided). Among these features, the four features with the largest positive SHAP values are selected. To determine the magnitude of change, this approach runs a linear search algorithm to determine the smallest amount, in terms of percentiles, required to get the applicant accepted. This algorithm tries step-wise increments and stops the moment an applicant has a predicted default probability at or below the approval threshold. If the suggested changes exceed the maximum or minimum changes observed in the data (according to the summary statistics the research team provided), the algorithm returns no feasible changes. This approach assumes that all features should be changed by an equal amount (percentile-wise) in the absence of some stronger criteria for favoring changes in a particular feature.

A second approach starts with the features with the top-6 SHAP values and uses a trial-and-error measure to find the best set of feature changes. For continuous features, this approach tries different changes between the minimum and maximum change over the 12-month period supplied by the research team. For categorical features, changes are only considered if the transition probability to another level of the categorical feature was greater than 10%. If none of these changes bring the applicant over the approval threshold, this approach concludes that no

feasible path to acceptance exists.

A third approach relies on finding a cluster of approved applicants whose credit characteristics look the most attainable for a particular rejected applicant in light of feasibility/plausibility constraints. This approach first defines clusters among approved applicants using an unsupervised machine learning algorithm. It first finds the clusters that look closest to the rejected applicant and then diagnoses on which features the reject looks worse than the approved applicants in the cluster. Among those features, this approach selects the ones that satisfy the feasibility/plausibility constraints. In particular, any immutable features that would need to change in a direction that is not feasible are dropped from consideration. Based on the remaining features, the model prediction is computed as if the reject were given the median value of the applicants in the good cluster. Among all clusters, the one is chosen that leads to approval and/or the highest likelihood of an approval.

Two additional approaches are based on automated, algorithmic approaches to generating counterfactual explanations. One is based on a genetic algorithm (Schleich et al., 2021) and another on an algorithmic recourse approach (Verma et al., 2021). Both approaches imposed basic immutability constraints excluding features that were classified as immutable over a 12-month horizon. One approach also imposed two additional constraints (a) that the sign of the change is consistent with the constraints described by the research team and (b) that all features in a feature family (defined by baseline feature, outlier flag and missing value indicator) have to move together.

4.4 EVALUATION: FIDELITY

Our first dimension of evaluation is fidelity, that is, the ability to reliably identify features that are in fact the principal drivers of the adverse credit decision. We first describe the fidelity tests on which our analysis is based and then present results.

Fidelity: the ability to reliably identify features that can help describe how models take adverse credit decisions.

4.4.1 EVALUATION DESCRIPTION

Our evaluation of the fidelity of model diagnostic tools in the context of consumer disclosure is based on two tests. Each test evaluates how well the four drivers of an adverse credit decision provided by a model diagnostic tool identify features in the model that are in fact indicative of the adverse decision. As there is no ground truth in our experiment, our analysis is based on evaluating (a) the relative performance of tools and (b) the performance of tools relative to a benchmark based on either randomly chosen or closely correlated drivers of an adverse credit decision. The first test asks whether applicants that look similar to the rejected applicant with regard to the four identified drivers for an adverse decision are also rejected by the model – in the sense that the model predicts a default probability above our approval threshold of 10%. The second test perturbs each identified driver in a favorable direction (where possible) and records how large the resulting drop in the model's default prediction is as compared to perturbations of other features. We describe each test in detail below.

FIDELITY TESTS

We use two tests to evaluate the fidelity of model diagnostic tools in the context of consumer disclosure. The nearest neighbor test asks if other applicants in the data who look identical on the four drivers of an adverse credit decision also get assigned a high default prediction by the model. If this is not the case, we learn that other aspects of the prediction model – beyond the four drivers identified – must have been important for the adverse decision. We would conclude that fidelity is low. The perturbation test ask whether changing – or perturbing – the four features identified as drivers of the adverse decision has a greater impact on the model's prediction than (a) changing other, randomly selected, features in the model or (b) changing other features that are closely correlated with the identified driver. If this is not the case, we learn that other aspects of the prediction model are at least as important in explaining the adverse decision. We would conclude that fidelity is low.

Nearest neighbor test This first fidelity test evaluates the following hypothetical scenario: a loan applicant subject to an adverse credit decision receives four drivers for that decision, that is, four features in the model. The applicant then sets out to find his or her "neighbors" among other applicants who share the four attributes identified as drivers for the adverse decision. When we select the nearest neighbor we ensure that this neighbor has as similar feature values on the four drivers as possible. The applicant then compares himself or herself to these neighbors: did the neighbors get a different score from the underwriting model? Were they approved for a loan?

High fidelity corresponds to the neighbors also being rejected by the model. Intuitively, if only the four drivers mattered for the loan rejection, then we should not see another applicant who looks similar based on those dimensions but who also has a much lower predicted probability of default. We can compare the fidelity of different diagnostic tools by evaluating how many of the neighbors are accepted — with lower acceptance rates corresponding to higher fidelity. We provide a common benchmark to evaluate the fidelity properties by randomly choosing four features on which to select a nearest neighbor. High fidelity implies that the diagnostic tools perform better – have fewer accepted neighbors – than the benchmark that uses a set of random features.

Perturbation test The second fidelity test evaluates the following hypothetical scenario: if we change each identified driver for adverse decisions in the favorable direction, how much can we reduce the predicted probability of default? High fidelity implies that this perturbation leads to a large drop in the predicted probability of default. We can compare the fidelity of different diagnostic tools by evaluating which tools lead to larger reductions in default predictions. Another way to compare the relative performance of tools is to ask whether perturbations reduce default predictions enough to bring the applicant below the approval threshold. The higher the fraction of approved applicants, the higher the fidelity of the tool. However, we regard this as a challenging benchmark to meet for two reasons: First, many of the rejected applicants have high predicted probabilities of default, which effectively means that perturbing only four features would have to produce very large changes in predictions to produce a result that would meet the threshold for acceptance. Second, for complex models that have hundreds of features, it is challenging for only four features to have a quantitatively large impact on predictions. We also create a common benchmark to evaluate the fidelity performance of the tools by comparing the results to a 'random' benchmark. We perform the same perturbation on four randomly chosen features. Fidelity implies that we should see larger changes in probabilities when perturbing the four drivers relative to four randomly chosen features.

We use two different perturbation schemes described in detail in Appendix C. The first scheme emphasizes inducing as many changes as possible (even if this might mean an unfavorable change), while the second scheme does not perturb a feature if the change would be unfavorable. We show results for the perturbation scheme that induces as many changes as possible in the main text. Results for the second perturbation scheme are qualitatively similar.

We also test whether perturbing the drivers of an adverse decision induce larger changes than highly correlated features. To implement this test, we choose the feature that is mostly highly correlated with each driver. We then conduct the same perturbation tests with the four correlates. High fidelity corresponds to the four closely correlated features leading to lower changes in predictions relative to the four drivers. Intuitively, one challenge for diagnostic tools when faced with complex models is the ability to pick out the most important features among a set of potentially highly correlated features. This test offers us insight into the ability of diagnostic tools to differentiate among correlated features.

Finally, we test whether the drivers of an adverse decision satisfy a form of monotonicity. Intuitively, we expect the first driver to matter more than the fourth driver. To test a weak form of monotonicity, we ask whether perturbations of the first two drivers lead to larger changes in prediction than those of the third and fourth drivers.

4.4.2 FIDELITY RESULTS

KEY FINDINGS

We find substantial variation in the fidelity of diagnostic tools that provide four drivers for an adverse credit decision. One cluster of tools exhibits high fidelity in the nearest neighbor test for both simple and complex models, in the sense that these tools are able to identify features that differentiate rejected applicants from other applicants. In the perturbation test, these tools are also able to identify features that, when changed in a favorable direction, reduce predicted default probabilities by more than randomly chosen or even closely correlated features. However, our results suggest that changing these features is often not sufficient to overturn the adverse credit decision. Instead, these features should be understood in the context of their feature correlations: only moving them together with correlated features shows the full effect on credit approvals.

Our first dimension of evaluation is fidelity, that is, the ability to reliably identify features that are in fact related to the adverse credit decision. We find substantial variation in the fidelity of diagnostic tools that provide four drivers for an adverse credit decision. One cluster of tools exhibits high fidelity and work equally well for both simple and complex models. These tools all derive the four drivers by ordering features by raw Shapley values. This cluster contains three company responses and one open-source tool. A second cluster exhibits more mixed fidelity performance. This second cluster performs significantly better for simple than for complex models, but overall fidelity is lower than for the first cluster. This second cluster contains responses that derive drivers either from LIME or from responses that order features by absolute SHAP values. This cluster contains two company responses and three open-source responses. We also find variation in fidelity within both clusters – although this variation is quantitatively much smaller than across clusters. In general, we find that open-source tools do not perform significantly worse (or better) than the company responses. This finding implies that the performance of each cluster is not driven by the open-source responses. The within-cluster differences in fidelity suggest that micro-choices that govern the

way a specific tool is used matter less than the overall choice of diagnostic tool.

Our fidelity tests suggest that the best tools identify features that indeed relate to the adverse credit decision. However, our results suggest that changing these features is often not sufficient to overturn the adverse credit decision. Instead, these features should be understood in context of their feature correlations: only moving them together with correlated features shows the full effect on credit approvals. We further explore attempts to focus adverse action notices specifically on factors that are actionable by consumers in the following section.

Complex models We find substantial variation in fidelity in both nearest neighbor and perturbation tests. Starting with the complex models in Table 3, we find that responses based on the SHAP feature importance package show the highest fidelity. Among the SHAP responses, the four responses ordering feature importance by raw SHAP feature importance values perform the best.

Table 3: AAN: FIDELITY TEST – COMPLEX MODELS

	Nearest neighbor test Fraction			Perturbation test Fraction					
	Neighbors accepted	Beat random	Ranking	Change in PD	Beat random	Change to accept	Beat correlated	Monotonic	Ranking
All SHAP									
Average	0.09	1.00	3.83	-0.03	1.00	0.11	0.96	1.00	3.50
Best	0.03	1.00	1.00	-0.05	1.00	0.15	1.00	1.00	1.00
Worst	0.24	1.00	8.00	0.00	1.00	0.02	0.75	1.00	6.00
N	6								
Raw SHAP									
Average	0.04	1.00	2.50	-0.04	1.00	0.13	1.00	1.00	2.50
Best	0.03	1.00	1.00	-0.05	1.00	0.15	1.00	1.00	1.00
Worst	0.06	1.00	4.00	-0.03	1.00	0.13	1.00	1.00	4.00
N	4								
LIME									
Average	0.28	0.78	7.33	0.01	0.56	0.03	0.56	0.89	8.00
Best	0.20	1.00	6.00	0.00	0.50	0.05	0.67	1.00	7.00
Worst	0.42	0.33	9.00	0.02	0.67	0.01	0.50	0.67	9.00
N	3								
Random									
Average	0.33			0.00		0.02			

Note: The table shows results from two fidelity tests for the four complex models by type of model diagnostic tool. Raw SHAP is a subset of the All SHAP results. The first three columns show results from the nearest neighbors test. High fidelity corresponds to a low number of nearest neighbors that are accepted and a fraction of responses that beat random close to 1. The remaining columns show results for the perturbation test using the first perturbation scheme. Change in PD refers to the differences in predictions induced by the perturbation. High fidelity corresponds to more negative changes in predictions as this implies the perturbation led to a large drop in the predicted probability of default. Beat random and beat correlated refer to the fraction of models for which the drivers identified by the diagnostic tools induce a larger change in probabilities than random drivers and correlated drivers, respectively. Monotonic refers to how often the top 2 features induce larger probability changes than top 3 and 4 features. The final row provides descriptive information about the collection of random features used for the benchmark comparison.

The first set of columns show fidelity results for the nearest neighbor test. All SHAP responses perform better than the benchmark based on random features, while only 78% of LIME responses beat this benchmark. The responses based on sorting by raw SHAP values achieve high overall fidelity with 6% – or fewer – of neighbors having a predicted probability of default low enough to be accepted (recall that high fidelity implies values close to zero). For the remaining tools, the fraction of accepted neighbors ranges from 15-24%. These numbers suggest that, for a significant portion of rejects, drivers other than the four named by the response are relatively important drivers of the

adverse decision. We summarize these performance differences in a ranking that first sorts responses by whether they perform better than the benchmark and then sorts by the lowest share of accepted neighbors. A ranking of 1 corresponds to the best performance. Equal performance is assigned the same rank. The ranking results confirm the relative performance of the diagnostic tools.

The second set of columns show results from the perturbation test. SHAP responses, in particular those based on raw values, again display the highest fidelity. These tools consistently beat the benchmark that perturbs random features. They also induce more negative changes in predicted probability of default, in line with the perturbation reducing the probability of default for the applicants. These tools have the highest fraction of rejects who cross the approval threshold following the perturbation. In contrast, LIME-based responses only beat the benchmark about half of the time and often induce changes of the wrong sign, that is, the probability of default goes up and not down after the perturbation. The fraction of rejects who are accepted following the perturbation is in the low single digits. We again summarize these performance differences in a ranking that first sorts responses by whether they perform better than the benchmark and then sorts by the largest drop in predicted default probability. A ranking of 1 corresponds to the best performance. Equal performance is assigned the same rank.

Even the best diagnostic tools lead to quantitatively small changes in predictions. To give a sense of scale, the average change in prediction resulting from the perturbation is about 17% of the standard deviation of model predictions in the sample of 3000 rejects. Intuitively, the variation across predicted probabilities of default among the 3000 rejects is about 5 times larger than the change in prediction we observe as a result of the perturbation. These small magnitudes are also reflected in the majority of rejected applicants not crossing over the approval threshold. The best response leads to approval for 15% of the rejects. These small magnitudes are likely due to two reasons. First, our perturbation scheme respects the bounds of the data; for example, we do not allow perturbations that push a feature below (above) the minimum (maximum) values observed in the data. Second, our complex models include hundreds of features, which makes it challenging for only four features to lead to large changes in probabilities. Third, some of our applicants might have default predictions far from the approval threshold.

We find similar patterns in the relative performance of diagnostic tools when it comes to the ability to differentiate drivers from closely correlated features. The majority of SHAP responses do better than the set of four correlated features. That is, perturbing the four drivers reduces predicted default probabilities by more than perturbing the four most highly correlated features. Among the raw SHAP responses, all responses do better than the correlated features. For LIME-based approaches, close to 60% do better than the correlates.

Finally, we test the monotonicity properties of the provided drivers. This test asks whether the ordering of the four drivers corresponds to the size of the changes in predictions. If monotonicity holds, then the first two drivers induce larger changes on average than the third and fourth. We find that this form of monotonicity is satisfied by all SHAP responses (both raw and absolute feature ordering) and by the majority of LIME responses.

Simple Models Fidelity is more equal across diagnostic tools for the simple models. We also observe higher fidelity for simple models than for complex models but the magnitude of improvement is often small. The tools that perform best for complex models also perform well for simple models. In contrast, tools that perform less well on complex models perform well when applied to simple models. These findings suggest that diagnostic tools exist that can handle both simple and complex black-box models for the purpose of creating drivers of adverse credit decisions.

Table 4 shows results for the logit model. The first set of columns shows results for the nearest neighbor fidelity test. We find that all responses perform better than the benchmark that matches neighbors on randomly selected

Table 4: AAN: FIDELITY TEST – LOGIT MODEL

	Nearest neighbor test Fraction			Perturbation test Fraction					
	Neighbors accepted	Beat random	Ranking	Change in PD	Beat random	Change to accept	Beat correlated	Monotonic	Ranking
All SHAP									
Average	0.08	1.00	3.83	-0.05	0.83	0.08	0.92	0.83	5.67
Best	0.03	1.00	1.00	-0.17	1.00	0.27	1.00	1.00	2.00
Worst	0.17	1.00	8.00	0.08	0.00	0.01	0.50	0.00	9.00
N	6								
Raw SHAP									
Average	0.09	1.00	4.50	-0.07	1.00	0.09	1.00	0.75	5.25
Best	0.03	1.00	2.00	-0.17	1.00	0.27	1.00	1.00	2.00
Worst	0.17	1.00	8.00	-0.01	1.00	0.02	1.00	0.00	8.00
N	4								
LIME									
Average	0.18	1.00	7.33	-0.12	1.00	0.16	1.00	1.00	3.67
Best	0.16	1.00	6.00	-0.19	1.00	0.32	1.00	1.00	1.00
Worst	0.21	1.00	9.00	-0.04	1.00	0.07	1.00	1.00	7.00
N	3								
Random									
Average	0.27			0.01		0.02			

Note: The table shows results from two fidelity tests for the logit model by type of model diagnostic tool. Raw SHAP is a subset of the All SHAP results. The first three columns show results from the nearest neighbors test. High fidelity corresponds to a low number of nearest neighbors that are accepted and a fraction of responses that beat random close to 1. The remaining columns show results for the perturbation test using the first perturbation scheme. Change in PD refers to the differences in predictions induced by the perturbation. High fidelity corresponds to more negative changes in predictions as this implies the perturbation led to a large drop in the predicted probability of default. Beat random and beat correlated refer to the fraction of responses for which the drivers identified by the model diagnostic tools induce a larger change in probabilities than the random drivers and correlated drivers, respectively. Monotonic refers to how often the top 2 features induce larger probability changes than top 3 and 4 features. Ranking refers to ranking of all 9 responses by fidelity performance (1 being best and 9 worst). See text for details. The final row provides descriptive information about the collection of random features for the benchmark comparison.

features. Overall, results are more homogeneous than for complex models. SHAP-based responses still perform better than the LIME-based responses but the gap is smaller with 8% versus 18% of neighbors being accepted (relative to 9% versus 28% for complex models). Responses based on raw SHAP values now have a more mixed performance, including some that perform very well while some perform less well than the best responses using LIME. All LIME-based responses have higher fidelity than in the case of complex models.

The second set of columns in Table 4 shows results for the perturbation fidelity test. Fidelity performance is more similar for SHAP and LIME responses. LIME-based responses perform much better for the logit than for the complex models. This finding might reflect that LIME relies on local linear approximation to determine feature importance. With the exception of two SHAP responses, this statement is also true for the SHAP-based responses. However, the improvements in fidelity are much smaller for the SHAP responses given their high fidelity for complex models. These patterns also extend to the identified features' ability to produce larger changes than the correlated features, as well as for the monotonicity analysis.

The magnitudes of changes in predictions are larger for the logit model than for the complex models. This difference is to be expected given that the logit models contain only a small number of features making it easier for a single feature to have a large impact on model predictions. However, in theory, this property also makes it more challenging to beat the random benchmark since randomly chosen features are more likely to have a high impact on

model predictions than randomly chosen features in a model with hundreds of features. However, in practice we find that model diagnostic tools are able to beat the random benchmark even in the case of simple models.

Company models Table 5 shows the results of the perturbation test when applied to responses we received for the Company Models. The responses for the Company Models exhibit a fidelity performance that is weaker than the high-fidelity responses for the Baseline Models. Across all responses, we find that the drop in probabilities (and the share of applicants who move across the approval threshold) is relatively high – especially when compared to the results for complex Baseline Models. In particular, responses for one model led to acceptance for close to 35% of rejects, which exceeds the performance of all responses for the Baseline Model. We also find that the identified drivers induce higher changes in predicted probabilities relative to closely related features. We were not able to conduct this test for one company model since this company used a set of new, engineered features for which it was not feasible to find close correlates. While fidelity is generally high, there is variation among the responses for the Company Models. This variation likely reflects differences in model complexity as well as the use of different types of model diagnostic tools.

Table 5: AAN: FIDELITY TEST – COMPANY MODELS

Four drivers	Change in PD	Change to accept	Beat correlated	Monotonic
Alpha	-0.08	0.20	0.62	0.52
Beta	-0.10	0.13	n/a	0.52
Gamma	-0.06	0.11	0.80	0.78
Delta	-0.08	0.16	0.93	0.50
Epsilon	-0.10	0.38	1.00	0.66

Note: The table shows results from the perturbation fidelity test for Company Models using the first perturbation scheme. Change in PD refers to the differences in predictions induced by the perturbation. Change to accept refers to the fraction of rejected applicants who move below the approval threshold as a result of the perturbation. High fidelity corresponds to more negative changes in predictions as this implies the perturbation led to a large drop in the predicted probability of default. Beat correlated refers to the fraction of models for which the drivers identified by the model diagnostic tools induce a larger change in probabilities than the correlated drivers respectively. Monotonic refers to how often the top 2 features induce larger probability changes than the top 3 and 4 features.

4.5 EVALUATION: CONSISTENCY

The second dimension of our evaluation is consistency. We first define consistency, then describe how we evaluate consistency, and finally present results.

Consistency: We test two notions of consistency. Consistency across tools asks how often two participating companies – or open-source tools – identify the same features as drivers of rejection decisions. Consistency across models asks how often a given model diagnostic tool identifies the same features as drivers of rejection decisions across different underwriting models.

4.5.1 EVALUATION DESCRIPTION

We consider two types of consistency: (1) consistency of drivers of an adverse credit decision for the same applicant and model *across different model diagnostic tools* and (2) consistency of drivers of an adverse credit decision provided by the same tool *across different models* for individual applicants.

CONSISTENCY TESTS

We test consistency across tools by tabulating how often the same features are identified by different responses. Similarly, we test consistency across models by tabulating how often the same features are identified by the same tool across different models.

Both types of consistency are helpful to gain insights into diagnostic tools commonly used to diagnose drivers of an adverse credit decision. However, consistency either across tools or across models is not necessarily a desirable property. If the model diagnostic tools analyzed in this study all exhibit high fidelity, then consistency is likely a favorable property – we obtain similar answers regardless of the precise tool used. If, however, some tools perform worse than others, it is not clear that we would expect (or want) consistency. As an extreme example, consider the case where one tool simply randomly draws features. Clearly, we would not want consistency with this random draw. For this reason, we present results by the type of diagnostic tool to reflect the differences in fidelity we presented in the preceding section. Similarly, if we believe that different models learn similar fundamental (causal) relationships about the world and our goal is to identify the most important of those relationships for driving a credit decision, then consistency is desirable. In other words, if we are hoping to learn about relationships in the world, we would expect a good model diagnostic tool to consistently identify these facts about the world. If, however, we believe that different models learn different correlation patterns in the data and that consumer protection regulations are interested in why a *particular model* exhibits adverse impact, then it is not clear that we would want (or expect) consistency across models. If models are learning different patterns, we would prefer the model diagnostic tool to correctly identify the pattern that drive adverse credit decisions for that particular model.

We evaluate consistency across models by tabulating how often the same features are identified as drivers of an adverse decision by the same tool across different (but similar) underwriting models. In our baseline results, we only treat two responses as consistent if they identify exactly the same features. We then extend our results to considering (a) feature families – such as outlier flags and missing value indicators – as well as (b) the strength of statistical association.

Roll-up Analysis While our baseline consistency tests are conducted at the feature-level, we also repeat the consistency tests using coarser feature categories. This analysis helps us understand whether disagreement at the feature level nevertheless reflects the tools' identification of two features that capture similar information in a consumer's credit information. This approach reflects the widespread practice of providing information on Adverse Action Notices that aggregate feature-level model explanations to higher-level categories in order to give consumers information that they might understand more readily. For example, two companies could identify model behavior being driven by an applicant's average monthly balances but might pick slightly different features – such as the average change in all outstanding monthly balances and the average change in outstanding *retail* credit card balances over the same period. In our initial test, those features would be deemed inconsistent. The roll-up test helps us understand how meaningful that result is and whether our consistency results improve once feature correlations are accounted

for in more robust ways.

We manually created the following categories based on the feature description provided by the credit bureau. The first roll-up uses categories that group features based on their description in the underlying credit bureau data set; the second roll-up divides them based on the type of loan product, and the third roll-up focuses on the combination of the product type and information type. The categories used to group features based on their descriptions in first roll-up are: loan balance; change in loan balance; number of trades; credit limit; overall delinquency (no information on severity provided), mild delinquency (less than 60 days), moderate delinquency (60-90 days), severe delinquency/default (more than 90 days, all collections, foreclosures and repossessions); credit card revolver status; credit utilization; loan payments; age of account; mortgage speciality (*e.g.*, type of mortgage account); and credit card speciality (*e.g.*, type of credit card account). The second set of roll-up categories are: mortgage; HELOC; credit card; and all (referring no specific type of loan product). The third set of categories combines the first and second set of categories to generate groupings that reflect both the feature descriptions and the product for which that data point is relevant, *e.g.*, "credit card – loan balance" and "consumer loan – loan balance."

Statistical Association Analysis The statistical association analysis helps us understand how much the differences across diagnostic tools reflect features that in fact capture similar information in the model. We conduct this analysis at the feature-level (not the rolled-up categories). We use the Pearson correlation coefficient as well as a mutual information approach to capture the extent to which two features express similar information about the data. Pearson coefficients are commonly used to measure linear correlations between data. The latter approach is based on the idea that the strength of the association between two features can be measured by the extent to which knowing one feature reduces uncertainty about the behavior of the other feature. In other words, we ask whether information about one feature helps us predict the other feature. Our implementation follows Kratzer and Furer (2018).[26] The advantage of the mutual information approach is that it handles well both non-linear relationships and categorical features in the data. Both the correlation and mutual information metrics range from 0 to 100 with 0 meaning that the two features have no association and changes to the value of one feature has no effect on the other. For both metrics, 100 reflects perfect association of two features such that the two features are perfectly correlated with each other. Our baseline approach compares features in their ranked position. That is, if two tools provide four drivers of an adverse loan decision, we compute the statistical association between the two drivers listed first, then between the two drivers listed second, and so forth. In unreported results, we also run a test where we compute the statistical association between any two drivers – regardless of the position in which they are listed. We find similar results.

[26] See Ramakrishnan (2021).

4.5.2 CONSISTENCY RESULTS

KEY FINDINGS

We find the following results with regard to consistency across tools and across models. Tools that exhibit higher fidelity also have more drivers of an adverse credit decision in common. Much of the feature-level disagreement among high-fidelity tools disappears when either rolling up features into broader feature families or considering the strength of the statistical association between features. Tools that exhibit lower fidelity have almost no drivers in common with each other nor with the high-fidelity tools. All tools exhibit low to moderate consistency across models.

Consistency across tools Table 6 shows pairwise overlap across tools for each model type to show consistency in drivers identified by each company's tool when assessed against each of the other participants. The table shows feature-level results (first column) as well as the results using each of the three roll-up categories (remaining columns). The table shows the average level of consistency in identified drivers for the entire population of 3000 rejects. We order responses by type of diagnostic tool with each cell in the table representing the fraction of drivers that overlap between a pair of responses. To understand our measure of consistency, suppose that an applicant sent an identical credit application to two lenders and received two adverse credit decisions. She also receives two explanations for the decisions that contain four features each. She compares the two explanations for the adverse credit decisions by comparing how many features are included in both explanations. At most the two responses she received can have 4 features in common (or 100% overlap), which corresponds to the highest possible consistency. At the other extreme, two responses can have no features in common (0% overlap), which corresponds to zero consistency. In 6, we show consistency numbers averaged across the 3000 rejected applicants that were provided to each participating company. Consistency numbers are expressed as the fraction of features that agree which ranges from 1 (100% overlap) to 0 (0% overlap).[27]

As expected, responses for the logit model exhibit higher overlap – with many responses agreeing on 50% features or more. The fidelity performance of the diagnostic tool is a good predictor of the consistency performance. SHAP responses have high overlap regardless of whether responses used the based on raw and absolute feature importance. The LIME responses exhibit much lower overlap with only 1.1 features out of 4 agreeing across responses. Note that because we take averages across the responses for 3000 rejected applicants, our averages can be a fraction of a feature.

Agreement patterns for the Complex Baseline Models are qualitatively similar to the results for the logit model, although the magnitude of consistency declines. SHAP responses now have 1.3 out of 4 features in common compared to 2.4 for the logit model. The responses based on the raw SHAP feature importance now have higher agreement (1.8 out of 4) relative to overall group of SHAP responses, which also includes responses using the absolute SHAP feature importance. LIME responses now exhibit very little overlap, agreeing on less than 1 out of 4 features.

The roll-up based on feature descriptions increases consistency by a little less than 1 feature for both the logit and Complex Baseline Models. For example, the drivers of an adverse credit decision computed using some form of SHAP tool for the logit model now agree on average on 2.8 out of 4 drivers up from 2.4 in our baseline results. In the product-driven roll-up exercise, the agreement increases by closer to 1 feature. For example, the drivers computed

[27] See Ramakrishnan (2021).

using some form of SHAP tool now agree on average on 3.3 out of 4 drivers. This large increase reflects that this roll-up approach only has four distinct categories compared to 14 distinct categories for the first roll-up scheme. The final column shows the results from roll-up scheme that combine feature descriptions and product type using 56 distinct categories. The results are similar to the description-based roll-up scheme with consistency increasing by approximately half a feature.

4.6 STATISTICAL ASSOCIATION ANALYSIS

We now consider the strength of the statistical association between two drivers of an adverse credit decision in our consistency tests. Table 7 shows the results for the two metrics of statistical association: the Pearson correlation coefficient and the mutual information metric.

Overall, we find that the statistical association across responses exceeds the benchmark we obtain by randomly choosing pairs of features in the data. The fidelity performance of the diagnostic tool is again a good predictor of the consistency performance when considering the strength of statistical association between features. For both Simple and Complex Baseline Models, we find that the high-fidelity tools, in particular raw SHAP, suggest features that are closely associated, both in terms of the correlation coefficient and mutual information. In contrast, low-fidelity tools, in particular LIME, suggest features whose statistical association is not too different from randomly choosing features.

Consistency across models Table 8 focuses on the degree to which a particular type of tool (such as SHAP raw) identifies two or more of the same drivers as most influential in causing a particular consumer to be rejected under different pairs of models (such as logit regression vs. neural net). The results are averaged across all 3000 rejected applicants in the sample and all tools of the same type. We group results by three model diagnostic types: SHAP based on raw feature importance values, SHAP based on absolute feature importance values, and LIME. It is important to note that while this test may provide insights into how the tools work, we would not necessarily expect a high degree of consistency to the extent that we believe that different model types are learning different correlational patterns in the data. In other words, if two models give extremely different predictions, then we would expect low consistency (regardless of diagnostic tool).

Overall consistency across models is relatively low with a maximum overlap of 30% and overlap often in the single digits. The model consistency patterns are different across the three families of diagnostic tools. For SHAP based on raw feature importance, the highest consistency is between the two Complex Baseline Models (30%) followed by the logit and XGBoost models (25%), followed by the logit and neural net models (16%). There is little overlap with the simple neural network model. SHAP based on absolute values has the highest overlap between the simple neural network and logit model (21%) followed by the XGBoost–logit and XGBoost–neural net pairs (both 13%). LIME responses see the highest overlap between neural net and logit (17%) followed by neural net-simple neural network pair (15%).

Table 6: AAN: CONSISTENCY WITH ROLL-UP

(a) PANEL A: LOGIT MODEL

	N	Baseline	Roll-up 1	Roll-up 2	Roll-up 3
All	36	1.76	2.60	3.33	2.46
All SHAP	15	2.40	2.75	3.36	2.67
Raw SHAP	6	2.38	2.71	3.17	2.60
LIME	3	1.14	2.64	3.60	2.44

(b) PANEL B: COMPLEX MODELS

	N	Baseline	Roll-up 1	Roll-up 2	Roll-up 3
All	36	0.74	1.50	3.51	1.31
All SHAP	15	1.29	1.80	2.77	1.68
Raw SHAP	6	1.79	2.25	3.10	2.13
LIME	3	0.31	1.51	2.41	1.31

Note: The table shows results for the consistency test of drivers of feature-level explanations aggregated across the roll-up categories. Each number represents the number of drivers of an adverse credit decision a response pair has in common. 0 indicates that they have no drivers in common. 4 indicates that they have all 4 drivers in common. Panel A shows results for the logit model and panel B aggregates across the XGBoost and neural network models. The first column shows the baseline results (no roll-up), and the remaining columns show the three roll-up schemes discussed in the text. Roll-up scheme 1 uses feature descriptions. Roll-up scheme 2 uses product type. Roll-up scheme 3 groups features into categories that combine feature descriptions and product types. Each column shows the number of features a pair of responses has in common - averaged across all response-pairs in that group.

Table 7: AAN: CONSISTENCY WITH CORRELATION ANALYSIS

(a) PANEL A: LOGIT MODEL

Logit		Pearson coefficient			Mutual Information		
All	36	61.39	40.39	89.96	35.82	18.83	72.91
All SHAP	15	70.71	52.09	89.96	43.23	30.93	72.91
Raw SHAP	6	66.91	52.09	83.00	42.06	30.93	72.75
LIME	3	57.77	44.84	83.51	21.29	18.83	24.38
		Mean	Standard dev.		Mean	Standard dev.	
Random benchmark		29.08	33.54		17.05	23.40	

(b) PANEL B: COMPLEX MODELS

Complex		Pearson coefficient			Mutual Information		
	N	Mean	Min	Max	Mean	Min	Max
All	36	25.86	4.05	64.64	15.7	2.6	51.36
All SHAP	15	37.85	20.12	64.64	25.4	10.55	51.36
Raw SHAP	6	45.33	37.58	52.65	31.1	24.84	39.34
LIME	3	12.66	4.05	25.03	7.99	2.6	18.72
		Mean	Standard dev.		Mean	Standard dev.	
Random benchmark		17.06	19.88		9.88	18.89	

Note: The table shows results for the strength of the statistical association between the drivers of feature-level explanations provided by different responses. The two metrics are the Pearson correlation coefficient and a measure of mutual information described in the text. Both metrics range from 0-100 with zero indicating no statistical association and 100 indicating perfect association. Benchmark refers to the statistical association of 100 (44 for the simple models) randomly chosen features in the data. Panel A shows results for the logit model and panel B aggregates across the XGBoost and neural network models. We show two different ways of aggregating responses (by tool used as well as by their fidelity performance). N refers to the number of response-pairs (*e.g.*, company A and company B constitute one pair).

Table 8: AAN: CONSISTENCY ACROSS MODELS

(a) PANEL A: SHAP (RAW VALUE)

	logit	simple nn	xgb	nn
logit	1.00			
simple nn	0.07	1.00		
xgb	0.25	0.03	1.00	
nn	0.16	0.02	0.30	1.00
N	4			

(b) PANEL B: SHAP (ABS VALUE)

	logreg	simple nn	xgb	nn
logit	1.00			
simple nn	0.21	1.00		
xgb	0.13	0.08	1.00	
nn	0.09	0.03	0.13	1.00
N	2			

(c) PANEL C: LIME (ALL VALUE)

	logit	simple nn	xgb	nn
logit	1.00			
simple nn	0.00	1.00		
xgb	0.09	0.00	1.00	
nn	0.17	0.15	0.11	1.00
N	3			

Note: The table shows results from the consistency test across different model types. Each cell shows the average fraction of overlap across 4 features identified as the drivers for an adverse decision when two different models make predictions for the same set of applicants. Panel A shows results for the tools using raw SHAP values. Panel B shows results for the tools using absolute SHAP values. Panel C shows results for tools using LIME. We average across the responses. 1 indicates perfect overlap (four features in common), and 0 indicates no overlap (0 features in common). 0.25 implies that two responses have 1 feature in common; 0.5 implies that two responses have 2 features in common; and 0.95 implies that two responses have 3 features in common. N denotes the number of responses considered in that panel.

4.7 EVALUATION: USABILITY

As a third dimension we now consider usability, that is, the ability to identify drivers that provide a rejected applicant with a feasible path to acceptance within one year. There is growing interest in whether adverse action notices can or should provide *actionable* information, rather than simply providing descriptive information. We first describe how we evaluated usability and then present results.

Usability: the ability of a model diagnostic tool to provide actionable information that helps an applicant subject to an adverse credit decision satisfy the criteria for approval within one year.

4.7.1 EVALUATION DESCRIPTION

USABILITY TESTS

We use three tests to evaluate the usability of model diagnostic tools for the purpose of providing actionable explanations for a rejected consumer. The first test checks whether the proposed actions indeed lower the prediction of default enough to overturn the adverse credit decision. The second test checks how feasible to proposed changes are given what we know about typical changes in our data. Finally, we compare the proposed actions to the principal drivers of an adverse credit decision.

We evaluate usability based on the following tests. Recall that each company was asked to provide a small set of actions (preferably no more than four but not a hard limit) that a rejected applicant could take to obtain a positive credit decision in the next 12 months. We first ask how many of the suggested changes lead to an approval decision. We obtain this information by computing model prediction after "implementing" the proposed changes in the data. We can then determine whether the new prediction indeed drops below the 10% approval threshold.

We then ask how realistic the proposed changes are considering the observed changes in the data. We compute additional statistics on a typical 12-month change observed in the data by drawing on additional data on the same applicants 12 months and 24 months prior to their credit card application. Recall that this information was provided to the participating companies as part of the usability tasks. We can then compute whether the proposed changes lie in the center or the tails of the observed distribution of changes. If all proposed changes look extreme relative to the changes typically observed in the data, we conclude that the suggested path is unlikely to be feasible for a given applicant. Finally, we document how much the proposed changes differ from the drivers of the adverse credit decisions identified in the first part of this analysis.

4.7.2 USABILITY RESULTS

KEY FINDINGS

While there are tools that perform well on our fidelity tests, our usability analysis shows that these tools do not necessarily do well in deriving actionable paths to acceptance. Changing only a few features in isolation is unlikely to overcome a rejection, although the tools performed better on this task with simple models. If we want to ensure that a proposed path to acceptance leads to a large enough drop in the predicted probability of default in more complex models, we have to allow the proposed changes to be large in number or very large in magnitude (relative to what is observed in the data). These results point to a fundamental challenge of describing feasible changes by focusing on a few features in isolation, rather than in the context of their correlation and causal relationships with each other.

Table 9 shows summary statistics for the proposed changes that provide a path to loan approval. For the simple Baseline Models, most approaches suggest an average of 7–8 changes per applicant. For the complex Baseline Model, the average number of changes are much higher: 62 for the XGBoost model and 137 for the neural network.

The results in Table 9 highlight two important trade-offs when it comes to usability (*i.e.*, feasible paths to acceptance). If we want to ensure that a proposed path to acceptance leads to a large enough drop in the predicted probability of default, we have to allow the proposed changes to either suggest many actions and/or we have to allow the proposed changes to be very large (relative to what is observed in the data). We find that it is generally challenging to generate paths to acceptance with only a few feature changes that also lie within feasible bounds of change observed in the data.

Table 9 shows the average acceptance ratios after implementing the proposed changes. For the logit and simple neural network models, respectively 80% and 66% of rejected applicants are approved. These paths on average reduce the probability of default by 20–30 percentage points. However, there is significant heterogeneity across model diagnostic tools. One tool achieves an acceptance rate of 100% for both models while other tools only reach an acceptance rate of 6% and 28%, respectively. The high acceptance rates however come at the price of suggesting a large number of changes that lie outside of the bounds observed in the data – 52% and 32% of changes, respectively.

For the XGBoost and Neural Network models, we find average acceptance rates of 36% and 27%, respectively. Again, there is significant heterogeneity across tools. One tool induces acceptance rates of 98% and 73% for the two complex models. The tools with the lowest acceptance rates, in contrast, only achieve acceptance rates of 7% and 16%, respectively. High acceptance rates for the XGBoost and Neural Network models are attainable in our setting only with a large number of changes (over 200 average changes per applicant for the XGBoost and over 500 changes for the Neural Network). The results for the Company Models exhibit acceptance rates around 30% with two notable outliers – one acceptance rate close to 70% and the other close to zero.

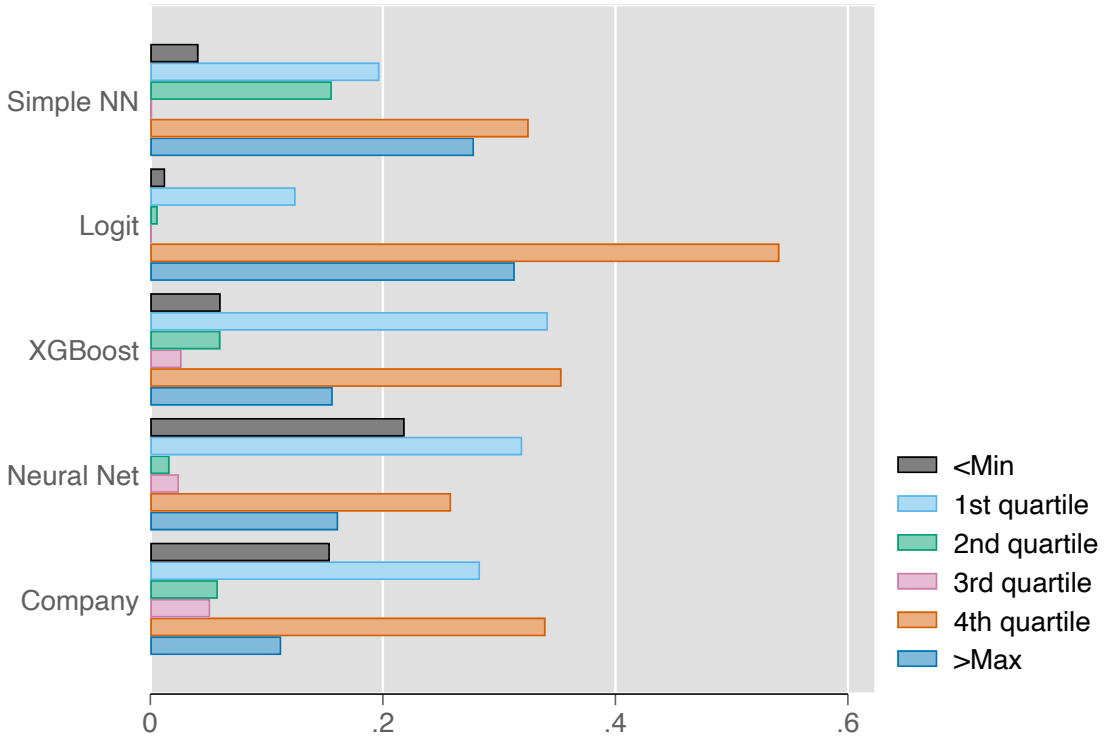
Comparing these results to the perturbation test in the preceding section, we find that the changes observed in the usability test are about 10x larger than the changes in predictions we documented in the fidelity tests in Table 3. This increase in magnitude reflects two differences. First, many proposed changes exceed their plausible historical ranges. Figure 1 shows the distribution of the proposed changes. The figure shows that there is little mass in the center of the distribution, that is, few proposed changes lie in the second or third quartile of typically observed changes. A significant number of proposed changes exceed either the maximum or minimum changes observed

Table 9: AAN: USABILITY

	Avg. # of changes per applicant	Ratio accepted	Avg. PD change	Fraction outside bounds
Logit				
Average	8	0.80	-0.24	0.27
Highest acceptance	4	1.00	-0.34	0.52
Lowest acceptance	4	0.06	-0.08	0.21
N	5			
Simple NN				
Average	7	0.66	-0.19	0.32
Highest acceptance	4	1.00	-0.27	0.32
Lowest acceptance	4	0.28	-0.07	0.11
N	4			
XGBoost				
Average	62	0.36	-0.09	0.25
Highest acceptance	235	0.98	-0.26	0.28
Lowest acceptance	4	0.07	-0.01	0.16
N	4			
Neural net				
Average	137	0.27	-0.06	0.43
Highest acceptance	538	0.73	-0.20	0.31
Lowest acceptance	4	0.16	-0.03	0.27
N	4			
Company models				
Alpha	4	0.67	-0.21	0.65
Beta	3	0.00	0.00	0.49
Gamma	4	0.33	-0.15	0.18
Delta	3	0.31	-0.17	0.01
Epsilon	3	0.27	-0.02	0.00

Note: The table shows results from the usability test for the drivers of an adverse credit decision. The first column shows the average number of changes suggested per applicant. Ratio accepted refers to the fraction of rejected applicants for whom the suggested changes lead to a reduction in default prediction large enough to bring them below the 10% approval threshold. Average PD change describes the average change in the default prediction (in probability units) that results from the change. Fraction outside bounds refers to the fraction of proposed changes that is greater (smaller) than the maximum (minimum) change observed in the data. N refers to the number of responses we received for each model type. Highest acceptance refers to the response that had the largest 'ratio accepted' while lowest acceptance refers to the response with the lowest 'ratio accepted'.

Figure 1: AAN: USABILITY



Note: The figure shows the distribution of proposed changes for the rejected applicants relative to the typical changes observed over a 12-month horizon in the data. 1st-4th quartiles show the fraction of proposed changes that fall in that respective quartile of changes observed in the data. $<$ Min and $>$ Max show the fraction that exceed the minimum/maximum change observed in the data. Each bar in the graph averages over all responses received for that model type. The company bar averages over all Company Models.

in the data. This finding is particularly pronounced for the XGBoost and Neural Network models. In contrast, the perturbation scheme we applied in the fidelity test in the preceding section always respected the bounds of the data. Second, some diagnostic tools proposed many more than the four drivers that we changed as part of the fidelity test in the preceding section. Altering a larger number of features also drives the larger changes in predictions observed in this usability test.

To gain further insight into the provided changes, we consider two dimensions of consistency. The first two columns of Table 10 show the degree of overlap between the two sets of responses that the participating companies provided for the two disclosure tasks. The first set represents the four drivers of an adverse credit decision, and the second set represents the feasible path to acceptance. In other words, we show how often a company suggests the same features for the same applicant across the two disclosure tasks. The second two columns of Table 10 then compare the overlap across the responses for the usability test only. In other words, we show how often two different companies suggest the same feasible path for the same applicant. We exclude responses that did not provide responses for both parts of the analysis from this table.

Table 10 shows that both types of consistency are low. The median overlap between the two disclosure tasks

Table 10: AAN: CONSISTENCY OF DRIVERS FOR USABILITY ANALYSIS

	Consistency tasks 1-2		Consistency across responses	
	Median overlap	Frac. of reject w/out overlap	Median overlap	Frac. of rejects w/out overlap
Logit	2	0.34	0	0.62
Simple NN	1	0.25	0	0.77
XGBoost	1	0.37	0	0.88
Neural net	1	0.36	0	0.82
N	4			

Note: The table shows overlap of drivers across the two tasks in this study for the same response (first two columns) and across responses to the second task (last two columns). Median overlap describes the number of drivers in common for a rejected applicant, with zero indicating no overlap and four indicating perfect overlap. Fraction of rejects with no applicant reports the fraction of the 3000 rejects for which two responses do not have any drivers in common. All statistics are averaged by model type.

suggests that about 35% of rejected applicants have no overlap in the drivers of an adverse decision and the path to feasible acceptance. We find that considering two potential objectives to the adverse action disclosures – identifying the primary basis for rejection and proposing feasible path to acceptance – lead to the identification of quite different features. The median overlap between two different sets of feasible paths is even lower with a median overlap of zero. A given pair of responses has on average no features in common for 60-80% of rejects. These overlap numbers are much smaller than those observed in our consistency tests in the disparate impact evaluation (as reported below in subsection 4.5.2). This finding suggests that the divergence in feasible paths to acceptance is greater than the divergence in the identification of the primary driver of an adverse credit decision.

There are three important limitations to our analysis. First, one important limitation is that we did not extend this analysis to test whether the features identified are in practice closely related – either because they belong to the same feature family or are closely correlated in the data. Second, our sample of rejected applicants was randomly drawn from the sample of rejected applicants in the training data. This approach implies that some rejects will likely be far away from a 10% approval threshold and finding a feasible path to acceptance would be challenging in practice. The upside of the random draw is that we represent the whole spectrum of rejects in our analysis. Further breaking down our results by marginal versus inframarginal rejects, that is rejects close to and further away from the approval threshold, is a fruitful avenue for future research. Third, we work with an identical approval threshold of 10% for all models. Because the models in our analysis differ in the distribution of predictions for the same set of rejects, a 10% approval threshold represents different degrees of leniency (or target default rates).

5 RESULTS: FAIR LENDING AND DISPARATE IMPACT

This section presents our evaluation of the participating model diagnostic tools with respect to fair lending and disparate impact requirements. It consists of two main sections: (1) background and (2) empirical evaluation. The former section provides an overview of relevant policy considerations, legal and regulatory expectations, and operational considerations. The latter section first describes the fairness and disparate impact properties of the credit underwriting models considered in this analysis. We then present our analysis that evaluates how well diagnostic tools identify drivers of disparities and help lenders identify less discriminatory model specifications.

5.1 BACKGROUND

5.1.1 LEGAL AND REGULATORY OVERVIEW

Lenders are subject to broad anti-discrimination requirements regardless of what type of model they use to predict an applicant's likelihood of default.[28] Two fair lending doctrines reflect these requirements: disparate treatment and disparate impact.[29] Disparate treatment focuses on whether lenders have treated applicants differently based on protected characteristics, and generally prohibit consideration of race, gender, or other protected characteristics in underwriting models. Disparate impact prohibits lenders' use of facially neutral practices that have a disproportionately negative effect on protected classes, unless those practices meet a legitimate business need that cannot reasonably be achieved through alternative means with a smaller discriminatory effect.[30] The legal analysis for disparate impact has three parts:

1. Adverse Impact: A plaintiff (such as a consumer or a regulatory agency) must make an initial showing that a particular act or practice causes a disproportionate adverse effect on a prohibited basis. This is typically analyzed by looking at whether use of particular features or other lending practices cause approval rates or pricing patterns to differ substantially by race, gender, or other protected characteristics;

2. Business Justification: In response, the creditor must then show that the practice furthers a legitimate business need, such as that the variable helps to predict the risk of default; and

3. Less Discriminatory Alternative: To prevail on a claim, the plaintiff must then demonstrate that the legitimate business need cited by the creditor can reasonably be achieved by using an alternative practice that would have less adverse impact.

A finding of disparate impact discrimination thus depends on whether the statistical disparities are driven by factors that lack a business necessity or by a showing that there is a less discriminatory alternative that does not substantially reduce predictive power. This business necessity analysis generally involves: (1) identifying the relevant factors, (2) determining whether those factors have a statistically significant relationship with creditworthiness, and

[28] The Equal Credit Opportunity Act (ECOA) prohibits discrimination in "any aspect of a credit transaction" for both consumer and commercial credit on the basis of race, color, national origin, religion, sex, marital status, age, or certain other protected characteristics, and the Fair Housing Act (FHA) prohibits discrimination on many of the same bases in connection with residential mortgage lending. See 15 U.S.C. §1691(a) (also prohibiting discrimination based on the receipt of public assistance and the good faith exercise of certain rights under federal consumer financial law); 42 U.S.C. §3605 (prohibiting discrimination on the basis of race, color, national origin, religion, sex, familial status or disability).

[29] The Supreme Court has confirmed that both doctrines are available under the Fair Housing Act, but has not yet ruled on whether disparate impact analysis applies under ECOA. See Texas Dep't of Housing & Community Affairs v. Inclusive Communities Project, Inc., 576 U.S. 519 (2015). Federal regulations, agency guidance, and lower court decisions have recognized the doctrine under ECOA for decades, in part based on legislative history. See, *e.g.*,12 C.F.R. §1002.6(a), 12 C.F.R. §1002.6(a), Supp. I, cmt. 1002.6(a)-2

[30] For a general overview of the two doctrines and the ways that they overlap, see Evans (2017).

(3) determining whether such factors have an intuitive logical relationship with creditworthiness.[31] Identification of less discriminatory alternatives can take several forms, most of which begin with scrutinizing the contribution of individual features to the relevant disparities. One option is to transform features that drive disparate impact. The widespread use of debt-to-income ratio rather than income alone is an example of such a transformation. Where such a transformation is not available, lenders may choose to omit the offending feature entirely, even if that sacrifices all of its predictive value. These approaches to identifying, measuring, and mitigating disparate impact risk implicitly require that models be sufficiently transparent to permit access to relevant information about the model's behavior.

In practice, determining whether disparities constitute an impermissible disparate impact requires thorough, careful analysis and judgment. Given that credit histories for many racial and ethnic groups – due in significant part to historical discrimination and its compounding effects[32] – reflect higher credit default rates than whites (Avtar et al., 2021; Emmons and Ricketts, 2016), developing a model that accurately predicts group-level default requires features that also differ, on average, between these groups. If analysis indicates that a given feature affects minority groups differently than others but such disparities account for the difference in average credit default rates, the lender will have to consider tradeoffs between fair lending risk and credit risk and weigh competing concerns: a desire to minimize risk of discrimination and fair lending problems and a desire to avoid giving applicants loans they are unlikely to repay and being criticized for lending practices that could be interpreted as both predatory and unsound.

5.1.2 OPERATIONAL CONSIDERATIONS

Financial institutions rely on statistical analyses to help them comply with both legal fair lending doctrines.[33] With the advent of advanced prediction tools, there has been heightened interest in how these statistical tests and analyses can be adapted to Complex Baseline Models developed by machine learning algorithms, which often have large numbers of features and capture non-monotonic and/or non-linear relationships in the data. For example, the identification and management of features that may proxy for protected class status under both disparate treatment and disparate impact theories of discrimination may require a high degree of transparency into how the models are built and how they make predictions. There are also concerns that machine learning models may effectively reverse-engineer protected class status from correlations in data, even though consideration of such status is prohibited. Thus, particularly where machine learning models rely on data from more varied sources or on more complex features, there are open questions concerning whether lenders and regulators may need new tools and face new limitations in efforts to diagnose disparate impact.[34] There is an ongoing debate in the machine learning community about how to define and measure fairness and whether awareness of protected class features can increase the

[31] See Office of the Comptroller of the Currency, Bulletin 1997-24: Credit Scoring Models: Examination Guidance (May 20, 1997); ("Developers should be able to demonstrate that such data and information are suitable for the model and that they are consistent with the theory behind the approach and with the chosen methodology.")

[32] Advocacy groups often point to historical discriminatory lending practices as the basis for these default rates for minority groups. These groups identify a form of label bias that results from the effects of minority borrowers being unfairly charged higher prices for credit, when they could obtain it at all. If, for example, because of such discrimination, minorities have paid an average extra 5% relative to similarly situated non-minority borrowers, we would expect minorities to experience higher rates of default because of the burden of that higher pricing. When data reflecting this pattern is later used to build a model, the model will likely reflect and perpetuate discrimination, because the reported defaults in the training data do not properly measure true expected default for groups unfairly charged higher prices.

[33] For an overview of processes to assess disparate impact risks related to credit underwriting, see NAACP Legal Defense and Education Fund, et al., Fair Lending Monitorship of Upstart Network's Fair Lending Model: Report of the Independent Monitor (2021); NAACP Legal Defense and Education Fund, et al., Fair Lending Monitorship of Upstart Network's Fair Lending Model: Second Report of the Independent Monitor (2021).

[34] Historically, regulators have looked at whether particular features have an "understandable relationship to an individual applicant's creditworthiness" as well as a statistical relationship to loan performance in determining whether they meet a legitimate business need. See Office of the Comptroller of the Currency, Bulletin 1997-24: Credit Scoring Models: Examination Guidance (May 20, 1997).

accuracy and fairness of machine learning underwriting models.Kleinberg et al. (2018a). However, while there are certain methods of debiasing that use protected class information in either pre-, in-, or post-processing, which are accepted for use broadly within the machine learning community and in other sectors, they may not be suitable for use in financial services.[35] Some lenders are concerned that these approaches might be cited as violations of the prohibition on disparate treatment and therefore may avoid such methods unless or until regulators address this question directly in regulation or guidance.

Diagnostic tools like those evaluated in this research can be used to support compliance with fair lending requirements in a variety of ways.[36] Some lenders may use such tools to identify the features or groups of features that make the largest contributions in the model, so that they can assess the effect of transforming or removing those features to document that they have explored the business necessity of individual features and taken pains to identify less discriminatory alternatives. This approach is a common and accepted fair lending risk management practice with incumbent logistic regression underwriting models, given the small number of variables and simple structures in those models.

Generating a list of top drivers of disparate impact may also be useful for users of machine learning models – to help developers identify those features and interrelationships identified by the algorithm that require close investigation to document their fair lending impact and justify their use. Information about the relative effect of disparities (*i.e.*, the ranking of features by disparities), when used in combination with measures of feature importance, is valuable for identifying groups of features that can be added or dropped to decrease disparate impact. Further, qualitative feature reviews may continue to be important for reasons beyond disparate impact risk management, including the need to identify whether features important to the model's prediction could expose lenders to severe reputational harm even if not technically illegal and to identify features that are tantamount to proxies under disparate treatment theories of discrimination.

Lenders can also use model diagnostic tools like those in this evaluation to produce fairer alternative models by reweighting features, using adversarial models, or other in-processing steps. These approaches can help lenders identify a set of possible models that trade off fairness and predictive performance.[37] In practice, lenders presented with these alternative model specifications must choose between different fairness and predictive performance outcomes with little explicit guidance under fair lending laws. Substantial uncertainty about how to make these choices may be a significant factor in chilling adoption of new modeling technologies or data sources.

5.2 EMPIRICAL EVALUATION: SUMMARY

This section describes results from our fair lending and disparate impact evaluation. Our analysis proceeds in two steps. In the first step, we evaluate the fairness and disparate impact properties of the Baseline and Company Models. In the second step, we evaluate the diagnostic tools used to identify the features or combination of features that drive disparities in a prediction model.

Our analysis can help answer three high-level questions: (1) how large are disparities generated in "out-of-the-box" prediction models trained on credit bureau data without knowledge of protected class status; (2) how well can we describe features that drive disparities in these models; and (3) does this information help generate alternate model specifications with lower disparities and similar predictive performance.

[35] Schwartz et al. (2022); Gill et al. (2020).
[36] The approaches used by companies participating in this study are described in greater detail in Section 5.4.
[37] Hall et al. (2021); Schmidt and Stephens (2019).

Our analysis focuses on a single protected class indicator (racial/ethnic minority) to simplify the analysis and to sidestep, for now, uncertainty about regulatory expectations regarding firms' efforts to mitigate bias in models where a proposed alteration of the model has differential effects (in size and/or direction) for different protected class groups.[38] Our evaluation methodology can easily be applied to more dis-aggregated racial/ethnic groups as well as to other protected class characteristics, and we expect the general insights to be applicable when the analysis is run to incorporate other protected class characteristics.

Fairness and Disparate Impact Properties We find the following results with regards to the fairness and disparate impact properties of the credit underwriting models in our study.

KEY FINDINGS: FAIRNESS PROPERTIES

As expected given constraints on our model development resources, all models exhibit relatively large disparities compared to industry standards associated with typical production-grade models. No single model performs best across a range of possible fairness metrics, but Complex Baseline Models consistently outperform simpler models that rely on relatively few features both in terms of fairness and predictive performance. The relative patterns of predictive performance and adverse impact are preserved when evaluating underwriting models on a held-out data set with a different applicant composition.

1. All models in this study are associated with relatively large disparities across a range of fairness metrics. These disparities are likely larger than those associated with typical production-grade models that lenders use to make credit decisions since our models represent a starting point prior to undergoing extensive fair lending testing and revision.[39] Furthermore, our results are specific to the given applicant distribution and approval threshold, and consider only unadjusted disparities that do not correct for compositional differences across groups.

2. Models differ with respect to their fairness properties across different metrics, and there is no single best model across a range of possible fairness metrics. This finding is consistent with the inherent trade-offs between different fairness metrics, as suggested by prior theoretical work on algorithmic fairness.[40]

3. More Complex Baseline Models exhibit higher predictive performance *and* smaller disparities across all metrics relative to most simple models considered in our analysis. Comparisons among the Complex Baseline Models suggest trade-offs between predictive performance and fairness properties, although the magnitude of these trade-offs is small. That is, Complex Baseline Models produced by a variety of different machine learning algorithms on the same data exhibit very similar performance for both predictive accuracy and fairness.

4. The relative patterns across prediction accuracy and adverse impact are largely preserved when evaluating the same models on a data set that has a different composition of applicants. We did not observe that simple models extrapolate to new contexts more robustly than Complex Baseline Models do, in the sense that both predictive accuracy and adverse impact can change by a similar, or even larger, amount. This finding is in tension with prevailing experience reported by some lenders and other stakeholders.

[38] Our data set did not contain protected class information. For more information on our imputation methodology, see Section 3.3 above and Appendix D.
[39] Stakeholder feedback broadly supports this characterization, although our use of a single minority classification obscures direct comparison.
[40] Kleinberg et al. (2016); Chouldechova (2017).

Model Diagnostic Tools in Fair Lending Analysis We evaluate tools for identifying drivers of disparities on three dimensions: (1) fidelity, that is, the ability to reliably identify features that relate to disparities in model predictions or decisions; (2) consistency, that is, the degree to which tools identify the same drivers; and (3) usability, that is, the ability to identify information that enables lenders to comply with anti-discrimination requirements regarding business justification and less discriminatory alternatives. Concretely, we evaluate two dimensions of usability. The first dimension asks to what extent identifying drivers of disparities helps us to create alternative models that have smaller disparities and comparable predictive performance. The second dimension asks how much drivers of disparities generalize to settings in which the model is applied to a different composition of applicants.

KEY FINDINGS: MODEL DIAGNOSTIC TOOLS

Among the model diagnostic tools we evaluated, some tools can identify features that make significant contributions to disparities in the default predictions produced by underwriting models. These tools are able to reliably identify features that are related to the model's disparities such that equalizing the distribution of these features across groups or perturbing these features in a favorable direction sizably reduces disparities on the basis of protected characteristics. These tools are also able to identify features that, when changed in a favorable direction, reduce predicted disparities by more than randomly chosen or even closely correlated features. Careful interpretation of the output of model diagnostic tools is central to their effective use in fair lending analysis – in particular, attempts to change model outputs by manipulating a small set of key features is challenging when we do not also account for interdependent or correlated attributes. While model diagnostic tools are able to identify plausible drivers of model behavior, these descriptions do not automatically translate into straightforward ways of improving model properties. Strategies that leave out identified drivers of disparities do not lead to revised models that have significantly smaller disparities in predicted defaults but often result in substantial performance deterioration. In contrast, tools that rely on some degree of automation can do better in generating a menu of models with reduced disparities in default predictions, while minimizing performance losses. Importantly, these more automated tools generalize well to never-seen-before settings that have a different applicant composition.

We find the following main results.

1. There are a set of diagnostic tools that exhibit high fidelity across both Simple and Complex Baseline Models. These tools are able to reliably identify features that are related to the model's disparities, in the sense that equalizing the distribution of these features across groups or perturbing these features in a favorable direction sizably reduce disparities across protected classes.

2. Lender choices about which diagnostic tools to use and how to deploy them can be important to achieving fair lending goals, particularly for Complex Baseline Models. The high-fidelity tools combine information about how a feature is correlated with protected class and how important the feature is for the model's prediction ('feature importance'). But there are also diagnostic tools that perform poorly on the fidelity tests. This latter group of tools either only uses information about whether a feature correlates with protected class (but do not consider feature importance) or uses an experimentation strategy based on dropping features that drive disparities (which is often called "leave-one-feature-out" analysis). The tools that perform best exhibit a substantial, but not perfect, degree of consistency with each other, often agreeing on at least 5 out of 10 drivers of disparities. The low-fidelity approaches in contrast have almost no drivers in common with each other. Unsurprisingly,

high and low fidelity tools also do not exhibit much agreement with each other when identifying drivers of disparities.

3. Careful interpretation of the output of model diagnostic tools is central to their effective use in fair lending analysis. Our two fidelity tests show that it is possible to describe the drivers of disparities in terms of a few key features *as long as we do so in the context of their feature correlations*. Our fidelity test based on feature reweighting implicitly accounts for these correlations when equalizing the distribution of a particular driver of disparities. Correspondingly, we find large reductions in disparities as a result. In contrast, our perturbation test manipulates drivers of disparities *in isolation* and exhibits smaller reductions in disparities. In addition, we find that tools that perform equally well often identify a different set of features that drives disparities. However, much of the feature-level disagreement among high-fidelity tools disappears when either rolling up features into broader feature families or considering the strength of the statistical association between features.

4. While model diagnostic tools are able to identify plausible drivers of model behavior, these descriptions do not automatically translate into straightforward ways of improving model properties. These findings emerge from our usability analysis that asks how different diagnostic tools contribute to the search for alternate model specifications with smaller disparities and, if possible, similar predictive performance. We find that automated methods that do not depend on evaluating and managing individual features outperform methods that depend on generation of a list of key drivers of disparities. More specifically, we find that the ability to explain what drives disparities does not automatically help lenders generate models that have smaller disparities in predicted default when that information is used mechanically. A strategy that leaves out drivers of disparities does not lead to revised models that have significantly smaller disparities in predicted defaults but often leads to substantial performance deterioration. In contrast, automated tools that do not require *ex ante* knowledge of what drives disparities do better in generating a menu of models with reduced disparities in default predictions, while minimizing performance losses. These automated tools differ in whether and how they use protected class information in the search for less discriminatory alternatives (LDAs), reflecting an evolving regulatory landscape.[41]

Taken together, the differences between Simple and Complex Baseline Models in our study suggest that adding complexity does not necessarily increase disparities, but can lead to improvements in both predictive power and fairness given current capabilities for identifying less discriminatory alternatives. These findings are subject to the limitations of our models, which do not reflect significant manual feature curation and improvements from iterative revision and testing as a lender's models would. More importantly, the ability to realize the promise of improving both predictive power and reducing disparities depends on a series of complex and interrelated choices that individual lenders must make about what type of machine learning model to build, how much complexity to enable in that model, and how to manage transparency and fairness considerations in model development and model monitoring.

5.3 FAIRNESS AND DISPARATE IMPACT PROPERTIES

[41] As noted above, concern about inclusion of features that can pose reputational harm even if they are not illegal may mean that lenders using automated methods to generate LDAs might still value the ability to perform qualitative feature reviews and therefore place value on the ability to identify key drivers of adverse impact. The same is true for reviews to identify proxies under disparate treatment theories.

5.3.1 DESCRIPTION OF FAIRNESS METRICS

There is a robust debate in the data science and machine learning communities about the many ways to define fairness in the context of prediction and classification models. Metrics differ by what information they incorporate, which can include model predictions, approval decisions, and default outcomes ('labels') (Verma and Rubin, 2018; Pessach and Shmueli, 2020).[42]

TYPES OF FAIRNESS METRICS

There are three types of fairness metrics considered in this study. Threshold-based metrics consider disparities in model decisions after applying a (hypothetical) approval threshold to the model's prediction. Examples include the Adverse Impact Ratio (AIR) and differences in false positive rates (FPR) and true positive rates (TPR). Non-threshold based metrics consider disparities in model predictions. Examples include statistical parity, conditional statistical parity, and the standardized mean difference (SMD). Hybrid metrics include differences in predictive performance, *e.g.* based on the AUC metric of predictive performance. This study focuses on the AIR and SMD metrics for most of the analysis.

Below we offer an overview of fairness metrics we consider in this analysis.

Threshold-based metrics Threshold-based metrics consider a (hypothetical) approval cutoff that is applied to the predictions of a model. One advantage of threshold-based metrics is that they are often very intuitive and correspond to a realistic use case. These metrics focus on relevant outcomes by considering the approval threshold used in practice. Disparities in extreme tails of the model might not matter much for observed disparities in outcomes. These metrics are thus closer to the meaning of fairness intended by disparate impact requirements.

The downside of threshold-based metrics is that they are specific to a decision threshold. If lenders change the decision threshold, the measured value of disparities also changes. For this reason, larger lenders often test multiple cutoffs using adverse impact ratio ("AIR") if the usage of the model is uncertain. We focus on a decision threshold for each model that targets a predicted default rate of 5% among approved applicants. The implied approval thresholds vary from 15-25% for the Baseline Models and from 14-20% for the Company Models.

We note that these metrics can also be sensitive to changes in applicant distribution, as well as strategic considerations related to the relevant product, business line, or loan portfolio. A model can appear to have low disparities when faced with an applicant pool that contains many minority applicants who are assigned low risk scores by the model. That same model can have high disparities when faced with an applicant pool that contains many minority applicants who are assigned high risk scores by the model and consequently rejected. This sensitivity property is not specific to threshold-based metrics, but instead applies to any unconditional metric. It can complicate the monitoring of disparate impact risks and the articulation of universally applicable standards.

We consider the following threshold-based fairness metrics.

1. **Adverse impact ratio (AIR)**. This metric represents the industry standard for lenders evaluating disparate impact in a variety of contexts including credit and hiring. It is defined as the ratio of the acceptance rate for the minority group to the acceptance rate of the majority group. AIR values closer 1 correspond to more parity.

[42] For more extended explanation of individual fairness metrics discussed in this section, see Section 5.2 and Appendix C of our Market & Data Science Context Report. FinRegLab (2021).

There is no set standard for appropriate AIR values in part due to recognition that appropriate target values will vary based on a variety of factors including protected class under consideration, data, and particular product and market conditions. However, AIR above 0.8 is a frequently used industry benchmark, although stakeholders at some firms report that 0.9 may be more common as an internal risk management benchmark.

2. **Differences in True Positive Rates ("TPR") or False Positive Rates ("FPR")**. The TPR is the fraction of defaults that are correctly predicted while the FPR refers to the fraction of non-defaults that are incorrectly predicted as defaults. Unlike the AIR, these measures also take outcome labels (here, defaults) into account, not only decisions (here, approvals). Values closer to zero correspond to more parity.

In practice, lenders may use some or all of these threshold-based metrics jointly in evaluating model fairness. Considering AIR in the context of TPR and FPR allows practitioners to determine whether greater approval rate parity (AIR) is gained at the expense of approving people who have insufficient ability to repay the loan, which is reflected in decreased TPR.

Non-threshold-based metrics Another class of metrics is based on the underlying model predictions as opposed to discrete classifications or implied decisions of the model. The three key metrics we employ are statistical or demographic parity, conditional statistical parity, and standardized mean difference.

⇒ **Statistical or demographic parity** is defined as the difference in the average predicted probabilities by protected classes. The closer to zero, the more parity.

⇒ **Conditional statistical parity** follows the same idea as statistical parity but "controls" for the impact of key features that might skew the probability distribution across protected class. For example, if one group has more bankruptcies and bankruptcy is an important feature in the model, we might want to control for the effect of bankruptcy by effectively comparing model scores across applicants with similar number of bankruptcies. We implement conditional statistical parity by first identifying the top-5 global features based on ordering features by their absolute SHAP feature importance. We then split each of these five features at the median feature value creating two bins. We compute the average model prediction in each of the 32 sub-samples defined by the top-5 feature bins. We compute the statistical parity metric in each sub-sample and finally take the weighted average across sub-samples, with the weights representing the number of applicants in that sub-sample across groups. The closer to zero, the more parity..

⇒ **Standardized mean difference ("SMD")** is a scaled version of statistical parity that is widely used by industry in fair lending compliance, as well as in other anti-discrimination contexts like employment. It is defined as the average difference in predictions between protected classes, divided by the standard deviation of the model predictions. The closer to zero, the more parity.

Hybrid metrics There are some additional metrics that combine model predictions and decisions but are not threshold-based. A key example of such a hybrid metric is AUC parity.[43]

⇒ **AUC parity** is defined as the difference in predictive performance as measured by AUC by protected class. In principle, we could define similar parity metrics with regard to other measures of model performance. Considering parity in performance metrics is important because models can fulfill some of the above fairness metrics

[43] Area under the curve ("AUC") provides an aggregate measure of performance across all possible classification thresholds. AUC can be interpreted as the probability that the model ranks a random positive example more highly than a random negative example.

and yet be differentially predictive across groups. Differences in predictive performance induce more classification mistakes – more incorrect rejections and/or incorrect acceptances – among protected classes. This inequity can drive differences in credit misallocation and approval rates across groups.[44]

Fairness metrics in this report We focus on the AIR and SMD metrics in our evaluation of particular diagnostic tools to represent one threshold-based and non-threshold based metric. We chose these metrics since they are commonly used inputs in disparate impact analysis in the United States. For consumer lenders in the United States, law, regulation, and agency guidance define processes for identifying and measuring fair lending risks. Whether a particular statistical disparity constitutes an impermissible disparate impact is a legal conclusion based on the three-part analysis outlined in Section 5.1. While this analysis begins with a statistical fairness assessment, such a conclusion depends on a judgmental, multi-stage analysis rather than the mechanical application of a statistical test. For this reason, we generally refer to disparities or adverse impacts, rather than to disparate impact, throughout this study.

5.3.2 RESULTS: FAIRNESS PROPERTIES

KEY FINDINGS: FAIRNESS PROPERTIES

We present four main results. First, as expected, the models in this study exhibit relatively large disparities across a range of fairness metrics. However, these disparities are likely larger than those associated with typical production-grade models that lenders use to make credit decisions since our models represent a starting point prior to undergoing extensive fair lending review. Second, models differ with respect to their fairness properties across different metrics, and there is no single best model across all fairness metrics considered herein. Third, more Complex Baseline Models exhibit higher predictive performance and smaller disparities across all metrics relative to most of the Simple Baseline Models considered in our analysis. Fourth, the relative patterns across prediction accuracy and adverse impact are largely preserved when evaluating the same models on a data set that has a different composition of applicants.

We evaluate the fairness properties of both the Baseline Models and six Company Models. We evaluate these properties on test data that was drawn from the same population as the data used to train the Baseline Models and the Company Models. To understand how the fairness properties of the models generalize to a context with a different composition of applicants, we also evaluate all models on a second test data set (the "deployment" data) that over-sampled credit card applicants from geographies that have a higher proportion of Black and Hispanic households.

Fairness Performance We find that all models in the study are associated with relatively large disparities across all metrics. Table 11 reports fairness metrics for both the Baseline Models and Company Models. We find that disparities are both large and relatively similar across models. The notable outliers are the simple Baseline Models that perform less well across several fairness metrics. At an approval threshold that targets a predicted default rate of 5% among approved applicants, all models have adverse impact ratios (AIR) of around 0.7 (we report the associated approval thresholds in column 1 of Table 11).

[44] Blattner and Nelson (2021).

We note that these results are not representative of values that would result from production-grade models. Neither the Baseline Models nor the Company Models were subjected to the kind of extensive fair lending compliance work that lenders would conduct prior to use in this evaluation. In Section 3.4, we assess how well the tools support identification of less discriminatory alternatives. The high disparities may also reflect persistent, historical differences in the availability and quality of credit histories for different racial groups in the U.S. and the associated underlying financial and economic disparities. Furthermore, our results are specific to the given applicant distribution and approval threshold, and only consider unadjusted disparities that do not correct for compositional differences across groups.

Figure 2 shows how the AIR metric changes as we vary the approval threshold with each line representing one model. The figure shows that AIR metrics increase to 0.8 at approval thresholds of around 30% to 40%. An AIR of 0.8 is commonly considered an acceptable target for managing disparate impact risks. However, we note that these approval thresholds are not directly comparable across models since the distribution of default predictions differ across models. For reference, an approval threshold of 40% corresponds to default rates of 10-12% in our data. SMD statistics are close to 0.6 suggesting a significant degree of disparities. SMD values of 0.5 are typically considered medium to large values of disparities with acceptable values being closer to 0.2.

Table 11: FAIRNESS METRICS

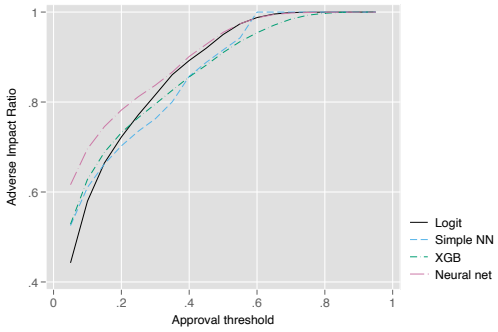
Model	Threshold	AIR	Statistical parity	Std. mean dif	Cond. stat. parity	Δ TPR	Δ FPR	Δ AUC
Ideal value		1	0	0	0	0	0	0
Direction		min/maj	min-maj	min-maj	min-maj	min-maj	min-maj	maj-min
Baseline Models								
Logit	0.15	0.66	0.10	0.57	0.02	0.09	0.19	0.06
Simple NN	0.20	0.68	0.09	0.56	0.01	0.09	0.19	0.05
XGBoost	0.20	0.73	0.12	0.57	0.02	0.08	0.16	0.05
Neural Net	0.25	0.81	0.10	0.55	0.01	0.10	0.11	0.05
Company models								
Alpha	0.19	0.72	0.12	0.56	0.02	0.08	0.17	0.05
Beta	0.20	0.73	0.12	0.57	0.02	0.08	0.16	0.05
Gamma	0.20	0.73	0.12	0.57	0.02	0.09	0.16	0.05
Delta	0.19	0.73	0.12	0.57	0.02	0.09	0.16	0.05
Epsilon	0.14	0.68	0.10	0.56	0.03	0.08	0.19	0.05
Zeta	0.18	0.70	0.11	0.58	0.01	0.07	0.18	0.05

Note: The table shows results for fairness metrics across both the Baseline Models and models built by participating companies. Please see text for the definition of each metric. All values are computed on the test data set. The ideal value row describes the value if there is perfect fairness. Direction describes the direction of the difference (or ratio) to facilitate interpretation. 'Min' denotes the minority group and 'maj' denotes the non-minority (or majority) group. All threshold-based metrics use an approval threshold that targets a 5% predicted default rate among the approved applicants for that model.

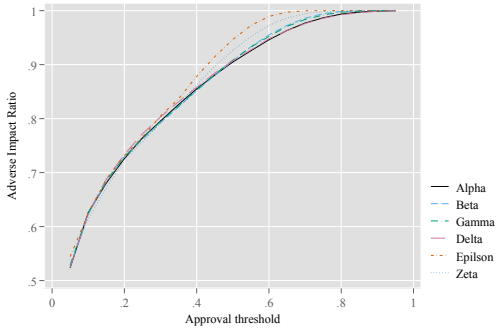
Trade-offs across fairness metrics Our second finding is that models differ with respect to their fairness properties, and there is no single best model across all fairness metrics considered herein. This finding suggests that there are indeed inherent trade-offs between different fairness metrics, as suggested by prior theoretical work in this area.[45] Figure 3 shows a graphic representation of five key fairness metrics across the four Baseline Models and the six Company Models. All metrics are standardized such that 0 corresponds to the least fair and 1 to the fairest in the set

[45] Kleinberg et al. (2016); Chouldechova (2017).

Figure 2: ADVERSE IMPACT RATIO BY THRESHOLD



(a) BASELINE MODELS



(b) COMPANY MODELS

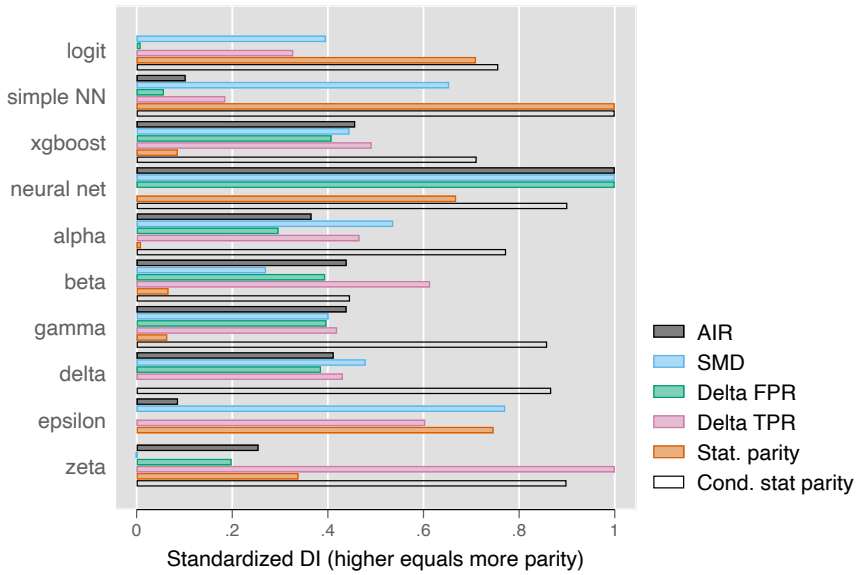
Note: The figure shows the adverse impact ratio as a function of the approval threshold. Each line corresponds to a model. All statistics are computed on the test set. Panel A shows the Baseline Models and panel B shows the Company Models.

of models. Table 12 further shows the correlation matrix across the standardized fairness metrics. Figure 3 shows that the various metrics rank the models somewhat differently, so that no one model is ranked consistently across all metrics as being the most fair.

However, the magnitude of these differences is small, suggesting that most of the models are clustered around similar magnitudes of fairness. For the simple Baseline Models, these differences are larger than for more Complex Baseline Models, indicating that Simple Baseline Models are associated with greater disparities for many of the metrics. We note that this finding may be contrary to industry experience with production-grade Simple Baseline Models. This may reflect a variety of factors, including that incumbent underwriting models reflect approaches developed and features selected over several decades and that each successive iteration had to meet fairness requirements to be put into production. As a result, the results using our models may not generalize well.

We find high correlations between the AUC and TPR and FPR statistics. This correlation may make some sense given that the AUC combines the information from the TPR and FPR metrics, albeit at a specific threshold. However, TPR and FPR reflect different aspects of the model and thus exhibit a low correlation with each other. The AIR moves closely with the FPR differences and SMD statistic.

Figure 3: STANDARDIZED FAIRNESS METRICS



Note: The figure shows different fairness metrics across both Baseline and Company Models. Company Models are anonymized. All metrics are standardized to range from 0 and 1 in the set of models. An increasing metric corresponds to more fairness, that is, more parity across groups.

Table 12: FAIRNESS METRICS: CORRELATION

	AIR	SMD	Cond. stat. parity	Δ FPR	ΔTPR	Δ AUC
AIR	1.00					
SMD	0.55	1.00				
Cond. stat. parity	0.03	-0.15	1.00			
Δ FPR	0.86	0.10	0.20	1.00		
Δ TPR	-0.12	-0.84	0.09	0.31	1.00	
Δ AUC	0.39	-0.19	0.28	0.47	0.53	1.00

Note: The table shows correlations across the standardized fairness metrics across both the Baseline Models and models built by participating companies. The standardization ensures that higher values are associated with more fairness and all metrics range from 0 to 1. Please see text for the definition of each metric. All values are computed on the test data set. All threshold-based metrics use an approval threshold that targets a 5% predicted default rate among the approved applicants for that model.

Fairness and Predictive Performance More Complex Baseline Models tend to have higher predictive performance *and* smaller disparities relative to the majority of Simple Baseline Models. Comparisons among the Complex Baseline Models suggested trade-offs between predictive performance and fairness, though the differences are relatively small. Figure 4 shows how model predictive performance and adverse impact compare across the set of models in this study. Across both AIR and SMD metrics, we find that the Simple Baseline Models, perform less well in terms of *both* adverse impact and predictive performance. However, one of the simpler models built by a participating company has comparable fairness and predictive performance to some of the more Complex Baseline Models. This

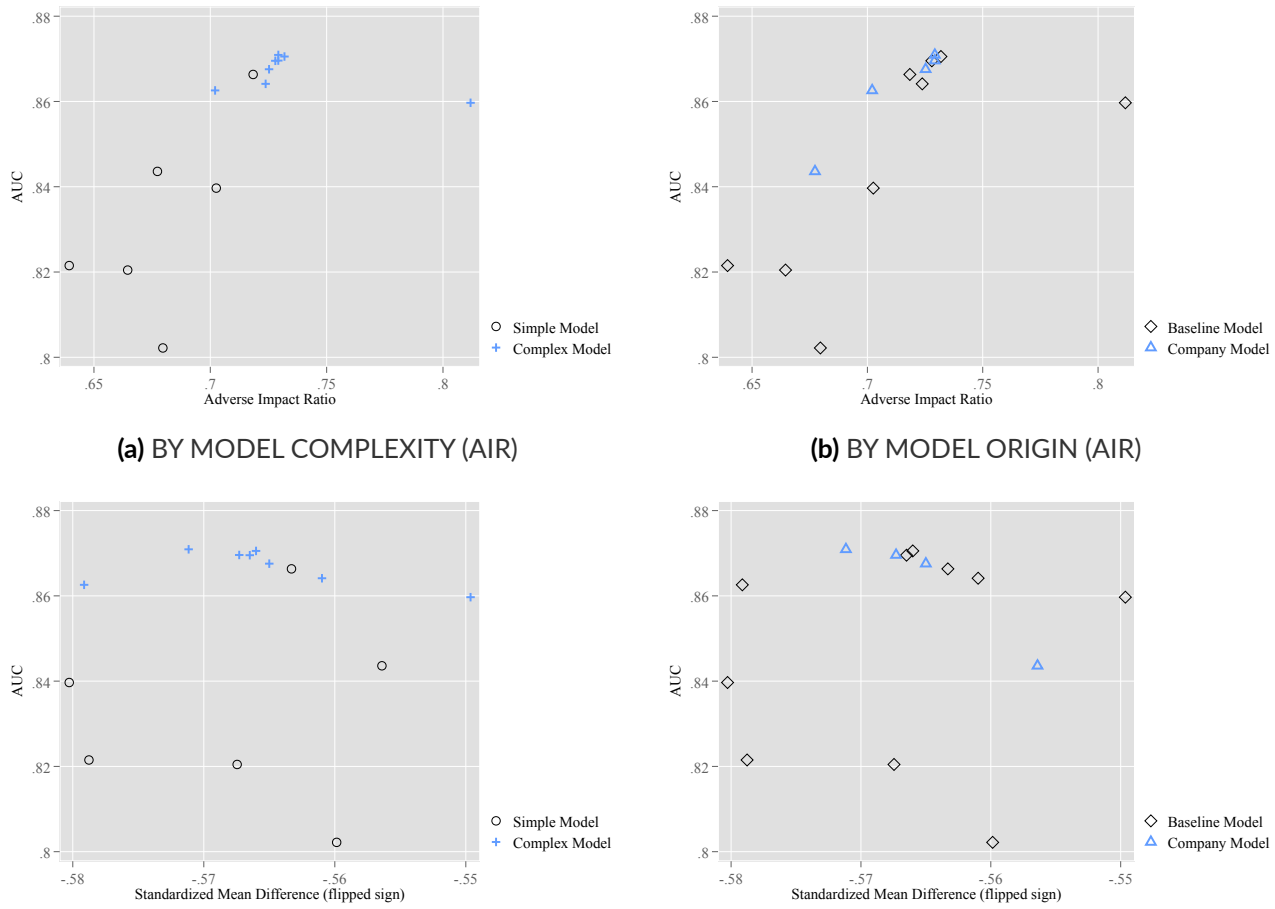
finding suggests that how simpler models are constructed matters more for fairness/predictive performance than in the case of Complex Baseline Models. The higher adverse impact of the Baseline Simple Models might reflect that models with a smaller number of features will have features that carry more weight with respect to determining the model's prediction. In other words, feature selection might have resulted in features that have high predictive power but that also exhibit significant correlation with protected class. In contrast, Complex Baseline Models distribute more weight on features that might contain similar predictive information but are less correlated with protected class. Alternatively, these differences in performance may be driven by under-fitting: by extrapolating simpler, mis-specified models with global trends across all applicants, differences between groups that differ in their features may be overstated.

In contrast, the Complex Baseline Models trace out a frontier that trades off predictive performance and adverse impact, although the magnitude of this trade-off is small. In other words, there is little difference between the models that have the strongest predictive performance and the models that have the lowest adverse impact. For example, the second row of Figure 4 shows that the model with the highest predictive power has an AUC of .87 and a SMD statistic of .57, while the model with the best adverse impact performance has an AUC of .86 and a SMD statistic of .55. Inspecting the AIR metric, we find even smaller trade-offs. In the figure, we apply the approval threshold that target a 5% predicted default rate among approved applicants. Almost all Complex Baseline Models cluster on the same performance-adverse impact point – the exception appears to be the neural network in the set of Baseline Models, which performs much better on adverse impact with little sacrifice in predictive performance. Using a more lenient approval threshold of 50%, we find more dispersion in adverse impact across the Complex Baseline Models. Interestingly, here we find very little trade-off in predictive performance.

Fairness and Performance on the Deployment Data Finally, we find that the relative patterns of performance across prediction accuracy and adverse impact were largely preserved when evaluating the same models on a data set that has a different composition of applicants. We did not observe that the Simple Baseline Models extrapolate to new contexts more robustly than Complex Baseline Models do, in the sense that both predictive power and adverse impact can change by a similar or even larger amount. This finding is in tension with prevailing experience reported by some lenders and other stakeholders. To understand how the fairness properties of the models generalize to a context with a different composition of applicants, we built a second data set (the "deployment" data) that over-sampled credit card applicants from geographies that have a higher proportion of minority applicants. We divide the deployment data into "train" data and "test" data, and we use the "train" data later in Section 5.7. Table 13 shows basic summary statistics for the deployment data set. Compared to the baseline test data, the fraction of minority applicants increases from roughly 20 to 30%, and the default rate increases for both the minority and non-minority groups. It also increases the average differences in default rate between the groups from 9% to 15%.

Figure 5 shows the adverse impact and performance metrics of the Baseline Models and Company Models on both the test and deployment test set. For this exercise, the models were not re-trained but simply evaluated on two different test data sets. We use the same model-specific AIR thresholds as in Figure 4. Both Baseline Models and Company Models exhibit higher adverse impact and lower predictive performance on the deployment test data set. However, the relative patterns of performance across prediction accuracy and adverse impact are largely preserved. The outliers are some of the Simple Baseline Models that have less adverse impact on the deployment test data than on the original test data *when considering the SMD metric*. For example, row 2 of Figure 5 shows that while predictive performance of the Baseline Simple Baseline Models declines on the deployment data set, the adverse impact performance improves. This property appears to be driven by the behavior of predictions among higher-risk

Figure 4: ADVERSE IMPACT AND MODEL PERFORMANCE



(a) BY MODEL COMPLEXITY (AIR)

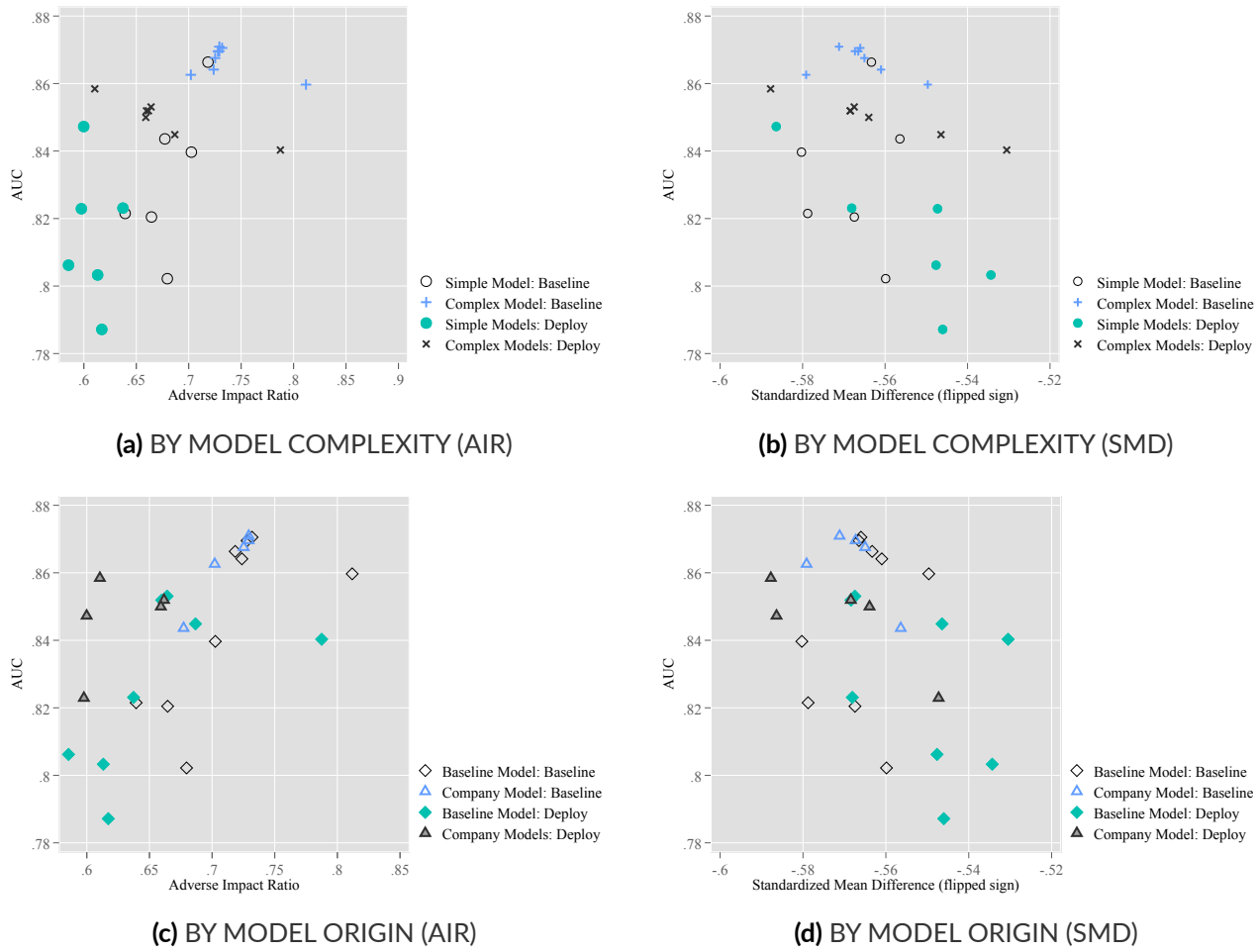
(b) BY MODEL ORIGIN (AIR)

(c) BY MODEL COMPLEXITY (SMD)

(d) BY MODEL ORIGIN (SMD)

Note: Each panel of the figure shows a scatter plot depicting the trade-off between model performance and adverse impact on the test data. Each row shows a different statistic for identifying disparities. The first row shows AIR using a model-specific approval threshold (see Table 11) and the second row shows the SMD statistic. The first column groups models by whether the model is simple or complex. The second column groups models by whether the model was built by the research team (Baseline Model) or by a participating research company. Note that higher AIR values correspond to lower adverse impact (perfect parity corresponds to an AIR equal to 1). Lower SMD values correspond to lower adverse impact. We flipped the SMD signs such that points further to the right always correspond to models with lower adverse impact across all figures. The y-axis across all panels is the ROC AUC, with 1 indicating the best possible predictive performance of a model.

Figure 5: ADVERSE IMPACT AND MODEL PERFORMANCE: DEPLOYMENT



(a) BY MODEL COMPLEXITY (AIR)

(b) BY MODEL COMPLEXITY (SMD)

(c) BY MODEL ORIGIN (AIR)

(d) BY MODEL ORIGIN (SMD)

Note: Each panel of the figure shows a scatter plot depicting the trade-off between model performance and adverse impact on the test and deployment test data. Each panel shows a different combination of adverse impact and split by model characteristic. We group models either by whether the model is simple or complex or by whether they are Baseline Models generated by the research team or models generated by the participating companies. Note that higher AIR values correspond to lower adverse impact (perfect parity corresponds to an AIR equal to 1). We show results both for a model-specific AIR threshold (see Table 11). Lower SMD values correspond to lower adverse impact. We flipped the SMD signs such that points further to the right always correspond to models with lower adverse impact across all figures. The y-axis across all panels is the ROC AUC, with 1 indicating the best possible predictive performance of a model.

Table 13: DISPARATE IMPACT: DEPLOYMENT DATA

	Baseline		Deployment	
	Train	Test	Train	Test
Fraction minority	0.22	0.20	0.28	0.28
Default rate				
Minority	0.21	0.22	0.34	0.34
Non-minority	0.11	0.13	0.19	0.19
$Delta$ default rate	0.10	0.09	0.15	0.15
N	312,715	229,722	312,715	337,737

Note: The table shows summary statistics for the baseline and deployment data set.

applicants.

Notably, we do not observe that Simple Baseline Models extrapolate to new contexts more robustly than Complex Baseline Models do, in the sense that both predictive power and adverse impact can change by a similar or even larger amounts.

5.4 PARTICIPATION DETAILS AND DIAGNOSTIC TOOLS

In the second portion of our analysis, we evaluate how well diagnostic tools identify drivers of disparities. Unlike the preceding section, our focus is now solely on the AIR and SMD metrics that are the focus of the three-step disparate impact analysis, which is at the core of legal and regulatory anti-discrimination requirements applicable to lenders in the United States.

We evaluate the diagnostic tools' outputs on three dimensions: (1) fidelity, that is, the ability to reliably identify features that are in fact related to a model's adverse impact; (2) consistency, that is, the degree to which different tools identify the same drivers for the same model and to which the tools identify the same drivers for different models; and (3) usability, that is, the ability to identify information that enables lenders to comply with the goals and purposes of consumer protection regulation. Concretely, we evaluate two dimensions of usability. The first dimension asks to what extent identifying drivers of adverse impact helps us to create alternative models that have smaller disparities but, if possible, comparable predictive performance. The second dimension asks how much drivers of adverse impact generalize to settings in which the model sees a different composition of applicants.

Our results show that there is a set of diagnostic tools that exhibit high fidelity across both simple and Complex Baseline Models. That is, these tools appear to be able to reliably identify features that are related to the model's adverse impact. These tools combine information about how a feature correlates with protected class status and how important the feature is for the model's prediction. The tools that perform best exhibit a substantial, but not perfect, degree of consistency with each other, often agreeing on at least 5 out of 10 drivers of disparities. The tools that perform poorly in contrast have almost no drivers in common. This latter group of tools either only uses information about whether a feature moves together with protected class (but does not consider feature importance); or uses an experimentation strategy based on a leave-one-feature-out analysis. Unsurprisingly, high and low fidelity tools also do not exhibit much agreement with each other in drivers of disparities.

Our findings to date suggest that the traditional nexus between being able to identify key drivers of disparities and using that information for mitigation may be less applicable to managing disparate impact risks in machine learning underwriting models. Methods that rely on more automated approaches in their search for less discriminatory

alternative models offer a notable contrast. These approaches differ in whether and how they use protected class information in the search for and construction of less discriminatory models. Complex Baseline Models in combination with tools that rely on some degree of automation can produce a menu of model specifications that efficiently trade off fairness and predictive performance because they assess a broader range of features and incorporate fairness considerations into the model's development from the start.

5.4.1 DESCRIPTION OF TASKS

Participating companies were asked to complete the following tasks with regard to fair lending and disparate impact.

1. First, each company was asked to provide the ten most important drivers of disparities for each of the Baseline Models as well as their Company Model(s) where applicable. Each company was given information on a single protected class indicator for the training data (see Section 3.3 for details how this indicator was constructed). We repeated this analysis on a second data set ("deployment data") that purposely over-sampled from regions with high proportions of households of color to test how the results generalize to an environment with a different applicant composition.

2. Second, after identifying drivers of disparities, each company was asked to provide recommendations on alternatives to the Baselines Models that would have lower adverse impact but similar predictive performance. This task approximates how many of these companies would interact with clients in helping them identify and mitigate bias in underwriting models. We refer to this ask as a search for less discriminatory alternative models. We asked for three sets of LDA recommendations: one based on analysis of the models' performance in processing the baseline training data with protected class information available for the analysis; one based on the models' performance in processing the deployment training data but withholding protected class information; and finally one based on the models' performance in processing the deployment training data but with protected class information available for the analysis.

5.4.2 PARTICIPATION DETAILS

Six out of seven companies participated in the first task by computing drivers of disparities, with several companies providing two separate responses. In addition, we built three open-source approaches to identify key drivers of disparities for purposes of the first task. Companies that built their own credit underwriting models also provided drivers of disparities for their own models. Three out of six companies provided drivers of disparities on the deployment data.

Five out of seven companies participated in the second task and provided recommendations for less discriminatory alternative models. Companies responded to this task in various ways, given operational challenges and resource limitations. For example, one company limited itself to conducting LDA analysis on only features that were identifiable given masking requirements from our data vendor, so that their personnel could review the analysis at the feature level. Since some of the masked variables excluded from their LDA analysis were important in the model, this affected what their LDA analysis could achieve. Two companies ran the LDA search in-house and provided predictions from the resulting models. For the remaining three company responses, we implemented the recommendations we received. Four out of seven companies provided recommendations for less discriminatory alternative models on the deployment data. Most responses were limited to a subset of the Baseline Models. Companies that built their

own credit underwriting models also provided LDA results for their own models. We included one open-source tool developed by the research team in the LDA analysis.

5.4.3 DESCRIPTION OF TOOLS

We offer a brief description of the approaches the participating companies took to both disparate impact and fair lending tasks.

Drivers of disparities Unlike in the case of adverse action notices, all companies (and open source tools) used different approaches to computing drivers of disparities. These approaches differ along the following dimensions: (1) whether they consider feature importance alongside information about how correlated a feature is with protected class status; (2) how feature importance is determined; (3) how correlation with protected class attributes is determined; and (4) how information about the two types of correlations is combined where applicable. The advantages or disadvantages we point out are based on commentary provided during the course of the evaluation by each participating company.

DRIVERS OF DISPARITIES

Model diagnostic tools used to compute the drivers of disparities with regards to protected class differ along the following dimensions: (1) whether they consider feature importance alongside information about how correlated a feature is with protected class status, (2) how feature importance is determined, (3) how correlation with protected class attributes is determined, and (4) how information about the two types of correlations is combined, where applicable.

We present the tools in three clusters: The first set of tools considers feature importance alongside information about how correlated a feature is with protected class. This set of tools uses SHAP to compute feature importance. The second set of tools is similar to the first but uses feature importance tools other than SHAP. The third set of tools does not consider feature importance.

Tools based on SHAP feature importance Six responses use some form of SHAP feature importance values. These approaches all combine information about feature importance and feature correlation with protected class. Intuitively, the features responsible for disparities are those that are both important for model predictions *and* are highly correlated with protected class. However, the responses differ in the details of how they combine SHAP feature importance with the information about which features are related to protected class status.

⇒ The open-source implementation first ranks features according to their correlation with our protected class indicator. In a second step, this approach then selects the ten features from this list that have the highest (absolute) SHAP feature importance. This approach is easy to implement but ignores how combinations of features might interact with protected class status.

⇒ A second approach builds a prediction model for protected class status and selects the features that have the highest (absolute) SHAP feature importance in this model. This approach then computes the SHAP feature importance values for the default prediction model and then multiplies the two SHAP feature importance values. The drivers of disparities are the features with the largest combined SHAP values.

⇒ Two additional approaches are similar to the second approach above but use the AIR metric to evaluate the effect on adverse impact. One of these additional approaches first creates feature families, grouping related features together. For example, for one feature, the missing value flag, the outlier flag, and the value itself are commonly grouped together.

⇒ Two final approaches compute the difference in average SHAP values for each feature by protected class. The drivers of disparities are identified as the features with the largest (absolute) difference in SHAP values across groups. This basic approach can be applied to both continuous predictions as well as binary predictions obtained by thresholding the continuous predictions. Both are considered in our analysis. These approaches are different from the second SHAP approach listed above since they do not first build a proxy model for protected classes.

Tools based on non-SHAP feature importance Two responses use feature importance values based on LIME and permutation importance, respectively. Both responses are open-source implementations computed by the research team. These approaches combine information about feature importance and feature correlation with protected class in a similar way to the tools above.

The open-source implementations first rank features according to their correlation with the protected class indicator. In a second step, the ten features are selected from this list that have the highest (absolute) feature importance according either to LIME or permutation importance. These approaches are easy to implement but ignore how potential interactions between combinations of features and protected class status can affect the model's predictions.

Tools not based on feature importance Three responses use approaches that do not rely on feature importance scores.

The first two responses use a leave-one-out approach. The general approach of leave-one-out is to modify or drop one feature and re-compute the adverse impact statistic (leaving the model otherwise unchanged). As discussed above in Section 5.1, this approach is widely used by lenders developing and operating logit regression underwriting models and was commonly used among early adopters of ML underwriting models, though its use may be waning. This approach identifies drivers of disparities by determining those features that produce the largest improvements in the model's fairness when left out of the analysis. The leave-one-out approach considered in this research was conducted in two different ways. The first drops all rows from the data where the feature is outside the interquartile range, that is, the feature value exceeds the 3rd quartile or falls below the first quartile.[46] The second approach replaces values outside of the interquartile range with the feature average.[47] Leave-one-out-approaches offer users simplicity and intuitiveness in attributing feature importance by removal. The disadvantage to such strategies is that they are not well suited to feature interactions and may not work well across the full distribution of applicants. For instance, on the first approach, parts of the applicant distribution are dropped altogether. The second case also makes an implicit assumption that features have an independent impact on model predictions, which means that feature interactions are not taken into account.

The third approach to identifying drivers of adverse impact uses a prediction model to determine which 10 features in the dataset are jointly most predictive of protected class membership. The particular implementation in this evaluation uses a rule-mining technique to develop the prediction model but in principle it could be implemented

[46] Note that this response excluded the missing-value features, the outlier flags, and any logarithm-transformed features.

[47] Other leave-one-out approaches are possible but beyond the scope of this analysis. This includes a commonly used approach in which the variable under study is replaced with "missing" or "0" and the model is scored on all the rows.

using any of the model types that the research team used to build the baseline prediction models. The results of this approach are specific to the data set used, not the specific default prediction model that processes the data set.

Approaches to LDA search We now describe the approaches used to find less discriminatory alternatives rather than merely to produce information about how the model behaves– the second task we asked the participating companies to complete. Some LDA strategies focus on identifying key drivers of disparities and then modifying or dropping them to reduce the disparities in the model. Others step away from explainability-driven approaches and instead rely on automated tools that search for a range of less discriminatory alternative models.

APPROACHES TO LDA SEARCH

Approaches to find less discriminatory alternative (LDA) models evaluated herein fall into three groups. The first group adopts a feature-drop strategy that re-trains the underwriting models after dropping features that were identified as key drivers of adverse impact. The second strategy reweights the training data to give less weight to observations in the disadvantaged group that default. The third group relies on more automated approaches in their search for less discriminatory alternative models.

Two company responses suggested dropping the features most related to disparities, thereby directly leveraging the diagnostic tools described in the preceding section. These approaches differ both in the number and identity of the features that were dropped. One response suggested a feature reweighting approach based on an open-source fairness tool. In particular, this response suggested new sample weights to be used in model re-training.

Three methods that rely on some degree of automation in their search for less discriminatory alternative models offer a notable contrast. These approaches differ in whether and how they use protected class information in the search for and construction of less discriminatory models, which reflects different judgments about the permissibility of protected class information in the model building process.

One approach, built by the research team, explicitly incorporates a version of the SMD statistic into the model training process. Intuitively, with the statistic built in, the machine learning algorithm now optimizes for a weighted sum of high predictive performance and low adverse impact. By varying the weight on the adverse impact in the algorithm's objective function, this 'joint optimization' approach can trace out a menu of models that have different disparate impact and performance properties. The second automated approach combines this joint optimization approach with an adversarial debiasing technique. A third automated approach incorporates automation but does not explicitly consider protected class information in the search for alternative models. Rather, this method searches over possible feature and hyperparameter combinations to identify a set of alternative models which can then – by a separate compliance team for example – be evaluated on the basis of their predictive performance and fairness properties.

For the company models, one company used an additional automated approach. This approach is a dual objective optimization approach. This algorithm is similar to adversarial debiasing, albeit with different implementation details. This debiasing routine considers two objective functions. The first function computes the AUC (or similar) metric which needs to be maximized, conditional on reaching the goal on the second objective which computes the bias metric (in this case, the AIR metric) is minimized below a target threshold. Both objectives are functions of model parameters. The resulting optimization problem is solved by an iterative mixed gradient approach.

5.5 EVALUATION: FIDELITY

Our first dimension of evaluation is fidelity, that is, the ability to reliably identify features that are in fact driving disparities in model prediction across protected classes. We first describe the fidelity tests on which our analysis is based and then present results.

Fidelity: the ability to reliably identify features that can help describe disparities in model prediction across protected class.

5.5.1 EVALUATION DESCRIPTION

Our evaluation of the fidelity of model diagnostic tools in the context of fair lending and disparate impact analysis is based on two tests. The first test asks whether equalizing the distribution of a feature that has been identified as a driver of disparities significantly reduces adverse impact in the predictions of an underwriting model. The second test perturbs all ten drivers of adverse impact in a favorable direction and records the resulting change in disparities in the model's prediction.

FIDELITY TESTS

We conduct two fidelity tests. The reweighting test asks whether equalizing the distribution of a feature that has been identified as a driver of adverse impact in fact reduces the adverse impact of the model. High fidelity corresponds to a large drop in adverse impact as a result of the reweighting. The perturbation test asks whether changing – or perturbing – the features identified as drivers of adverse impact in a favorable direction reduces disparities on the basis of protected class. High fidelity corresponds to a large drop in adverse impact as a result of the perturbation. The reweighting test implicitly takes feature correlations and interdependencies into account by over-sampling applicants based on the drivers of adverse impact. In contrast, the perturbation test manipulates features in isolation and does not consider features in the context of their correlated features or feature families.

The first test reweights features identified as key drivers of disparities. This fidelity test evaluates the following hypothetical scenario. Assume bankruptcy is a key driver of adverse impact.[48] We then create a hypothetical data set in which the distribution of bankruptcies is the same for the minority and majority groups. For example, in this new hypothetical data set we no longer observe a larger number of bankruptcies for the minority group. We then evaluate the model's disparities on this hypothetical data set. High fidelity corresponds to large improvement in the model's adverse impact on this hypothetical data set. Intuitively, if a given feature drives a substantial portion of the disparities, then equalizing the disparities in the data should reduce disparities. We provide a common benchmark to evaluate the fidelity properties by randomly choosing features as drivers of adverse impact and repeating the reweighting test. We repeat the random draw 100 times and average across the results.[49] High fidelity implies that the diagnostic tools perform better, that is induce larger improvements in adverse impact, than the benchmark that uses a set of random features.

[48] For simplicity, we excluded from the test consideration of patterns, such as periods of high utilization or an increase in balance transfers or new cards that might commonly precede bankruptcies in consumer credit histories.
[49] Due to computational limitations, the random benchmark for the Company Models consists only of a single random draw.

We implement this reweighting test by sampling more non-minority applicants until the point that we observe approximately equal distribution across drivers of disparities. We then re-run the adverse impact assessment using the modified data set to see if the disparities are reduced. To achieve this equal distribution for continuous features when reweighting the data, we divide continuous features into bins and reweight such that the proportion of applicants from each group in each bin is equal. We perform this test both for each individual driver of disparities separately and run a test that simultaneously reweights all top three drivers of disparities.

The second fidelity test is a perturbation test that changes all ten drivers of disparities in the direction that should induce a favorable relative change in default probabilities for the minority group. We re-compute adverse impact metrics after each perturbation to test how much adverse impact has improved. Perturbation-based fidelity tests are more challenging to analyze than local feature importance, which we considered in the case of adverse action notices. To induce a perturbation we need to know whether the feature increases or decreases adverse impact. A challenge for Complex Baseline Models that are not constrained to be monotonic is that the same feature can have a positive impact on default for some (minority) applicants and a negative one for others. Even if the average direction of a feature is determined to be favorable for reducing adverse impact, we would not expect to see an increase in that feature to lead to a lower default prediction for all applicants. Nevertheless, perturbation tests can provide some insights into the fidelity of identified drivers of disparities.

We use two different perturbation schemes described in detail in Appendix C. The first scheme emphasizes inducing as many changes as possible (even if this might mean an unfavorable change) while the second scheme does not perturb a feature if the change would be unfavorable. Our baseline results are for the perturbation scheme that induces as many changes as possible. Results for the second perturbation scheme are qualitatively similar.

A key difference between these two tests is their treatment of correlated or interdependent features. The reweighting test implicitly allows other features to change as well as we reweight with respect to the feature identified as a driver of disparate impact. Assume we have identified the number of bankruptcies as a driver of disparities in the model. As we over-sample applicants with many bankruptcies to equalize the distribution of bankruptcies, we are also changing other aspects of the data. For example, we might expect that applicants with many bankruptcies also have a history of default. For this reason, the reweighting test considers features in the context of their correlated features. In contrast, the perturbation test considers features in isolation and only changes the ten features that were identified as drivers of disparities. For this reason, we would expect to see large changes in model predictions – and adverse impact – resulting from the reweighting test relative to the perturbation test.

5.5.2 FIDELITY RESULTS

KEY FINDINGS

Among the model diagnostic tools we evaluated, some tools can identify features that make significant contributions to disparities in the default predictions produced by underwriting models and support fair lending risk management approaches that rely on managing individual features. These tools are able to reliably identify features that are related to the model's disparities such that equalizing the distribution of these features across groups or perturbing these features in a favorable direction sizably reduces disparities on the basis of protected characteristics. These tools are also able to identify features that, when changed in a favorable direction, reduce predicted disparities by more than randomly chosen or even closely correlated features. The careful interpretation of outputs of these tools is central to their use. We find that changing a small set of features in isolation is not sufficient to account for observed disparities. However, an approach that considers this set of features in context with correlated or interdependent features *can* express main model differences across protected class.

When considering the reweighting analysis, we find that most responses are associated with improvements in disparity metrics. The tools performed well for all model types represented in the set of Baseline and Company Models.

Table 14 shows that fidelity is high across all models – including Complex Baseline Models with hundreds of features. That is, we observe significant reductions in disparities when equalizing the distribution of the features identified as drivers of disparities. We obtain an average increase in the AIR metric by 17 percentage points for a single-feature change and an average increase by 25 percentage points for the joint change in the top-3 drivers of disparities.[50] These improvements are large given that the across model differences in AIR in Table 11 were largely in the range of 2–7 percentage points.[51] We note however an important qualification for the interpretation of these results: Since we change full distributions, this test changes individual features as well as any correlated features. A chosen "driver" of adverse impact should thus be understood as a change in applicant distribution rather than an isolated change of just one input feature or three in isolation.

The majority of responses – 80% or more – beat the benchmark that reweights randomly chosen features. Beating the random benchmark is likely more challenging for the Simple Baseline Models because we chose the random features for Simple Baseline Models from the relatively small subset of features included in those models. Intuitively, when there are few features in the model, choosing another random feature is more likely to have a substantial effect on adverse impact relative to the complex case where we have many features that each only have a marginal effect on the level of adverse impact in the model's prediction.

Table 15 shows fidelity across types of diagnostic tools. We group diagnostic tools into three groups: tools that use SHAP feature importance alongside a feature's correlation with protected class; tools that use non-SHAP feature importance alongside a feature's correlation with protected class; and finally tools that do not use any feature importance. The first group contains 5 company responses and 1 open-source tool, the second group contains two open-source tools that use LIME and permutation importance respectively, and the third group contains 3 company responses, including both the leave-one-out approaches as well as the approach using only the protected class prediction model.

[50] Recall the no disparities AIR benchmark is 1.
[51] Recall that the drivers of disparities frequently vary by response to such a degree that there is no single set of top-3 drivers.

Table 14: FIDELITY TEST - REWEIGHTING TEST

		AIR			SMD		
	# resp	Beat random	Avg. change	Random change	Beat random	Avg. change	Random change
Simple Models							
Logit							
Single driver	22	0.82	0.18	0.14	0.82	0.29	0.23
Top-3 drivers	22	0.73	0.33	0.25	0.86	0.45	0.37
Simple NN							
Single driver	20	0.70	0.16	0.14	0.65	0.27	0.24
Top-3 drivers	20	0.75	0.27	0.25	0.65	0.40	0.38
Complex Models							
XGBoost							
Single driver	22	0.77	0.12	0.04	0.77	0.24	0.10
Top-3 drivers	22	0.77	0.24	0.10	0.77	0.38	0.22
Neural net							
Single driver	20	0.90	0.14	0.05	0.90	0.25	0.11
Top-3 drivers	19	0.84	0.23	0.12	0.84	0.38	0.24
Company models							
Single driver	4	1.00	0.16	0.05	1.00	0.27	0.11
Top-3 drivers	4	1.00	0.32	0.13	1.00	0.43	0.23

Note: The table shows results for the fidelity tests based on feature reweighting by model type. We equalize the distribution of one driver of DI at a time ("Single driver") or the top-3 drivers at a time ("Top-3 drivers"). We then re-compute the adverse impact metric (either AIR or SMD). In both cases, we compute a random benchmark that reweights 1 (or 3) randomly chosen features in the model – we repeat this 100 times and average over the outcomes. # responses indicates how many company and open source responses we received for that diagnostic tool. Beat random refers to the fraction of responses that achieve larger reductions in adverse impact than the random benchmark. Average and random change show the average (random) change in the DI metric achieved by reweighting. All AIR statistics are calculated with a threshold of 0.1.

We observe high fidelity among both the SHAP and non-SHAP tools but lower fidelity for the third group of tools that do not use feature importance to identify drivers of adverse impact. The first two sets of tools perform better than the random benchmark about 90% of the time while the third set of tools beats the benchmark only half of the time. We do observe heterogeneity within the first two sets of tools, with the best tool consistently performing better than the random benchmark but the worst tool beating the random benchmark less than 50% of the time. Note that there are high-performing tools both in the set of tools that use SHAP importance as well as the set that uses other feature importance packages. This finding suggests that the exact feature importance package used is less important than incorporating this information in the diagnostic tool in the first place.

Perturbation test Table 16 and Table 17 show results for the perturbation-based fidelity test. For the SMD statistic, we find that about 70% of responses lead to a drop in the SMD statistic relative to the benchmark, that is, the perturbation successfully reduces the disparities in the model predictions. Between 50% and 80% of responses (depending on the model) perform better than the random benchmark, which perturbs 10 randomly chosen features in the model. Between 60% and 77% of responses induce bigger drops in the SMD statistic relative to perturbing 10 features that are closely correlated with the 10 drivers of disparities indicated by the response. Results for the AIR

Table 15: FIDELITY BY DIAGNOSTIC TOOLS - REWEIGHTING TEST

	Single driver				Top-3 drivers			
	AIR		SMD		AIR		SMD	
	Beat random	Avg. change	Beat random	Avg. change	Beat random	Avg. change	Beat random	Avg. change
SHAP feat. importance								
Average	0.98	0.17	0.96	0.28	0.83	0.28	0.88	0.42
Best	1.00	0.21	1.00	0.33	1.00	0.33	1.00	0.48
Worst	0.88	0.14	0.75	0.23	0.25	0.17	0.50	0.33
N	6							
Non-SHAP feat. importance								
Average	0.88	0.17	0.88	0.29	0.94	0.30	0.88	0.43
Best	1.00	0.18	1.00	0.31	1.00	0.31	1.00	0.46
Worst	0.75	0.16	0.75	0.28	0.88	0.29	0.75	0.41
N	2							
No feat. importance								
Average	0.19	0.06	0.19	0.12	0.38	0.18	0.46	0.32
Best	0.38	0.09	0.38	0.17	0.88	0.29	0.88	0.42
Worst	0.00	0.03	0.00	0.07	0.00	0.09	0.00	0.21
N	3							

Note: The table shows results for the fidelity tests based on feature reweighting by type of diagnostic tools. We consider three clusters: responses that use (1) feature importance based on SHAP, (2) feature importance on tools other than SHAP (namely, LIME and permutation importance) and (3) tools that do not incorporate feature importance scores. N indicates how many company and open source responses we received for a particular model. The reweighting test equalizes the distribution of one driver of DI at a time ("Single driver") or the top-3 drivers at a time ("Top-3 drivers"). We then re-compute the adverse impact metric (either AIR or SMD). In both cases, we compute a random benchmark that reweights 1 (or 3) randomly chosen features in the model – we repeat this 100 times and average over the outcomes. Beat random refers to the fraction of responses that achieve larger reductions in adverse impact than the random benchmark. Average change shows the average change in the DI metric achieved by reweighting. All AIR statistics are calculated with a threshold of 0.1.

statistic (at a 10% threshold) exhibit lower fidelity patterns with the percentages of responses that reduce disparity levels, beat a benchmark of random features, and beat a benchmark of correlated features.

The relative performance by type of diagnostic tool (see Table 17) is similar to the results based on the first fidelity test. Model diagnostic tools that incorporate information about feature importance generally perform better than the tools that do not. We find that model diagnostic tools for the Company Models generally perform very well and often better than the same approaches applied to the Baseline Models.

Table 16: FIDELITY - PERTURBATION TEST

SMD	Perturbation	Beat benchmark	Original value	Beat random	Random	Beat correlated	Correlated
Logit	0.05	0.73	0.57	0.64	0.42	0.77	0.52
Simple NN	0.18	0.70	0.57	0.50	0.04	0.60	0.40
XGBoost	0.51	0.68	0.57	0.73	0.57	0.59	0.56
Neural net	0.48	0.75	0.56	0.80	0.57	0.70	0.57

AIR (0.1)	Perturbation	Beat benchmark	Original value	Beat random	Random	Beat correlated	Correlated
Logit	0.84	0.63	0.57	0.58	0.54	0.71	0.60
Simple NN	0.91	0.59	0.60	0.55	0.58	0.64	0.92
XGBoost	0.51	0.04	0.63	0.42	0.53	0.38	0.53
Neural net	0.60	0.14	0.67	0.50	0.57	0.64	0.58

SMD	Perturbation	Beat benchmark	Original value	Beat random	Random	Beat correlated	Correlated
Alpha	0.47	1.00	0.56	1.00	0.51	1.00	0.55
Beta	0.51	1.00	0.57	1.00	0.57	1.00	0.56
Gamma	0.68	0.00	0.57	0.00	0.57	0.00	0.55
Delta	0.53	1.00	0.57	1.00	0.58	1.00	0.59
Epsilon	0.08	1.00	0.56	1.00	0.58	1.00	0.56

AIR (0.1)	Perturbation	Beat benchmark	Original value	Beat random	Random	Beat correlated	Correlated
Alpha	0.53	0.00	0.54	0.00	0.57	0.00	0.55
Beta	0.56	1.00	0.54	1.00	0.54	1.00	0.54
Gamma	0.37	0.00	0.54	0.00	0.52	0.00	0.54
Delta	0.49	0.00	0.54	0.00	0.54	0.00	0.52
Epsilon	0.37	0.00	0.54	0.00	1.21	0.00	0.54

Note: The table shows results for the fidelity tests based on feature perturbation by model type. The perturbation test changes each driver of adverse impact in a favorable direction for the minority group. We then re-compute the adverse impact metric (either AIR or SMD). Beat benchmark refers to the percent of responses that successfully reduce the disparities in the model predictions. Beat random and beat correlated refer to the fraction of responses that achieve larger reductions in adverse impacts than perturbing groups of random features or features that are closely correlated to the features identified in the responses. Alpha, Beta, Delta, Epsilon, and Gamma are the company models.

Deployment data We now ask how well drivers of disparities generalize to a new data set with a different applicant composition. We repeat the fidelity reweighting test on the new data set. Each company was also given the opportunity to identify a set of new drivers after observing the training sample drawn from the deployment data. We answer two key questions: (1) do the existing drivers of disparities perform significantly worse in terms of fidelity on the new data set and (2) do drivers computed for the deployment data set better represent the true drivers.

Table 17: FIDELITY BY DIAGNOSTIC TOOLS - PERTURBATION TEST

	SMD			AIR		
	Beat benchmark	Beat random	Beat correlated	Beat benchmark	Beat random	Beat correlated
SHAP feat. importance						
Average	0.86	0.80	0.82	0.41	0.55	0.64
Best	1.00	1.00	1.00	1.00	1.00	1.00
Worst	0.00	0.00	0.00	0.00	0.00	0.00
Non-SHAP feat. importance						
Average	0.88	0.81	0.81	0.38	0.63	0.63
Best	1.00	1.00	1.00	1.00	1.00	1.00
Worst	0.00	0.00	0.00	0.00	0.00	0.00
No feat. importance						
Average	0.46	0.42	0.42	0.17	0.33	0.42
Best	1.00	1.00	1.00	1.00	1.00	1.00
Worst	0.00	0.00	0.00	0.00	0.00	0.00

Note: The table shows results for the fidelity tests based on feature perturbation by type of diagnostic tools. We consider three clusters: responses that use (1) feature importance based on SHAP, (2) feature importance on tools other than SHAP (namely, LIME and permutation importance) and (3) tools that do not incorporate feature importance scores. The perturbation test changes each driver of adverse impact in a favorable direction for the minority group. We then re-compute the adverse impact metric (either AIR or SMD). In both cases, we compute a random benchmark that perturbs 10 randomly chosen features in the model – we repeat this 100 times and average over the outcomes. Beat random refers to the fraction of responses that achieve larger reductions in adverse impact than the random benchmark. Average change shows the average change in the DI metric achieved by perturbing the drivers. All AIR statistics are calculated with a threshold of 0.1.

Deployment Data: a data set that was purposely designed to represent a different composition of applicants. The deployment data allows us to test how characteristics of model diagnostic tools generalize to never-seen-before settings, mimicking how an underwriting model might encounter a changed environment once deployed.

Table 18 show the results from repeating the reweighting test but using the deployment test data. We find that drivers of disparities on the baseline data continue to perform well relative to the benchmark of randomly choosing features. The fidelity performance on the deployment data is comparable to that on the baseline data shown in Table 14.

Table 18: DISPARATE IMPACT: REWEIGHTING TEST ON DEPLOYMENT

		AIR			SMD		
	# resp	Beat random	Avg. change	Random change	Beat random	Avg. change	Random change
Simple Models							
Logit							
Single driver	11	0.82	0.15	0.09	0.82	0.34	0.27
Top-3 drivers	11	0.73	0.27	0.21	0.73	0.46	0.41
Simple NN							
Single driver	10	0.80	0.13	0.08	0.75	0.31	0.26
Top-3 drivers	10	0.70	0.23	0.21	0.75	0.43	0.41
Complex Models							
XGBoost							
Single driver	11	0.68	0.05	0.01	0.77	0.25	0.12
Top-3 drivers	11	0.77	0.17	0.08	0.77	0.38	0.24
Neural net							
Single driver	10	0.65	0.10	0.12	0.90	0.28	0.16
Top-3 drivers	10	0.60	0.19	0.20	0.85	0.40	0.29
Company models							
Single driver	5	1.00	0.19	0.09	1.00	0.28	0.15
Top-3 drivers	5	1.00	0.34	0.15	1.00	0.46	0.26

Note: The table shows results for the fidelity tests based on feature reweighting on the deployment data. We equalize the distribution of one driver of DI at a time ("Single driver") or the top-3 drivers at a time ("Top-3 drivers"). We then re-compute the adverse impact metric (either AIR or SMD). In both cases, we compute a random benchmark that reweights 1 (or 3) randomly chosen features in the model – we repeat this 100 times and average over the outcomes. # responses indicates how many company and open source responses we received for the model. Beat random refers to the fraction of responses that achieve larger reductions in adverse impact than the random benchmark. Average and random change show the average (random) change in the adverse metric achieved by reweighting. All AIR statistics are calculated with a threshold of 0.1.

5.6 EVALUATION: CONSISTENCY

The second dimension of evaluation is consistency. We consider two types of consistency: (1) consistency of the drivers of disparities for the same model *across different model diagnostic tools* and (2) consistency of drivers of disparities provided by the same tool *across different models*. We first describe the consistency tests on which our analysis is based and then present results.

Consistency: Consistency across tools refers to how often two participating companies – or open-source tools – identify the same features as drivers of adverse impact. Consistency across models refers to how often a given model diagnostic tool identifies the same features as drivers of adverse impact across different underwriting models.

5.6.1 EVALUATION DESCRIPTION

We evaluate consistency across tools by tabulating how often the same features are identified as drivers of disparities by different responses. In our baseline results, we only treat two responses as consistent if they identify exactly the same features. We also extend our results to consider (a) feature families – such as outlier flags and missing value indicators – as well as (b) the extent to which different features display high statistical association.

CONSISTENCY TESTS

We evaluate consistency across tools by tabulating how often the same features are identified as drivers of disparities by different responses. We evaluate consistency across models by tabulating how often the same features are identified as drivers of disparities by the same tool across different (but similar) underwriting models.

Both types of consistency are helpful to gain insights into diagnostic tools commonly used to diagnose sources of model disparities. However, consistency either across tools or across model is not necessarily a desirable property. If the model diagnostic tools analyzed in this study all exhibit high fidelity, then consistency is likely a favorable property – we obtain similar answers regardless of the precise tool used. If, however, some tools perform worse than others, it is not clear that we would expect (or want) consistency. As an extreme example, consider the case where one tool simply randomly draws features. Clearly, we would not want consistency with this random draw. For this reason, we present results by the type of diagnostic tool to reflect the differences in fidelity we presented in the preceding section. Similarly, if we believe that different models learn similar fundamental (causal) relationships about the world and our goal is to identify the most important of those relationships for disparities, then consistency is desirable. In other words, if we are hoping to learn about relationships in the world, we would expect a good model diagnostic tool to consistently identify these facts about the world. If, however, we believe that different models learn different correlation patterns in the data and that consumer protection regulations are interested in why a *particular model* exhibits adverse impact, then it is not clear that we would want (or expect) consistency across models. If models are learning different patterns, we would prefer the model diagnostic tool to correctly identify the pattern that drive disparities for that particular model.

We evaluate consistency across models by tabulating how often the same features are identified as drivers of disparities by the same tool across different (but similar) underwriting models. In our baseline results, we only treat two responses as consistent if they identify exactly the same features. We then extend our results to considering (a) feature families – such as outlier flags and missing value indicators – as well as (b) highly correlated features.

Roll-up Analysis While our baseline consistency tests are conducted at the feature-level, we also repeat the consistency tests using coarser feature categories. This analysis helps us understand whether disagreement at the feature level nevertheless reflects the tools' identification of two features that capture similar information in a consumer's credit information. For example, two companies could identify model behavior being driven by an applicant's average monthly balances but might pick slightly different features – such as the average change in all outstanding monthly balances and the average change in outstanding *retail* credit card balances over the same period. In our initial test, those features would be deemed inconsistent. The roll-up help us understand how meaningful that result is and whether our consistency results improve once feature correlations are accounted for in more robust ways.

We manually created the following categories based on the feature description provided by the credit bureau. The first roll-up uses categories that group features based on their description in the underlying credit bureau data set;

the second roll-up divides them based on the type of loan product, and the third roll-up focuses on the combination of the product type and information type. More specifically,

⇒ In the first roll-up, the categories used to group features are loan balance, change in loan balance, number of trades, credit limit, overall delinquency (no in- formation on severity provided), mild delinquency (less than 60 days), moderate delinquency (60-90 days), severe delinquency/default (more than 90 days, all collections, foreclosures and repossessions), credit card revolver status, credit utilization; loan payments, age of account; mortgage speciality (*e.g.*, type of mortgage account), and credit card speciality (*e.g.*, type of credit card account).

⇒ In the second roll-up, the categories used to group features are mortgage, HELOC, credit card; and all (referring no specific type of loan product).

⇒ In the third roll up, the first and second set of categories are combined to generate groupings that reflect both the feature descriptions and the product for which that data point is relevant, *e.g.*, "credit card – loan balance" and "consumer loan – loan balance."

Statistical Association Analysis The statistical association analysis helps us understand how much the differences across diagnostic tools reflect features that in fact capture similar information in the model. We conduct this analysis at the feature-level (not the rolled-up categories). We use the Pearson correlation coefficient as well as a mutual information approach to capture the extent to which two features express similar information about the data. Pearson coefficients are commonly used to measure linear correlations between data. The latter approach is based on the idea that the strength of the association between two features can be measured by the extent to which knowing one feature reduces uncertainty about the behavior of the other feature. In other words, we ask whether information about one feature help us predict the other feature. Our implementation follows Kratzer and Furer (2018).[52] The advantage of the mutual information approach is that it handles well both non-linear relationships and categorical features in the data. Both the correlation and mutual information metrics range from 0 to 100 with 0 meaning that the two features have no association and changes to the value of one feature has no effect on the other. For both metrics, 100 reflects perfect association of two features, in other words, the two features are perfectly correlated with each other. Our baseline approach compares features in their ranked position. That is, if two tools provide four drivers of an adverse loan decision, we compute the statistical association between the two drivers listed first, then between the two drivers listed second, and so forth. In unreported results, we also run a test where we compute the statistical association between any two drivers – regardless of the position in which they are listed. We find similar results.

[52] Ramakrishnan (2021).

5.6.2 CONSISTENCY: RESULTS

KEY FINDINGS

We find the following results with regard to consistency across tools and consistency across models. Tools that exhibit higher fidelity also have more drivers of disparities in common. Much of the feature-level disagreement among high-fidelity tools disappears when either rolling up features into broader feature families or considering the strength of the statistical association between features. Tools that exhibit lower fidelity have almost no drivers in common with each other nor with the high-fidelity tools. All tools exhibit low to moderate consistency across models.

Consistency across tools

Roll-up analysis Table 19 shows the baseline consistency results as well as the results using each of the three roll-up categories. The first column shows the feature-level agreement across pairs of responses. We find that consistency is high among tools that exhibit high fidelity. Among high fidelity tools, the responses identified 8 out of 10 drivers in common on average for the Simple Baseline Models and almost 6 out of 10 drivers in common for the Complex Baseline Models. In contrast, low fidelity responses have almost no drivers in common. Unsurprisingly, high-fidelity and low-fidelity tools have almost no drivers in common. SHAP and non-SHAP tools have on average 4 drivers in common within as well as across groups. This result highlights that what matters for consistency across two tools is how well they perform on fidelity – not if they use the same feature importance package. In contrast, the group of responses that does not use feature importance have few drivers in common with each other nor with the other two groups of diagnostic tools.

The next three columns in Table 19 show how consistency changes across the three roll-up categories. The roll-up using feature descriptions increases consistency by between 1-2 features for both Simple and Complex Baseline Models. For example, the drivers of disparity computed using some form of SHAP tool for the Simple Baseline Models now agree on average on 6.7 out of 10 drivers – up from 5.5 in our baseline results. In the product-driven roll-up exercise, the agreement increases by about 3 features. For example, the drivers computed using some form of SHAP tool now agree on average on 8.9 out of 10 drivers. This large increase reflects that this roll-up approach only has four distinct categories compared to 14 distinct categories for the first roll-up scheme. The final panel shows the third roll-up scheme – the one that combines product type and feature description of the two categories (56 distinct categories). The results are similar to the first roll-up scheme with consistency increasing by 1-2 features on average.

On average, there is more agreement across responses for the Simple Baseline Model than for the Complex Baseline Models. However, the fidelity of the diagnostic tools is more important in driving agreement than the level of model complexity. Low-fidelity tools exhibit low agreement – even after feature roll-up – for both Simple and Complex Baseline Models. On average, their agreement never exceeds 2-3 features even with the most generous roll-up. In contrast, high fidelity tools exhibit high agreement for all model types: High-fidelity tools agree on almost 8 out of 10 features for the Simple Baseline Model and agree on 6 out of 10 features of the Complex Baseline Models. The increase in agreement after the roll-up for high-fidelity tools is similar across model types. With both model types, some disagreement on drivers of disparity remains even after accounting for the roll-up.

Statistical Association Analysis We now consider the strength of the statistical association between two drivers of disparity in our consistency tests. Table 20 shows the results for the two metrics of the statistical association: the Pearson correlation coefficient and the mutual information metric.

Our results confirm the patterns observed for the roll-up analysis. Overall, we find that the statistical association across responses exceeds the benchmark we obtain by randomly choosing pairs of features in the data – with the exception of the responses that use LIME for the Complex Baseline Models. The fidelity performance of the diagnostic tool is again a good predictor of the consistency performance when considering the strength of statistical association between features. For both Simple and Complex Baseline Models, we find that the high-fidelity tools suggest features that are closely associated on both in terms of the correlation coefficient and mutual information. In contrast, low-fidelity tools suggest features whose statistical association is not too different from randomly choosing features.

Consistency across models Table 21 show how often the same features are mentioned by the same diagnostic tools *across different (but similar) models*. Each row in the table compares responses for two different models that are built in a similar fashion. Concretely, this means we compare responses for the two Simple Baseline Models (Logit and simple neural network) as well as responses for the two Complex Baseline Models (XGBoost and neural network). On average, responses agree on 4–5 drivers of DI for the Simple Baseline Models and close to 4 drivers for the more Complex Baseline Models. The maximum agreement ranges between 5–10, and the minimum is often no overlap at all. Our findings suggest that the diagnostic tools identify different correlation patterns that drive disparities across models.

5.7 EVALUATION: USABILITY

The final component of our evaluation relates to the usability of information provided by the diagnostic tools, that is, the ability to identify information that enables lenders to comply with the goals and purposes of consumer protection regulation. In particular, this evaluation asks to what extent identifying drivers of disparities can help lenders develop alternative models that have comparable predictive performance and less disparities than the original models. We test the ability to identify less discriminatory alternative models both on the baseline data on which the models were trained as well as on a new data set with a different applicant composition. We first describe the tests on which our analysis is based and then present results.

Usability: the ability to identify information that enables lenders to comply with the goals and purposes of consumer protection regulation. In particular, the ability of a tool to identify less discriminatory alternative (LDA) models – a key component of fair lending requirements.

Table 19: DISPARATE IMPACT: CONSISTENCY WITH ROLL-UP

(a) PANEL A: SIMPLE BASELINE MODELS

	Baseline	Roll-up 1	Roll-up 2	Roll-up 3
SHAP feat. importance	5.53	6.67	8.90	6.47
Non-SHAP feat. importance	8.00	8.00	9.50	8.00
No feat. importance	0.67	2.17	3.00	1.83
SHAP <> no SHAP	6.30	7.46	8.92	7.13
SHAP <> No feat	1.81	3.47	4.75	3.03
High fidelity	7.67	7.83	9.33	7.83
Medium fidelity	5.35	6.60	9.00	6.40
Low fidelity	0.67	2.17	3.00	1.83

(b) PANEL B: COMPLEX BASELINE MODELS

Complex	Baseline	Roll-up 1	Roll-up 2	Roll-up 3
SHAP feat. importance	2.12	3.60	7.18	3.27
Non-SHAP feat. importance	3.25	3.50	5.75	3.50
No feat. importance	0.92	1.92	3.92	1.33
SHAP <> no SHAP	2.94	4.56	7.13	4.40
SHAP <> No feat	0.77	2.47	5.15	1.68
High fidelity	5.67	6.83	8.25	6.67
Medium fidelity	2.25	3.58	7.30	3.45
Low fidelity	0.92	1.92	3.92	1.33

Note: The table shows results for the consistency test of drivers of disparities aggregated across the roll-up categories. Each number represents the number of drivers of disparity a response pair has in common. 0 indicates that they have no drivers in common. 10 indicates that they have all 10 drivers in common. Panel A shows results for the Simple Baseline Model and panel B aggregates across the XGBoost and neural network models. The first column shows the baseline results (no roll-up), the remaining columns show the three roll-up schemes discussed in the text. Roll-up scheme 1 uses feature descriptions. Roll-up scheme 2 uses product type. Roll-up scheme 3 groups features into categories that combine feature descriptions and product types. Each column shows the number of features a pair of responses has in common - averaged across all response-pairs in that group.

Table 20: DISPARATE IMPACT: CONSISTENCY WITH CORRELATION ANALYSIS

(a) PANEL A: SIMPLE BASELINE MODELS

		Pearson correlation coefficient			Mutual Information		
	N	Mean	Min	Max	Mean	Min	Max
All	55	43.99	6.82	94.16	26.10	1.44	58.34
SHAP feat. importance	15	58.57	50.70	74.52	32.21	19.73	46.26
Non-SHAP feat. importance	1	58.29	58.29	58.29	41.01	41.01	41.01
No feat. importance	3	11.42	7.64	17.73	5.74	2.14	11.27
SHAP <> no SHAP	12	59.88	49.09	94.16	35.90	19.08	49.78
SHAP <> No feat	18	30.51	6.82	79.73	19.21	1.44	58.34
High fidelity	3	72.55	60.81	94.16	36.28	24.43	49.78
Medium fidelity	10	58.23	49.09	74.52	32.54	19.08	43.43
Low fidelity	3	11.42	7.64	17.73	5.74	2.14	11.27
		Mean	Standard dev.		Mean	Standard dev.	
Benchmark		29.08	33.54		17.05	23.4	

(b) PANEL B: COMPLEX MODELS

		Pearson correlation coefficient			Mutual Information		
	N	Mean	Min	Max	Mean	Min	Max
All	55	24.60	4.65	63.19	14.39	2.01	44.51
SHAP feat. importance	15	27.02	15.78	48.67	17.06	9.11	44.51
Non-SHAP feat. importance	1	39.25	39.25	39.25	21.42	21.42	21.42
No feat. importance	3	13.02	5.81	25.68	7.05	3.09	12.12
SHAP <> no SHAP	12	32.87	19.71	63.19	20.37	8.88	43.15
SHAP <> No feat	18	18.40	4.65	47.50	9.81	2.14	30.47
High fidelity	3	31.29	19.71	45.63	28.97	18.87	44.51
Medium fidelity	10	34.23	15.78	63.19	20.20	10.66	43.15
Low fidelity	3	13.02	5.81	25.68	7.05	3.09	12.12
		Mean	Standard dev.		Mean	Standard dev.	
Benchmark		17.06	19.88		9.88	18.89	

Note: The table shows results for the strength of the statistical association between the drivers of disparities provided by different responses. The two metrics are the Pearson correlation coefficient and a measure of mutual information described in the text. Both metrics range from 0-100, with zero indicating no statistical association and 100 indicating perfect association. Benchmark refers to the statistical association of 100 (44 for the Simple Baseline Models) randomly chosen features in the data. Panel A shows results for the Simple Baseline Models and panel B aggregates across the XGBoost and neural network models. We show two different ways of aggregating responses (by tool used as well as by their fidelity performance). N refers to the number of response-pairs (*e.g.*, company A and company B constitute one pair).

Table 21: DISPARATE IMPACT: CONSISTENCY ACROSS MODELS

			Features in common (out of 10)		
Model 1	Model 2	# resp.	Mean	Max	Min
Logit	Simple NN	10	3.5	5	0
Neural Net	XGBoost	10	3.6	5	2

Note: The table shows results for consistency tests of drivers of disparities across prediction models. Each row compares responses for two models of similar complexity.

5.7.1 EVALUATION DESCRIPTION

USABILITY TESTS

Our usability tests proceed in four stages. We first evaluate whether the less discriminatory alternative (LDA) models proposed by each participating company reduce adverse impact when additional test data are run through the models – and at what cost to predictive performance. We then asked the companies to generate LDA improvements using never-seen-before data set with a different applicant composition. This test allows us to assess how well tools generalize to a different environment. We then evaluate how LDA improvements compare when companies are given a sample of the never-seen-before data that *omits* protected class information. Finally, we evaluate how the LDA improvements compare when companies incorporate a sample of the never-seen-before data in their analysis that also includes protected class. This final stage allows us to test how much having access to protected class information matters for the ability to identify promising LDA candidates.

Our evaluation of the usability of model diagnostic tools in the context of fair lending and disparate impact analysis is based on evaluating each company's recommendation for developing a fairer model by exploring one or more less discriminatory alternatives (LDAs). We present results for the following four stages.

1. Stage I: We present results of the LDA search that was performed by the participating companies on the training data. We evaluate all LDA models on a test data set that was not used in the construction of the LDA models. We separately present results for the Baseline Models and for the Company Models.

2. Stage II: We evaluate the performance of the original LDA recommendations when the adjusted models are applied to the deployment test data. Recall that, as part of the usability analysis, we construct a new data set with a different applicant composition ("deployment data"). Table 13 shows basic summary statistics for this deployment data set. The over-sampling increases the fraction of minority applicants from roughly 20% to 30% and increases the default rate in both groups. It also increases the average differences in default rate between both groups from 10% to 15%. The deployment test data was not used in the construction of the LDA models. We separately present results for the Baseline Models and for the Company Models. The Stage II evaluation helps us assess the stability of the LDA improvements found in Stage I when evaluated in a different context. In other words, we are asking whether the LDA improvements found in Stage I generalize to a new environment when the model is confronted with a different applicant distribution.

3. Stage III: We evaluate a new set of LDA recommendations generated based on applying the models to a training sample of the deployment data – a dataset that differs from the Stage I training data and represents a different composition of applicants. This deployment data set is identical to the Stage II deployment data. Notably, participating companies were *not* given protected class information for the training sample of the deployment data. As in Stage II, we evaluate the performance of the LDA search on the deployment test data. The deployment test data is distinct from the deployment training data and only the latter was used in the construction of the LDA models. We separately present results for the Baseline Models and for the Company Models. Stage III helps us evaluate how much re-running the LDA search with additional information about the deployment data improves the LDA results relative to Stage II.

4. Stage IV: We evaluate a new set of LDA recommendations generated based on applying the models to the same deployment data as used in Stage II. In this stage, however, participating companies were *also* given protected class information for this new deployment data. As in Stage II and III, we evaluate the performance of the LDA search on the deployment test data. The deployment test data was not used in the construction of the LDA models. We separately present results for the Baseline Models and for the Company Models. Similar to Stage III, Stage IV helps us evaluate how much re-running the LDA search with additional information about the deployment data improves the LDA results relative to Stage II. Since not all companies participated in Stage III, Stage IV contains important additional information about the improvement of the LDA search when incorporating information about the new data environment. In combination with Stage III, we can also learn about the importance of the presence of imputed protected class information to the performance of the LDA search.

We focus our analysis on the Logit and XGBoost models since we received the most complete responses for these models.

5.7.2 USABILITY: RESULTS

KEY FINDINGS

We present three key findings. First, the ability to describe features that drive disparities with respect to a protected class does not automatically lead to models that are less discriminatory alternatives (LDA) when this information is used mechanically. Automated tools perform significantly better than strategies based on dropping features that were identified as drivers of disparities in the model. Second, among the more automated tools, no single approach does best across all model types and fairness metrics. Third, all automated tools generalize well to new environments. In particular, we find that initial LDA improvements continue to present comparable solutions on a never-seen-before data set with a different applicant composition.

Key results We present the following three key results from our usability analysis.

1. The ability to explain what drives disparities does not lead to less discriminatory alternatives models when this information is used mechanically. A strategy based on dropping features that drive disparities does not lead to models that have significantly smaller disparities but often leads to substantial performance deterioration.

In contrast, automated tools that search for a range of less discriminatory alternative models, instead of just dropping important features, can successfully improve fairness metrics.

2. Among the more automated tools considered in this study, there is no tool that always performs best at identifying a fairer alternative model at the lowest cost to predictive performance. Rather, we find that which automated tool performs best depends on both the type of underwriting model as well as the specific metric of adverse impact we consider.

3. All of the more automated tools considered in this study generalize well to new environments. In particular, these tools exhibit two desirable properties when applied to a data set that has a different loan applicant composition. First, the LDA improvements carry over to a new environment that was not used in the model building process. Even though the *level* of adverse impact and predictive performance deteriorate on the new data, the LDA models still represent a significant fairness improvement while minimizing predictive performance cost. In fact, these solutions are comparable with the models resulting from a search for LDAs on a training sample from the new data. Second, we find that these tools are able to identify good LDA models on a data set with a very different applicant composition. While we observe efficiency gains for these tools when protected class data *is* provided, the automated tools considered in this study are able to attain good results even when this information is withheld.

Structure of results We present the LDA results according to the four stages described in subsubsection 5.7.1. For each stage, we separately present results for Baseline and Company Models (see correspondence in the enumeration below).

1. Stage I: Figures 6 and 7
2. Stage II: Figures 8 and 9
3. Stage III: Figures 10 and 11
4. Stage IV: Figures 12 and 13

Each LDA figure shows both predictive performance and fairness metrics on different axes of the graphs to visualize the trade-offs between these two key metrics. The y-axis on all graphs shows a measure of predictive performance – the Area-under-the-Curve (AUC). Higher AUC numbers reflect higher predictive performance – with 1 indicating perfect prediction and 0.5 performance that is no better than a random guess. The x-axis shows a measure of disparities with values further to the right corresponding to models with less disparities. For all figures in the main text, we use the AIR metric and present results for both a 10% and 20% threshold. In Appendix B, we also present all corresponding figures with the SMD metric. Points further to the top-right of the graph represent more desirable models, that is, these models exhibit both high predictive performance and low disparities. Points closer to the bottom left of the graph show less desirable models, that is, these models exhibit less predictive performance and more disparities. As we move from the top-left point to the bottom-right of the graph, we trade off predictive performance against the disparities of the model.

Each line or set of points corresponds to a model evaluated on the test data set (either test or deployment test set). For each method, we show both the starting point and the alternative models found as part of the LDA search. We refer to the starting point as the "baseline." The starting point represents the models that were trained without

protected class information and that we evaluated in the preceding section on fairness properties. Some approaches suggested multiple points, and we represent these approaches as a line that interpolates between the set of points received. The top-left point on each line represents the model prior to any LDA search. Note that some companies first built their own Baseline Models, that is, these approaches built replica models of the Baseline Models prior to the LDA search. One company producing these replica models restricted themselves to only features that were not required to be masked by the data vendor. For those LDA results, we separately show the new baseline point generated by this method. All results should be evaluated relative to their own baseline model since some approaches produced replica models that perform slightly differently.

We now discuss each of the four stages in turn.

Stage I

Baseline Models Stage I represents the first round of LDA searches that were performed by the participating companies on the training data. Starting with the XGBoost model, we find that three approaches work well in tracing out a frontier of models that trade off predictive performance and adverse impact (in particular when considering the AIR metric and a 20% threshold). All of these three methods suggested a series of LDA models. These three methods are depicted as the two lines (solid and dotted) as well as the series of x symbols in the figures. Each of these three approaches start from a slightly different baseline model due to two companies first building their own replica of the Baseline Model.[53] In Figure 6, these baselines are depicted as the blue x and the top-left points of the dotted and solid lines, respectively. All three methods involve a degree of automation in their search for LDA models. While all three methods start from slightly different baseline models, all three methods suggest LDA models that perform broadly similarly in terms of predictive performance and disparities metrics. However, no single method does best across all models and fairness metrics. We show additional results replacing the threshold-specific AIR metric with the SMD metric in Figure B.2 in Appendix B. We hypothesize that this finding is driven by the fact that each LDA method considered in this study optimized for a different fairness metric and performs best for the fairness metric that it targeted.

All other approaches remain within these frontiers. These approaches are characterized by the diamond, circle, and square points in the figure. All of these points should be evaluated relative to the Baseline Model, which is represented by the top-left point on the solid line. The feature reweighting leads to virtually no change. We hypothesize that this likely reflects that only four distinct sample weights were supplied. A more complex (conditional) reweighting approach might lead to larger observed changes. The two feature dropping approaches appear generally inefficient. The best among them generates models with less adverse impact but at high performance cost. The worst simply reduce predictive performance but do not improve adverse impact.

To understand the magnitude of these changes, it is helpful to translate the points on the figures into (a) the number of minority and non-minority applicants who are approved at a given approval threshold, (b) the number of applicants who are "incorrectly" accepted at a given approval threshold, that is, are accepted but then default ("false approval") and (c) the number of applicants are "incorrectly" rejected, that is, are rejected at a given approval threshold but would not have defaulted ("false rejects"). We consider only the three approaches that propose models that lead to significant fairness improvement. At a (relatively generous) 20% approval threshold, all three baseline (or

[53] As noted above, one company conducted the LDA analysis only on features that were identifiable given masking requirements from our data vendor. Since some of the excluded variables were important in the model, this choice affected the relationship of their LDA outputs to the frontier.

starting point) models approve about 56% of minority applicants and 78% of non-minority applicants. About 52% of minority applicants are "incorrectly" approved and 20% are "incorrectly" rejected. The corresponding numbers for the non-minority group are 75% and 10%. The fairest alternative for each approach increases approval rates for minorities by 10, 11, and 20 percentage points, respectively. The non-minority approval rates of the fairest model increase by 2 percentage points, 0 percentage points and 8 percentage points, respectively. These fairest alternative models correspond to the points furthest to the right on the solid and dotted lines as well the "x" marker furthest to the right in Figure 6. The increased approval rates are also reflected in fewer false rejects. The false reject rate for the minority group drops by 6 percentage points, 9 percentage points, and 23 percentage points, respectively. The corresponding drops for the non-minority groups are 4 percentage points, 3 percentage points, and 24 percentage points. However, these relatively larger increases in approval rates for the minority group come at the cost of more false approvals. Intuitively, as more applicants are approved, we are less likely to falsely reject a qualified applicant but more likely to approve an applicant who will later default. The false approval rate for the minority group increases by 7 percentage points, 8 percentage points and 16 percentage points, respectively. This increase in false approval rates is reflected in the lower predictive performance of these alternative models, that is, they have lower AUC metrics in Figure 6. Note that the AUC captures false approvals and false rejects *at all possible approval thresholds* while this specific example considers a single approval threshold of 20%.

We find qualitatively similar results for the Simple Baseline Model. The reweighting approach again has virtually no effect on the model. Some of the feature drop approaches now lead to improvements in predictive performance and, in some cases, to improvements in adverse impact. The latter is partially an artifact of how we implemented the feature-dropping approach for Simple Baseline Models. Given the small number of features in the model, we used a LASSO-based feature selection procedure to replace the dropped features as opposed to simply omitting them from the model. Given the large number of features that were recommended to be dropped, simply omitting these features from the model would have led to significantly lower model complexity and undermined the comparison needed to evaluate the results.

The Simple Baseline Models starting points for the three automated approaches again differ since two companies built replica versions of the Logit provided by the research team. Again considering a 20% approval threshold, the three baseline models have minority approval rates of 58%, 56%, and 57%, respectively. Approval rates for the non-minority group are 78%, 76%, and 80%, respectively. In Figure 6, these baseline models are depicted as the blue x and the top-left points of the dotted and solid lines, respectively. The fairest alternative model increases minority approval rates by 21 percentage points, 10 percentage points, and 22 percentage points, respectively. Approval rates for the non-minority group again increase by less, namely by 6 percentage points, 3 percentage points, and 2 percentage points, respectively. The relatively larger increase in approval rates, which drives the improvements in the AIR metrics, is again reflected in larger drops in the false reject rates. False reject rates decrease by 25 percentage points, 10 percentage points, 22 percentage points for the minority group and by 21 percentage points, 8 percentage points and 15 percentage points for non-minority group. Again, higher approval rates come at the cost of higher false approval rates, which increase by 15 percentage points, 7 percentage points, and 8 percentage points for the minority group and by 4 percentage points, 2 percentage points and 0 percentage points for non-minority group. We note that one approach also provided two models not depicted in the Figure 6 for easier graphical representation. These models represent relatively extreme points of almost perfect fairness but relatively low predictive performance. On the graph, these models would lie to the very far right of the graph at an AIR of close to 1 (at a 20% threshold) but at a very low predictive performance of an AUC of 0.65.

Company models We now discuss the LDA search for five Company Models, since one company which built a prediction model did not participate in the disparate impact analysis. We also show LDA results for the open-source tool applied to the XGBoost Baseline Model as a reference ("Method X"). We find similar results for the company LDA search as for the Baseline Models. Unlike in the first two figures, we now compare LDA results with each method *also corresponding to a different model*. The focus of the analysis is therefore on the improvement of each method/model relative to its baseline. As before all baseline points are depicted in blue and all LDA models are depicted in black. Baseline-LDA pairs share the same symbol on the graph.

We find qualitatively similar results for the Company Models. The three LDA searches that relied on a feature drop strategy do not consistently improve the model's fairness and reduce predictive performance. One unusual case here is the third feature-drop solution, which improves adverse impact substantially relative to a small performance drop in the case of AIR, although the starting point model does not have the same predictive power. The two LDA searches that relied upon automated adversarial debiasing strategies perform well at generating a range of models with lower adverse impact. These LDA models look comparable to the XGBoost open-source LDA search generated by the research team.

For the Baseline XGBoost model, the best LDA results attain an AIR of around 0.8 (at a 20% threshold) with an AUC of 0.85. Two Company Models combined with an automated debiasing attain similar predictive performance-fairness combinations. In contrast, some of the LDA results for the Company Models extend the frontier further and propose models higher AIR but lower predictive performance costs. Note that this finding also holds with the threshold-independent SMD metric (see Figure B.3 in Appendix B). One company that employed a feature-drop strategy for their LDA search offers significantly higher fairness when considering the 20% AIR threshold but not when evaluated on the SMD metric.

Stage II

Baseline Models Across both Baseline and Company Models, we find that both predictive performance and disparities worsen on the new data set. The performance deterioration is unsurprising since the data set is no longer drawn from the same population as the one on which the models were trained. The deterioration in disparities is partly by design as we purposely over-sampled minority applicants for this new data set. If there are more minority applicants whom the models predict pose a higher risk of default, this difference will translate into lower AIR and SMD metrics.

However, we find that the LDA models continue to represent significant improvements when evaluated on the deployment data set. The patterns we observed in Stage I largely generalize to the new environment. This finding suggests that although the underlying applicant composition has changed, the model improvements found as part of the LDA search continue to hold on the deployment data. In other words, improvements in disparities are not specific to the particular environment on which they were built. Below we compare these improvements relative to the LDA models that were proposed incorporating information about this new data set.

Stage III and IV Stage III presents results of the LDA search that was performed by the participating companies on the deployment data. However, participating companies were *not* given protected class information for this new deployment data. Note that we received fewer responses for this stage relative to Stage III due to the perceived difficulty of performing this task without additional protected class information. Stage IV presents results of the

LDA search that was performed by the participating companies on the deployment data. In this stage, participating companies were given protected class information for this new deployment data.

Baseline Models We find broadly similar results in Stage IV on the relative performance of different methods to perform LDA searches. The two automated tools continue to outperform feature-dropping and reweighting approaches (one automated tool did not participate in this part of the research). In Stage I, the automated approaches were able to suggest models that led to a roughly 10 percentage point increase in the AIR metric (regardless of approval threshold). This improvement came at the cost of a roughly 2.5 percentage point drop in AUC. On the new deployment data in Stage IV, the automated tools produce comparable improvements in fairness. The predictive performance costs are slightly larger and around 3 percentage points of AUC. Overall, however the performance is remarkably similar.

For the Simple Baseline Model, we find that one of the feature drop approaches leads to improvements in predictive performance and, in some case, to improvements in adverse impact. We again highlight that the latter is partially an artifact of how we implemented the feature-dropping approach for Simple Baseline Models. Given the small number of features in the model, we used a LASSO-based feature selection procedure to replace the dropped features as opposed to simply omitting them from the model. Given the large number of features that were recommended to be dropped, simply omitting these features from the model would have led to significantly lower model complexity and undermined the comparison needed to evaluate the results.

Translating our results to approval rates, we find that the fairest models resulting from the two automated approaches in Stage IV increase minority approval rates by 16 percentage points and 12 percentage points, respectively. These results describe the points in Figure 12. One approach leads to an increase in approval rates for the non-minority group by 3 percentage points while the other decreases approval rates for non-minority group by 2 percentage points. False reject rates for the minority group again drop – by 7 percentage points and 6 percentage points for the approaches, respectively. For the non-minority group, false reject rates drop by 5 percentage points and 1 percentage points. We again observe a trade-off with the number of false approvals. For the minority group, the two approaches lead to increases by 12 percentage points and 9 percentage points, respectively. Reflecting on the non-minority approval rate changes, we find that false approvals increase by 2 percentage points for one method and decrease by 2 percentage points for the other method.

Comparing Stage III and IV results, presented in Figure 10 and Figure 12, shows that the presence of protected class information on the new deployment data set provides surprisingly little value. That is, the two more automated approaches are still able to find comparable LDA models even when relying only on a prediction of protected class instead of the imputed protected class information used in this study. The same amount of model fairness requires a roughly 1-2 percentage points higher AUC reduction in Stage III – where protected class is not known – relative to Stage IV, where protected class information is known. The result holds across all fairness metrics (Figure B.6 and Figure B.8 in Appendix B provide results for the SMD metric).

Comparing the results in terms of approval rates, the two more automated approaches increase minority approval rates by 13 percentage points and 4 percentage points, respectively. The non-minority approval rates are reduced by 1 percentage point and 0 percentage points, respectively. False reject rates for minorities drop by 7 percentage points and 2 percentage points and minority false acceptance rate increase by 9 percentage points and 3 percentage points, respectively. These smaller magnitudes likely reflect two different forces. First, the lack of protected class reduces the ability to efficiently find LDA alternatives. However, these losses appear small. Second, the smaller magnitudes likely reflect that one of the more automated approaches presented a narrower range of LDA models in

Stage III relative to Stage IV. This difference likely does not reflect a technical limitation but rather an *ad hoc* decision in the parameters used in the LDA search.

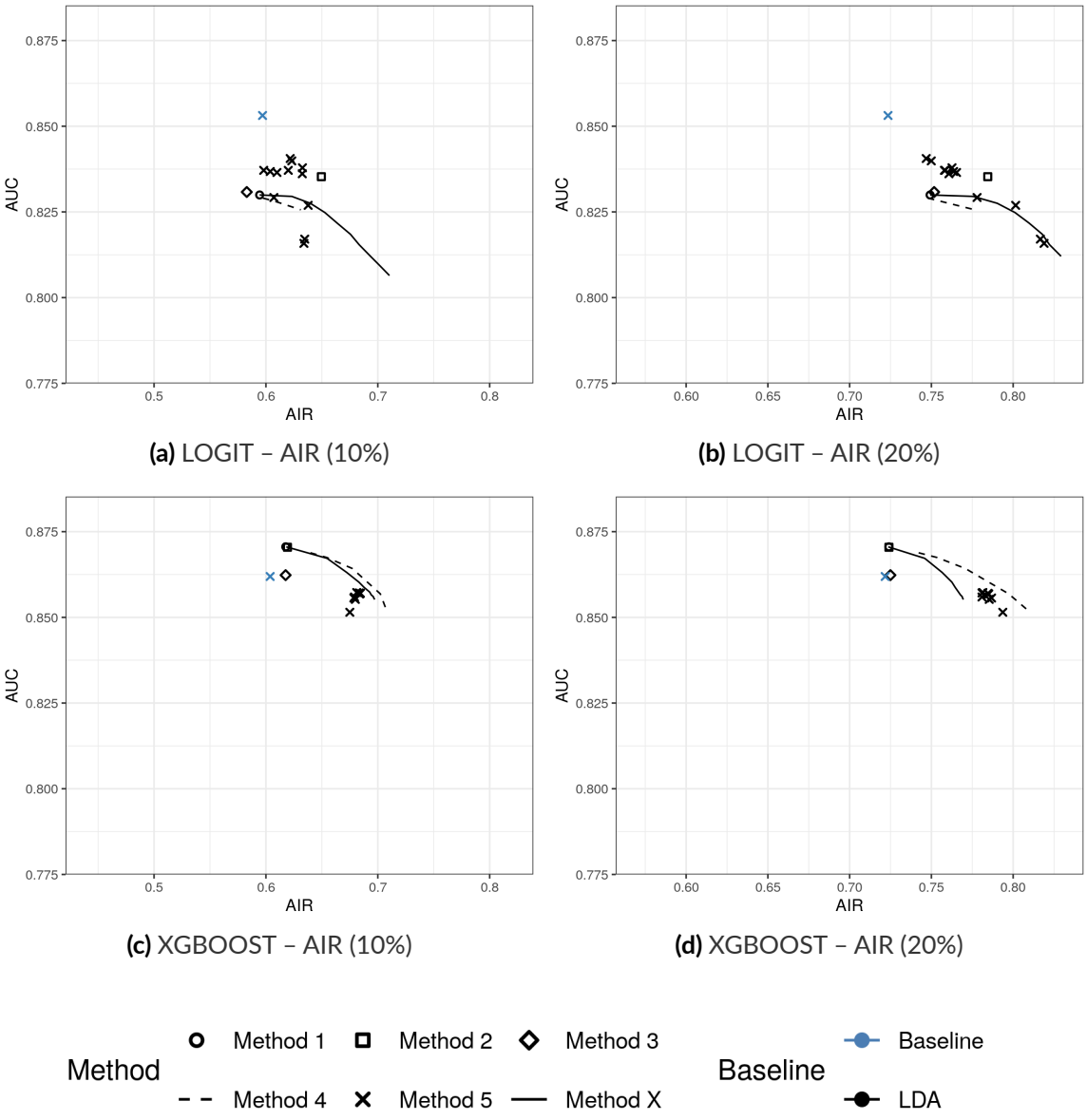
We find that the LDA models found in Stage IV do not necessarily represent unambiguous improvements over the LDA solutions found in Stage I. In other words, the gains to re-running the LDA search on a new data set appear quite small if present at all. One automated solution proposes LDA models in Stage IV that have higher predictive performance but do not necessarily improve on the AIR or SMD metrics. This finding holds true for both the Logit and XGBoost Baseline Models. The second approach similarly proposes LDA models in Stage IV with higher predictive performance but slightly more disparate impact. In contrast, the Stage IV solutions for the XGBoost model sacrifice some predictive performance but decrease disparate impact. These differences however are quantitatively small – with less than a percentage point difference in AUC and less than 5 percentage point differences on the AIR and SMD metrics.

Company models Our Stage I results also broadly hold for the Stage III and IV results. The automated methods outperform the feature-drop approaches although we observe relatively larger improvements for the feature-drop approaches in Stage IV relative to Stage I. However, this difference in magnitudes is difficult to interpret since the underlying data sets differ significantly. One feature-drop method-model combination continues to provide alternative models with large fairness improvements *when considering the AIR metrics*. However, this improvement does not hold when looking at the SMD metric and both the initial model as well as the LDA model do not have predictive power on par with LDA models produced by other methods.

Similar to our finding for the Baseline Models, there are small efficiency gains from providing protected class information. However, these gains are small, and all methods still perform well when no protected class information is available. One difference is that the set of automated tools provides a wider range of LDA alternatives when given protected class information in Stage IV. This difference likely does not reflect a technical limitation but rather an *ad hoc* decision in the parameters used in the LDA search.

Similar to Stage I, we find that the automated tools applied to the Company Models propose LDA models with similar predictive accuracy and fairness properties to the automated tools applied to the Baseline XGBoost model. This finding holds both for the AIR metrics as well as the threshold-independent SMD metric (see Figure B.9 in Appendix B). This finding suggests that the ability of automated tools to find efficient LDA alternatives is not model-specific.

Figure 6: LDA RESULTS STAGE I – BASELINE MODELS



(a) LOGIT – AIR (10%)

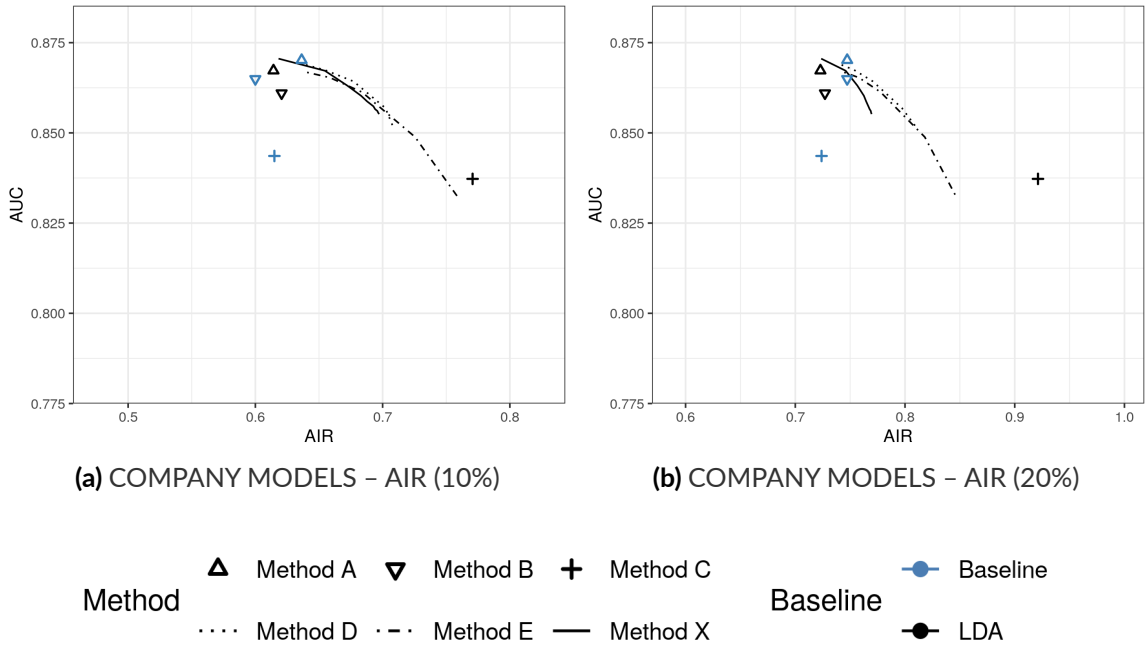
(b) LOGIT – AIR (20%)

(c) XGBOOST – AIR (10%)

(d) XGBOOST – AIR (20%)

Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metrics, using adverse impact ratios at either a 10% or 20% approval threshold. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure 7: LDA RESULTS STAGE I – COMPANY MODELS

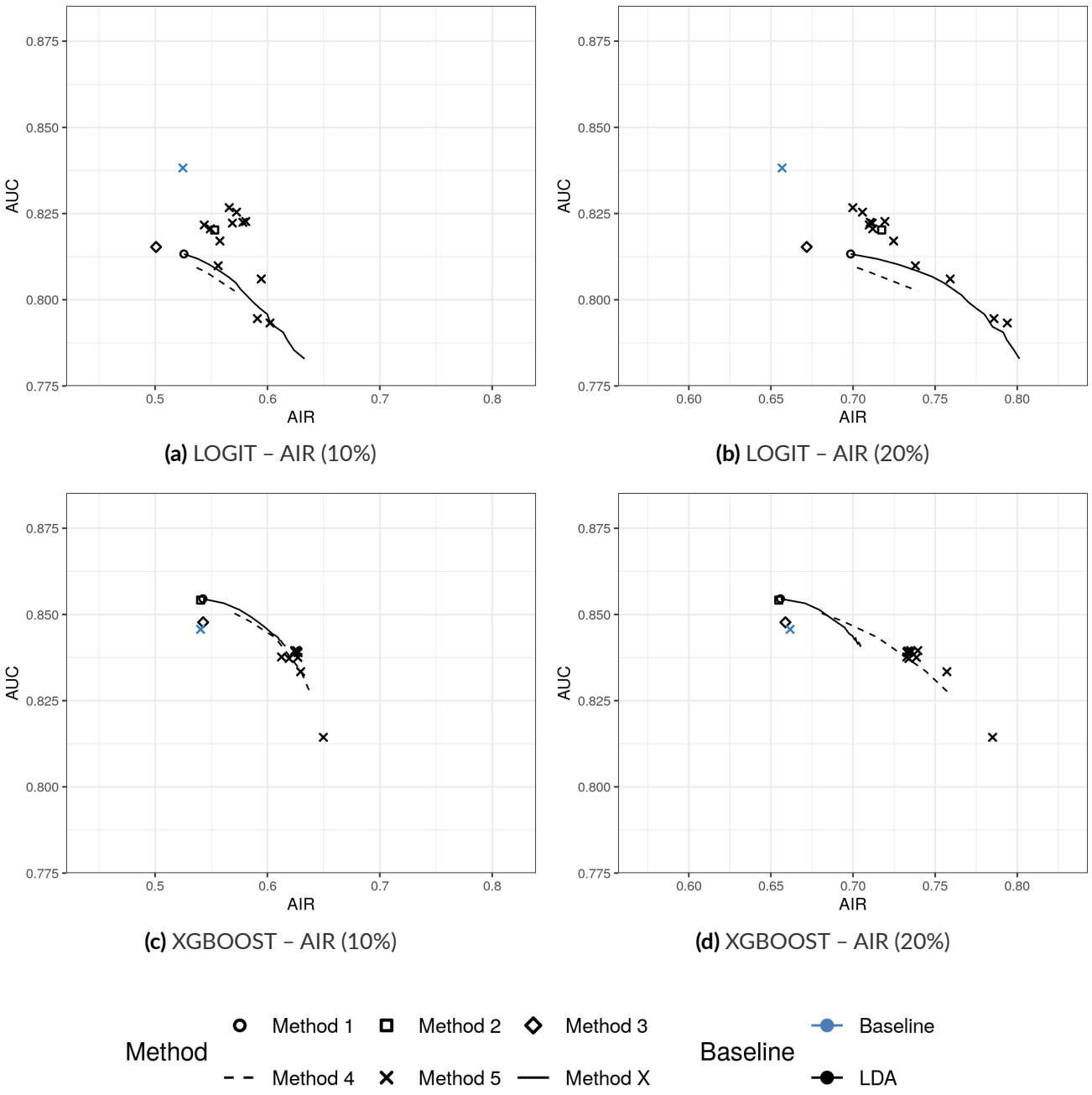


(a) COMPANY MODELS – AIR (10%)

(b) COMPANY MODELS – AIR (20%)

Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) search for the Company Models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metrics, using adverse impact ratios at either a 10% or 20% approval threshold. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For example, the blue triangle represents the starting point for Method A and the black triangle represents the LDA model for Method A. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure 8: LDA RESULTS STAGE II – BASELINE MODELS



(a) LOGIT – AIR (10%)

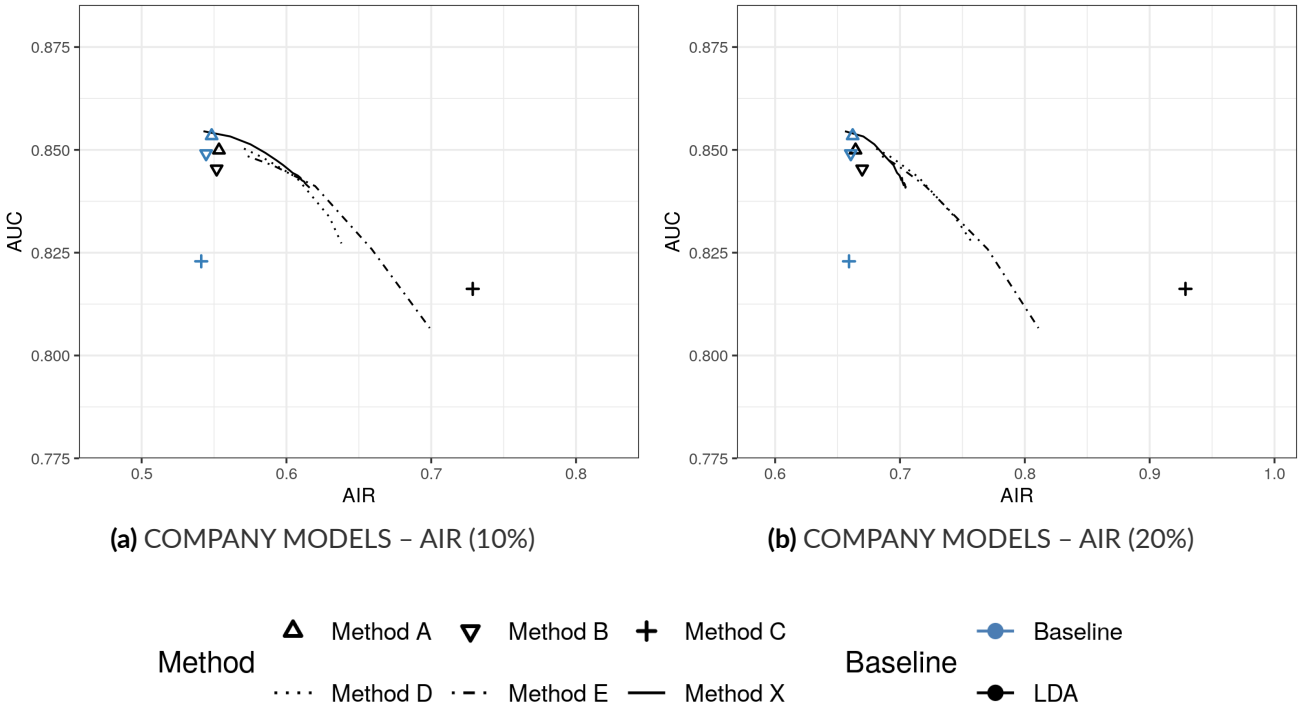
(b) LOGIT – AIR (20%)

(c) XGBOOST – AIR (10%)

(d) XGBOOST – AIR (20%)

Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metrics, using adverse impact ratios at either a 10% or 20% approval threshold. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure 9: LDA RESULTS STAGE II – COMPANY MODELS

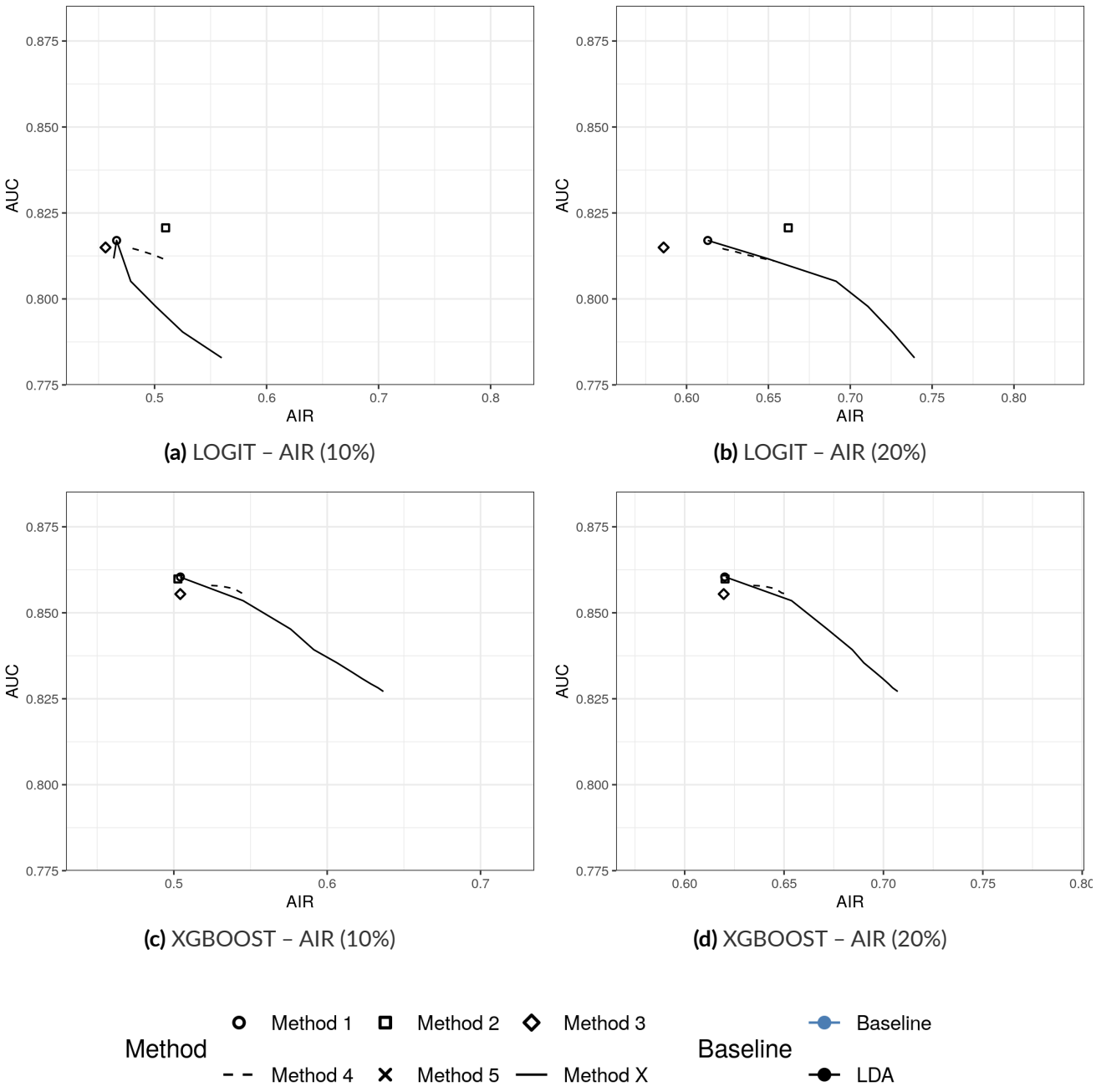


(a) COMPANY MODELS – AIR (10%)

(b) COMPANY MODELS – AIR (20%)

Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) search for the Company Models. All statistics are computed on the deployment test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metrics, using adverse impact ratios at either a 10% or 20% approval threshold. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For example, the blue triangle represents the starting point for Method A and the black triangle represents the LDA model for Method A. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure 10: LDA RESULTS STAGE III – BASELINE MODELS



(a) LOGIT – AIR (10%)

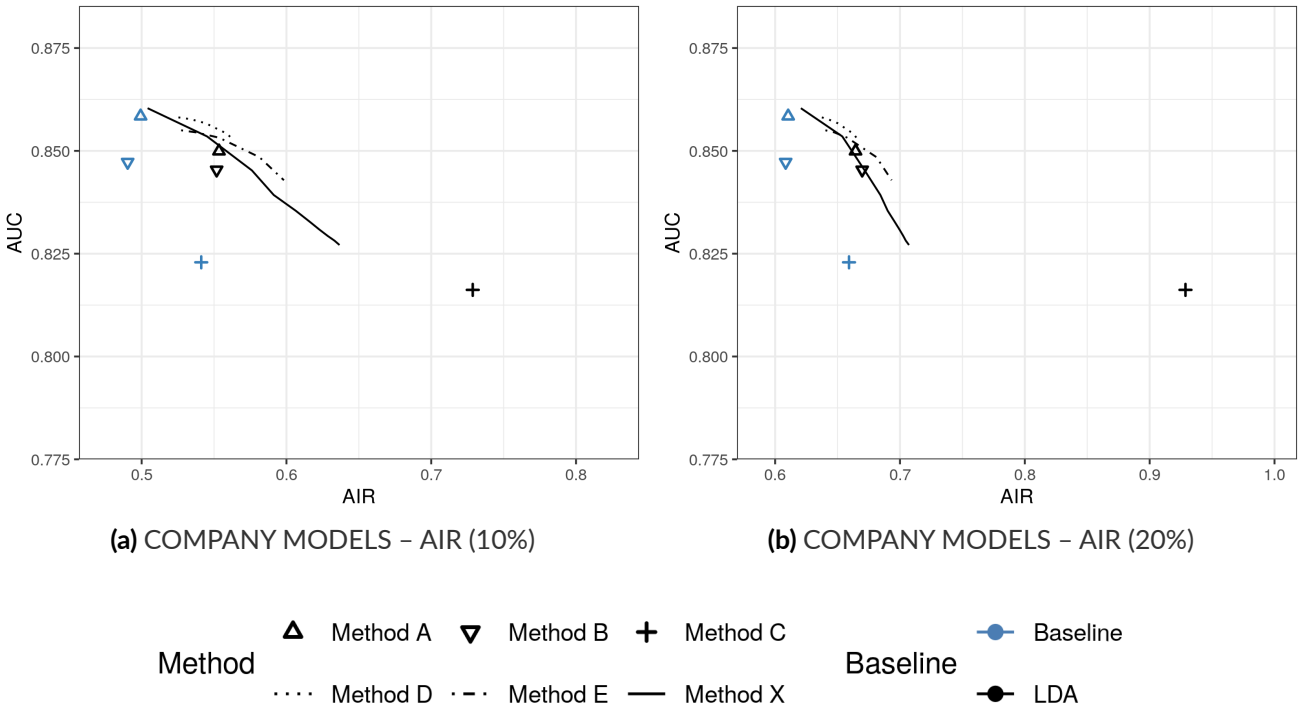
(b) LOGIT – AIR (20%)

(c) XGBOOST – AIR (10%)

(d) XGBOOST – AIR (20%)

Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) search for the Baseline Models. All statistics are computed on the deployment test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metrics, using adverse impact ratios at either a 10% or 20% approval threshold. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure 11: LDA RESULTS STAGE III – COMPANY MODELS

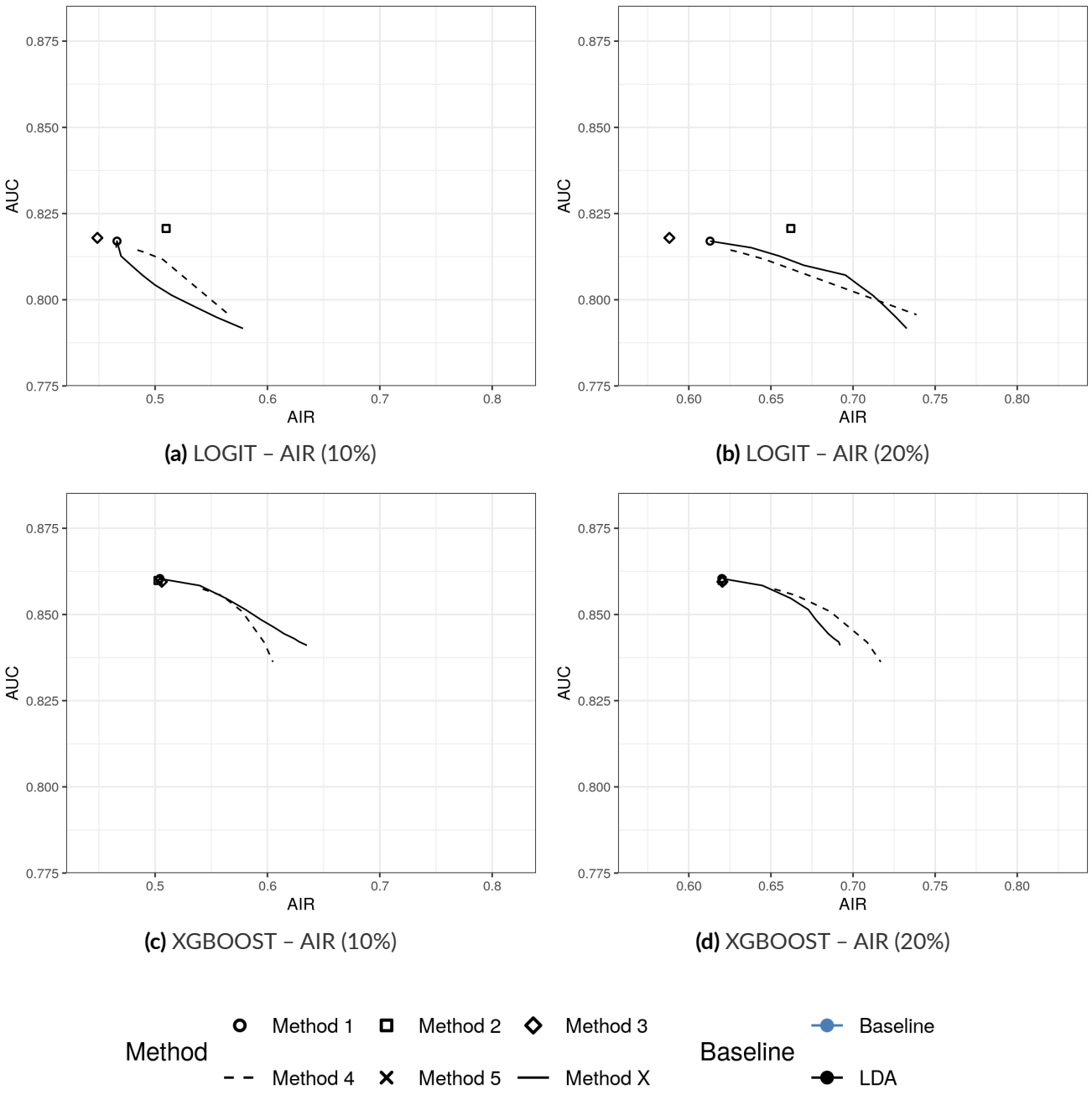


(a) COMPANY MODELS – AIR (10%)

(b) COMPANY MODELS – AIR (20%)

Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) search for the Company Models. All statistics are computed on the deployment test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metrics, using adverse impact ratios at either a 10% or 20% approval threshold. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure 12: LDA RESULTS STAGE IV – BASELINE MODELS



(a) LOGIT – AIR (10%)

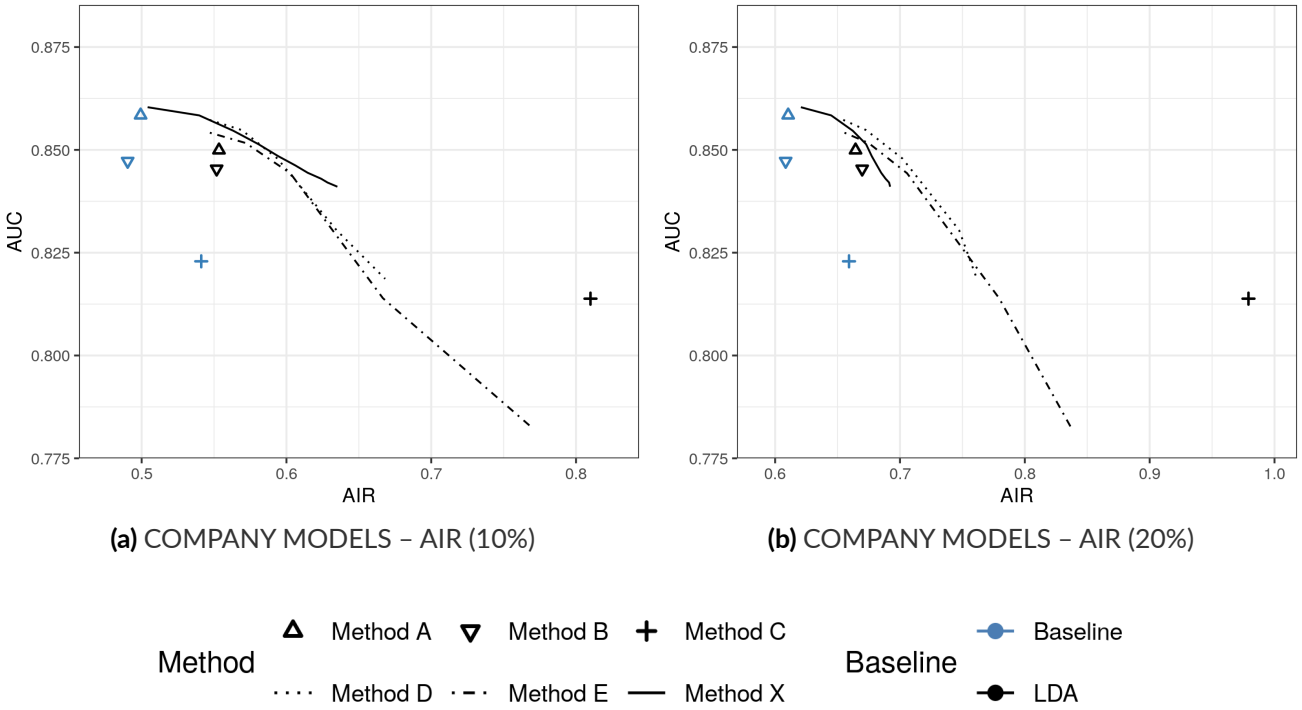
(b) LOGIT – AIR (20%)

(c) XGBOOST – AIR (10%)

(d) XGBOOST – AIR (20%)

Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) search for the Baseline Models. All statistics are computed on the deployment test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metrics, using adverse impact ratios at either a 10% or 20% approval threshold. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure 13: LDA RESULTS STAGE IV – COMPANY MODELS



(a) COMPANY MODELS – AIR (10%)

(b) COMPANY MODELS – AIR (20%)

Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) search for the Company Models. All statistics are computed on the deployment test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metrics, using adverse impact ratios at either a 10% or 20% approval threshold. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For example, the blue triangle represents the starting point for Method A and the black triangle represents the LDA model for Method A. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

6 RESULTS: MODEL RISK MANAGEMENT

This section presents our evaluation of the participating model diagnostic tools with respect to model risk management. It consists of two main sections: (1) background and (2) empirical evaluation. The former section provides an overview of relevant policy considerations, legal and regulatory expectations, and operational considerations. The latter section presents our analysis that evaluates how well diagnostic tools identify drivers of model behavior for purposes of model risk management

6.1 BACKGROUND

This section considers in turn the legal and regulatory requirements regarding model risk management and operational considerations relevant to this evaluation.

6.1.1 LEGAL AND REGULATORY OVERVIEW

Federal prudential regulators have issued extensive guidance outlining their expectations for steps that banks should take to identify and manage the risks related to the use of models in all aspects of their business and operations.[54] This guidance applies broadly to all models that expose the firm to unexpected losses, compliance problems, or other negative outcomes. These expectations call for development of enterprise-wide model risk management programs that consist of governance processes, policies, and controls. Banks are expected to identify potential sources of model risk,[55] assess the magnitude of those risks, and mitigate them appropriately, both at the individual model level and in the aggregate across business lines and legal entities.

Prudential model risk management expectations emphasize model transparency in a variety of ways, many of which can provide additional challenges for those who choose to develop and use machine learning underwriting models. For example, the guidance creates an expectation that developers will evaluate whether models are relying on relationships in the data that are intuitive and defensible with regard to the outcome that they are attempting to predict,[56] that firms will conduct appropriate sensitivity analyses to establish the soundness of the model for use,[57] and that lenders will establish appropriate processes for identifying and mitigating risks relevant to the model's use, including compliance with applicable consumer protection laws.[58]

[54] Although each agency has its own issuance, the Federal Reserve Board's Supervisory & Regulation Letter 11-7 is often used as a shorthand to refer to all three agencies' guidance. See Board of Governors of the Federal Reserve System, Supervisory & Regulation Letter 11-7: Supervisory Guidance on Model Risk Management (Apr. 4, 2011) (hereinafter "FRB, SR 11-7"); Office of the Comptroller of the Currency, Bulletin 2011-12: Sound Practices for Model Risk Management: Supervisory Guidance on Model Risk Management (Apr. 4, 2011); Federal Deposit Insurance Corporation, Financial Institution Letter 22-2017: Adoption of Supervisory Guidance on Model Risk Management (Jun. 7, 2017). Even though SR 11-7 applies only to banks, risk management practices at nonbank financial institutions may nevertheless reflect bank-specific requirements and norms. Funding and securitization counterparties often seek to impose some of the model risk management practices as contractual requirements.

[55] Model risk is defined to include "the potential for adverse consequences from decisions based on incorrect or misused model outputs and reports," which can lead to financial loss, poor business and strategic decision-making, or damage to a bank's reputation. *See* FRB, SR 11-7 attachment at 3-4.

[56] *Id.* FRB, SR 11-7 (evaluating conceptual soundness involves assessing "documentation and empirical evidence supporting the methods used and variables selected for the model [to] ensure that judgment exercised in model design and construction is well informed, carefully considered, and consistent with published research and with sound industry practice."); *id.* attachment at 6 ("Developers should be able to demonstrate that such data and information are suitable for the model and that they are consistent with the theory behind the approach and with the chosen methodology"); *id.* attachment at 11 ("Key assumptions and the choice of variables should be assessed, with analysis of their impact on model outputs and particular focus on any potential limitations. The relevance of the data used to build the model should be evaluated...").

[57] *Id.* attachment at 11-13 (stating that sensitivity analyses should be performed during both development and deployment).

[58] *Id.* attachment at 17-18.

6.1.2 OPERATIONAL CONSIDERATIONS

Models that help firms decide whether and in what circumstances to extend consumer credit are typically subject to heightened standards in model risk reviews depending upon the composition of a particular firm's businesses and its size. Similarly, the decision to develop a machine learning model will often heighten the degree of internal scrutiny applied in pre-deployment reviews and post-deployment monitoring since use of machine learning can suggest greater potential risk based on the model's methodology, complexity, data usage, and operational structure, among other factors.

In the context of both pre-deployment validation and post-deployment monitoring, use of machine learning models requires particular attention to understanding how changing data conditions are likely to affect the performance of the model. Machine learning models may be prone to overfitting (*i.e.*, they may in effect reflect the training data too closely and not generalize to deployment conditions that differ from that data). Managing models in the presence of these kinds of data or model shifts is an important component of demonstrating a model's robustness and fitness for use.

The emergence of various *post hoc* explainability techniques has led to the development of diagnostic tools that are relevant to a number of assessments that lenders make when developing and validating machine learning underwriting models. For example, having the capacity to describe model behavior can be useful when assessing whether the relationships between each feature in the model and the model's predictions are sufficiently justifiable, when determining whether to use and how to construct appropriate constraints, or when trying to understand performance variations across marketing or application channels or in different time periods across out-of-time or deployment data.

This evaluation focuses on one such application of model diagnostics tools in the context of managing model risk management requirements. We consider whether model diagnostic tools can help lenders detect two ways in which model behavior can change in an out-of-time or deployment context: (a) the distribution of model predictions can change as a result of the composition of the population changing (*e.g.*, now we suddenly have many more applicants with higher predicted default rates); and (b) the model's ability to make accurate predictions can change because the fundamental relationships between variables and behaviors change, for example if credit utilization has a different relationship to default risk in a recession than in stable economic conditions. Our tests assess whether tools can figure out why these two phenomena, which we call "data shift"[59] and "model shift," respectively, occur. In this context, the tools evaluated in this study could conceivably help lenders address performance variations due to both causes. We expect that lenders' established practice for establishing the fitness-for-use of any one machine learning underwriting model may vary widely and in any one case deploy a wide range of tests to establish the robustness of a proposed model's performance. Nevertheless, we hypothesize that the tests used herein are a plausible application of available model diagnostic as part of that effort to validate models.

6.2 EMPIRICAL EVALUATION: SUMMARY

This section describes key results from our evaluation of model diagnostic tools in the context of model risk management.

We evaluate the diagnostic tools' outputs on three dimensions: (1) fidelity, that is, the ability to reliably identify features that are in fact related to global model behavior; (2) consistency, that is, the degree to which different tools

[59] Data shift is also often referred to as covariate shift or feature shift.

identify the same drivers for the same model and the degree to which the tools identify the same drivers across models; and (3) usability, that is, the ability to identify information that enables lenders to comply with the goals and purposes of model risk management. Concretely, we evaluate to what extent diagnostic tools can identify drivers of changes in model behavior in out-of-time data.

KEY FINDINGS

Among the model diagnostic tools we evaluated, some tools can identify features that drive overall model behavior. These tools are able to reliably identify features that are related to the model's behavior such that perturbing these features in a favorable direction sizably affects the model's predictions. These tools are also able to identify features that, when changed in a favorable direction, reduce predicted default probabilities by more than randomly chosen or even closely correlated features. We also find that the same tools are able to identify the sources of changes in model predictions and model performance in an out-of-time (or "deployment") context. We do not find evidence that more targeted approaches that have access to a sample of the deployment data outperform the responses based on the initial diagnostic tools.

6.3 PARTICIPATION DETAILS AND DIAGNOSTIC TOOLS

6.3.1 DESCRIPTION OF TASKS

Participating companies were asked to complete the following two tasks.

1. First, each company was asked to provide the ten most important drivers of overall model behavior for each of the Baseline Models as well as their Company Model(s) where applicable.
2. Second, each company was given a new dataset drawn from a different time period and asked to explain what could account for the change in model behavior. We refer to this out-of-time dataset as "deployment data."

6.3.2 PARTICIPATION DETAILS

All of the research companies participated in the model risk management part of the research. One company provided two different responses. In addition, we built three open-source approaches to identify key drivers of global model behavior. The three companies that built their own credit underwriting models also provided global drivers of model behavior for their models. Four of seven companies supplied responses for the second task that aimed to diagnose sources of changes in model behavior in an out-of-time context. In addition, we computed three open-source responses for the second task.

6.3.3 DESCRIPTION OF TOOLS

We offer a brief description of the approaches that the participating companies took to solving the two model risk management tasks.

DIAGNOSTIC TOOLS

Four types of diagnostic tools were deployed to identify sources of global model behavior: tools that aggregate local SHAP feature importance values, tools that aggregate local LIME feature importance values, tools that aggregate counterfactual explanations, and, finally, tools leveraging permutation importance.

Global model explanations The tools used for the first task can be grouped into three clusters. The first set of tools uses SHAP to compute global feature importance. The second set of tools relies on permutation importance to compute global feature importance. One approach aggregates counterfactual explanations while a final approach aggregates explanations derived from LIME. We group the last two approaches in the "Other" category.

Tools based on SHAP feature importance Six responses aggregated SHAP feature importance values to arrive at global feature importance. Each response then selected the 10 features with the largest global feature importance. The responses differ in their implementation choices along the following dimensions.

First, some responses limited the feature set to globally monotonic features, that is, features that unambiguously increase (or decrease) a model's default prediction as one varies the feature values.

Second, companies employed different SHAP implementations, *e.g.*, kernel SHAP or tree SHAP, depending on the model that is being explained and their assessment of potential trade-offs that come with SHAP implementations that are model agnostic versus an implementation designed for a specific kind of model. Most responses used tree SHAP for the XGBoost model and kernel SHAP for the remaining models. A key difference between these versions of SHAP is how feature values are chosen to obtain predicted values for different combinations (or "coalitions") of features in the model. The core idea behind the Shapley value is to compute the average marginal contribution of a given feature to the predictions of the model. This approach requires computing predictions for all possible coalitions of features. If a coalition does not include a particular feature, that feature's value is replaced with a randomly chosen value. If multiple features do not form part of that coalition, the random draw typically happens individually for each feature and does not take into account the functional relationships between features. For example, if both debt and the debt-to-income ratio needed to be replaced with random values, the kernel SHAP approach would ignore the fact that a change in the debt balance would automatically imply a change to the debt-to-income ratio. The tree SHAP implementation partially addresses this problem but is likely limited in how many of these functional relationships can be considered in practice.

Third, since SHAP requires two sampling steps, different choices were made regarding sampling procedure at each step. Concretely, since an exact computation of Shapley values requires the evaluation of all possible combination of features – which is time and cost prohibitive in many applications, including consumer lending – the SHAP implementation uses approximations. These approximations rely on (a) sampling from the number of possible feature coalitions and (b) potentially limiting the number of individuals in the data set for whom a Shapley value is computed. Both sampling choices can lead to variation in the response.

Tools based on permutation importance Three responses used a form of permutation feature importance. Permutation feature importance is global explanation method that determines how much shuffling – or permuting – one feature at a time affects the prediction error of the model. Unlike the tools in the first and third cluster, permutation importance requires access to the default outcome in order to compute prediction errors.

Remaining tools Two additional tools were used to generate global drivers of model behavior. The first approach uses local explanations generated by LIME and aggregates these responses across instances in the training data. The second approach uses local counterfactual explanations and aggregates these responses across instances in the training data. Both approaches limit computational complexity by first drawing a random sample of instances in the training data, then computing a local explanation for the sampled instances, and finally averaging the feature importance scores across all individual instances to obtain a global importance score.

Directional feature importance In addition to providing the 10 most important features, companies were also asked to provide the direction of feature importance. Directional feature importance indicates whether a feature tends to increase or decrease the default prediction of the model. Determining directional feature importance is much more challenging in the case of global explanations than in the case of local explanations, which we explored in the section on Adverse Action Notices. The reason is that the same feature can have a positive impact for some instances and a negative impact for others. Taking an average across these effects might obscure the variability in the effect and, in some cases, be an unreliable indicator of the overall behavior of the feature. Due to this challenge, one company did not provide global directional feature importance. Another company chose to restrict their responses to globally monotonic features in the data which ensures that the sign of the directional feature importance is unambiguous. While we acknowledge these challenges, our fidelity tests rely on feature perturbation, which requires a directional input.

Several approaches exist to computing global directional feature importance. First, feature importance tools typically also provide the direction of importance. For methods that aggregate local importance scores, global feature directions can be obtained by aggregating these local directional impacts. Aggregation methods include averaging or inspecting dependence plots that show how local feature importance changes as the feature values vary. Second, directional impact can be obtained independently of the feature importance score by either computing whether model predictions increase or decrease as the feature value varies, or by computing whether default rates increase or decrease as the feature value varies.

Out-of-Time Tests For the second task, we evaluate three sets of responses.

1. The first set of responses are the global feature importance responses that companies provided in response to the first task. To be clear, none of the participating companies suggested that we re-use their responses to the first task. Rather, we include these initial responses in the analysis to quantify the additional benefit provided by responses targeted explicitly to the second task.

2. The second set of responses identifies 10 features that (individually) exhibit the largest shift in distribution between the baseline and deployment data sets. These responses were computed by the research team following the suggestions of several participating companies. We used three different statistics to identify the 10 features with the largest data shift (sometimes referred to as covariate or feature shift):

   (a) the Kolmogorov–Smirnov test (or KS test), which is a non-parametric test of the equality of two distributions; (b) the Population-Stability Index (PSI), which is a non-parametric test based on comparing two distributions (and related to the Kullback-Leiber information criterion); and (c) a standardized mean difference between the two distributions. The first two metrics consider the whole distribution of a feature while the third metric only compares differences in feature averages.

3. The third set of responses uses changes in global feature importance scores to identify features that account for the change in model behavior on the deployment data. Responses computed the global feature importance scores on the deployment data and then identified features with the largest change in importance score relative to the baseline data. One response used SHAP feature importance while a second response used permutation importance.

6.4 EVALUATION: FIDELITY

Our first dimension of evaluation is fidelity, that is, the ability to reliably identify features that are in fact driving global model behavior. We first describe the fidelity tests on which our analysis is based and then present results.

Fidelity: the ability of a diagnostic tool to reliably identify features that can help describe drivers of global model behavior.

6.4.1 EVALUATION DESCRIPTION

Our evaluation of the fidelity of model diagnostic tools in the context of model risk management is based on a perturbation test that perturbs all ten drivers of global model behavior in a favorable direction (that is, in the direction that should *reduce* the default prediction) and records the resulting change in the model's prediction. High fidelity for a particular tool is associated with material favorable changes in a model's prediction.

FIDELITY TEST

We conduct a perturbation test to gauge the fidelity of model diagnostic tools. The perturbation test asks whether changing – or perturbing – the features identified as drivers of global model behavior in a favorable direction reduces the model's default prediction on average. High fidelity corresponds to a large drop in predicted default as a result of the perturbation.

Perturbation test A model diagnostic tool's fidelity can be tested by perturbing all ten drivers of global model behavior identified by the tool in the direction that should induce a favorable relative change in default probabilities. We run this perturbation for each loan applicant in the training data and then re-compute the model's prediction to see how much the model's default prediction has decreased. High fidelity of a diagnostic tool corresponds to a large improvement (or reduction) in the predicted default probability suggesting that this feature has a large impact on the model's behavior.

Perturbation-based fidelity tests are more challenging to analyze for global feature importance compared to local feature importance, which we analyze in the section on Adverse Action Notices. To induce a feature perturbation, we need to know whether the feature increases or decreases the model's prediction. We obtain this information from the participating companies or open-source tools (see description of diagnostic tools). A challenge for models that are not constrained to be monotonic is that the same feature can have a positive impact on default for some applicants and a negative one for others. Note that none of the Baseline Model and only one Company Model were monotonicity-constrained. Even if the average direction of a feature is determined to be favorable, we would not

expect to see an increase in that feature to always lower the model's default prediction. Nevertheless, perturbation tests can provide useful insights into the fidelity of identified drivers of global model behavior.

Our perturbation schemes are described in detail in Appendix C. Our preferred approach emphasizes producing as many changes as possible (even if this might mean an unfavorable change).[60]

Benchmarks We evaluate the fidelity performance relative to a random benchmark that selects 10 features at random and performs both fidelity tests. We run 100 iterations of the random selection and average results to obtain a "random" benchmark. We also repeat the fidelity test for the 10 most closely correlated features to the original responses submitted. For each of the 10 features submitted, we find the feature that is most closely correlated in the data but is not already included in the list of drivers. We then repeat our fidelity test on these correlated features to test whether the original responses exhibit better fidelity performance than the most closely correlated features.

6.4.2 FIDELITY RESULTS

KEY FINDINGS

The results showed that SHAP-based responses performed well compared to benchmarks across all three models. Specifically, 14 of the 16 SHAP-based responses outperformed the benchmarks, while only two performed poorly. However, the degree to which the features chosen by SHAP together affect the predictions, as well as their reliability, varies across underlying prediction models. The performance of permutation-based and "Other" tools was mixed. Despite the good performance of SHAP-based tools, the study suggests two considerations when choosing a specific tool for model risk management. First, due to the tendency of complex models with a large number of features to allocate explanatory power among highly correlated features, it is important to group highly correlated features to gain a better understanding of the model's behavior. Second, different SHAP implementations in conjunction with different sampling methods could lead to variations in the response. Therefore, it is important to evaluate a specific implementation before deploying a new SHAP-based tool for model risk management.

Baseline Models Figure 14 presents the results for the perturbation fidelity test. Each row in the figure corresponds to a different model. Within each row, the left panel displays the fidelity performance of the diagnostic tools, and

[60] In our experiment, we added a second step to our perturbation test that involves finding the nearest neighbor of a perturbed applicant and comparing their risk levels. This second step was introduced to address the possibility that our perturbation scheme might generate unrealistic loan applicants. Although our scheme respects logical bounds in the data, such as preventing loan balances from becoming negative (see Appendix C), perturbing 10 features simultaneously could still result in an applicant that does not resemble any actual individual in our dataset.

To mitigate this issue, we identify a "nearest neighbor" for each perturbed applicant. This is an individual in our credit bureau data who has not been perturbed but has the most similar feature values to the perturbed applicant across the 10 features identified by the response we are evaluating. The idea is to find an existing applicant who resembles the perturbed applicant as closely as possible.

However, this second step's success is not guaranteed due to the curse of dimensionality and the inability to control the directions of the features involved. First, finding a neighbor that is "close" to the perturbed test case in a 10-dimensional space is challenging, as the expected size of the 10-dimensional cube containing the nearest neighbor is 0.4/1. Second, there is no assurance that the induced changes match the intended perturbation directions. The directions identified for each of the top 10 features differ depending on the diagnostic tools used in the responses. The data correlation structure may lead to a nearest neighbor having one or more features going in the opposite direction, resulting in uncertainty about the predicted default rate's change direction.

Therefore, it is crucial to measure match quality by examining the difference in the predicted default rate between the perturbed applicant and their nearest neighbor. Our analysis revealed that the match quality was indeed poor (see Figure E.1), leading us to conclude that this second step cannot effectively mitigate the risks associated with the perturbation test in the context of identifying drivers of global model behavior.

the right panel displays the fidelity performance of the correlated benchmarks. The fidelity performance is measured as the difference in the model prediction for the original applicant and the model prediction for the perturbed applicant. The result for the random benchmark is displayed in each panel for reference. Since in each test the top 10 features identified as the most important drivers of global model behavior are perturbed in the direction that reduces predicted default rates, positive differences that are on average larger than the benchmarks indicate that the diagnostic tool effectively identifies important model drivers, with larger differences indicating higher fidelity.

The box plots in Figure 14 summarize the distribution of the differences across all applicants in the data. Each box corresponds to a different diagnostic tool, identified by the type of software used. The green triangle represents the average difference in predicted default rates, while the box around the triangle shows the 25th and 75th percentiles. To preserve anonymity, we do not reveal the names of individual companies.

To determine which diagnostic tools exhibit high fidelity, we use two criteria. First, a tool is considered to have high fidelity if perturbing features suggested by the tool induce a change in predicted defaults that is larger than the random benchmark. This is indicated in Figure 14 by the green triangle and the box around it being above the horizontal line at zero and higher than the box labeled "random." Second, high fidelity requires the change in predicted defaults to be larger than that induced by perturbing the ten most closely correlated features. This is shown by the box around the green triangle in the left panel being higher than the corresponding box in the right panel.

For the logistic model, both SHAP and permutation-based responses perform well compared to the benchmark. The responses that use "Other" techniques perform worse than the random benchmark, however, and are also outperformed by the closely correlated features, suggesting that either features of little importance are picked or the directions in which the features are considered to have influenced the output are wrong, or both.

For the complex neural network model, SHAP-based responses perform well compared to the benchmark. Only one out of the three permutation-based responses outperforms both the random benchmark and the correlated benchmark. However, we note that the average changes induced by the perturbation in the case of complex neural network is much smaller than in the case of logistic regression, which uses a total of 44 features. This is likely because the neural network combines predictive information across a large number of variables, with the impact of each individual variable (and even of ten variables together) being small.

For the XGBoost Model, a majority of the SHAP-based tools perform well compared to the benchmark in the fidelity test. However, the results are not as homogeneous for the SHAP-based tools. Four of the six SHAP-based tools outperform both the random benchmark and the correlated benchmark while the other two perform poorly. One out of the three perturbation-based tools and one out the two "Other" tools perform well compared to benchmark. As in the case of the complex neural network model and for the same reason, the average changes induced by the perturbation tend to be much smaller than in the case of the logistic regression.

In general, SHAP-based responses outperform the benchmark across all three models (logistic regression, complex neural network, and XGBoost). Specifically, 14 of the 16 SHAP-based responses across the three models outperform the random and correlated benchmarks while two perform poorly. In comparison to other techniques, the SHAP-based responses consistently exhibit fidelity, indicating their effectiveness in determining feature importance for these models. However, the degree to which the chosen features together affect the predictions is limited for neural networks, and the results seem somewhat less reliable for boosted trees.

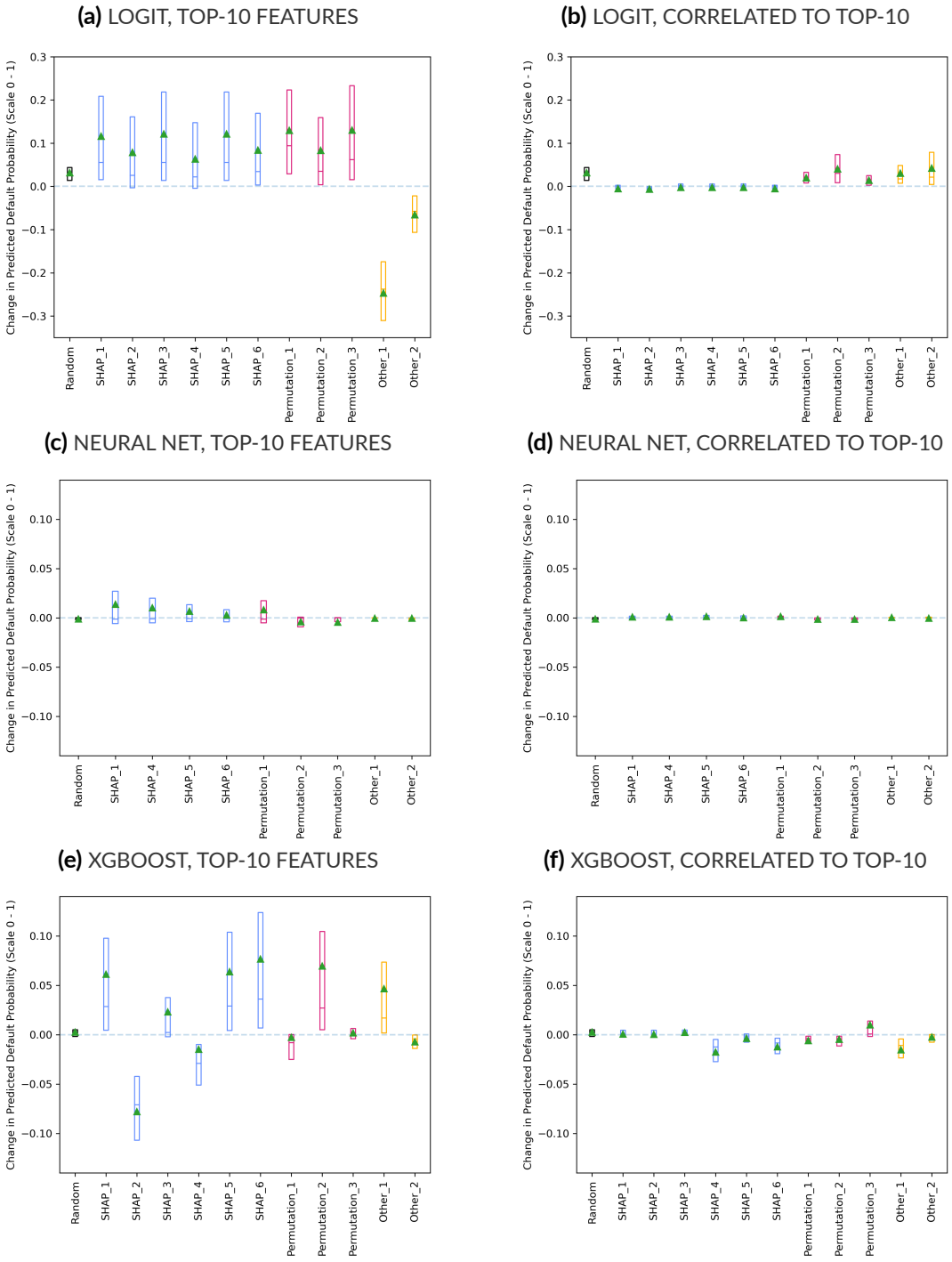
Notwithstanding the encouraging performance of SHAP-based tools, the study suggests two considerations when choosing a specific SHAP-based tool for model risk management. First, due to the tendency of complex models with a large number of features to allocate explanatory power among highly correlated features, a user should

consider grouping features that are highly correlated to gain a better understanding of the model's behavior. Focusing on just the top *n* variables may lead to under-appreciation of the importance of certain closely linked dimensions. Second, as noted in Section 6.3.3, the combination of different SHAP implementations and different sampling methods could lead to variations in the response. Therefore, it is desirable to evaluate the specific implementation for a particular use case before deploying a new SHAP-based tool for model risk management.

Company Models Figure 15 shows results for the fidelity test for the Company Models. Each company submitted one response for each model they built such that we are comparing both different diagnostic tools *and* different models in Figure 15.

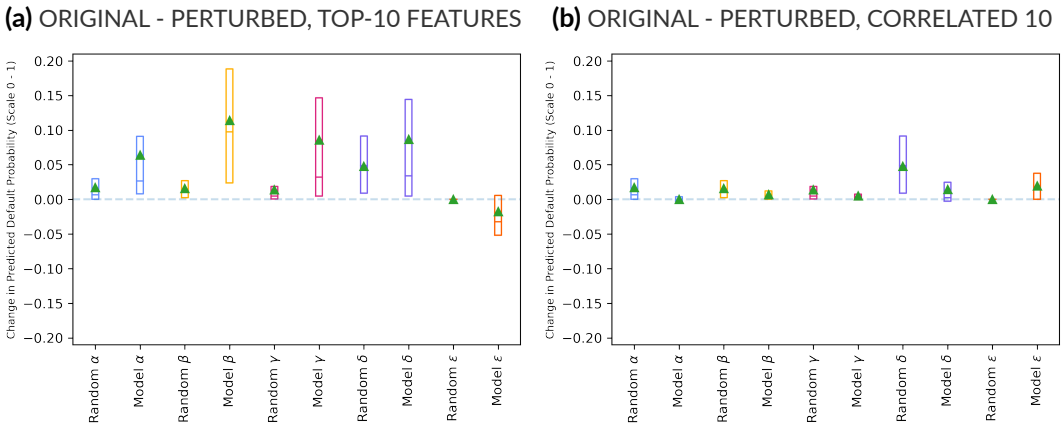
Responses submitted for four of the five Company Models perform well compared to benchmarks on the perturbation test in the sense that they induce favorable changes in predictions with magnitudes comparable to the high-fidelity tools for the XGBoost models. Responses for these four models also perform better than both the random benchmark and the correlated benchmark. Note that we only ran a single random draw for the Company Models. As a result, the random benchmarks in this test are less stable than in the case of the Baseline Models where we repeated the random draw one hundred times.

Figure 14: FIDELITY: PERTURBATION TEST



Note: This figure shows results of the perturbation tests the participating companies provided for the Baseline Models. Panel (a) shows the results for perturbing the top 10 features and Panel (b) shows the results for perturbing the 10 features most closely correlated with each of the top 10 features. These panels show the distribution of changes in predictions that result from the perturbation. The boxes display the 25th and 75th percentiles, while the horizontal lines display the medians. The green triangles represent the averages. The results for random benchmarks, which are generated by perturbing 10 features randomly drawn from all input features, are displayed in each panel alongside the results for each of the company models

Figure 15: FIDELITY TEST: COMPANY MODELS



Note: This figure shows results of the perturbation tests the participating companies provided for the Company Underwriting Models. The first column shows the results for perturbing the top 10 features, and the second column shows the results for perturbing the 10 features most closely correlated with each of the top 10 features. These panels show the distribution of changes in predictions from the perturbation. The boxes display the 25th and 75th percentiles, while the horizontal lines display the medians. The green triangles represent the averages. We received a single response for each Company Model (which we refer to by Greek letters to preserve anonymity). The results for random benchmarks, which are generated by perturbing 10 features randomly drawn from all input features, are displayed in both panels alongside the results for each of the company models.

6.5 EVALUATION: CONSISTENCY

The second dimension of our evaluation is consistency. We consider two types of consistency: (1) consistency of the drivers of global model behavior for the same model across different model diagnostic tools and (2) consistency of drivers of global model behavior provided by the same tool across different models. We first describe the consistency tests on which our analysis is based and then present results.

Consistency: Consistency across tools refers to how often two participating companies – or open-source tools – identify the same features as drivers of global model behavior. Consistency across models refers to how often a given model diagnostic tool identifies the same features as drivers of global model behavior across different underwriting models.

6.5.1 EVALUATION DESCRIPTION

Consistency *across tools* is not obviously desirable. Where model diagnostic tools analyzed in this study exhibit high fidelity, then consistency is likely to be a positive attribute in that we obtain similar answers regardless of the precise methodology used to describe the model's behavior. If, however, some tools perform worse than others in reliably identifying features that can help describe drivers of global model behavior, it is not clear that we would expect – *or want* – consistency. As an extreme example, consider the case where one tool simply randomly draws features. Clearly, consistency in this random draw would not improve the quality of information being provided by the tool. For this reason, we present results by the type of diagnostic tool to reflect the differences in fidelity we presented in the preceding section.

We evaluate consistency across tools by tabulating how often the same features are identified by different responses. In our baseline results, we only treat two responses as consistent if they identify exactly the same features. We also extend our results to consider broader feature families ("roll-up analysis") as well as the degree to which features are highly correlated with each other ("correlation analysis").

CONSISTENCY TESTS

We test consistency across tools by tabulating how often the same features are identified by different responses. Similarly, we test consistency across models by tabulating how often the same features are identified by the same tool across different models.

Consistency *across models* is helpful to gain insights into how diagnostic tools work but again is not necessarily a desirable property for users of machine learning underwriting models or these diagnostic tools. If we believe that different models learn similar fundamental (causal) relationships about the world and our goal is to identify the most important of those relationships, then consistency is desirable. In other words, if we are hoping to learn about relationships in the world, we would expect a good model diagnostic tool to consistently identify these facts about the world. If, however, we believe that different models learn different correlation patterns in the data and that consumer protection regulations are interested in why a particular model exhibits a specific behavior, then it is not clear that we would want (or expect) consistency across models. If models are learning different patterns, we would prefer the model diagnostic tool to correctly identify the pattern that drives behavior for that particular model. We

evaluate consistency across models by tabulating how often the same features are identified as drivers of model behavior by the same tool across different (but similar) underwriting models. In our baseline results, we only treat two responses as consistent if they identify exactly the same features.

6.5.2 CONSISTENCY: RESULTS

KEY FINDINGS

We find the following results with regard to consistency across tools and consistency across models. Tools that exhibit higher fidelity also have more drivers of global model behavior in common. The agreement grows as we allow for coarser feature families and incorporate information about the strength of statistical association between features. Tools that exhibit lower fidelity have almost no drivers in common with each other nor with the high-fidelity tools. High-fidelity tools have more agreement across models than low-fidelity tools.

Consistency across Responses

Roll-up Analysis Table 22 shows the baseline results for our consistency results across diagnostic tools and how these results vary as we consider broader feature categories. For the simple models, we find that a given pair of responses on average agrees on about 7 out of 10 features. For the Complex Baseline Models, a given pair of responses will on average agree on approximately 2 out of 10 features. This difference in consistency reflects that responses can only choose from 44 features in the simple models, while responses can choose from over 600 different features in the Complex Baseline Models. For the logistic model, the SHAP tools exhibit the highest agreement amongst each other – agreeing on a little over 8 out of 10 features on average. Permutation-based approaches agree on 7 out of 10 features on average. Both SHAP-based and permutation-based approaches agree on 3 out of 10 features on average for the Complex Baseline Models.

The roll-up analysis suggests that much of the disagreement between responses disappears once we consider broader feature families, namely the type of credit bureau information (roll-up 1), the type of loan product (roll-up 2), and the combination of the two (roll-up 3). For the logistic models, the roll-up analysis suggests that responses tend to agree on an additional 1 or 2 features out of 10 when considering broader categories. For example, the second and third roll-up categories lead to an average agreement of 8 to 9 out of 10 features. For the Complex Baseline Models, the roll-up leads to drastic increases in agreement. The first roll-up category increases the average agreement from 2 out of 10 features to over 5 out of 10 features. This increase is even more pronounced for the family of SHAP responses that now agree on 8 out of 10 features on average.

With both model types, some fundamental disagreement on drivers of global model behavior remains even after accounting for the roll-up.

Statistical Association Analysis We now consider the strength of the statistical association between two drivers of global model behavior in our consistency tests. Table 23 shows the results for the two metrics of the statistical association: the Pearson correlation coefficient and the mutual information metric.

Our results confirm the patterns observed for the roll-up analysis. Overall, we find that the statistical association across responses exceeds the benchmark we obtain by randomly choosing pairs of features in the data – with the

Table 22: MRM: CONSISTENCY WITH ROLL-UP

(a) PANEL A: SIMPLE BASELINE MODELS

	N	Baseline	Roll-up 1	Roll-up 2	Roll-up 3
All	55	6.75	8.22	9.11	7.98
SHAP	15	8.47	8.80	9.13	8.80
Permutation	3	7.33	8.00	10.00	8.00
Other	1	3.00	7.00	8.00	6.00

(b) PANEL B: COMPLEX BASELINE MODELS

	N	Baseline	Roll-up 1	Roll-up 2	Roll-up 3
All	91	1.98	5.37	7.41	5.02
SHAP	21	2.67	8.00	8.33	7.42
Permutation	6	2.67	4.00	6.33	3.83
Other	2	1.00	3.00	7.00	2.50

Note: The table shows results for the consistency test across diagnostic tools for the baseline feature-level analysis as well as aggregated across the roll-up categories. Each number represents the number of drivers a response pair has in common. 0 indicates that they have no drivers in common. 10 indicates that they have all 10 drivers in common. Panel A shows results for the Simple Baseline Models and panel B aggregates across the XGBoost and neural network models. The first column shows the baseline results (no roll-up), the remaining columns shows the three roll-up schemes discussed in the text. Each column shows the number of features a pair of responses has in common - averaged across all response-pairs in that group.

exception of the responses that use either LIME or counterfactual tools for the Complex Baseline Models. For example, the average feature correlation across all responses pairs is 25% for Complex Baseline Models compared to 17% if we randomly draw two sets of features. The feature correlations are higher for both SHAP-based and permutation-based response-pairs, namely 28% and 32%, respectively. Tools using permutation importance display the highest feature associations for both logistic and Complex Baseline Models.

Consistency across models We now consider a second type of consistency: How similar are responses that a company suggests across different models? Table 24 shows the results. SHAP-based responses show the highest consistency across models with 5.5 out of 10 features in common for Simple Baseline Models and 3.4 out of 10 features in common for Complex Baseline Models. Permutation-based responses are next with an average agreement of 3.2 out of 10 features for Simple Baseline Models and 2.2 out of 10 features for Complex Baseline Models. The other two types of tools (LIME and counterfactual explanations) show almost no overlap across models.

Table 23: MRM: CONSISTENCY WITH CORRELATION ANALYSIS

(a) PANEL A: SIMPLE BASELINE MODELS

		Pearson correlation coefficient				Mutual Information			
	N	Mean	Min	Max	Std	Mean	Min	Max	Std
All	55	47.48	29.92	57.62	7.74	26.28	16.02	35.03	4.77
SHAP	15	47.92	38.82	57.27	6.86	26.45	21.44	30.71	3.19
Permutation	3	54.43	52.12	56.84	2.36	30.71	28.27	32.76	2.27
Other	1	36.11	36.11	36.11		21.69	21.69	21.69	
Random Benchmark		29.08			33.54	17.05			23.4

(b) PANEL B: COMPLEX BASELINE MODELS

		Pearson correlation coefficient				Mutual Information			
	N	Mean	Min	Max	Std Dev	Mean	Min	Max	Std Dev
All	91	24.5	11.08	47.48	7.09	15.15	4.43	32.55	5.05
SHAP	21	27.82	18.7	34.17	4.44	18.73	12.54	23.18	3.08
Permutation	6	31.56	22.68	43.56	7.00	17.41	11.14	21.45	3.70
Other	2	13.33	11.84	14.83	2.11	8.76	8.74	8.77	2.22
Random benchmark		17.06			19.88	9.88			18.89

Note: The table shows results for the strength of the statistical association between the drivers of global model behavior provided by different responses. The two metrics are the Pearson correlation coefficient and a measure of mutual information described in the text. Both metrics range from 0-100 with zero indicating no statistical association and 100 indicating perfect association. Random benchmark refers to the statistical association of 100 (44 for the Simple Baseline Models) randomly chosen features in the data. Panel A shows results for the Simple Baseline Models, and panel B aggregates across the XGBoost and neural network models. We aggregate responses by type of diagnostic tool used. N refers to the number of response-pairs (*e.g.*, company A and company B constitute one pair).

Table 24: MRM: CONSISTENCY ACROSS MODELS

(a) PANEL A: SIMPLE BASELINE MODELS

	Average	Min	Max
SHAP	5.50	2.00	10.00
Permutation	3.17	1.00	5.00
Other	0.50	0.00	1.00

(b) PANEL B: COMPLEX BASELINE MODELS

	Average	Min	Max
SHAP	3.43	2.00	5.00
Permutation	2.17	0.00	4.00
Other	1.50	0.00	3.00

Note: The table shows results for consistency tests for the drivers of global model behavior across prediction models. Panel A shows results for the simple Baseline Models while Panel B shows results for the complex Baseline Models. The first column shows an average across diagnostic tools. This average is created by first computing the number of features that overlap in a given tool's response for two models and then taking an average across tools. 0 indicates that tools provide completely different responses for different models while 10 indicates the tools provide identical responses for different models. Min and max are computed similarly. The types of diagnostic tools are described in the text.

6.6 EVALUATION: USABILITY

The final component of our evaluation relates to the usability of information provided by the diagnostic tools, that is, the ability to identify information that enables lenders to comply with the goals and purposes of model risk management. In particular, we evaluate whether diagnostic tools can accurately diagnose the sources of changes in model behavior on an out-of-time data set. Our analysis centers on whether diagnostic tools can help anticipate the answers to the following two questions: (1) How have the predictions of the model changed? (2) How has the performance of the model changed?

Usability: the ability to identify information that enables lenders to comply with the goals and purposes of prudential regulation. In particular, the ability of a tool to identify sources of changes in model behavior in an out-of-time context.

Both questions are important for model risk management because changes in model performance or predictions might lead to undesirable changes in the risk profile of the loan portfolio and potential financial losses. A large shift in model predictions can lead to changes in the number of applicants who are approved at a given threshold and can shift the risk profile of a loan portfolio in unexpected ways. For example, the deployment context might have more applicants with a predicted probability such that they are marginally approved for a loan. This increase in marginally approved loan applicants could lead to an overall increase in the risk of the loan portfolio. A deterioration in model performance might lead to larger hidden risks as more high-risk applicants will erroneously be predicted to present low risk of default by the underwriting model.

In both cases, it is important to be able to diagnose what features in the model are driving the change in model behavior to evaluate the severity of the problem, to mitigate the problem by changing the model, and to anticipate distortions in different contexts.

6.6.1 EVALUATION DESCRIPTION

Our evaluation focused on the two types of relevant model changes that can occur in an out-of-time (or "deployment") context: a change in the distribution of the model's predictions and a change in the model's predictive performance.

Out-of-Time Data We created the out-of-time data by sampling credit card applicants from different time periods. Our baseline data uses 2012 data while our deployment data mixes data from 2010 and 2015. This strategy implies that we do not deliberately induce any feature or model shift in the data. Rather, we allow the deployment data to reflect any changes that would have occurred when deploying our Baseline Models in a different time period.

USABILITY TESTS

Our usability tests ask whether diagnostic tools can identify the sources of (1) a change in the distribution of model predictions and (2) a change in model performance in an out-of-time ("deployment") context. We perform two types of tests. The first type of test creates a new hypothetical dataset that matches the distribution of the 10 features named in a response to the one observed in the new deployment dataset. High fidelity corresponds to the distribution of model predictions (and model performance) in this new hypothetical dataset closely matching the distribution (and model performance) observed in the actual deployment data. The second type of test creates proxy models for the Baseline Underwriting Models using only the 10 features named in a response. High fidelity corresponds to the proxy model producing prediction performance reductions that are similar in magnitude to the predictive performance reduction of the Baseline Underwriting Models from the baseline to the deployment data.

Model predictions Changes in model predictions are driven by a shift in the underlying features, a phenomenon often referred to as data shift or feature shift. Our evaluation asks whether diagnostic tools can identify potential sources of feature shift. Our evaluation is based on the following test. For each response, that is, the 10 features identified as important for driving the change in model behavior, we create a new, hypothetical dataset that pretends that only those 10 features have changed. In order to create this hypothetical dataset, we start with the original training data but sample loan applicants such that the distribution of the 10 features matches that of the out-of-time (or "deployment") data. Our procedure first bins each of the 10 features, then computes the distribution of these binned features on the deployment data, and finally samples applicants in the training data to match the distribution on the deployment data.

We then compute model predictions on this new hypothetical data. If the original response accurately diagnosed the sources of feature shift, we should find that the change in model predictions on the new hypothetical dataset closely resembles that observed on the actual deployment data.

As in our fidelity tests, we repeat this test 100 times for 10 randomly chosen features. High-performing diagnostic tools should perform better than this random benchmark.

Note that we cannot perform this test for the company models since our test requires re-computing model predictions on our hypothetical datasets. Since we did not have access to the underlying company models, we could only perform this task for the Baseline Models.

Model performance Changes in model performance can be driven *both* by what we call data shift and by what we call model shift. Data shift occurs when the composition of the population to which a model is applied changes. Model shifts refers to a change in the relationship between the outcome we are trying to predict, that is, loan default, and the features in the model. For example, the role that credit utilization plays in predicting default might switch from lowering default predictions in good economic times (when it signals the ability to access credit) to increasing default prediction in bad economic times (when it signals liquidity challenges for the household).

To assess the ability of diagnostic tools to identify potential culprits for model performance deteriorations, we run two distinct tests. The first test is identical to the test we use to evaluate changes in model predictions. The difference is that we now focus on changes in model performance metrics on the hypothetical data set. Note that we cannot perform this test for the company models since our test requires re-computing model predictions on our

hypothetical datasets. Since we did not have access to the underlying company models, we could only perform this task for the Baseline Models.

The second test incorporates the effect of model shift in addition to the effect of data shifts. This test relies on building proxy models that allow us to assess how much model performance change is manifested by a proxy model using only the 10 features identified in a given response. We proceed as follows. For a given set of 10 features identified as most important drivers of the change in model performance, we project the model's prediction in the training data on the 10 features using a cross-validated XGBoost model (the "proxy model"). We then evaluate this proxy model on both baseline test and deployment test data. If the proxy model's performance deterioration on the deployment resembles that of the full model, we conclude that the 10 features could potentially contribute a lot to the model performance deterioration once model shift is also taken into account. To generate a benchmark for our results, we repeat the proxy model test for 100 runs of 10 randomly drawn features.

6.6.2 USABILITY: RESULTS

We now present results on the usability property. We first show how both Baseline and Company Underwriting Models perform on the deployment data. We then present results for the three tests designed to test whether diagnostic tools can account for possible data and model shift on the deployment data.

USABILITY RESULTS

All diagnostic tools were able to identify sources of changes in model behavior across all usability tests. Tools that were specifically designed to diagnose changes in model behavior (and had access to sample deployment data) did not systematically outperform the initial drivers of global model behavior that companies submitted for the first model risk management task. Our results suggest that tools that successfully identify drivers of global model behavior also hold useful information about what drives changes in model behavior in an out-of-time (or "deployment") context.

Model performance Table 25 shows how model performance and model predictions change in the deployment data. All models predict higher default rates on the deployment data and exhibit worse predictive performance. However, the differences are quantitatively small. The small differences partly reflect that we used real-world data for this exercise and our goal was not necessarily to engineer large differences between training and deployment data.

The first set of columns show the difference in performance metrics. Larger differences indicate worse performance on the deployment data. For example, we find that the AUC performance metric of the simple Baseline Models drop by 1.7 percentage points and the default rate among approved applicants increases by 0.6 percentage points.

We do not see any evidence that simpler models generalize better to out-of-time data. For the Baseline Models, the Complex Baseline Models exhibit smaller differences in both prediction changes and their predictive performance deteriorates less. For the Company Underwriting Models, simple and complex models exhibit similar drops in performance.

The second set of columns show how the predictions of the models change on the deployment data. We find that default predictions increase but their dispersion drops slightly. This change is in line with the fact that the actual

Table 25: MODEL PERFORMANCE AND PREDICTIONS ON DEPLOYMENT

	Change in Performance				Change in Predictions				
	AUC	MSE	Log Loss	Default| Approved	Mean	Std	p(25)	p(50)	p(75)
Baseline Models									
Simple models	0.017	0.011	0.032	0.006	0.008	-0.002	0.005	0.015	0.015
Complex models	0.014	0.010	0.031	0.008	0.006	-0.002	0.002	0.010	0.017
Company Models									
Simple models	0.015	0.009	0.027	0.007	0.006	-0.002	0.003	0.010	0.014
Complex models	0.016	0.009	0.027	0.008	0.005	-0.004	0.004	0.010	0.010
All Models									
Simple models	0.016	0.010	0.029	0.007	0.007	-0.002	0.004	0.012	0.015
Complex models	0.014	0.010	0.030	0.008	0.005	-0.003	0.003	0.010	0.015
Change in default rate	0.017								

Note: The table shows how both predictive performance and model predictions change on the deployment test data. All numbers are differences of deployment statistic minus the baseline statistic (with the exception of the AUC metric which is flipped for ease of interpretation). The first three columns show how three common measures of model predictive performance change. The remaining columns show how the predictions of the model change. Mean shows the difference in the average predicted default probabilities of the same model across the baseline and deployment data. Similarly, std, p(25), p(50), p(75) refer to differences in the standard deviation, and the 25th, 50th and 75th percentiles of predictions, respectively. The first two rows show results for the Baseline Model. The next two rows show results for prediction models built by participating companies. The next rows then aggregate Baseline and Company Models. The final row shows how actual default rates change in the deployment relative to the baseline data.

default rates increase on the deployment data (see the last row of Table 25).

Changes in Model Prediction: Test 1 The first three columns in Table 26 show results for the first usability test. Panel A shows results for the Simple Baseline Models while panel B shows results for the Complex Baseline models. We compare three sets of responses: (a) the ten most important features driving global model behavior that companies submitted for the first model risk management task – we refer to these responses as 'Baseline' in the tables; (b) the features whose distribution changes the most according across the two datasets – we refer to these responses as 'Feature shift' in the tables; and (c) the two responses that use the change in global feature importance scores to identify drivers of change in model behavior. We refer to these as 'Targeted' in the table. We further disaggregate the baseline responses by type of diagnostic tool. Further, we show the best and worst performing tools among categories of diagnostic tools where we observe large variation in performance. The last row labelled 'Random' refers to the random benchmark that performs the test 100 times on 10 randomly chosen features.

Each number in the table represents a ratio. That is, we take the difference of some evaluation metric between the hypothetical data and the baseline training data and divide it by the difference of the same metric between the deployment data and the baseline training data. A ratio equal to 1 suggests that the changes observed in our hypothetical data align perfectly with the changes we observe in the deployment data relative to the baseline data. A ratio less than 1 suggests that the changes observed in our hypothetical data are less extreme than the changes we observe in the deployment data. A ratio greater than 1 suggests that the changes observed in our hypothetical data are more extreme than those observed in the deployment data. It is important to note that these ratios merely indicate the degree of alignment between the hypothetical and observed data, and do not definitively establish the underlying causes of these changes. A ratio closer to 1 indicates that a diagnostic tool suggests drivers that together capture the actual change in model performance well. But note that even ratios close to 1 do not necessarily imply a causal relationship between these factors and the changes observed in model performance.

Across all model types, all three types of responses significantly outperform the random benchmark. In other words, despite taking very different approaches, all three types of approaches provide useful information about what could have contributed to the change in model predictions from training to deployment.

For Simple Baseline Models, the feature shift responses show the greatest alignment, corresponding to 94% of the change in average predictions and 33% of the change in the population-stability index (PSI). The best baseline responses demonstrate similar alignment, corresponding to over 90% of the change in average predictions and close to 30% of the PSI. However, some baseline tools demonstrate significantly less alignment corresponding to only 23% of the change in average predictions and 3% of the change in PSI.

For Complex Baseline Models, responses vary, sometimes exceeding the observed changes in average predictions or undershooting them. There is a set of responses (including the feature shift and targeted responses as well as some of the baseline responses) that tend to produce a more extreme change in model predictions than what is actually observed in the data (ratio above 1). These responses appear to correctly identify sources of changing predictions, however the 10 identified features do not tell the full story. Rather, there are other features in the model that counteract the effect of the 10 identified features. In contrast, another set of baseline responses produce changes that are only a small fraction of the observed changes – corresponding to just 11% to 15% of the observed changes in predictions. This set of responses did not appear to be able to identify potential drivers of observed changes in model predictions.

Table 26: MRM: USABILITY TEST PART I

(a) PANEL A: SIMPLE BASELINE MODELS

	Change in probabilities			Change in performance	
	Mean(PD)	Std(PD)	PSI	AUC	Log Loss
Baseline					
SHAP	0.91	1.39	0.28	0.29	0.40
Best	0.92	1.34	0.29	0.40	0.43
Worst	0.90	1.58	0.26	0.13	0.33
Other	0.54	0.89	0.15	0.02	0.10
Best	0.86	1.21	0.27	0.11	0.18
Worst	0.23	0.57	0.03	0.02	0.07
Permutation Importance	0.67	1.03	0.19	0.20	0.29
Feature shift	0.94	1.30	0.33	0.36	0.43
Targeted	2.17	9.40	23.92	0.97	0.77
Random	0.04	0.31	0.03	0.04	0.09

(b) PANEL B: COMPLEX BASELINE MODELS

	Change in probabilities			Change in performance	
Complex models	Mean(PD)	Std(PD)	PSI	AUC	Log Loss
Baseline					
SHAP	1.47	1.12	0.66	0.26	0.38
Best	1.95	1.30	0.60	0.31	0.45
Worst	0.39	0.50	0.05	0.24	0.22
Other	0.15	0.59	0.14	0.05	0.11
Best	0.18	0.59	0.23	0.09	0.14
Worst	0.11	0.59	0.05	0.01	0.09
Permutation Importance	0.62	0.26	0.55	0.18	0.24
Feature shift	3.02	2.29	1.81	0.36	0.61
Targeted	0.89	12.44	5.79	0.71	0.70
Random	0.04	0.31	0.22	0.09	0.04

Note: The table shows results for the first and second usability tests. The first three columns show results for the first test (can diagnostic tools account for changes in model predictions) while the last two columns show results for the second test (can diagnostic tools account for changes in model performance). Each number in the table represents a ratio. That is, we take the difference of some evaluation metric between the hypothetical data and the baseline training data and divide it by the difference of the same metric between the deployment data and the baseline training data. A ratio equal to 1 suggests that our hypothetical data can account for the entire change we observe in the deployment data relative to the baseline data. Baseline refers to the 11 responses submitted to Task 1 (global feature importance). Feature shift refers to the 3 responses that identify the 10 features whose distribution has shifted the most. Targeted refers to 2 responses that use changes in feature importance scores to identify drivers of changes in model behavior. Mean(PD) and Std(PD) refer to the mean and standard deviation of predicted probabilities. PSI refers to the Population-Stability-Index. AUC and log loss are common performance metrics.

Changes in Model Performance: Test 2 The last two columns in Table 26 shows results for the second usability test. We again compare three sets of responses. Similar to Test 1, the ratios displayed in the tables are again constructed as the difference of some evaluation metric between the hypothetical data and the baseline test data divided by the difference of the same metric between the deployment test data and the baseline test data. However, we now evaluate whether a diagnostic tool can effectively identify potential drivers of for the change in model *performance*, instead of the change in model predictions as in the preceding section. This ratio is meant to show alignment between the hypothetical data and the deployment data relative to the baseline data. A ratio equal to 1 suggests that changes observed in our hypothetical data align perfectly with the changes we observe in the deployment data relative to the baseline data. A ratio less than 1 suggests that the changes observed in our hypothetical data are less extreme than the changes we observe in the deployment data. A ratio greater than 1 suggests that the changes observed in our hypothetical data are more extreme than those observed in the deployment data. These ratios merely suggest the degree of alignment between the hypothetical and observed data, but do not conclusively establish the underlying causes of these changes. A ratio closer to 1 indicates that a diagnostic tool suggests drivers that together capture the actual change in model performance well. But note that even ratios close to 1 do not necessarily imply a causal relationship between these factors and the changes observed in model performance.

Much like in Test 1, all three types of responses comfortably outperform the random benchmark for both model types. However, unlike in Test 1, there is now a clear performance ranking, with targeted responses performing the best, followed by feature shift responses, and the baseline responses last. These findings suggest that all three responses deliver useful information in identifying potential drivers that could have contributed to the drop in model performance on the deployment data. However, responses generated by tools that are explicitly designed for this purpose now outperform the baseline responses.

Changes in Model Performance: Test 3 Table 27 show results for the third usability test which relies on building proxy models of the underlying Baseline Underwriting Models using only the 10 features provided by a diagnostic tool. Each number in the table represents a ratio of the performance difference of a proxy model between deployment and baseline test data to the performance difference of the Baseline Underwriting Model. A ratio equal to 1 suggests that the performance change observed in a proxy model aligns perfectly with the change observed in the corresponding Baseline Model. A ratio less than 1 suggests that the change observed in a proxy model is less extreme than the change we observe in the corresponding Baseline Model. A ratio greater than 1 suggests that the change observed in a proxy model is more extreme than the change we observe in the corresponding Baseline Model. Similar to in Tests 1 and 2, these ratios merely suggest the degree of alignment between the proxy models and Baseline Models, but do not conclusively establish the underlying causes of these changes. A higher ratio, either less than or exceeding 1, indicates that a diagnostic tool may be better at identifying the likely contributing factors, but it does not confirm a causal relationship between these factors and the changes observed in model performance.

In contrast to the second test in Table 26 which captures only the effect of feature shift on changes in model performance, this test incorporates in addition the effect of model shift. The differences between test 2 and test 3 is thus informative about the extent to which the 10 features identified in a response help explain the sources of model shift.

Overall, we observe the proxy models produce performance changes either on par or larger than the performance changes of the corresponding Baseline Models. This is true across both AUC and log loss measures of predictive performance.

Again, all three types of responses comfortably outperform the random benchmark across all model types sug-

gesting that all responses deliver useful information about the sources of changes in model performance in the deployment setting we study. Note that the random benchmarks produce performance changes that correspond to 65 to 87% of the change observed for the Baseline Models, indicating that some of this performance difference may originate from the way the test was designed.

Unlike in the previous test, no clear winner (or loser) emerges from the usability test. This result partly reflects that the ordering in performance across the two performance metrics (AUC and log loss) does not perfectly coincide. Both targeted and feature shift responses perform better than the random benchmarks for both Simple and Complex Baseline Models. Specifically, the proxy models using only the top 10 features produce performance changes that are close to 100% of the observed changes in model performance when the Simple Baseline Models are applied to the deployment data, and slightly more than 100% of the change in model performance when the Complex Baseline Models are applied to the deployment data. Notably, however, many of the baseline diagnostic tools offer comparable performance. Overall, we do not observe much gain from targeted responses that had access to a sample of deployment data over the initial drivers of global model behavior submitted by the companies.

Table 27: MRM: USABILITY TEST PART II

(a) PANEL A: SIMPLE BASELINE MODELS

		Change in performance AUC	Log Loss
Baseline		1.067	1.050
SHAP		0.825	1.010
	Best	0.888	0.974
	Worst	0.796	1.020
Other		0.832	0.972
	Best	0.886	0.997
	Worst	0.787	0.967
Permutation Importance		0.839	0.972
Feature shift		0.962	0.986
Targeted		0.860	0.949
Random		0.654	0.704

(b) PANEL B: COMPLEX BASELINE MODELS

		Change in performance AUC	Log Loss
Baseline		0.582	1.002
SHAP		1.212	0.661
	Best	1.343	1.055
	Worst	1.061	0.316
Other		1.223	1.007
	Best	1.525	1.081
	Worst	1.007	0.956
Permutation Importance		0.988	0.946
Feature shift		2.306	1.203
Targeted		1.109	1.301
Random		0.719	0.870

Note: The table shows results for the third usability tests based on building proxy models with only the 10 features named in a given response. Each number in the table represents a ratio. That is, we take the difference of some evaluation metric of the proxy model between baseline and deployment data and divide it by the difference of the same metric evaluated for the underlying Baseline Model between the deployment data and the baseline data. All evaluations are performed on test data. A ratio equal to 1 suggests that our hypothetical data can account for the entire change we observe in the deployment data relative to the baseline data. Baseline refers to the 11 responses submitted to Task 1 (global feature importance). Feature shift refers to the 3 responses that identify the 10 features whose distribution has shifted the most. Targeted refers to 2 responses that use changes in feature importance scores to identify drivers of changes in model behavior AUC and log loss are common performance metrics.

7 CONCLUSION

As the use of machine learning underwriting models continues to expand, lenders and their regulators will gain institutional experience with, and confidence in, the kinds of model diagnostic tools evaluated herein as well as the underlying models themselves. For now, we hope that this evaluation provides insight as stakeholders make choices about how to develop, implement, and manage fair and responsible machine learning underwriting models. This analysis is intended to help stakeholders as they reconsider current expectations and practices and the articulation of policy and market practices.

Key Contributions While encouraging, our evaluation suggests that there are no universal or "one size fits all" model diagnostic tools that lenders can use to help them explain, understand, and manage all aspects of machine learning underwriting models. Rather, the specific context of the model's operation is critical to selecting the right tool and implementing that tool correctly. Information about a model that is appropriate and useful to respond to one regulatory requirement may not necessarily be similarly responsive to a different requirement. As a result, lenders must make careful judgments when they select and implement tools designed to help them understand and manage machine learning underwriting models. The same is true when lenders make decisions about how to respond to and use the information that those tools produce. Thus, the responsible use of model diagnostic tools adds another dimension to the many consequential decisions that lenders must make – and be responsible for – when designing, implementing, and operating machine learning underwriting models.

To that end, we offer two primary contributions:

1. A framework for evaluating the quality and usability of information produced about machine learning models' behavior. Notably, this evaluation methodology enables the assessment of specific tools without access to "ground truth" explanations of the model's behavior, which opens up the possibility for evaluating the capabilities, limitations, and performance of model diagnostics in a range of applied contexts like those considered in this study.

2. A rigorous case study regarding the use of various model diagnostic tools in the context of specific legal and regulatory requirements that prompt lenders and their regulators to address transparency challenges associated with machine learning models and to continue to explore the potential of using more complex models responsibly and fairly.

Both of these contributions may be relevant to other sectors, such as medicine, criminal justice, and employment, where AI and machine learning models are being used to help make highly consequential decisions. Implementing machine learning in the context of extending consumer credit brings into play regulatory requirements focused on promoting responsible risk-taking and providing consumers broad, non-discriminatory access. Efforts to ensure that emerging uses of advanced prediction technology – to explain, understand, and debias machine learning models – satisfy these requirements gives technologies that gain acceptance in finance disproportionate potential to shape how other sectors answer the same questions.

While this paper represents the final findings in this evaluation, our results point to further research on the explainability and fairness of machine learning when used to extend consumer credit and in other financial services use cases. Avenues for further exploration by practitioners, academics, policymakers, and other stakeholders include:

1. Deeper evaluation of specific implementation choices that make one model diagnostic tool higher performing on certain tasks – such as, identifying whether and how the definition of the baseline set to which a rejected applicant is compared (such as, all approved borrowers, the top cohort of approved borrowers, or the bottom cohort of borrowers) affects the quality of information given to consumers on an adverse action notice.

2. Deeper evaluation of specific debiasing approaches within the category of more automated methods to illuminate promising methodologies for mitigating bias in machine learning underwriting models and the specific choices that lenders make when deploying those methods, including whether and how protected class characteristics can be responsibly used to improve the fairness of credit decisions.

3. Assessing with rigor the transparency costs related to the use of more complex machine learning underwriting models to the related tradeoffs that lenders make to improve the transparency of underwriting models.

4. Evaluating whether inclusion of additional types of underwriting data affects the fairness and inclusiveness of credit decisions as well as the performance, capabilities, and limitations of the kinds of model diagnostic tools evaluated herein.

5. Examining whether the performance-fairness tradeoffs identified in our less discriminatory model searches are robust – that is, whether there is a band in which lenders can improve the fairness of models without incurring significant loss of performance and how potential performance tradeoffs distribute across populations of interest.

6. Reanalysis of the consistency data presented herein showing that some disagreement remains in information provided by different tools about the same model (even after accounting for correlations) to classify the types of disagreements that persist and to consider whether those types of disagreements have a material effect on the regulatory compliance tasks considered in this evaluation.

7. Refinement of the assessment framework for evaluating capabilities, limitations, and performance of model diagnostic tools. Refinement may consider, for example, whether different or additional qualities warrant investigation to help identify when information from such tools can be trusted and used in high-stakes contexts or whether different tests can help establish how well individual tools perform as to qualities assessed in the framework.

8 LITERATURE

Aumann, R. J. and L. S. Shapley (2015): "Values of non-atomic games," in *Values of Non-Atomic Games*, Princeton University Press.

Avtar, R., R. Chakrabarti, K. Chatterji-Len, et al. (2021): "Unequal Distribution of Delinquencies by Gender, Race, and Education," Tech. rep., Federal Reserve Bank of New York.

Barocas, S., M. Hardt, and A. Narayanan (2017): "Fairness in machine learning," *Nips tutorial*, 1, 2.

Barocas, S. and A. D. Selbst (2016): "Big data's disparate impact," *Calif. L. Rev.*, 104, 671.

Bartlett, R., A. Morse, R. Stanton, and N. Wallace (2022): "Consumer-lending discrimination in the FinTech era," *Journal of Financial Economics*, 143, 30–56.

Blattner, L. and S. Nelson (2021): "How Costly Is Noise? Data and Disparities in Consumer Credit," .

Broniatowski, D. A. et al. (2021): "Psychological foundations of explainability and interpretability in artificial intelligence," *NIST: National Institute of Standards and Technology, US Department of Commerce*.

Chouldechova, A. (2017): "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big data*, 5, 153–163.

Coston, A., A. Rambachan, and A. Chouldechova (2021): "Characterizing fairness over the set of good models under selective labels," in *International Conference on Machine Learning*, PMLR, 2144–2155.

Dieber, J. and S. Kirrane (2020): "Why model why? Assessing the strengths and limitations of LIME," *arXiv preprint arXiv:2012.00093*.

Dwork, C., M. Hardt, T. Pitassi, O. Reingold, and R. Zemel (2012): "Fairness through awareness," in *Proceedings of the 3rd innovations in theoretical computer science conference*, 214–226.

Elliott, M. N., P. A. Morrison, A. Fremont, D. F. McCaffrey, P. Pantoja, and N. Lurie (2009): "Using the Census Bureau's surname list to improve estimates of race/ethnicity and associated disparities," *Health Services and Outcomes Research Methodology*, 9, 69–83.

Emmons, W. R. and L. R. Ricketts (2016): "The Demographics of Loan Delinquency: Tipping points or tip of the iceberg?" *Center for Household Financial Stability, Federal Reserve Bank of St. Louis (Oct. 2016)*.

Evans, C. (2017): "Keeping Fintech fair: Thinking about fair lending and UDAP risks," *Consumer Compliance Outlook*, 2, 1–9.

Ficklin, P. A., T. Pahl, and P. Watkins (2020): "Innovation spotlight: Providing adverse action notices when using AI/ML models," *Consumer Financial Protection Bureau*, July 7.

FinRegLab (2021): "The Use of Machine Learning for Credit Underwriting: Market and Data Science Context," .

Fuster, A., P. Goldsmith-Pinkham, T. Ramadorai, and A. Walther (2022): "Predictably unequal? The effects of machine learning on credit markets," *The Journal of Finance*, 77, 5–47.

Gill, N., P. Hall, K. Montgomery, and N. Schmidt (2020): "A Responsible Machine Learning Workflow with Focus on Interpretable Models, Post-hoc Explanation, and Discrimination Testing," *Information*, 11.

Gillis, T. B. (2022): "The input fallacy," *Minnesota Law Review, forthcoming*.

Gillis, T. B. and J. L. Spiess (2019): "Big data and discrimination," *The University of Chicago Law Review*, 86, 459–488.

Hall, P., B. Cox, S. Dickerson, A. Ravi Kannan, R. Kulkarni, and N. Schmidt (2021): "A United States Fair Lending Perspective on Machine Learning," *Frontiers in Artificial Intelligence*, 4, 78.

Hellman, D. (2020): "Measuring algorithmic fairness," *Virginia Law Review*, 106, 811–866.

Hutchinson, B. and M. Mitchell (2019): "50 years of test (un) fairness: Lessons for machine learning," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 49–58.

Kleinberg, J., J. Ludwig, S. Mullainathan, and A. Rambachan (2018a): "Algorithmic fairness," in *Aea papers and proceedings*, vol. 108, 22–27.

Kleinberg, J., J. Ludwig, S. Mullainathan, and C. R. Sunstein (2018b): "Discrimination in the Age of Algorithms," *Journal of Legal Analysis*, 10, 113–174.

Kleinberg, J., S. Mullainathan, and M. Raghavan (2016): "Inherent trade-offs in the fair determination of risk scores," *arXiv preprint arXiv:1609.05807*.

Krishna, S., T. Han, A. Gu, J. Pombra, S. Jabbari, S. Wu, and H. Lakkaraju (2022): "The Disagreement Problem in Explainable Machine Learning: A Practitioner's Perspective," *arXiv preprint arXiv:2202.01602*.

Kumar, I. E., S. Venkatasubramanian, C. Scheidegger, and S. Friedler (2020): "Problems with Shapley-value-based explanations as feature importance measures," in *International Conference on Machine Learning*, PMLR, 5491–5500.

Lundberg, S. M. and S.-I. Lee (2017): "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, 30.

Mateos, P. et al. (2014): "Names, ethnicity and Populations," *Advances in Spatial Science*.

Pessach, D. and E. Shmueli (2020): "Algorithmic fairness," *arXiv preprint arXiv:2001.09784*.

Race, P. (1805): "Ethnicity from the Sequence of Characters in a Name," *Gaurav Sood, Suriyan Laohaprapanon arXiv (2018-07-31) https://arxiv. org/abs*.

Ramakrishnan, R. (2021): "An Alternative to the Correlation Coefficient That Works For Numeric and Categorical Variables," *R Views*.

Rodolfa, K. T., H. Lamba, and R. Ghani (2021): "Empirical observation of negligible fairness–accuracy trade-offs in machine learning for public policy," *Nature Machine Intelligence*, 3, 896–904.

Schleich, M., Z. Geng, Y. Zhang, and D. Suciu (2021): "GeCo: Quality Counterfactual Explanations in Real Time," *CoRR*, abs/2101.01292.

Schmidt, N. and B. Stephens (2019): "An introduction to artificial intelligence and solutions to the problems of algorithmic discrimination," *arXiv preprint arXiv:1911.05755*.

Schwartz, R., A. Vassilev, K. Greene, L. Perine, A. Burt, P. Hall, et al. (2022): "Towards a Standard for Identifying and Managing Bias in Artificial Intelligence," .

Selbst, A. D. and S. Barocas (2018): "The intuitive appeal of explainable machines," *Fordham L. Rev.*, 87, 1085.

Shapley, L. S. (1951): "Notes on the n-Person Game—II: The Value of an n-Person Game," .

Slack, D., S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju (2019): "How can we fool LIME and SHAP? Adversarial Attacks on Post hoc Explanation Methods," .

Verma, S., K. Hines, and J. P. Dickerson (2021): "Amortized Generation of Sequential Counterfactual Explanations for Black-box Models," *CoRR*, abs/2106.03962.

Verma, S. and J. Rubin (2018): "Fairness definitions explained," in *2018 ieee/acm international workshop on software fairness (fairware)*, IEEE, 1–7.

A COMMON TERMS AND ACRONYMS

Adversarial debiasing: Adversarial models are models that can be used during training to debias machine learning models. In this context, adversarial models attempt to predict the protected class status of an individual based on the output of the underlying model, with the underlying model continuing to adjust until the ability of the adversary to correctly predict protected class characteristics diminishes to an appropriate point.

Adverse impact ratio (AIR): This metric represents the industry standard for evaluating disparities in a variety of contexts including credit and hiring. It is defined as the ratio of the acceptance rate for the minority group to the acceptance rate of the majority group. AIR values closer to 1 correspond to more parity.

Area under the curve (AUC): AUC provides an aggregate measure of performance across all possible classification thresholds. AUC can be interpreted as the probability that the model ranks a random positive example more highly than a random negative example.

Conditional statistical parity: Conditional statistical parity measures whether the likelihood of a predicted positive outcome is the same across protected classes, once a set of control variables has been accounted for. The variables are typically chosen because they have a close link to the outcome being predicted by the model, and thus increase the accuracy of its predictions.

Consistency: When applied to descriptions of model behavior, consistency refers to the degree to which tools identify the same drivers either for the same model or across models.

Disparate impact: Disparate impact is one of two theories for establishing legal liability for discrimination against classes of persons protected under the Equal Credit Opportunity Act (ECOA) or Fair Housing Act (FHA). It prohibits the use of facially neutral practices that have a disproportionately adverse effect on protected classes unless those practices meet a legitimate business need that cannot reasonably be achieved through alternative means with a less adverse effect.

Disparate treatment: Disparate treatment is one of two theories for establishing legal liability for discrimination against classes of persons protected under the Equal Credit Opportunity (ECOA) Act or Fair Housing Act (FHA). It prohibits treating individuals differently based on a protected characteristic. Establishing disparate treatment does not require any showing that the treatment was motivated by prejudice or a conscious intention to discriminate.

Equal Credit Opportunity Act (ECOA): The Equal Credit Opportunity Act of 1974 is a federal statute (codified at 15 U.S.C. § 1691 et seq.) that makes it unlawful for any creditor to discriminate against any applicant, with respect to any aspect of a credit transaction, on the basis of race, color, religion, national origin, sex, marital status, or age (provided the applicant has the capacity to contract); to the fact that all or part of the applicant's income derives from a public assistance program; or to the fact that the applicant has in good faith exercised any right under the Consumer Credit Protection Act. ECOA is implemented by the Consumer Financial Protection Bureau through Regulation B (codified at 12 C.F.R. Part 1002).

Explainability: In this report, model explainability refers to the ability of various stakeholders to understand how or why a particular decision was made or result was reached.

Explainability techniques: *Post hoc* explainability techniques are supplemental models, methods, and analyses used to improve the transparency of complex models. Since these tools are used after the model has been trained, they are often referred to as *post hoc* or indirect techniques. These methods do not generally affect the design or operation of the underlying model and can be used with a variety of machine learning model types.

Fair Credit Reporting Act (FCRA): The Fair Credit Reporting Act is a federal statute (codified at 15 U.S.C. § 1681 et seq.) enacted to protect consumers from the willful and/or negligent inclusion of inaccurate information in their credit reports and to promote the accuracy, fairness, and privacy of consumer information contained in the files of consumer reporting agencies. FCRA regulates the collection, dissemination, and use of consumer information for credit purposes as well as for activities such as employment, insurance, and housing. It is implemented by the Consumer Financial Protection Bureau through Regulation V (codified at 12 C.F.R. Part 1022).

False positive rates (FPR): The FPR refers to the fraction of non-defaults that are incorrectly predicted as defaults. FPR is a threshold-based metric that can be used to assess a model's fairness. It requires access to information about whether the borrower defaulted, instead of only taking approvals into consideration. Values closer to zero correspond to more parity (or less disparity).

Feature importance: Feature importance refers to how much impact an input variable has on the target prediction in a model. Various *post hoc* explainability techniques are designed to identify and quantify feature importance within more complex models.

Fidelity: When applied to descriptions of model behavior, fidelity refers the ability to reliably identify features that are relevant to a model's prediction.

Fitness-for-use: Fitness-for-use refers to the effectiveness of a model in serving its purpose, which can include model accuracy, fairness, and other factors, and the quality of the plan to appropriately manage risks related to operation of a particular model. Model risk management expectations require firms to determine that a model is fit for use prior to deployment.

Global explainability: Global explainability refers to the ability to identify a model's high-level decision-making processes and is relevant to evaluating a model's overall behavior and fitness-for-use.

Hyperparameter: Hyperparameters refer to aspects of a machine learning model that are not learned from the data, but rather are determined by model developers, such as the number of nodes in a decision tree. Hyperparameters can affect the predictiveness and explainability of the model and are often adjusted during model tuning.

Inherently interpretable models: An inherently interpretable model specifies the contribution that each input variable makes toward the output and enables stakeholders to understand its predictions without the use of secondary models, analyses, or methods. These models are also sometimes referred to as self-explanatory.

Interpretability: Model interpretability refers to the ability to understand a model's operations based largely on its formal notation and without reliance on secondary models, analyses, or methods. To be interpretable, a person should be able to infer the following: (1) the types of information or input variables that a model uses, (2) the relationship between the input variables and the model's predictions or outputs; and (3) the data conditions for which the model will return a specific result.

Local interpretable model-agnostic explanations (LIME): LIME is a feature importance explainability technique that uses local linear surrogate models around a particular data point to approximate a complex model's output. The resulting local surrogate models are used to both explain the model's behavior around individual data points and quantify feature importance for the overall model. LIME is generally used today as a baseline against which to compare the outputs and performance of other explainability tools or to generate insight into feature importance.

Linearity: Linearity is a property of models whereby changes in a particular input produce a consistent rate of change in the output.

Linear regression: Linear regression refers to a statistical technique that computes the best-fit linear relationship between input variables and a target variable.

Local explainability: Local explainability refers to the ability to identify the basis for specific decisions made by a model.

Model debiasing: Model debiasing refers to a range of methods to reduce bias in a model's predictions, either by transforming the input data, building a debiasing function into model training, or transforming a model's output. Debiasing techniques vary based on the model's use case, the data being used, model complexity, and other factors.

Monotonicity: Monotonicity refers to a relationship that is one-directional (*e.g.*, increasing the value of an input variable will always cause the output to increase or will always cause the output to decrease). Imposing monotonicity constraints can help model developers limit the complexity and improve the explainability of machine learning models while potentially distorting the true relationship between the input and output.

Non-threshold based metrics: The approaches to measuring model fairness use the underlying model predictions as opposed to discrete classifications or implied decisions of the model. Statistical or demographic parity, the standardized mean difference, and conditional statistical parity are illustrative non-threshold based metrics.

Perturbation: Perturbation is a technique used in various feature-based explainability methods that identifies key drivers of a model's prediction by assessing iteratively the effect on the model's prediction of a series of incremental changes to input data. For example, to identify what level debt-to-income an unsuccessful loan applicant would have needed to obtain a loan (assuming no changes in other credit characteristics), a perturbation-based method might consider the effect of a series of small changes on either side of the applicant's actual debt-to-income ratio to identify the point at which the model's predicted default was sufficient for approval.

Protected class: Like anti-discrimination statutes applicable in other areas, ECOA and FHA prohibit discrimination against people based on a common characteristic. Such characteristics include race, color, religion, national origin, sex, marital status, disability status, family status, or age (provided the applicant has the capacity to contract); reliance on a public assistance program; or in the good faith exercise good faith of any right under certain federal consumer financial laws.

Shapley Additive Explanations (SHAP): Shapley Additive Explanation is a feature importance explainability method that is used to explain complex models. SHAP does this by indicating the contributions of particular features in changing a model's outcome. It is similar to LIME in that it explains a model locally. This method measures feature importance by removing features from a data point and quantifying how much that effects the model's output.

Standardized mean difference: Standardized mean difference is a scaled version of statistical parity that is widely used by industry in fair lending compliance, as well as in other anti-discrimination contexts like employment. It is defined as the average difference in predictions between protected classes, divided by the standard deviation of the model predictions. The closer to zero, the more parity.

Statistical or demographic parity: Statistical or demographic parity is defined as the difference in the average predicted probabilities by protected class. The closer to zero, the more parity.

Threshold-based metrics: Threshold-based metrics apply a hypothetical approval cutoff to the predictions of model. These metrics focus on relevant outcomes by considering the approval threshold used in practice.

True positive rates (TPR): The TPR is a threshold-based metric that can assesses a model's fairness based on the fraction of defaults that are correctly predicted. It requires access to information about whether the borrower defaulted, instead of only taking approvals into consideration. Values closer to zero correspond to more parity or less disparity.

Usability: When applied to descriptions of model behavior, usability refers to the ability to identify information that helps users of the information accomplish particular tasks, such as lenders managing particular compliance tasks in accordance with the goals and purposes of the underlying regulation. Unlike fidelity and consistency, the exact definition of usability depends on the specific regulatory requirement in question.

B ADDITIONAL RESULTS

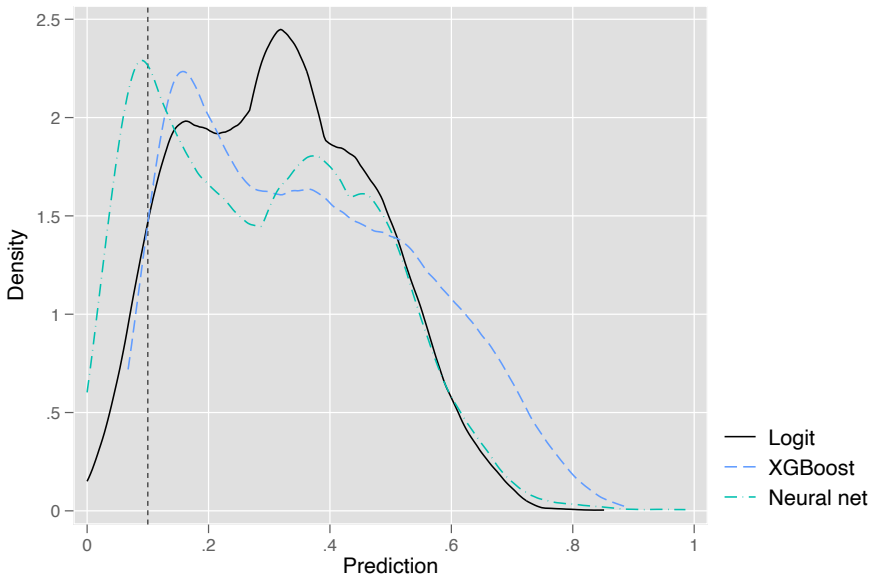
This appendix provides additional figures and tables.

Table B.1: ML HYPERPARAMETER TUNING OF BASELINE MODELS

Model	Score	Search Space	Best Model
XGBoost	Negative Log-Loss	max_depth: [1,2,3,..,11,12]	9
		min_child_weight: [1,2,3,..,9,10]	8
		n_estimators: [100, 500,1000]	1000
		learning_rate: [0.01, 0.1, 0.3]	0.01
		gamma: [0, 0.2, 0.4, 0.6, 0.8]	0.6
		max_delta_step: [0,1,5,10]	0
		subsample: [0.3, 0.5, 0.75, 1]	0.75
		colsample_bytree: [0.3, 0.5, 0.75, 1]	0.5
Neural Network	Binary Cross-Entropy	Batches: [5, 10, 20]	10
		Neurons: [10, 25, 50, 75, 100]	10

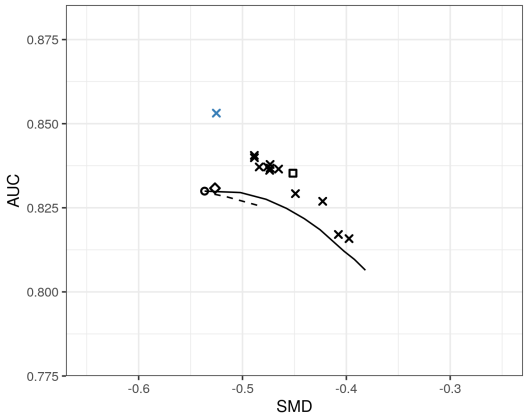
Note: The above table presents the final search space for each Baseline Model and chosen hyperparameters according to 5-fold cross-validation scores.

Figure B.1: DISTRIBUTION OF PREDICTION FOR REJECTED APPLICANTS

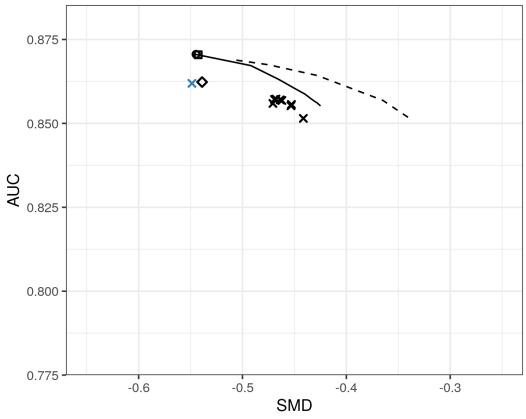


Note: The figure shows the distribution of model predictions for the 3000 rejected applicants considered in the Adverse Action Notice analysis.

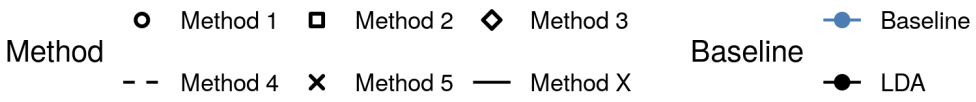
Figure B.2: LDA RESULTS STAGE I - BASELINE MODELS



(a) LOGIT - SMD

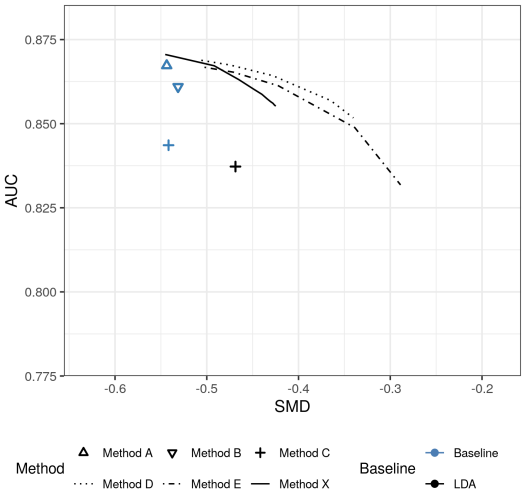


(b) XGBOOST - SMD



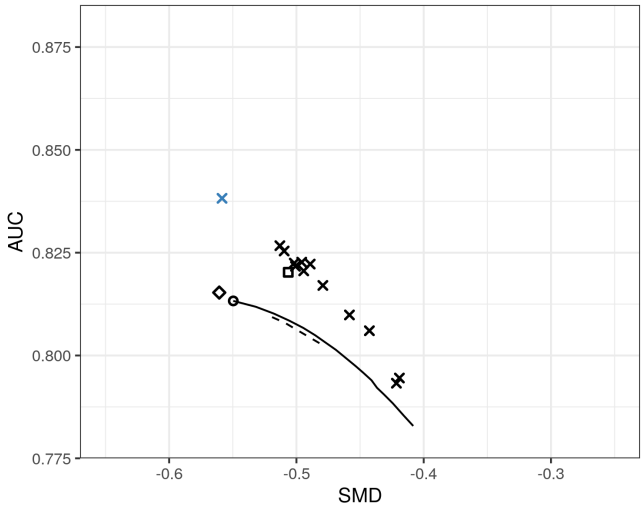
Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metric: the SMD metric with the sign flipped such that points further to the right represent models with less adverse impact. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure B.3: LDA RESULTS STAGE I - COMPANY MODELS

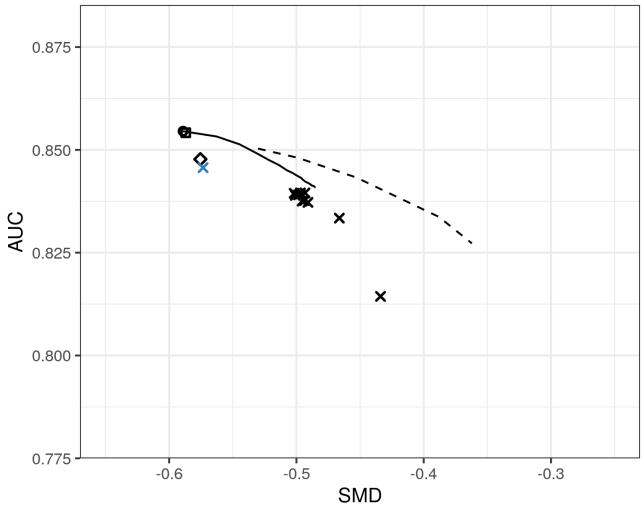


Notes. Each panel of the figure shows the performance of less discriminatory alternative (LDA) models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metric: the SMD metric with the sign flipped such that points further to the right represent models with less adverse impact. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure B.4: LDA RESULTS STAGE II - BASELINE MODELS



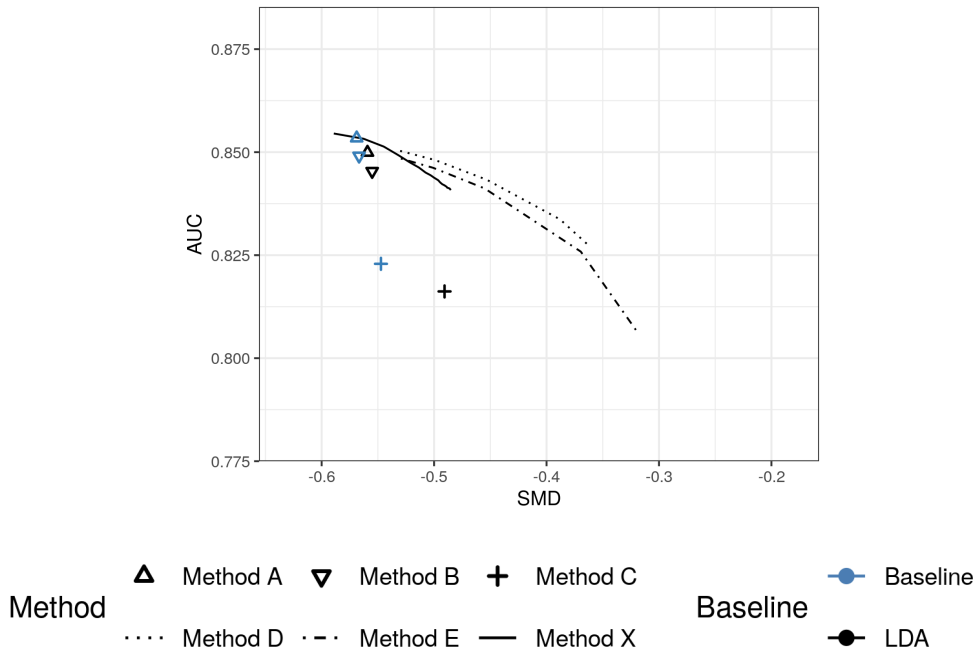
(a) LOGIT - SMD



(b) XGBOOST - SMD

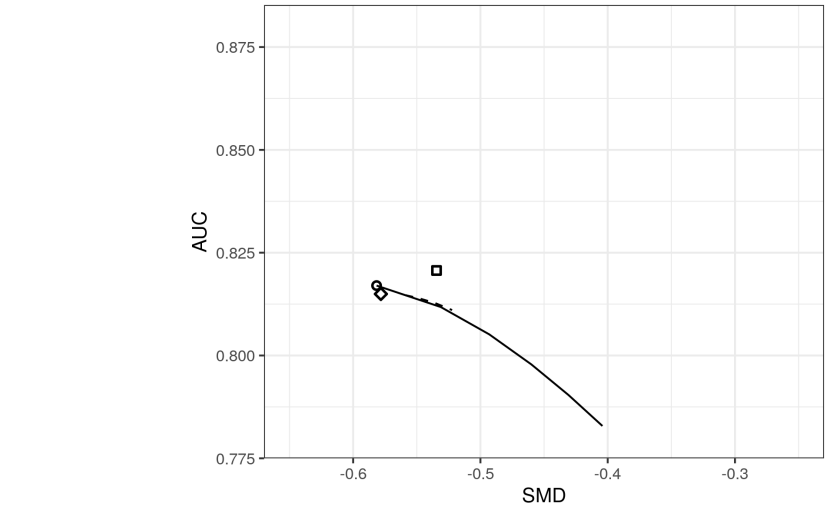
Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metric: the SMD metric with the sign flipped such that points further to the right represent models with less adverse impact. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure B.5: LDA RESULTS STAGE II - COMPANY MODELS

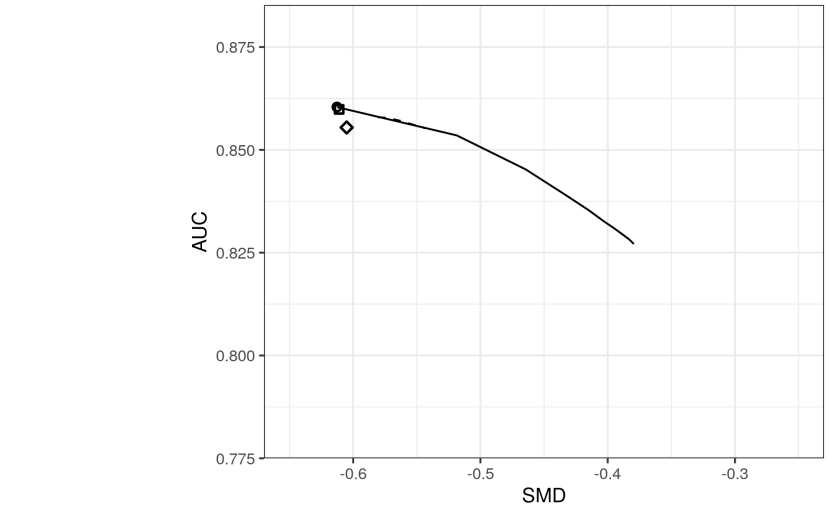


Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metric: the SMD metric with the sign flipped such that points further to the right represent models with less adverse impact. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

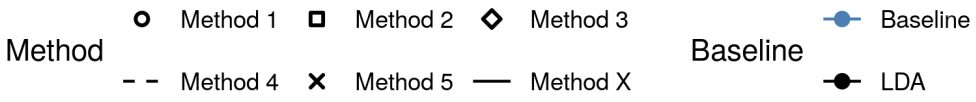
Figure B.6: LDA RESULTS STAGE III - BASELINE MODELS



(a) LOGIT - SMD

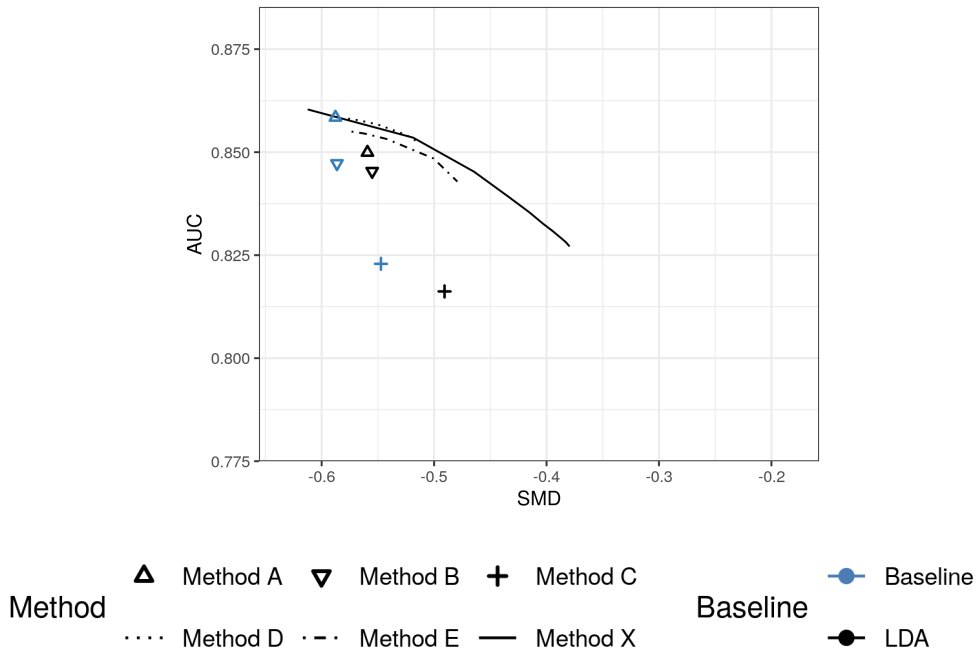


(b) XGBOOST -SMD



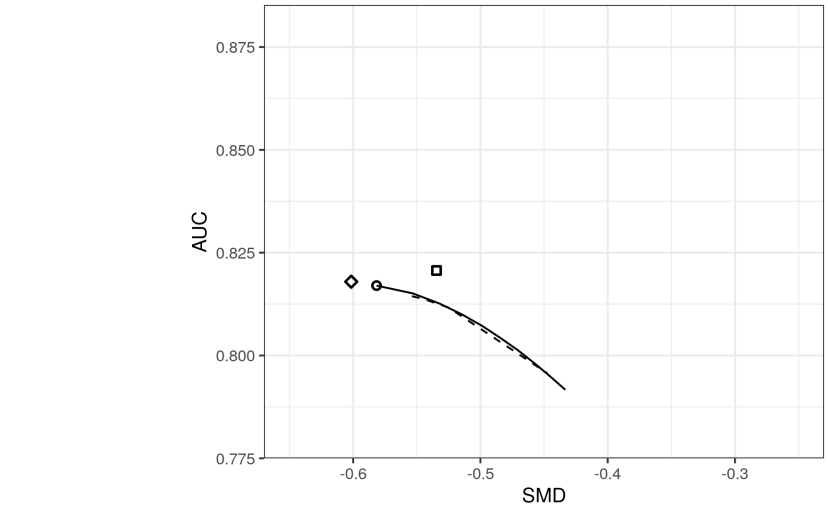
Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metric: the SMD metric with the sign flipped such that points further to the right represent models with less adverse impact. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure B.7: LDA RESULTS STAGE III - COMPANY MODELS

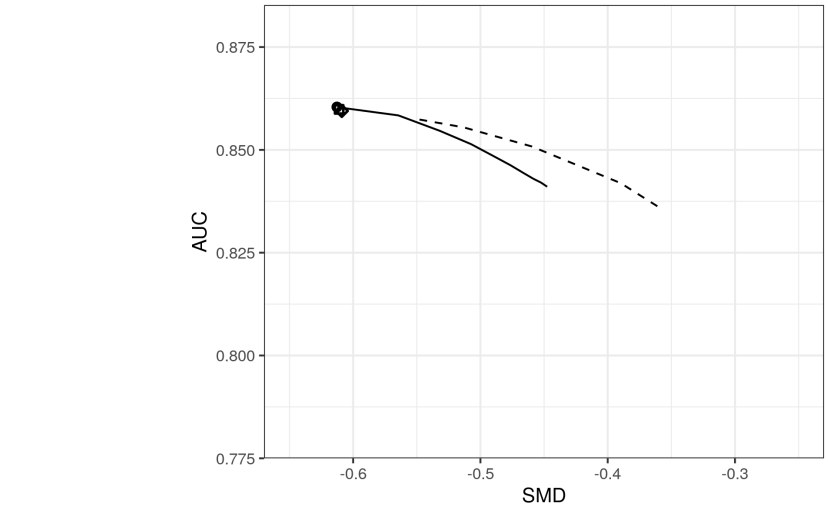


Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metric: the SMD metric with the sign flipped such that points further to the right represent models with less adverse impact. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

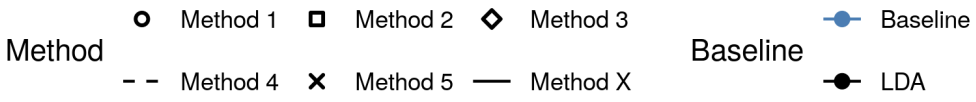
Figure B.8: LDA RESULTS STAGE IV - BASELINE MODELS



(a) LOGIT - SMD

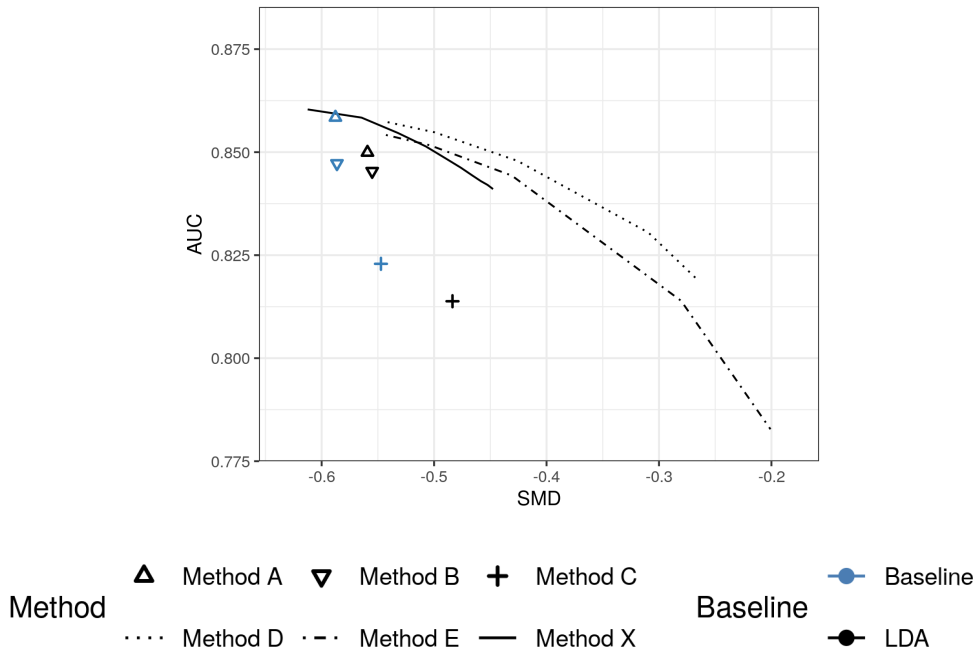


(b) XGBOOST - SMD



Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metric: the SMD metric with the sign flipped such that points further to the right represent models with less adverse impact. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

Figure B.9: LDA RESULTS STAGE IV - COMPANY MODELS



Note: Each panel of the figure shows the performance of less discriminatory alternative (LDA) models. All statistics are computed on the test data. The y-axis shows the AUC performance metric. Higher AUC numbers correspond to better predictive performance. The x-axis shows the adverse impact metric: the SMD metric with the sign flipped such that points further to the right represent models with less adverse impact. Colors distinguish between the starting point (baseline) and the LDA model. Different symbols distinguish between different methods. For the methods represented as lines (since they suggested many different LDA models), the Baseline Model always corresponds to the top-left point of the line.

C PERTURBATION

This appendix describes the perturbation schemes used for the fidelity tests across the three risk areas. The perturbation procedure differs by feature type (binary, categorical, or numerical/continuous). Binary features include all outlier flags. Categorical features include all features recording missing value codes and 9 categorical features supplied by the credit bureau data. All remaining features are numerical (these include balances, counts, and ratios). In our perturbation, we trade off two goals: inducing as many changes as possible and inducing changes into the favorable direction, that is, reducing the probability of default. We want to choose a consistent direction of change because we do not want changes in opposite direction to net out when we compare the final change in predictions. However, we also want to perturb as many features as possible to measure the fidelity of the explanation.

The direction of change is determined in one of two ways: For the baseline perturbation tests, participating companies and the research team using open-source tools were asked to report the direction of the feature impact. Usually, a negative direction implies a favorable impact since we are predicting default risk, given that higher values correspond to higher predicted default probabilities. For the correlated and random tests, the research team determines the direction by calculating SHAP feature importance. For local perturbation tests (Adverse Action Notice), we use the local SHAP values. For global perturbations, we want to mimic the global feature importance values provided by the responses. We therefore aggregate the local responses by computing the correlation of individual SHAP values with feature values. Usually, a positive direction implies a negative impact because higher values correspond to higher predicted default probabilities.

C.1 PARTIAL PERTURBATION

The partial perturbation scheme prioritizes the favorable direction and does not induce a change if it would not lead to a lower probability of default. We follow the procedure outlined below.

⇒ **Binary features**. We flip the value of a binary feature if the directional impact suggests a favorable impact on the prediction (*i.e.*, lower predicted probability of default). Otherwise, we leave the feature unchanged.

⇒ **Numerical features**. We change the feature by one standard deviation in the favorable direction subject to the change not violating feasible bounds of the feature in the data. We obtain the standard deviation in the test data set, omitting the missing value code "-1" in the calculation. The feasible bounds are the minimum and maximum values the feature takes in the test data.

⇒ **Categorical features**. We determine the perturbed level of categorical features by taking into account both the transition likelihood across levels in the data as well as the feature importance sign of each level. We first compute a transition matrix across levels of the categorical features by merging in data on the applicants in the test data from 12 months and 24 months prior to the application date. Based on this panel data, we calculate transition matrices across levels. We then determine the feature importance associated with each level of the categorical feature. Finally, we check whether there are levels that have higher favorable impact than the current level. If these exist, we choose the one with the highest transition probability from that set. If no such level exists, we leave the feature unchanged.

C.2 FULL PERTURBATION

The full perturbation scheme prioritizes inducing as many changes as possible. We follow the procedure outlined below.

⇒ **Binary features**. We always flip the value of a binary feature regardless of direction.

⇒ **Numerical features**. We change the feature by one standard deviation in the favorable direction subject to the change not violating feasible bounds of the feature in the data. We obtain the standard deviation in the test data set, omitting the missing value code "-1" in the calculation. The feasible bounds are the minimum and maximum values the feature takes in the test data. Empirically, we find that this procedure almost always leads to a perturbation.

⇒ **Categorical features**. We randomly choose another level of the categorical feature.

D INFERRING MINORITY STATUS

This appendix provides additional details on the procedure for inferring minority status. The process was performed as described below using a marketing data set that was acquired from Infutor that includes name and address information. After performing the calculations for inferring minority status, that information was provided to our data vendor, which performed a merge with the credit report data and then returned only de-identified data for analysis.

The process for inferring minority status proceeded in two steps: We first used first and last names to compute a baseline probability and then updated these probabilities using the racial make-up of the census block associated with the consumer's place of residence.

Our first step used two distinct software packages to compute baseline probabilities of race/ethnicity based on first and last names.

The first package is a proprietary commercial software (Onolytics) developed by (Mateos et al., 2014), which is based on anthropological research on the etymology of first and last names. It is based on a proprietary international database of over 1 million last names and 500,000 first names. We collapsed the detailed Onolytics categories to six categories used by the U.S. Census.[61]

The second package is the ethnicolr package developed by (Race, 1805).[62] They model the relationship between the sequence of characters in a name and race and ethnicity using Florida Voter Registration data as well as a database of 140k name-race associations from Wikipedia (Ambekar et al. (2009)).

Our second step updated each individual's baseline racial/ethnic probabilities with the racial and ethnic characteristics of the census block associated with the consumer's place of residence using Bayes' Rule. We computed posterior probabilities based on an individual's 2000 address and 2000 census data on racial and ethnic composition at the block level to create posterior probabilities for the four major racial/ethnic categories used by the U.S. Census (Hispanic, non-Hispanic white, non-Hispanic Black or African American, and non-Hispanic Asian/Pacific). The posterior probability that an individual with name s residing in geographic area g belongs to race or ethnicity r is then

$$\mathsf{Pr}(r|g,s) = \frac{\mathsf{Pr}(r|s)\mathsf{Pr}(g|r)}{\sum_{r' \in R} \mathsf{Pr}(r'|s) Pr(g|r')}$$

where R denotes the set of ethnic categories. We then updated this posterior again using individuals' 2010 address and the 2010 census data on racial and ethnic composition.

An individual was assigned to a racial/ethnic category if this category has the highest posterior probability according to both sets of posteriors and is equal to or above 0.8 on at least one of two sets of posteriors. We make one exception to this rule for non-Hispanic Black or African American. Onolytics has a relatively high mis-classification rate for African-American names who have last names whose etymological roots are European (*e.g.* "Washington") and are therefore classified as non-Hispanic white. To address this problem, we assigned an individual to the non-Hispanic Black or African American category if the posterior probability of this category based on the ethnicolr baseline is equal to or greater than 0.8 even when Onolytics-based posterior places a high probability on the non-Hispanic White category.

This two step method is similar to methods used by the CFPB to construct race in fair lending analysis. CFPB (2014) and (Elliott et al., 2009) show that combining geographic and name-based information outperforms methods using either of these sources of information alone.

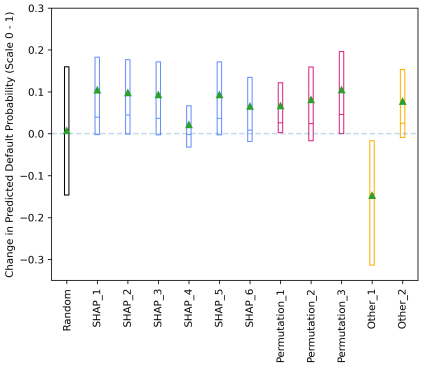
[61] These categories are Hispanic, non-Hispanic White, non-Hispanic Black or African American, non-Hispanic Asian/Pacific Islander, non-Hispanic American Indian and Alaska Native, non-Hispanic multi-racial.

[62] This is Python package is available at `https://github.com/appeler/ethnicolr`.

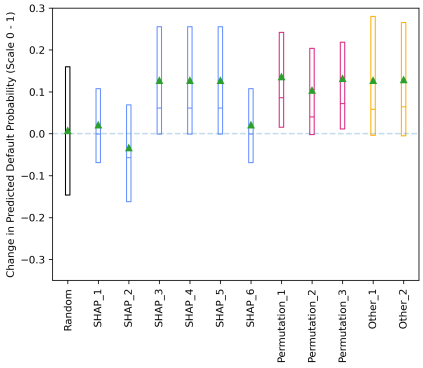
E MODEL RISK MANAGEMENT - FIDELITY: NEAREST NEIGHBOR TEST

Figure E.1: FIDELITY: NEAREST NEIGHBOR TEST

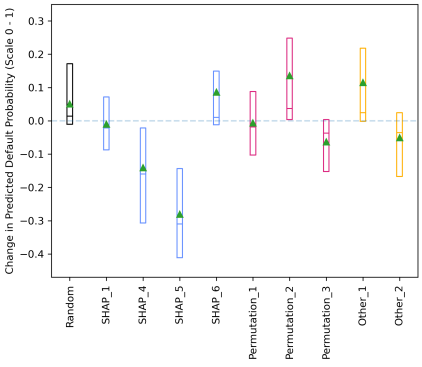
(a) LOGIT, TOP-10 FEATURES



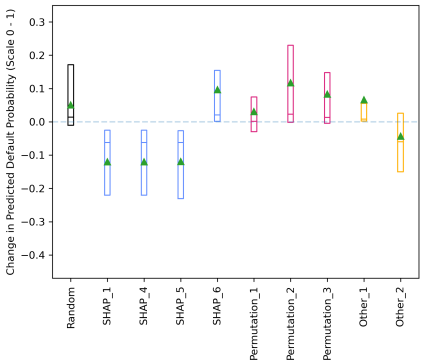
(b) LOGIT, CORRELATED TO TOP-10



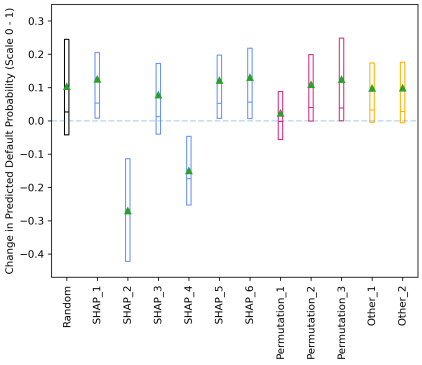
(c) NEURAL NET, TOP-10 FEATURES



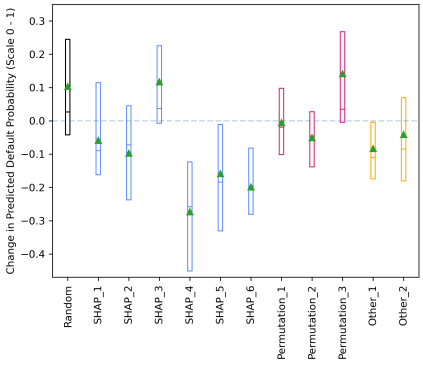
(d) NEURAL NET, CORRELATED TO TOP-10



(e) XGBOOST, TOP-10 FEATURES



(f) XGBOOST, CORRELATED TO TOP-10



Note: The figure displays the distribution of the change in predicted default probability between the original applicant and the nearest neighbor (NN) applicant. Each row represents a model. The right column shows the correlated benchmark. The boxes display the 25th and 75th percentiles, while the horizontal lines display the medians. The green triangles represent the averages. The results for random benchmarks, which are generated by perturbing 10 features randomly drawn from all input features, are displayed in both panels alongside the results for each of the company models.

Acknowledgments

FinRegLab would also like to recognize the presenters and members of our project Advisory Board who contributed to productive discussion of the development of the design, execution, and interpretation of this research. The Advisory Board consists of subject matter experts from computer science, economics, financial services, and regulatory backgrounds and includes representatives from approximately 40 major institutions including bank and nonbank financial institutions, technology firms, advocacy and civil society organizations, and academic institutions. State and federal regulators participated as observers in Advisory Board meetings.

We would also like to thank the following individuals who provided valuable feedback on portions of this report:

Alexei Alexandrov; Jay Budzik; Marsha Courchane and Adam Gailey, Charles River Associates; Steve Dickerson; Patrick Hall, bnh.ai; Stephen Hayes and Eric Sublett, Relman Colfax PLLC; Raghu Kulkarni, Discover Financial Services; Scott Lundberg, Microsoft Research; John Morgan, Capital One; Kate Prochaska; and Michael Umlauf and Gene Volcheck, TransUnion.

We would also like to acknowledge FinRegLab team members who worked on convenings and reports related to this project:

Natalia Bailey, Alex Bloomfield, Kelly Thompson Cochran, Colin Foos, Saurab Guatam, Gillous Harris, Tess Johnson, and Kerrigan Molland.

With support from:

Center for Inclusive Growth

The **Mastercard Center for Inclusive Growth** advances equitable and sustainable economic growth and financial inclusion around the world. The Center leverages the company's core assets and competencies, including data insights, expertise, and technology, while administering the philanthropic Mastercard Impact Fund, to produce independent research, scale global programs and empower a community of thinkers, leaders, and doers on the front lines of inclusive growth. The Center has provided funding to support this research.

JPMorgan Chase & Co.

JPMorgan Chase is committed to advancing an inclusive economy and racial equity. The firm uses its expertise in business, public policy and philanthropy, as well as its global presence, expertise and resources, to focus on four areas to drive opportunity: careers & skills, financial health and wealth creation, business growth & entrepreneurship, and community development.

flourish

Flourish, a venture of the Omidyar Group, has provided operating support to FinRegLab since its inception. Flourish is an evergreen fund investing in entrepreneurs whose innovations help people achieve financial health and prosperity. Established in 2019, Flourish is funded by Pam and Pierre Omidyar. Pierre is the founder of eBay. Managed by a global team, Flourish makes impact-oriented investments in challenger banks, personal finance, insurtech, regtech, and other technologies that empower people and foster a fairer, more inclusive economy.

Digital version available at finreglab.org

Published by FinRegLab, Inc.

1701 K Street NW, Suite 1150
Washington, DC 20006
United States