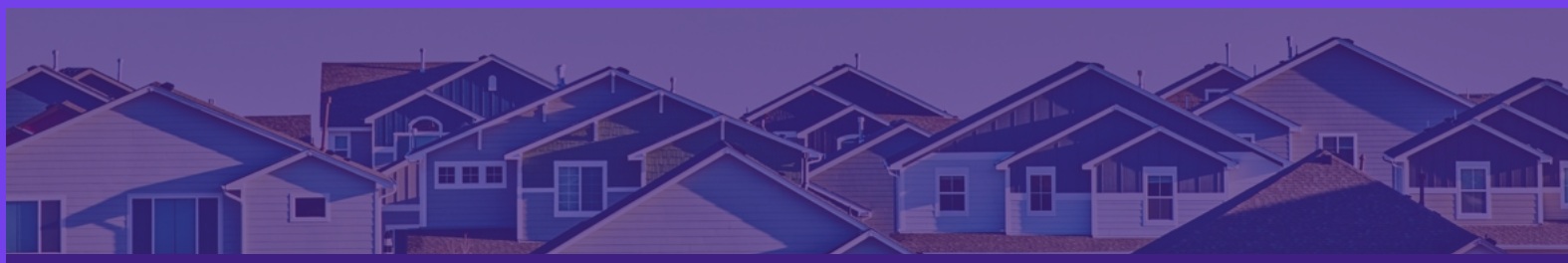


Improving Mortgage Underwriting and Pricing Outcomes for Protected Classes through Distribution Matching

EMBARGOED UNTIL APRIL 24, 2024

NFHA

fairplay

Improving Mortgage Underwriting and Pricing Outcomes for Protected Classes through Distribution Matching

A Joint Study by the National Fair Housing Alliance and FairPlay

Authors: Dr. John Merrill[1], Mark Jones[2], Mark Eberstein[3], Kareem Saleh[4], Dana Lockwood[5], Lusine Petrosyan[6], Dr. Michael Akinwumi[7]

Executive Summary

Mortgage underwriting disparities for historically underserved groups remain essentially unchanged despite several decades of legislative and policy interventions to improve them. Now with artificial intelligence, including machine learning, poised to augment or take over decision making across a range of domains, including in housing and lending, many mortgage market participants and stakeholders are focused on the question of whether algorithmic systems will create, exacerbate, or ameliorate disparities for protected groups. One potentially positive development in recent years is the emergence of algorithmic fairness techniques which aim to do a better job of predicting outcomes for populations that are not well-represented in data and/or negatively impacted by historical biases that might be contained in data used to develop models. To date, most studies of algorithmic fairness techniques applied to consumer credit have shown an accuracy-fairness tradeoff, a concept that suggests an increase in positive outcomes for

[1] Co-Founder and CTO, FairPlay AI
[2] VP, DevOps, FairPlay AI
[3] VP, Data Science, FairPlay AI
[4] Founder and CEO, FairPlay AI
[5] AI Data Engineer, National Fair Housing Alliance
[6] AI Policy Researcher, National Fair Housing Alliance
[7] Chief Responsible AI Officer, National Fair Housing Alliance, contact: makinwumi@nationalfairhousing.org

protected groups comes at the expense of a model's accuracy and consequently, in the view of some, at the expense of its profitability. In addition, many existing algorithmic fairness techniques rely on training methods that can be slow and unstable, which has tended to moderate their use in financial services. There are also concerns that some fairness techniques may result in increased fairness for some groups while decreasing fairness for others.

The goal of this study therefore was to evaluate whether there are algorithmic methods — using standard machine learning training techniques — that make it possible to build mortgage underwriting and pricing models that increase fairness without sacrificing accuracy and, by extension, without sacrificing profitability or introducing a bias for certain groups. Employees from the National Fair Housing Alliance (NFHA)'s Responsible AI Lab partnered with data scientists from FairPlay, an algorithmic fairness company based in Los Angeles, to conduct a fairness optimization study to explore this question.

NFHA's team members were pivotal in the successful execution of this project, dedicating substantial effort to the acquisition of vital data—a cornerstone of the research's feasibility. Their expertise was instrumental in developing and implementing robust systems for data management and security, ensuring the integrity and confidentiality of consumer information. Moreover, through meticulous data cleaning and analysis, they laid the groundwork for insightful research findings. Their comprehensive preparation and handling of the data were fundamental to the project's success. In addition, NFHA's employees provisioned the cloud resources used to perform the study including AWS RDS instance, EC2 instance, and VPN access.

When a task or result under discussion relates to model design, model development and model validation, "Project Team" means FairPlay's team of data scientists. NFHA's employees performed reviewer and project management roles while models were developed by FairPlay. As a reviewer, NFHA's Purpose, Process and Monitoring (PPM)[8] framework for algorithmic assessment was used to review the model source code and related documentations or mathematical formulations of the model constructs.

The Project Team built mortgage underwriting and pricing models trained for fairness using a novel algorithmic fairness methodology, Distribution Matching (DM), wherein machine learning models are trained to ensure that their outputs for protected groups should closely mirror the distribution of outputs for a corresponding control group. By incorporating one or more disparity minimization terms into a standard machine learning loss function, DM treats differences in outcomes between protected and control classes as a form of model error which, when minimized, reduces the differences in model outputs between those groups. This model was validated by NFHA with data provided courtesy of CoreLogic®.

[8] Akinwumi, M., Rice, L., & Sharma, S. (2022). Purpose Process and Monitoring: A New Framework for Auditing Algorithmic Bias in Housing & Lending. National Fair Housing Alliance. Retrieved from https://nationalfairhousing.org/wp-content/uploads/2022/02/PPM_Framework_02_17_2022.pdf

The Project Team's Preliminary Findings are that a DM-modified loss function can reduce the disparity of mortgage underwriting outcomes between Black and Hispanic applicants on the one hand and White, non-Hispanic applicants on the other, by upwards of 13 percent at the same rate of accuracy as models built using a standard loss function. Initial findings by the Project Team also suggest mortgage pricing disparities for Black and Hispanic borrowers could be reduced by upwards of 20 percent relative to White, non-Hispanic borrowers.

The Project Team's investigations and findings were limited by the absence of certain variables. For example, the Project Team did not have access to information about declined loans and loans that were approved but not taken by consumers. In addition, the data did not contain macroeconomic information, such as the prevailing 10-year U.S. Treasury bond rate at the time of loan origination, which is frequently used to set mortgage rates.

Despite these limitations, the Project Team's preliminary findings suggest that DM may be a viable pathway for integrating disparity minimization and other policy goals into algorithmic decision-making without sacrificing performance — an approach that aligns with emerging regulatory frameworks for artificial intelligence, as well as societal calls for less discriminatory housing and financial practices.

To ground its findings, the Project Team recommends repeating this study with an enriched dataset which includes information about declined loans, approved loans not taken, additional data about applicants, and the prevailing macroeconomic conditions at the time of application.

A SPECIAL THANKS

National Fair Housing Alliance extends its sincere appreciation to the Wells Fargo Foundation for their generous funding, which has greatly contributed to the work of the Responsible AI Lab and our collaborative research with FairPlay AI. While the insights and findings presented in this report are our own, they were made possible through the foundation's support. The views expressed within this report do not necessarily reflect those of Wells Fargo or the Foundation.

Introduction

Problem of Discrimination in Mortgage Origination

Disparities in mortgage underwriting and pricing outcomes have remained essentially unchanged for protected groups at least since 1990.[9] Industry experts disagree over whether Artificial Intelligence (AI), including Machine Learning (ML), combined with Big Data, could improve outcomes in mortgage lending for protected groups. The emergence of algorithmic fairness techniques in recent years has only furthered this debate, with skeptics arguing that the increased "fairness" achieved by debiasing methods must come at the expense of model accuracy and therefore profitability, and/or at the expense of increasing bias for some groups. This study investigates whether there are AI techniques that can increase fairness without sacrificing accuracy and, if there are, whether such methods are viable for use in the mortgage industry. The preliminary results of the study find that Distribution Matching (DM), whereby AI models learn that the distribution of algorithmic outcomes for any one protected group should closely resemble the distribution of the corresponding control group, can increase the fairness of mortgage underwriting to protected groups without a functional diminution in accuracy.

Literature Review

There are, in general, three different kinds of disparity-reducing machine learning algorithms:[10] pre-processing algorithms, in which the inputs to the system are transformed to mitigate or eliminate disparity; in-processing algorithms, in which the ML model is trained to reduce disparity; and post-processing algorithms, in which the output of the system is modified in order to reduce disparity. This study focuses on in-processing algorithms.

In-process techniques are proactive by design as they prioritize controlling the potential risk of biased outcomes, rather than delivering mitigations after an instance of unfairness. The following section will provide a closer overview of four related techniques — Prejudice Index Regularizer, Wasserstein Fairness, Counterfactual Fairness, and Fairness Though Awareness — that are open-source and can be compared with the DM technique. The application of these methods is context dependent, but similarly utilized for mitigating algorithmic bias and fostering fairer solutions.

Regardless of the sensitivity of the data, ML models can generate biased outcomes in myriad ways, including by associating seemingly discrete information with protected attributes. As a result, discriminatory outcomes may arise, disadvantaging one group over the other. To address this, Kamishima et al. proposed the Prejudice Index Regularizer (PIR) technique to enforce a

[9] https://www.usatoday.com/story/money/personalfinance/real-estate/2022/11/23/interest-rates-rise-mortgage-fairness-crisis/10748394002/?gnt-cfr=1
[10] https://dl.acm.org/doi/10.1145/3551390

classifier's independence from sensitive information.[11] Methodologically, PIR functions by penalizing predictions made relying on sensitive attributes during the training process. In contrast, the Wasserstein fair classification (WFC) is a regularization method focused on minimizing disparities in outcome distribution across different groups,[12] and it uses a measure of the distance between probability distributions for groups with different sensitive attributes to achieve this.[13] To ensure statistical independence between prediction and sensitive attributes, this method focuses on achieving fairness by optimally equalizing the prediction across various groups. Its approach to constraining models for fairness is similar to that of the DM technique.

Unlike the former methods that concentrate on eliminating disparate impact by ensuring similar outcomes across groups, additional techniques prioritize fairness measures at an individual level. For example, Counterfactual Fairness (CF) is a dominant method of assessing individual fairness. It aims to investigate if an outcome would be the same if certain protected attributes were different. In practice, a prediction would be considered fair to an individual if its outcome was the same for both the actual and counterfactual scenarios.[14] Under hypothetical cases, a quantified fairness measure is assigned when comparing a model's predictions when chosen characteristics (e.g., race or gender) are altered while the remaining data is unaltered. Additionally, the method of Fairness Through Awareness (FTA) is applied to integrate fairness constraints into the model optimization process. The objective is to ensure that similar individuals receive similar outcomes, correlating prediction with data input similarity.[15] This methodology utilizes the Lipschitz condition on the classifier, which is a constraint ensuring that the difference between the model's outcomes for similar individuals is bounded by the distance between their similarity difference.[16] This process can help more controlled behavior of the model and stabilize treatment consistency at the individual level.

In recent years, the current work on in-processing algorithms has been dominated by two broad approaches: approaches based on generative adversarial networks (GANs) and approaches based on constrained optimization. GAN-based approaches include the methods proposed by

[11] Kamishima, T., Akaho, S., Asoh, H., & Sakuma, J. (2012). Fairness-aware classifier with prejudice remover regularizer. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23* (pp. 35-50). Springer Berlin Heidelberg.

[12] Wan, M., Zha, D., Liu, N., & Zou, N. (2023). In-process modeling techniques for machine learning fairness: A survey. *ACM Transactions on Knowledge Discovery from Data*, *17*(3), 1-27.

[13] Jiang, R., Pacchiano, A., Stepleton, T., Jiang, H., & Chiappa, S. (2020, August). Wasserstein fair classification. In *Uncertainty in artificial intelligence* (pp. 862-872). PMLR.

[14] Wan, M., Zha, D., Liu, N., & Zou, N. (2023). In-process modeling techniques for machine learning fairness: A survey. *ACM Transactions on Knowledge Discovery from Data*, *17*(3), 1-27.

[15] *Id.*

[16] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., & Zemel, R. (2012, January). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference* (pp. 214-226).

Louppe et al. (2017)[17] which eliminate reliance on likelihood-inference data and attempt to treat protected class membership as a nuisance variable to be eliminated during training. In result, the model focuses on capturing the underlying patterns in the data without explicitly considering sensitive criteria. Constraint-based systems such as Zafar et al. (2017),[18] Cotter et al (2019)[19] or Narasimhan et al. (2019)[20] approach the problem differently: they directly treat disparity as a constraint upon the performance of the final model, and then perform a constrained optimization to build a less disparate model.

This study is most closely related to that of Jiang et al. (2020)[21] or Donini et al. (2018)[22] in which they consider the direct minimization of the difference between the distributions of scores. Like Jiang et al., and unlike Donini et al, this study focuses on adding a disparity minimization term to a standard loss function to measure and narrow the difference between the distributions of the classes between which disparities are to be reduced.

Comparing Distribution Matching to GANs

A GAN is a combination of two networks, a generator and an adversary, and finding an optimal set of weights for the two networks taken together entails finding a saddle point: a set of weights which simultaneously minimizes the error due to the generator while maximizing the error due to the adversary. Error minimization processes, such as back-propagation, cannot find a saddle point: they can only approximate it, and doing so requires slowing down learning as the saddle point is approached in order to not "miss" the saddle point and either rise forever (when the act of maximizing the error of the adversarial network dominates the act of minimizing the error of the generative network) or fall forever (when the act of minimizing the generative network's error dominates the act of maximizing the adversarial network's error.)[23]

In distribution matching, the adversary is replaced by an adaptive system which includes terms that measure and minimize the disparity between the distributions of the models' outputs. The key difference is that there is no adversary to be trained; instead, there is a directly observed difference between two distributions which does not have to be maximized. This transforms the min-max search for a saddle point into a minimization problem with an additional adjustment term, where the adjustment term can be incorporated into the training process.

[17] Louppe, G., Hermans, J., & Cranmer, K. (2019, April). Adversarial variational optimization of non-differentiable simulators. In The 22nd International Conference on Artificial Intelligence and Statistics (pp. 1438-1447). PMLR.
[18] https://doi.org/10.48550/arXiv.1610.08452
[19] https://proceedings.mlr.press/v97/cotter19b.html.
[20] https://ojs.aaai.org/index.php/AAAI/article/view/5970/5826
[21] https://arxiv.org/abs/1907.12059
[22] https://proceedings.neurips.cc/paper_files/paper/2018

This means that model developers can benefit from the performance of modern ML training tools with minimal changes to the configuration of their learning systems. In addition, because reducing any given disparity by adjusting the loss function simply consists of adding one more disparity minimization term, this method makes it relatively easy to incorporate multiple disparity targets by simply adding more distribution matching terms, making it applicable to a scenario where fairness is sought in a multidimensional space of protected classes or a scenario where there are other objectives to be jointly optimized alongside fairness.

Distribution Matching Can Exploit More Data than Constrained Optimization

Distribution matching techniques can make use of data that is not normally useful to other learning algorithms like constrained optimization.

First, distribution matching systems can use uncertain demographic data that other algorithmic fairness techniques cannot. In many cases, actual demographic data is not available during training, and demographic class membership must be evaluated using probabilities of membership. Many fairness-enhancing algorithms, including most GAN-based and many pre-processing and post-processing algorithms, depend on direct assignment of each input record to one and only one demographic class. Distribution matching techniques can accommodate uncertain demographic class membership data; most other algorithms cannot.

Second, and more importantly, distribution matching is an algorithm that maps one output distribution to another output distribution for any input, and not just for inputs that are associated with outcomes. This means DM can use records that other algorithms cannot consume — for example, loan applications that were declined and for which there is no performance data. By using the information implicit in these additional records, DM can learn more about the distribution of all records and build models that are more accurate than models built without such information.

Question Presented

This study explores whether a modified loss function that includes one or more distribution matching terms might overcome the challenges exhibited by other algorithmic fairness techniques and yield fairer models with no functional loss of accuracy.

Preliminary Findings

The Project Team's preliminary findings are that a learning system enhanced by distribution matching can reduce the disparity of underwriting outcomes between Black and Hispanic applicants on the one hand and White, non-Hispanic applicants on the other, while maintaining accuracy relative to an unenhanced system. These results can be quite dramatic: an increase in

the Adverse Impact Ratio (AIR)[24] for Black applicants of upwards of 13 percent when compared to a model built using the same base loss function without a distribution matching term. The Project Team also found that distribution matching may yield significant increases in pricing fairness for protected classes — a diminution in standardized mean difference on the order of 15 percent or higher.

Materials and Methods

Data Acquisition

NFHA acquired a dataset provided courtesy of CoreLogic® consisting of 5,926,182 records representing 3,797,678 distinct loans (the "dataset"). This dataset was assembled using a match key provided by CoreLogic® to merge the Home Mortgage Disclosure Act database (HMDA database) and a proprietary mortgage dataset provided courtesy of CoreLogic®.

Dataset Characteristics

The dataset contains records of loans originated from December 1987 through January 2021. These records include information about originated mortgages joined with data about the performance of those mortgages. Each record includes information about the property (e.g., appraisal price), information about the final loan terms (e.g., APR and loan-to-value ratio), and a limited amount of information, such as FICO score at the time of origination and the back-end ratio of the loan, about the principal borrower for each loan.

Data Partitioning and Analysis

The Project Team used loans from January 2010 to December 2016 for model development and validation. To facilitate an accurate analysis and to ensure that every record used for the study has observation data that covers at least a four-year period, records outside the seven-year interval were excluded from the development set. For the analyses that covered the whole United States, all loan records were used. For the analyses that covered only the Los Angeles Metropolitan Statistical Area, all 65,250 records from the 2010-2016 period were used. Records for loans originated between January 2010 and December 2014 were used for 'in-time' training and testing, and loans originated between January 2015 and December 2016 were used as an 'out-of-time' testing set. The in-time sets were subsequently divided into two subsets using a 70/30 ratio. The larger subset was used for training purposes, while the smaller subset served as an in-time testing dataset.

[24] See glossary for definition

Fairness Optimization of Linear and Non-Linear Models

The Project Team investigated whether a model's fairness could be improved via distribution matching (DM) without compromising accuracy. The project team used a Base Loss Function (BLF) and a Modified Loss Function (MLF), both described in greater detail below, to train underwriting and pricing models. In the case of models trained with a BLF, the model's objective was to accurately predict the default target, for example delinquency or interest rate. In the case of models trained with a MLF, the model's objective was to accurately predict the default target while also ensuring that the distribution of predicted outcomes for protected classes matched the distribution of predicted outcomes for control class applicants. The Project Team generated four underwriting and pricing models using a BLF (referred to hereafter as "Unconstrained Models") and four underwriting and pricing models using the MLF (referred to hereafter as "Constrained Models").

Each of these eight models could be described as belonging to one of two classes of neural networks: linear networks and non-linear networks. All non-linear models contained one hidden layer with 256 nodes. Both classes of models included a single layer of 10 nodes, with connections to the final output layer being linear. The activation functions on the hidden layer in the non-linear network were sigmoids. Finally, since the nominal pricing for any loan is never negative, the output layer of the pricing network was made up of a single rectified linear (ReLU) node instead of a linear node. (See diagrams below.)

Figure 1: Architecture of the nonlinear neural networks

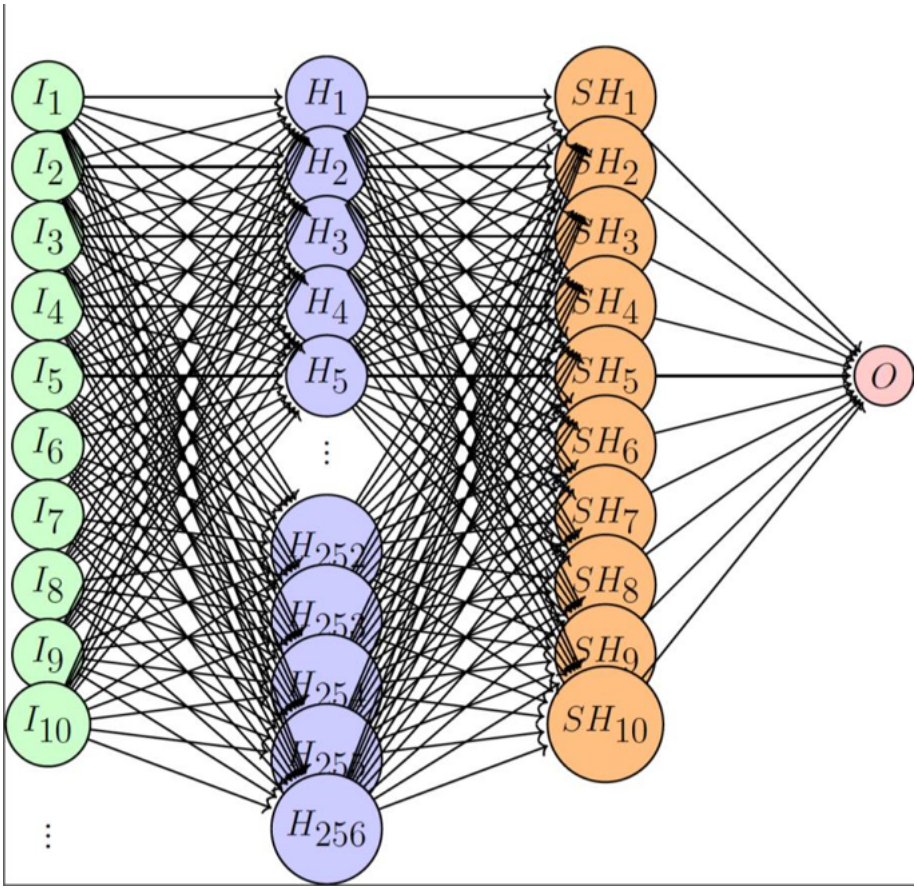
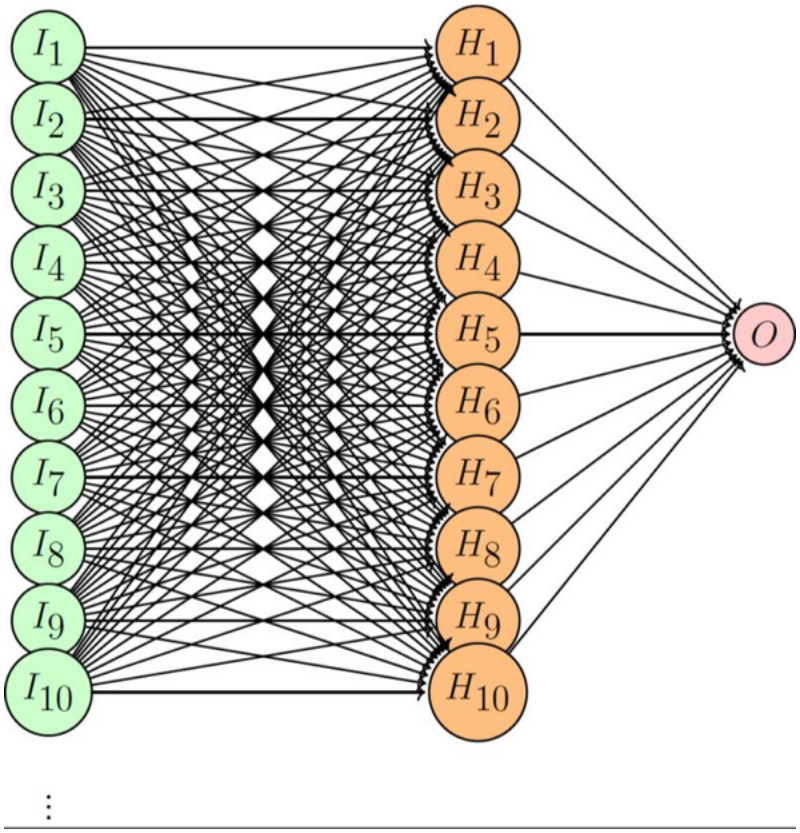


Figure 2: Architecture of the linear neural networks



Unconstrained Model Targets and Base Loss Functions

The target for underwriting models trained with the Base Loss Function (BLF) was set to identify loan applications that would become 60 days delinquent within the first 48 months after origination. The BLF for underwriting was the binary cross-entropy[25] between the network output and delinquency target. The target for pricing models was identified as the nominal Annual Percentage Rate (APR) for loans, with the mean squared error[26] between network output and APR target serving as the BLF for pricing.

Constrained Model Targets and Modified Loss Functions

The Constrained Models were trained with a modified loss function (MLF), which included additive terms added to the Base Loss Function (BLF) at the output layer to measure and narrow the Jensen-Shannon divergence[27] between protected and control class distributions and reduce the disparity between them. This process involved mapping each class of applicants to a distribution at the second-to-last layer which was linearly transformed to provide an output score in the top layer, optimized with the BLF. This methodology was applied to neutralize the difference between the target distributions for four racial groups (Black, American Indian and Alaska Native (AIAN), and Hispanic) that are all protected classes; one control class (White); and two gender groups, one protected (Female) and one control (Male). The output distributions of these classes were all forced to match simultaneously, rather than one at a time.

Analysis of CoreLogic Data

The Project Team conducted a total of eight separate analyses of the dataset. These analyses were designed to consider various combinations of objectives, geographical coverage, and the type of loss function utilized. The analyses can be broken down as follows:

Loss Function Types: Two different kinds of loss functions were employed:

+ Unconstrained: These models were trained only with accuracy terms and without any distribution matching terms.
+ Constrained: These models were trained with one or more distribution matching terms to promote accuracy and increased fairness outcomes for protected groups.

These Loss Function Types were used to build two kinds of models:

+ Underwriting Models: Algorithms to assess the risk associated with providing a mortgage loan.

[25] See glossary for definition
[26] See glossary for definition
[27] See glossary for definition

+ Pricing Models: Algorithms designed to generate an APR for an offer of mortgage credit that is likely to be accepted but that is also commensurate with the riskiness of the Applicant.

Geographies: The Project Team constructed fairness optimized underwriting and pricing models across two different geographic areas:

+ The Whole U.S.: A nationwide analysis that allowed for the creation of national underwriting and pricing models reflecting the distributional properties of the entire country.
+ Los Angeles (LA): A fairness optimization analysis specific to the LA metropolitan area, which provided insights into the fairness of the mortgage market in this significant urban center.

As discussed above, The Project Team conducted its investigation using data from a sample of records dated between January 2010 and December 2016. This period was chosen because it was the longest post-2008 financial crisis interval in the data that (1) would be fully seasoned by having performance data at least 48 months after origination, and (2) would allow third parties and the public to perform independent testing of the Project Team's results, for example, with the omitted 2017 records.

Fairness Optimization through Distribution Matching

The foundational principle behind distribution matching is: if the output score distributions corresponding to two distinct groups are identical, then the model outputs will exhibit no group disparity. Suppose that there are two sets of scores with the same distribution, one corresponding to a protected class and another corresponding to the control class. Assume further that output scores are used in applications like loan underwriting or other decisioning, and pricing or other loan term assignment. When underwriting decisions are made from such a score by thresholding (e.g., applicants whose scores are above a certain value are approved and applicants below that value are denied) and in a decisioning process where the output score distributions match, there will be no group disparity when the protected class performance is compared to the control class at any threshold; in other words, the fraction of applicants from the control group with a score above that threshold will be identical to the fraction of applicants from the protected group with a score above that threshold, resulting in a theoretical 100 percent AIR. In the reverse situation, where loans are approved if the model's score is below a certain threshold, for example in tenant scoring applications where a lower score implies a better outcome, thresholding on identical distributions will still lead to the same formal outcome.

The same principle holds true for pricing: if the distribution of pricing outcomes for a protected group and a control group are the same, their means and variances will be identical and therefore both the mean difference between the distributions will be zero and the standardized mean difference between the distributions will be zero.

Achieving this parity is non-trivial and requires specific mechanisms to measure the difference between the two distributions in question and to use this difference to encourage matched distributions. The difference between the distributions is measured using the Jensen-Shannon (JS) divergence,[28] a measure that quantifies the disparity between two probability distributions. It is always non-negative and is zero if and only if the two measures are identical.[29] Therefore, if a term proportional to the JS disparity between two distributions is added to a loss function and minimized, that term will tend to force the two output distributions to be identical.

Data Analysis

Input Transformation

The Project Team took several steps to pre-process the input data provisioned by NFHA. Only a specific subset of the variables was used as model inputs. This subset was selected after consultation with industry experts[30] about which data elements would be available at decision time, as well as which data elements are used for underwriting by the Government Sponsored Enterprises (GSEs), Freddie Mac and Fannie Mae.

Categorical variables were separated into a set of flag variables for each unique value in such an input, and a binary feature with a value of either zero or one was created, marking the presence or absence of that value in each given record.

Continuous variables were standardized with each input feature transformed linearly to have a mean of zero and a standard deviation of one. This step was taken because some learning algorithms can fail if the inputs are either too large or too small. In cases where data points were missing, zeros were used as replacements.

[28] The Jensen-Shannon divergence is a way to tell how similar or different two sets of data are. Imagine you're comparing two fruit bowls to see how similar they are in terms of fruit variety. If one bowl has a mix of apples and oranges, and the other has the same but with different amounts of each, the Jensen-Shannon divergence helps measure how much the mix of fruit in one bowl differs from the mix in the other. The divergence has some handy characteristics: it will always give you a result, it works the same both ways (comparing bowl A to bowl B is the same as B to A), and it behaves nicely in a geometric sense – just like you can measure distance with a ruler. This makes it very useful in fields like information theory, machine learning, and biology to compare all sorts of data, not just fruit bowls! It's like having a universal tool for measuring the difference between any two sets of things.

[29] Bruni, V., Rossi, E. & Vitulano, D. Jensen–Shannon divergence for visual quality assessment. SIViP 7, 411–421 (2013). https://doi.org/10.1007/s11760-013-0444-3

[30] Neither the CoreLogic data dictionary nor the data itself was shared in connection with these consultations.

Certain coded values signified missing data. Specifically, any FICO score below 250 or exceeding 800 was recognized as a missing value code. Such scores were replaced with zero after standardization to ensure accurate transformation.

The missingness statistics of the continuous input variables are shown in Table 1 below. There were no missing values in any of the discrete variables.

Table 1: Missingness rates for continuous variables

Variable	Missingness rate (%)
number_of_units	0.0
appraised_value	55.9
original_ltv	0.0
combined_ltv_at_origination	80.3
io_term	99.9
back_end_ratio	59.1
fico	2.1

Input Data

Table 2: Input variables used in training the underwriting and pricing models' scoring functions

Variable name	Interpretation	Values
fico	The borrower's FICO credit score at the time of loan origination	Continuous, derived from fico_score_at_origination by replacing all value below 250 or above 900 with NULL
number_of_units	The total number of units or housing units within the property.	Continuous: valid values are 0-99 or NULL.
appraised_value	The appraised value of the property	Continuous

Variable name	Interpretation	Values
original_ltv	The loan-to-value ratio at the time of application	Continuous
combined_ltv_at_origination	The combined loan-to-value ratio at the time of origination	Continuous
io_term	The term or duration of an interest-only payment period, if applicable	Continuous
back_end_ratio	The back-end debt-to-income ratio for the primary applicant	Continuous
coapplicant_present	Was there a coapplicant on the loan?	Boolean recast as continuous.
statecode	The code representing the state where the property is located	Discrete: 51 USPS codes or nan if not present
property_type	The type or category of the property (e.g., single-family home, condominium)	Discrete: 1 = SFR (Single Family Residence) 2 = Condominium 3 = Co-Operative 4 = Multi-Family (2-4 Units) 5 = Townhouse 6 = Planned Unit Development 7 = Multi-Family (5+ Units) 8 = Commercial Property 9 = Mixed Use Property L = Lot M = Manufactured Housing U = No Info Z = Other
occupancy_type	The type of occupancy for the property (e.g., owner-occupied, rental)	Discrete: 1. Principal residence 2. Second residence 3. Investment property

Variable name	Interpretation	Values
		U. Unknown/No info
payment_frequency	How often loan payments are made (e.g., monthly, biweekly)	Discrete: 1 = Weekly Payments 2 = Bi-weekly Payments 3 = Semi-Monthly Payments 4 = Monthly Payments 5 = Quarterly Payments 6 = Semi-Annual Payments 7 = Annual Payments U = No Info
channel	Lender's origination source of the loan.	Discrete: 1 = Retail Branch 2 = Wholesale 3 = Mortgage Broker 4 = Realtor Originated 5 = Relocation Corporate 6 = Relocation Mortgage Broker 7 = Builder 8 = Direct Mail 9 = Other Direct A = Internet B = Other Retail C = Mortgage Banker D = Corresponded Lender U = No Info
documentation_type	The documentation requirements used for underwriting the loan.	Discrete: 1 = Full Documentation 2 = Low or Minimal Documentation 3 = No Asset/Income Verification U = No Info

Variable name	Interpretation	Values
io_flag	A flag indicating whether the loan includes an interest-only payment period	Discrete: Y = Yes N = No U = No Info
product_type_category	Product type for the loan (Fixed, ARM, or Unknown)	Discrete: F = Fixed A = ARM U = No Info
gse_eligible_flag	A flag indicating whether the loan is eligible for purchase by government-sponsored enterprises (GSEs)	Discrete: 0 = Non-Conforming 1 = Conforming (standard GSE policy) 2 = Jumbo Conforming (expanded GSE policy starting in 2008) U = No Info

Target data

Table 3 – Target values

Variable name	Interpretation	Values
Underwriting	eventually_60dq	Was the loan ever 60 days delinquent?
Pricing	interest_rate	Nominal interest rate

Accuracy and Fairness Metrics

The Project Team used two metrics to identify the 'best' underwriting model: Area Under the Curve (AUC)[31] and Adverse Impact Ratio (AIR).[32] The best underwriting models were defined as having

[31] See glossary for definition
[32] See glossary for definition

an AUC value that was within a very close margin (0.01) of the highest AUC achieved amongst all the scoring functions. The Project Team also required that the AIR be maximized across all demographic categories such that the 'best' models were the ones with the highest AUC within the close margin and the highest minimal AIR among those with AUCs within the close margin.

The Project Team used two metrics to identify the 'best' pricing model: Mean Squared Error (MSE)[33] and Standardized Mean Difference (SMD).[34] The best pricing model was defined as having the lowest MSE and the SMD with the lowest absolute value across all demographic categories given that MSE.

Dealing with Absence of Information on Unapproved Loans

One of the challenges faced by the Project Team in conducting this study was the absence of information on unapproved loans. This omission would normally make it impossible to compute AIR and AUC statistics for an underwriting model, since there would be no way to determine how the classifier behaved on loan applications that were either denied or approved but not originated. To manage this limitation, the underwriting models were calibrated to approve only the top five-sixths of the dataset. The performance of the underwriting model could then be estimated by treating the lowest one-sixth as the class of unapproved loans.

The Project Team considered adding HMDA records associated with denied loans in the publicly available data but concluded that taking this step would not have resolved the data sufficiency problem because many of the declined applications in the publicly available HMDA database lack essential input features, making them inadequate for purposes of this analysis. A richer column space is available for post-2017 Loan Application Records in the HMDA database, which, if combined with a more contemporary dataset on originated loans, could further validate the results of this study.

Demographic Information and Categorization

The dataset had self-reported demographic class membership for each loan record. The demographic groups in the data were: six races or ethnicities (White, Black, Asian or Pacific Islander, American Indian or Alaska Native (AIAN), two or more non-White races, and Hispanic) and two genders (Male or Female). The distinction between race and ethnicity is often nuanced and they can be treated differently. However, many proxies for race and ethnicity treat them as a single feature. For this analysis, the Project Team treated Hispanic ethnicity as a stand-alone race. Consequently, regardless of the race an applicant may have self-identified with, if they also reported being of Hispanic ethnicity, they were recategorized exclusively as 'Hispanic' in this

[33] See glossary for definition
[34] See glossary for definition

analysis. Under this categorization schema, then, a 'non-Hispanic White' applicant was simply coded as 'White.' In contrast, a 'White Hispanic' applicant was coded as 'Hispanic.'

Table 4: Demographic values

Variable name	Interpretation	Values
applicant_race	Primary applicant race	1. American Indian or Alaska Native 2. Asian 3. Black or African American 4. Native Hawaiian or Other Pacific Islander 5. White 6. Information not provided by applicant in mail internet or telephone application 7. Not applicable.
coapplicant_race	Co-applicant race (if any)	1. American Indian or Alaska Native 2. Asian 3. Black or African American 4. Native Hawaiian or Other Pacific Islander 5. White 6. Information not provided by applicant in mail internet or telephone application 7. Not applicable 8. No co-applicant.
applicant_ethnicity	Primary applicant ethnicity	1.Hispanic or Latino 2. Not Hispanic or Latino 3. Information not provided by applicant in mail internet or telephone application 4. Not applicable.
coapplicant_ethnicity	Co-applicant ethnicity (if any)	1.Hispanic or Latino 2. Not Hispanic or Latino

Variable name	Interpretation	Values
		3. Information not provided by applicant in mail internet or telephone application 4. Not applicable 5. No co-applicant.
applicant_sex	Primary applicant gender	1. Male 2. Female 3. Information not provided by applicant in mail internet or telephone application 4. Not applicable 6. Applicant selected both male and female
coapplicant_sex	Co-applicant gender (if any)	1. Male 2. Female 3. Information not provided by applicant in mail internet or telephone application 4. Not applicable 5. No co-applicant 6. Co-applicant selected both male and female

Input Subset Selection

Nationwide Dataset

After trimming the dataset of years prior to 2010 and subsequent to 2016, the dataset contained 3,234,601 unique loan records, which were partitioned into an in-time training set and an out-of-time testing set with a 70/30 ratio.

Los Angeles Specific Dataset

The Los Angeles-specific dataset contained 65,250 total records in the dataset referring to loans within the LA Metropolitan Statistical Area. These records were partitioned into an in-time training set and an out-of-time test set with a 70/30 ratio.

Selecting Models for consideration

Underwriting models were selected by observing the highest AUC and AIR values achieved by each potential model on the out-of-time test set. All AUC and AIR scores were rounded to the nearest 0.05 in order to eliminate false precision. The best Unconstrained underwriting model was taken to be the one with the highest AUC and the highest AIR across all models with that same AUC. The best Constrained underwriting model was similarly selected. Pricing models were chosen in the same manner, except that the 'best' pricing models selected minimized the MSE and the absolute value of the associated SMD.

Training Details and Tools

All models were built using Python 2.10. Optimization was performed using TensorFlow v 2.10.0, numpy v 1.23.3, scipy v 1.91, and pandas v 1.5.0. The search for the optimal hyperparameters controlling the balance among the terms in the loss function was performed using hyperopt v 0.2.7. The hyperparameter search was run for 100 tests in each of the eight configurations for which networks were constructed, and the relevant accuracy and fairness statistics (AUC and AIR for underwriting and MSE and SMD for pricing) were displayed in each graph.

Due to the absence of any records reflecting applications that were not funded, the distribution matching algorithm was applied only to applications that were originated.

Results

Performance Figures

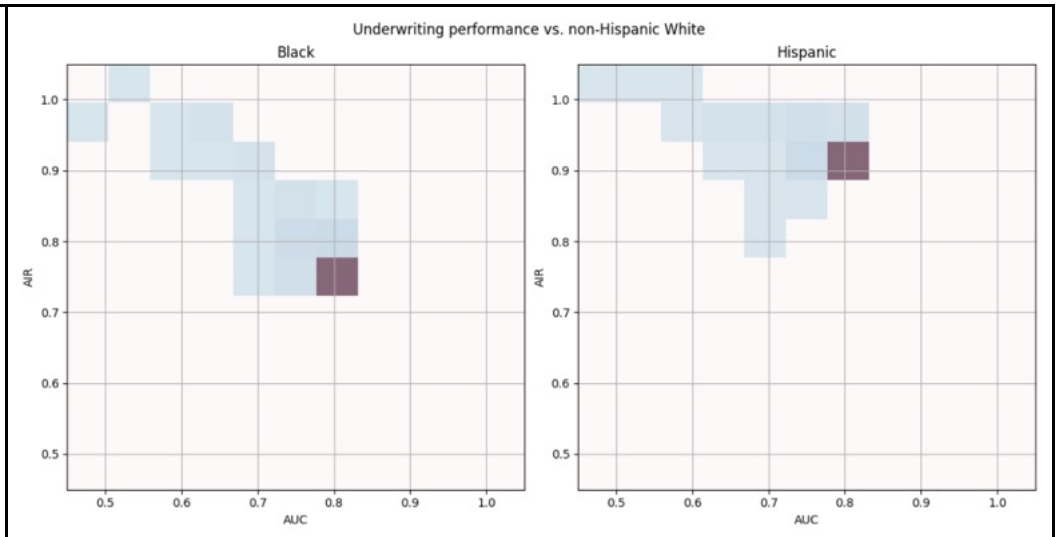
Figures 3(a) to 6(a) show the fairness and accuracy behavior of the Constrained and Unconstrained underwriting scoring functions by plotting the AUC and the AIR when the scoring function approves five-sixths of all applications. As mentioned above, there were no records in the dataset that were not approved, so it was not possible to create a meaningful swap-set. Results are shown for only two of the protected groups — Black and Hispanic applicants — by showing a dot for each potential scoring function. It is not easy to see the dots corresponding to Unconstrained Models here because they all fall inside a very small area that is completely covered by the results of Constrained Models.

Figures 3(b) to 6(b) show the fairness and accuracy behavior of the Constrained and Unconstrained pricing functions. Unlike the case of the underwriting scoring functions, a pricing model can only be trained on originated loans, so there was no loss of information in the analysis.

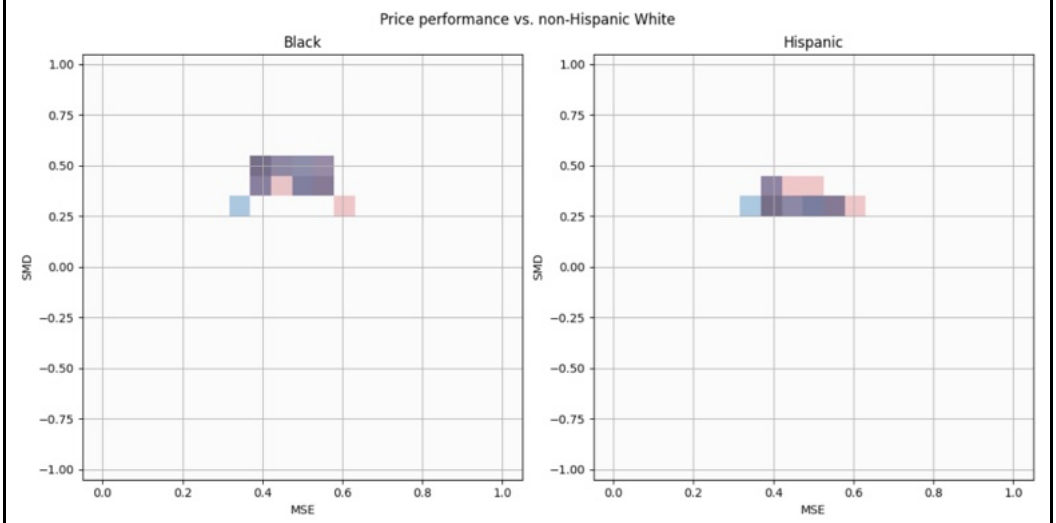
Figures 3 through 6 are two-dimensional histograms that display the number of models that fall into each block in the performance range. For underwriting models, the performance range is defined by each model's AUC and AIR. For pricing models, the performance range is defined by each model's MSE and SMD. Each plot contains squares denoting a performance range along these dimensions. In each plot, a blue square means that there were one or more Constrained models but no Unconstrained models in that performance range; a red square means that there were one or more Unconstrained models but no Constrained models in that performance range; and a purple box means there were both Constrained and Unconstrained models in that performance range.

Figure 3: Performance scatter plots – All US, nonlinear neural network

(a) Underwriting



(a) Pricing



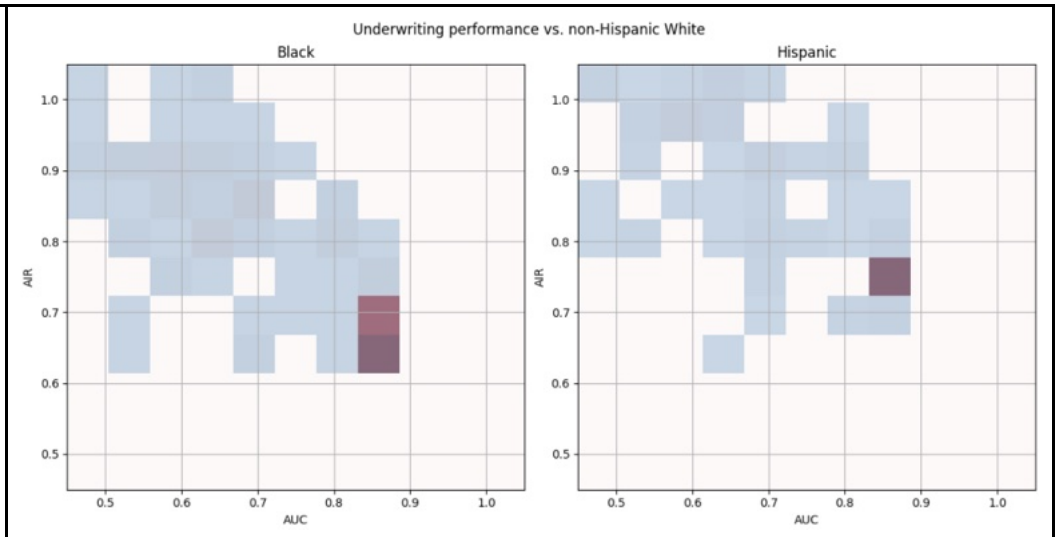
One or more **Constrained** models but no Unconstrained models in that performance range

One or more **Unconstrained** models but no Constrained models in that performance range

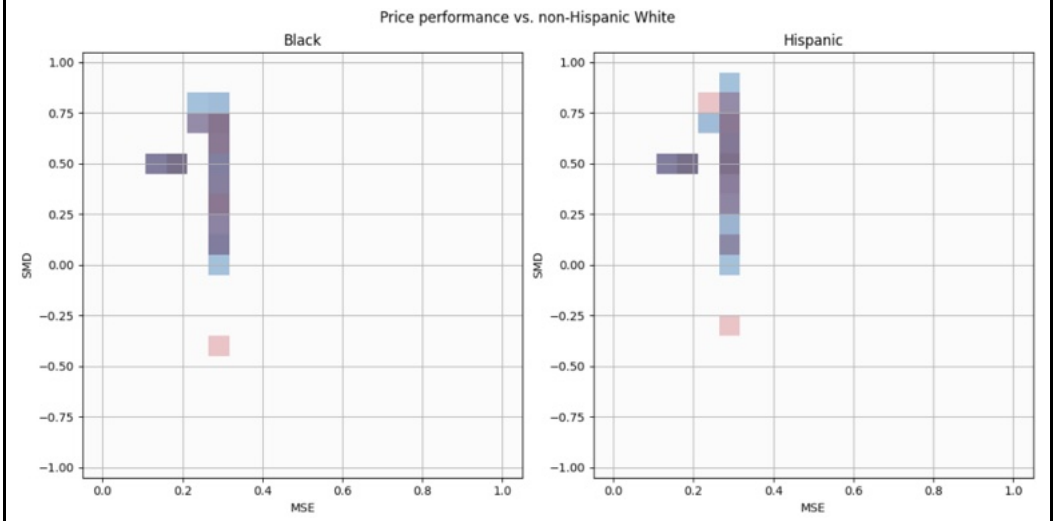
Both **Constrained and Unconstrained** models in that performance range.

Figure 4: LA County only, nonlinear neural network

(a) Underwriting



(b) Pricing



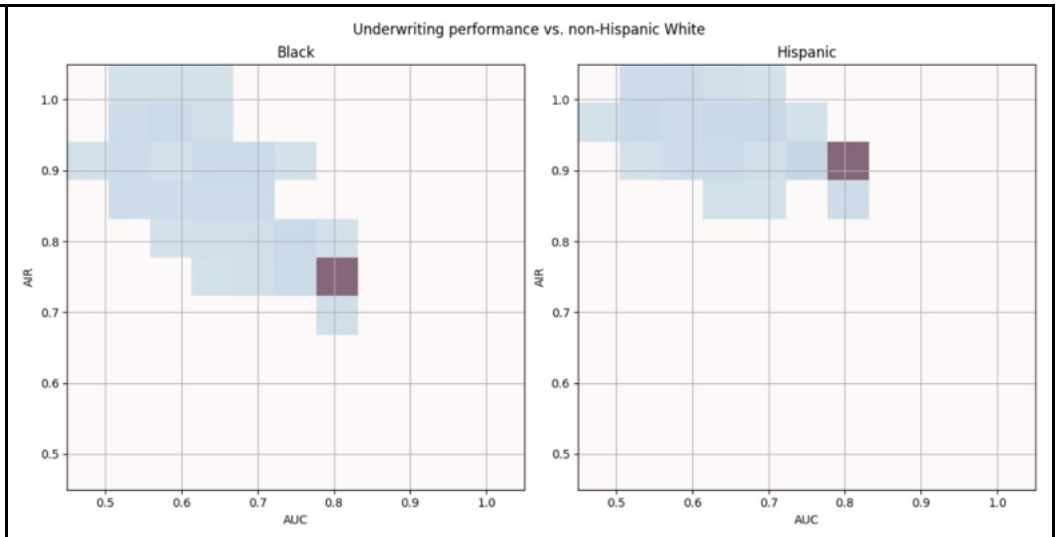
One or more **Constrained** models but no Unconstrained models in that performance range

One or more **Unconstrained** models but no Constrained models in that performance range

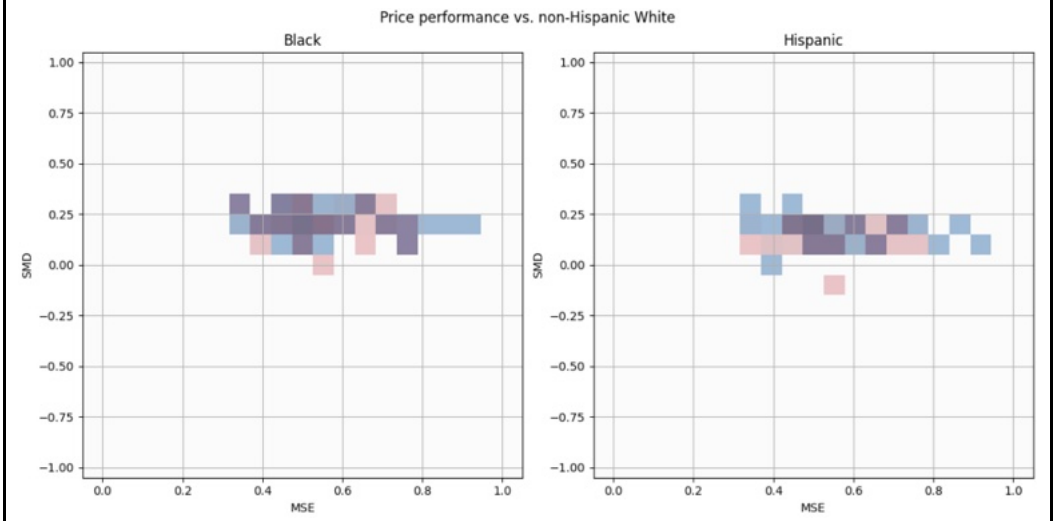
Both **Constrained and Unconstrained** models in that performance range.

Figure 5: All US, nonlinear logistic regression or ReLU fit

(a) Underwriting



(b) Pricing



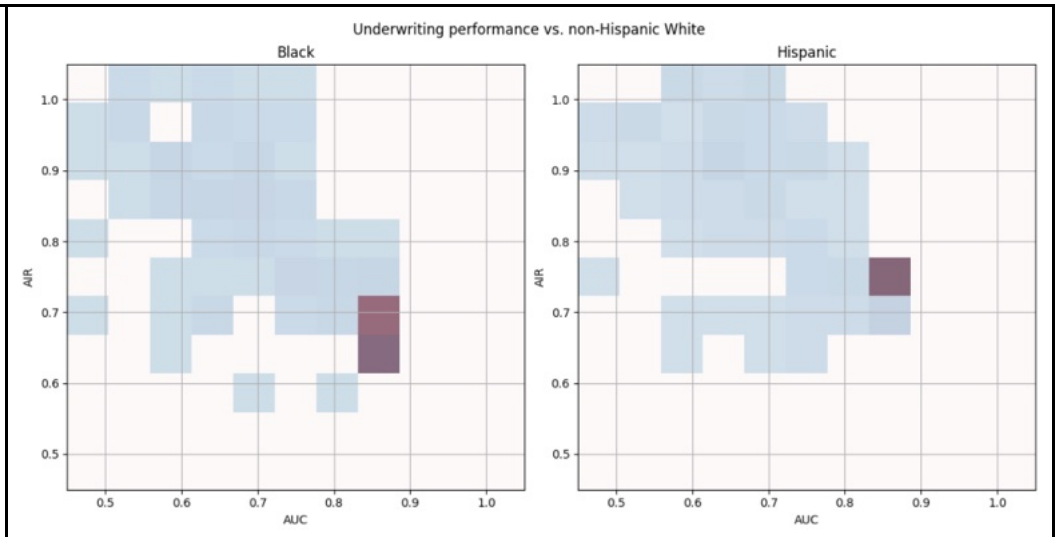
One or more **Constrained** models but no Unconstrained models in that performance range

One or more **Unconstrained** models but no Constrained models in that performance range

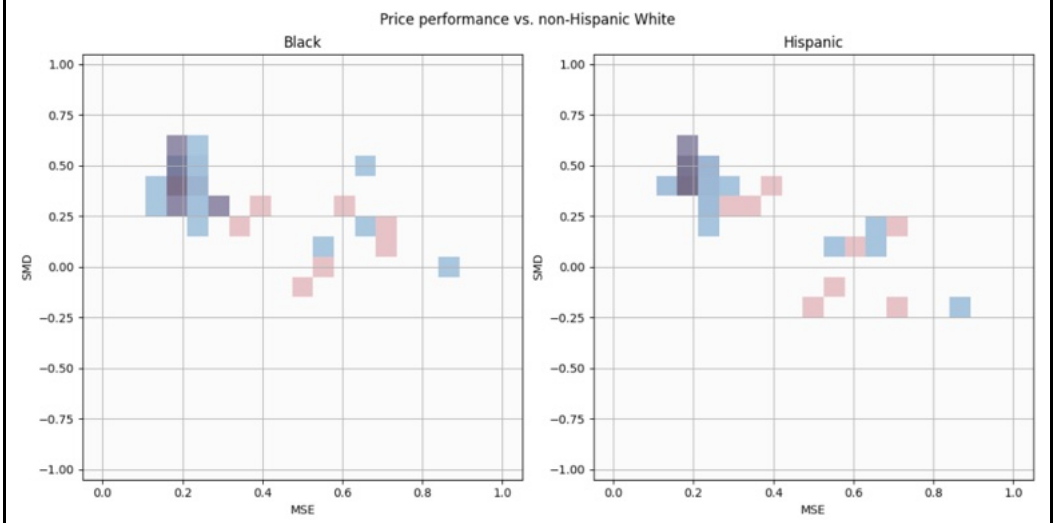
Both **Constrained and Unconstrained** models in that performance range.

Figure 6: LA county only, logistic regression or ReLU fit

(a) Underwriting



(b) Pricing



One or more **Constrained** models but no Unconstrained models in that performance range

One or more **Unconstrained** models but no Constrained models in that performance range

Both **Constrained and Unconstrained** models in that performance range.

Feature Importance

Contributors to the Final Outputs

Tables 5 through 8 show the estimated contributions of the most important variables to the output of the best linear and non-linear Unconstrained and Constrained underwriting models for both the entire U.S. and the LA metro area. Tables 9 through 12 show the estimated contributions of the 10 most important variables to the pricing models. Expanded tables covering more of these values are included in Appendix B. These variable importances were computed by using XGBoost to create a surrogate for each of the best underwriting and pricing models[35] and then using the SHAP package to generate a set of outcome contributors. These contributors were then organized in inverse order of their absolute values, and then the normalized sum of those was computed to measure the fractional impact of each variable.

Table 5: Variable contributions to the best linear underwriting functions[36] for the entire U.S.

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
fico	58.87	fico	56.66
original_ltv	20.06	original_ltv	17.41
back_end_ratio	5.14	back_end_ratio	4.69
appraised_value	2.70	channel_D	2.18
occupancy_type_U	2.12	payment_frequency_4	1.85
channel_D	1.62	property_type_2	1.69
payment_frequency_4	1.03	channel_1	1.26
property_type_2	0.96	channel_2	1.12
channel_2	0.72	appraised_value	1.09
statecode_CA	0.66	number_of_units	0.95

[35] To facilitate explainability, these models were trained to be surrogates for the original neural network or linear models. SHAP was then used to explain the impact of changes in the original variables on the outputs of the surrogate models and those values, in turn, were used to estimate the contributions of the input variables to the outputs of the original models.

[36] "Linear underwriting function" both here and in Tables 7, 9, and 11 refers to the output of a linear underwriting model.

Table 6: Variable contributions to the best nonlinear underwriting functions[37] for the entire U.S.

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
fico	58.79	fico	51.79
original_ltv	19.50	original_ltv	13.06
back_end_ratio	5.45	back_end_ratio	7.33
occupancy_type_U	2.92	channel_2	6.39
appraised_value	2.84	appraised_value	4.66
channel_D	1.28	number_of_units	4.24
property_type_2	1.24	payment_frequency_U	1.42
coapplicant_present	1.05	gse_eligible_flag_1	1.33
channel_2	0.94	coapplicant_present	1.22
payment_frequency_U	0.65	payment_frequency_4	1.02

Table 7: Variable contributions to the best linear underwriting functions for the LA Metro area

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
fico	50.23	fico	55.72
original_ltv	18.30	original_ltv	21.58
appraised_value	5.95	channel_1	2.60
back_end_ratio	5.06	appraised_value	2.34
occupancy_type_U	4.57	occupancy_type_1	2.18
gse_eligible_flag_1	4.47	occupancy_type_U	2.14
channel_D	1.77	gse_eligible_flag_1	1.98
payment_frequency_4	1.59	back_end_ratio	1.89
coapplicant_present	1.45	documentation_type_2	1.73
documentation_type_2	1.10	channel_D	1.09

[37] "Nonlinear underwriting function" both here and in Tables 8, 10, and 12 refers to the output of a nonlinear underwriting model.

Table 8: Variable contributions to the best nonlinear underwriting functions for the LA Metro area

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
fico	51.18	fico	49.16
original_ltv	21.73	original_ltv	12.09
occupancy_type_U	4.31	number_of_units	6.63
appraised_value	3.70	back_end_ratio	4.27
back_end_ratio	3.34	gse_eligible_flag_2	3.40
coapplicant_present	2.72	occupancy_type_U	3.27
gse_eligible_flag_1	2.34	property_type_4	3.14
channel_D	1.90	appraised_value	3.13
gse_eligible_flag_0	1.44	channel_2	3.10
documentation_type_2	1.27	property_type_1	2.51

Table 9: Variable contributions to the best linear pricing functions for the entire U.S.

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
appraised_value	24.25	fico	30.13
channel_D	8.40	occupancy_type_3	17.07
channel_1	6.26	property_type_5	6.10
statecode_NJ	6.11	appraised_value	6.05
back_end_ratio	6.09	occupancy_type_1	5.25
fico	5.58	property_type_6	4.03
gse_eligible_flag_0	4.97	channel_2	3.73
statecode_NY	3.41	property_type_2	3.42
statecode_CA	3.40	channel_D	3.17
product_type_category_A	3.35	channel_9	2.69

Table 10: Variable contributions to the best nonlinear pricing functions for the entire U.S.

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
back_end_ratio	17.84	back_end_ratio	21.80
product_type_category_A	16.71	appraised_value	15.61
channel_D	12.49	payment_frequency_4	13.40
channel_2	7.72	fico	12.95
fico	7.47	channel_2	8.00
coapplicant_present	6.54	coapplicant_present	5.17
occupancy_type_3	4.76	occupancy_type_1	4.00
appraised_value	4.62	product_type_category_A	3.83
product_type_category_F	4.19	channel_9	3.39
number_of_units	4.06	number_of_units	2.94

Table 11. Variable contributions to the best linear pricing functions for LA Metro area

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
appraised_value	24.64	number_of_units	13.34
gse_eligible_flag_0	18.76	channel_2	12.83
occupancy_type_3	5.89	property_type_4	10.78
gse_eligible_flag_2	5.24	channel_D	10.44
original_ltv	4.82	product_type_category_F	6.58
documentation_type_1	4.58	appraised_value	6.49
product_type_category_A	4.50	back_end_ratio	5.21
io_flag_N	3.77	property_type_1	4.92
gse_eligible_flag_1	3.73	occupancy_type_3	3.88
channel_D	3.70	gse_eligible_flag_0	3.54

Table 12. Variable contributions to the best nonlinear pricing functions for LA Metro area

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
gse_eligible_flag_1	23.94	channel_D	24.29
gse_eligible_flag_0	17.60	gse_eligible_flag_0	9.63
channel_D	16.34	occupancy_type_1	8.84
channel_2	7.66	occupancy_type_3	8.10
product_type_category_A	5.87	fico	6.91
fico	5.77	product_type_category_A	5.97
occupancy_type_3	4.19	product_type_category_F	4.81
occupancy_type_1	3.95	property_type_1	4.59
original_ltv	3.06	number_of_units	4.43
payment_frequency_4	2.24	property_type_U	3.58

Performance Summary

Comparing the Overall Performance of the Unconstrained and Constrained Models

Table 13 shows the overall performance of the best Unconstrained and Constrained models for both underwriting and pricing. The two left-hand columns show the statistics for the best Unconstrained and Constrained linear and nonlinear underwriting models, respectively. The two right-hand columns show the relative improvement or deterioration between the corresponding Constrained and Unconstrained models. Table 14 shows the corresponding values for the best Constrained and Unconstrained pricing models.

Table 13: Performance summary for Underwriting models

		Model type		Relative change	
		Linear	*Nonlinear*	*Linear*	*Nonlinear*
US	**AUC (unconstrained)**	0.8	0.8		
	AIR (unconstrained)	0.75	0.75		
	AUC (constrained)	0.8	0.8	0.00%	0.00%
	AIR (constrained)	0.8	0.85	6.67%	13.33%
LA	**AUC (unconstrained)**	0.85	0.85		
	AIR (unconstrained)	0.65	0.65		
	AUC (constrained)	0.85	0.85	0.00%	0.00%
	AIR (constrained)	0.75	0.75	15.38%	15.38%

Table 14: Performance summary for Pricing models

		Model type		Relative change	
		Linear	*Nonlinear*	*Linear*	*Nonlinear*
US	**MSE (unconstrained)**	0.35	0.4		
	SMD (unconstrained)	0.3	0.45		
	MSE (constrained)	0.35	0.35	0.00%	14.29%
	SMD (constrained)	0.2	0.35	50.00%	28.57%
LA	**MSE (unconstrained)**	0.15	0.15		
	SMD (unconstrained)	0.55	0.45		
	MSE (constrained)	0.15	0.15	0.00%	0.00%
	SMD (constrained)	0.4	0.45	37.50%	0.00%

Discussion

Behavior of Underwriting Scoring Functions

For an AUC versus AIR plot at a fixed approval rate, one would expect that an efficient frontier would trend up and to the left, indicating a tradeoff between higher fairness and lower accuracy, with a fairer model, in general, being less accurate than a less fair model. In the case of the Constrained underwriting models, there is a clear Pareto frontier of models that traverse the accuracy-fairness spectrum. Interestingly, there is also a multiplicity of fairer Constrained models with identical accuracy to the second decimal of the Unconstrained models. This can be observed in the Figures 3(a) through 6(a): the Unconstrained models all have their AUC/AIR points aggregated in the lower right of the plot, whereas there are Constrained models which have their AUC/AIR points in a vertical stripe directly above the Unconstrained models.

Behavior of Pricing Functions

For an MSE versus SMD plot with the same training and testing data, one would expect that an efficient frontier would run towards an SMD of 0 and towards a larger MSE, indicating a tradeoff between fairness and accuracy. As discussed below, this is not seen. However, as in the case of underwriting models, there are several Constrained models with accuracies as good or better than the most accurate Unconstrained models, but which are fairer.

Performance of the distribution matching technique

Interpreting the Constrained underwriting models

It's clear that the Constrained underwriting models depend on the distribution matching algorithm to find the Pareto frontier: the Unconstrained underwriting models form a small cluster around a certain AIR-AUC point whereas the Constrained models show a clear up-and-to-the-left fairness accuracy tradeoff. Importantly, there are many Constrained models with AUCs that are indistinguishable from those of several Unconstrained models, but which nevertheless have higher AIRs from whence the efficient frontier can originate, demonstrating that the distribution matching method can create underwriting fairness gains in excess of 13 percent without sacrificing accuracy.

In the cases where the performance of a Constrained model exceeds the performance of the best Unconstrained model, it is worth noting how similar the different models are. The variables that contribute the most to the two models are nearly identical. Moreover, the contributions of the two most important variables — the primary applicant's FICO score at time of origination and the original loan to value ratio — are very close together. Since these two variables contribute close

to 70 percent of the final decisions of both models, it is reasonable to conclude that the most accurate and fair Constrained model is only a small perturbation of the most accurate and fair Unconstrained model.

Interpreting the Constrained pricing models

It is challenging to interpret the Unconstrained or the Constrained pricing models that emerged from the training process. First, there does not appear to be an obvious Pareto frontier in the behavior of the pricing models. Second, the MSEs of both the Constrained and Unconstrained pricing models were unexceptional. Third, mortgage pricing is known to rely on data elements such as the prevailing rate on the 10-year U.S. Treasury bond, which were not available in the data. Nevertheless, the distribution matching technique appears to generate Constrained pricing models that are fairer and more accurate than any of the Unconstrained pricing models.

Limitations of the available data

The data available to perform this study had some limitations.

First, the dataset contained no unapproved loan applications, which makes it challenging to examine the behavior of the underwriting function accurately: it is impossible to determine which loans were denied historically but would have been approved with the new scoring function. In addition, the lack of unoriginated applications eliminates a powerful source of information that the Distribution Matching code could have exploited: the DM code can consume all applications, not merely those that were approved. In other work, the Project Team has found that the inclusion of data on unapproved loans can yield LDA candidates that are fairer and more accurate than those constructed without it.

As discussed above, the Project Team considered adding records of unapproved loans drawn from the HMDA database to strengthen its analysis but concluded that those records did not contain enough information about each application to be useful because they lacked essential input features.

In addition, there is virtually no information available in the dataset about the applicant or applicants: the two variables available at origination that directly relate to applicants are their FICO score and the back-end ratio of the loan. Normally, at underwriting time, lenders have direct access to information about the applicant's income, employment history, payment history and other financial indicators. Without such data about the applicant and co-applicant (if any), underwriting models will tend to underperform. In addition, the dataset contained no information about the macroeconomic environment at the time of application. That is a material omission since mortgages are, in part, often priced on the basis of the prevailing interest rate on the U.S. 10-year U.S. Treasury bond. Without data about that rate and the stability of that rate, mortgage pricing models will be less reliable.

Another limitation of the data used for this study: it was created via a statistical merge process which combined data from the HMDA database with a proprietary servicing dataset owned by CoreLogic®. The merge process by which the dataset was constructed resulted in individual records from either dataset matching more than one record from the other dataset. To give a sense of scale for this problem, about one HMDA record in 12 appears more than once in the training data, and, since each duplicated record winds up in the training or testing sets more than once, its presence distorts the statistics of the training set.

This is not an uncommon problem when using sampled data and is often ameliorated by weighting each record in the sampled dataset in a way that captures its probabilistic frequency. The Project Team could not, however, determine how to weight records: not only could a single HMDA record match multiple records from the database of originated loans, but a single record could match multiple HMDA records. Without the exact details of the merger counts, it is impossible to determine how to optimally weight each record in the resulting statistical merge.

Conclusion

The Project Team's Preliminary Findings are that Distribution Matching, whereby a model learns during the training process that the distribution of outputs for protected groups and control groups ought to closely resemble each other, can increase mortgage underwriting and pricing fairness in excess of 13 percent with no functional diminution in accuracy. Thus, the Preliminary Findings of this study demonstrate a potentially viable pathway for integrating disparity minimization and other public policy goals into algorithmic decision-making without sacrificing performance, an approach that aligns with emerging regulatory frameworks for artificial intelligence and societal calls for more equitable housing and financial practices.

Appendix A

Data field description for CoreLogic® dataset

Here is the interpretation of each variable name into English:

1. `loan_id`: A unique identifier for a loan.
2. `add_date`: The date when the loan information was added or recorded.
3. `property_zip`: The ZIP code of the property associated with the loan.
4. `statecode`: The US Postal Code representing the state where the property is located.
5. `property_type`: The type or category of the property (e.g., single-family home, condominium).
6. `number_of_units`: The total number of units or housing units within the property.
7. `occupancy_type`: The type of occupancy for the property (e.g., owner-occupied, rental).
8. `origination_date`: The date when the loan was initially originated or approved.
9. `maturity_date`: The date when the loan is scheduled to mature or be fully paid off.
10. `first_payment_date`: The date when the first payment on the loan is due.
11. `original_balance`: Funded amount for this loan provided at the time of origination.
12. `sale_price`: The price at which the property was sold.
13. `appraised_value`: Reported fair market value of a property.
14. `product_type`: The type of loan product (e.g., fixed-rate, adjustable-rate, Interest only).
15. `original_term`: The original duration of the loan in months or years.
16. `initial_interest_rate`: The initial interest rate on the loan at origination.
17. `back_end_ratio`: Total of all debt payments including the new mortgage payment (principal, interest, insurance and taxes, (PITI)) divided by the gross monthly income of the borrower(s).
18. `loan_type`: The type of loan (e.g., FHA, conventional).
19. `loan_purpose`: Borrower's stated purpose for the loan.
20. `payment_frequency`: How often loan payments are made (e.g., monthly, biweekly).
21. `channel`: Lender's origination source of the loan.
22. `buydown_flag`: Indicates situations where the borrower paid additional points at closing in order to obtain a reduction to the interest rate.
23. `documentation_type`: The type of documentation provided for the loan application.
24. `pmi_company_code`: The code representing the private mortgage insurance (PMI) company.
25. `pool_insurance_flag`: Indicates loans that are covered by a supplemental mortgage insurance
covering a pool of loans as opposed to loan-level mortgage insurance.
26. `original_ltv`: Original Loan To Value. Original mortgage amount divided by the lesser of the origination appraised value or the sales price..
27. `convertible_flag`:.Indicates whether the borrower has an option to convert their ARM mortgage to a fixed rate loan.

28. `origination_ltv`: The loan-to-value ratio at the time of origination. Original mortgage amount divided by the lesser of the origination appraised value or the sales price.
29. `negative_amortization_flag`: A flag indicating whether the loan has negative amortization.
30. `arm_index_id`: Published financial index name used as a basis to determine the interest rate of the loan.
31. `margin`: The margin added to the ARM index to determine the interest rate.
32. `periodic_rate_cap`: Limit on how much the interest rate can increase during any one adjustment period regardless of the margin and index.
33. `periodic_rate_floor`: Limit on how much the interest rate can decrease during any one adjustment period regardless of the margin and index.
34. `lifetime_rate_cap`: The maximum allowable interest rate adjustment over the life of an ARM.
35. `lifetime_rate_floor`: The minimum allowable interest rate adjustment over the life of an ARM.
36. `rate_reset_frequency`:Number of months between rate resets for adjustable rate loans.
37. `pay_reset_frequency`: How often the payment amount on an ARM is reset.
38. `first_rate_reset_period`: Number of months between payment resets for adjustable rate loans.
39. `fico_score_at_origination`: Borrower's FICO credit score at the time of loan origination.
40. `lien`: The type of lien associated with the loan (e.g., first lien, second lien).
41. `prepay_penalty_flag`: A flag indicating whether there is a prepayment penalty associated with the loan.
42. `prepay_penalty_term`: The term or duration of the prepayment penalty.
43. `combined_ltv_at_origination`: The combined loan-to-value ratio at the time of origination.
44. `cbsa`: The Core-Based Statistical Area where the property is located.
45. `io_term`: The term or duration of an interest-only payment period, if applicable.
46. `io_flag`: A flag indicating whether the loan includes an interest-only payment period.
47. `msa`: The Metropolitan Statistical Area where the property is located.
48. `paid_off_flag`: A flag indicating whether the loan has been paid off.
49. `inferred_collateral_type`: Identifies whether the loan is Prime or Subprime as defined by CoreLogic®.
50. `collateral_type`: Identifies whether the loan is Prime or Subprime as defined by contributor.
51. `orig_active_status`: Active status at Origination.
52. `period`: Reporting period as of the date of the data. (Represented as a CoreLogic® defined numeric value.).
53. `product_type_category`: Product type for the loan (Fixed, ARM, or Unknown).
54. `loan_purpose_category`: Summarized purpose of the loan based on aggregation of the primary Loan Purpose field.
55. `mortgage_insurance_flag`: Indicates the presence of mortgage insurance at origination.
56. `gse_eligible_flag`: A flag indicating whether the loan is eligible for purchase by government-sponsored enterprises (GSEs).

57. `data_year`: The year in which the loan data is recorded.
58. `applicant_ethnicity`: The ethnicity of the loan applicant.
59. `coapplicant_ethnicity`: The ethnicity of the co-applicant, if applicable.
60. `applicant_race`: The race of the loan applicant.
61. `coapplicant_race`: The race of the co-applicant, if applicable.
62. `applicant_sex`: The gender of the loan applicant.
63. `coapplicant_sex`: The gender of the co-applicant, if applicable.
64. `applicant_age`: The age of the loan applicant.
65. `co_applicant_age`: The age of the co-applicant, if applicable.
66. `applicant_age_above_62`: A flag indicating whether the loan applicant is above the age of 62.
67. `co_applicant_age_above_62`: A flag indicating whether the co-applicant is above the age of 62
68. `active_status`: The current active status of the loan.
69. `epd_fha`: Indicates early payment default using Federal Housing Administration (FHA) methodology.
70. `epd_gse`: Indicates early payment default using Government Sponsored Entity (GSE) methodology.
71. `foreclosure_start_date`: The date when a foreclosure process on the property started.
72. `foreclosure_end_date`: The date when a foreclosure process on the property ended.
73. `bankruptcy_start_date`: The date when a bankruptcy process started, if applicable.
74. `bankruptcy_end_date`: The date when a bankruptcy process ended, if applicable.
75. `bankruptcy_chapter`: The chapter of bankruptcy, if applicable.
76. `payoff_period`: Period when loan was first paid off since last time MBA Delinquency Status was 30, 60 or 90 Days delinquent, Current, Foreclosed and Current Balance was greater than $0.
77. `payoff_date`: The date when the loan is fully paid off.
78. `first_period_30_days_delinquent`: The first period when the loan became 30 days delinquent.
79. `first_period_60_days_delinquent`: The first period when the loan became 60 days delinquent.
80. `first_period_90_days_delinquent`: The first period when the loan became 90 days delinquent.

Appendix B

Variable contributions for nonlinear models for underwriting for the entire U.S. for the best Unconstrained and Constrained models

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
fico	58.79	fico	51.79
original_ltv	19.50	original_ltv	13.06
back_end_ratio	5.45	back_end_ratio	7.33
occupancy_type_U	2.92	channel_2	6.39
appraised_value	2.84	appraised_value	4.66
channel_D	1.28	number_of_units	4.24
property_type_2	1.24	payment_frequency_U	1.42
coapplicant_present	1.05	gse_eligible_flag_1	1.33
channel_2	0.94	coapplicant_present	1.22
payment_frequency_U	0.65	payment_frequency_4	1.02
payment_frequency_4	0.56	channel_U	0.91
statecode_CA	0.50	occupancy_type_1	0.82
gse_eligible_flag_2	0.47	channel_1	0.75
statecode_CO	0.38	channel_D	0.47
statecode_NV	0.33	statecode_CA	0.44
channel_U	0.29	statecode_CO	0.35
channel_9	0.25	property_type_5	0.34
statecode_IL	0.18	gse_eligible_flag_0	0.26
statecode_AZ	0.18	statecode_AR	0.25
io_flag_N	0.17	property_type_6	0.24
property_type_M	0.16	statecode_FL	0.23
statecode_WA	0.15	io_flag_N	0.20
statecode_CT	0.14	statecode_UT	0.18
statecode_NC	0.13	statecode_WA	0.18

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
combined_ltv_at_origination	0.13	property_type_4	0.15
statecode_MN	0.09	statecode_OH	0.15
statecode_OR	0.08	property_type_2	0.14
property_type_1	0.07	combined_ltv_at_origination	0.12
statecode_NY	0.07	statecode_AZ	0.12
statecode_NE	0.07	property_type_U	0.10
statecode_MD	0.06	statecode_OK	0.09
statecode_TX	0.06	statecode_IL	0.08
statecode_VT	0.06	gse_eligible_flag_2	0.08
statecode_FL	0.05	statecode_IN	0.08
documentation_type_2	0.05	statecode_NV	0.07
statecode_OH	0.05	occupancy_type_3	0.07
statecode_AL	0.05	statecode_NJ	0.06
statecode_LA	0.05	statecode_MI	0.05
gse_eligible_flag_0	0.04	statecode_CT	0.05
statecode_ID	0.04	statecode_WI	0.05
statecode_IN	0.03	statecode_MN	0.04
statecode_AK	0.03	statecode_NC	0.04
occupancy_type_1	0.03	statecode_TX	0.04
statecode_MO	0.03	channel_9	0.04
statecode_MT	0.03	statecode_NE	0.03
statecode_NJ	0.03	property_type_1	0.03
gse_eligible_flag_U	0.03	statecode_GA	0.03
property_type_U	0.03	statecode_MO	0.02

Variable contributions for linear models for underwriting for the entire U.S. for the best Unconstrained and Constrained models

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
fico	58.87	fico	56.66
original_ltv	20.06	original_ltv	17.41
back_end_ratio	5.14	back_end_ratio	4.69
appraised_value	2.70	channel_D	2.18
occupancy_type_U	2.12	payment_frequency_4	1.85
channel_D	1.62	property_type_2	1.69
payment_frequency_4	1.03	channel_1	1.26
property_type_2	0.96	channel_2	1.12
channel_2	0.72	appraised_value	1.09
statecode_CA	0.66	number_of_units	0.95
coapplicant_present	0.55	statecode_OK	0.89
channel_U	0.46	property_type_U	0.74
payment_frequency_U	0.45	statecode_CA	0.74
channel_9	0.43	occupancy_type_U	0.71
statecode_CO	0.40	gse_eligible_flag_0	0.68
gse_eligible_flag_2	0.33	combined_ltv_at_origination	0.68
statecode_NV	0.30	channel_U	0.63
io_flag_U	0.20	statecode_IL	0.53
combined_ltv_at_origination	0.19	statecode_IN	0.52
statecode_MT	0.19	coapplicant_present	0.46
statecode_AZ	0.18	channel_9	0.43
statecode_WA	0.16	statecode_NC	0.31
property_type_M	0.15	product_type_category_U	0.28
statecode_MN	0.14	statecode_ND	0.26
channel_1	0.12	statecode_AZ	0.22
statecode_MD	0.12	statecode_LA	0.22

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
occupancy_type_2	0.12	statecode_NE	0.21
statecode_TX	0.09	payment_frequency_U	0.19
statecode_CT	0.08	statecode_OH	0.19
statecode_NY	0.08	statecode_SD	0.18
statecode_AL	0.08	property_type_1	0.18
statecode_GA	0.08	occupancy_type_1	0.13
statecode_NC	0.08	statecode_OR	0.12
occupancy_type_1	0.07	statecode_ME	0.11
statecode_NE	0.06	statecode_VA	0.10
property_type_5	0.06	statecode_AR	0.10
property_type_4	0.06	property_type_4	0.10
statecode_OR	0.06	occupancy_type_3	0.10
io_flag_N	0.06	io_flag_N	0.10
statecode_MA	0.05	statecode_MN	0.10
statecode_AR	0.05	statecode_NJ	0.09
statecode_FL	0.05	statecode_NM	0.08
statecode_LA	0.05	statecode_AK	0.07
statecode_MO	0.04	statecode_MD	0.07
statecode_VT	0.04	statecode_AL	0.06
statecode_ID	0.04	statecode_KS	0.05
statecode_IN	0.04	statecode_MA	0.05
property_type_U	0.03	statecode_GA	0.05

Variable contributions for nonlinear models for underwriting for LA County

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
fico	51.18	fico	49.16
original_ltv	21.73	original_ltv	12.09
occupancy_type_U	4.31	number_of_units	6.63
appraised_value	3.70	back_end_ratio	4.27
back_end_ratio	3.34	gse_eligible_flag_2	3.40
coapplicant_present	2.72	occupancy_type_U	3.27
gse_eligible_flag_1	2.34	property_type_4	3.14
channel_D	1.90	appraised_value	3.13
gse_eligible_flag_0	1.44	channel_2	3.10
documentation_type_2	1.27	property_type_1	2.51
channel_2	1.21	payment_frequency_4	1.97
combined_ltv_at_origination	1.15	product_type_category_A	1.75
product_type_category_A	0.60	gse_eligible_flag_1	1.38
channel_9	0.44	occupancy_type_1	0.53
io_flag_U	0.40	channel_D	0.49
channel_1	0.36	payment_frequency_U	0.39
channel_U	0.34	property_type_2	0.37
property_type_6	0.31	coapplicant_present	0.35
property_type_1	0.30	occupancy_type_3	0.34
property_type_4	0.19	io_flag_U	0.31
payment_frequency_U	0.17	gse_eligible_flag_0	0.28
io_flag_N	0.11	property_type_6	0.22
payment_frequency_4	0.11	documentation_type_1	0.18
gse_eligible_flag_2	0.08	combined_ltv_at_origination	0.17
documentation_type_U	0.08	product_type_category_F	0.15
number_of_units	0.07	documentation_type_U	0.13
property_type_2	0.06	property_type_U	0.07

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
gse_eligible_flag_U	0.04	channel_U	0.06
product_type_category_F	0.02	io_flag_N	0.06
io_flag_Y	0.02	channel_1	0.03
channel_3	0.01	property_type_5	0.03
occupancy_type_1	0.01	io_flag_Y	0.01
property_type_U	0.01	documentation_type_2	0.01
documentation_type_1	0.01	occupancy_type_2	0.00
io_term	0.01	product_type_category_U	0.00
occupancy_type_2	0.00	product_type_category_nan	0.00
payment_frequency_2	0.00	io_flag_nan	0.00
statecode_CA	0.00	documentation_type_nan	0.00
statecode_nan	0.00	gse_eligible_flag_U	0.00
product_type_category_nan	0.00	channel_3	0.00
product_type_category_U	0.00	channel_nan	0.00
io_flag_nan	0.00	channel_9	0.00
documentation_type_nan	0.00	property_type_3	0.00
property_type_3	0.00	io_term	0.00
property_type_5	0.00	payment_frequency_nan	0.00
occupancy_type_nan	0.00	payment_frequency_2	0.00

Variable contributions for linear models for underwriting for LA County

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
fico	50.23	fico	55.72
original_ltv	18.30	original_ltv	21.58
appraised_value	5.95	channel_1	2.60
back_end_ratio	5.06	appraised_value	2.34
occupancy_type_U	4.57	occupancy_type_1	2.18
gse_eligible_flag_1	4.47	occupancy_type_U	2.14
channel_D	1.77	gse_eligible_flag_1	1.98
payment_frequency_4	1.59	back_end_ratio	1.89
coapplicant_present	1.45	documentation_type_2	1.73
documentation_type_2	1.10	channel_D	1.09
channel_1	0.96	gse_eligible_flag_0	1.02
combined_ltv_at_origination	0.78	channel_2	0.99
channel_9	0.56	property_type_2	0.81
channel_2	0.54	property_type_1	0.75
gse_eligible_flag_0	0.40	coapplicant_present	0.70
property_type_1	0.31	product_type_category_F	0.55
property_type_Z	0.28	payment_frequency_U	0.29
channel_U	0.25	property_type_4	0.25
number_of_units	0.22	property_type_6	0.24
io_flag_U	0.22	occupancy_type_3	0.20
property_type_6	0.18	product_type_category_A	0.19
io_flag_N	0.15	payment_frequency_4	0.19
occupancy_type_1	0.13	channel_U	0.14
property_type_2	0.12	property_type_U	0.11
documentation_type_1	0.11	combined_ltv_at_origination	0.10
io_flag_Y	0.08	documentation_type_1	0.10
documentation_type_U	0.06	number_of_units	0.08

gse_eligible_flag_2	0.05	io_flag_N	0.03
product_type_category_A	0.05	gse_eligible_flag_2	0.01
property_type_U	0.04	payment_frequency_2	0.00
payment_frequency_U	0.01	gse_eligible_flag_U	0.00
occupancy_type_2	0.01	io_term	0.00
gse_eligible_flag_U	0.01	statecode_CA	0.00
documentation_type_nan	0.00	product_type_category_nan	0.00
product_type_category_F	0.00	product_type_category_U	0.00
io_flag_nan	0.00	statecode_nan	0.00
product_type_category_nan	0.00	property_type_3	0.00
statecode_CA	0.00	io_flag_nan	0.00
io_term	0.00	io_flag_Y	0.00
product_type_category_U	0.00	io_flag_U	0.00
channel_nan	0.00	documentation_type_nan	0.00
statecode_nan	0.00	occupancy_type_nan	0.00
property_type_nan	0.00	documentation_type_U	0.00
property_type_3	0.00	property_type_5	0.00
property_type_4	0.00	property_type_7	0.00
channel_3	0.00	channel_nan	0.00
property_type_5	0.00	property_type_M	0.00
payment_frequency_nan	0.00	property_type_Z	0.00

Variable contributions for nonlinear models for pricing for the entire U.S.

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
back_end_ratio	17.84	back_end_ratio	21.80
product_type_category_A	16.71	appraised_value	15.61
channel_D	12.49	payment_frequency_4	13.40
channel_2	7.72	fico	12.95
fico	7.47	channel_2	8.00
coapplicant_present	6.54	coapplicant_present	5.17
occupancy_type_3	4.76	occupancy_type_1	4.00
appraised_value	4.62	product_type_category_A	3.83
product_type_category_F	4.19	channel_9	3.39
number_of_units	4.06	number_of_units	2.94
gse_eligible_flag_1	2.42	property_type_5	2.54
combined_ltv_at_origination	2.21	payment_frequency_U	1.85
gse_eligible_flag_0	1.97	statecode_MO	1.22
occupancy_type_1	1.32	product_type_category_F	0.78
statecode_MN	0.98	gse_eligible_flag_1	0.65
property_type_6	0.95	occupancy_type_2	0.54
statecode_MO	0.84	statecode_PA	0.44
statecode_CA	0.75	statecode_MN	0.33
original_ltv	0.41	original_ltv	0.32
payment_frequency_U	0.35	statecode_AZ	0.24
statecode_NC	0.25	property_type_nan	0.00
channel_1	0.25	occupancy_type_nan	0.00
statecode_IA	0.24	occupancy_type_U	0.00
statecode_MD	0.23	occupancy_type_3	0.00
statecode_OH	0.22	gse_eligible_flag_U	0.00
channel_U	0.16	product_type_category_nan	0.00
payment_frequency_4	0.04	property_type_Z	0.00

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
documentation_type_1	0.01	payment_frequency_2	0.00
statecode_AZ	0.00	property_type_M	0.00
io_flag_nan	0.00	property_type_7	0.00
occupancy_type_2	0.00	property_type_6	0.00
gse_eligible_flag_2	0.00	property_type_4	0.00
property_type_nan	0.00	property_type_3	0.00
property_type_Z	0.00	property_type_2	0.00
io_flag_N	0.00	property_type_1	0.00
property_type_U	0.00	property_type_U	0.00
property_type_M	0.00	channel_1	0.00
occupancy_type_U	0.00	gse_eligible_flag_2	0.00
property_type_7	0.00	payment_frequency_nan	0.00
product_type_category_nan	0.00	product_type_category_U	0.00
property_type_5	0.00	gse_eligible_flag_0	0.00
property_type_4	0.00	io_flag_nan	0.00
property_type_3	0.00	io_flag_Y	0.00
product_type_category_U	0.00	io_flag_U	0.00
io_flag_Y	0.00	io_flag_N	0.00
occupancy_type_nan	0.00	documentation_type_nan	0.00
payment_frequency_2	0.00	documentation_type_U	0.00
documentation_type_nan	0.00	documentation_type_3	0.00

Variable contributions for linear models for pricing for the entire U.S.

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
appraised_value	24.25	fico	30.13
channel_D	8.40	occupancy_type_3	17.07
channel_1	6.26	property_type_5	6.10
statecode_NJ	6.11	appraised_value	6.05
back_end_ratio	6.09	occupancy_type_1	5.25
fico	5.58	property_type_6	4.03
gse_eligible_flag_0	4.97	channel_2	3.73
statecode_NY	3.41	property_type_2	3.42
statecode_CA	3.40	channel_D	3.17
product_type_category_A	3.35	channel_9	2.69
statecode_AZ	2.86	statecode_IN	2.04
statecode_IN	2.80	gse_eligible_flag_2	1.73
statecode_OK	2.44	statecode_KS	1.36
channel_9	2.28	statecode_AL	1.31
statecode_MI	1.73	product_type_category_A	1.23
statecode_AL	1.71	product_type_category_F	1.20
gse_eligible_flag_2	1.20	statecode_NV	1.09
occupancy_type_1	1.06	statecode_AR	0.96
property_type_3	0.91	statecode_NH	0.90
documentation_type_2	0.88	combined_ltv_at_origination	0.88
statecode_WI	0.82	documentation_type_U	0.85
statecode_NV	0.81	statecode_GA	0.66
statecode_FL	0.77	statecode_NM	0.62
property_type_U	0.71	statecode_NJ	0.49
statecode_MO	0.65	statecode_WA	0.44
statecode_MD	0.64	number_of_units	0.42
product_type_category_U	0.64	property_type_1	0.36
channel_3	0.60	original_ltv	0.32

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
number_of_units	0.53	io_flag_Y	0.29
statecode_NC	0.51	statecode_PA	0.24
io_flag_Y	0.47	statecode_WI	0.19
statecode_VA	0.39	statecode_MD	0.18
channel_U	0.35	channel_3	0.13
statecode_NE	0.33	statecode_ID	0.11
statecode_TN	0.30	statecode_RI	0.10
original_ltv	0.22	payment_frequency_4	0.07
property_type_M	0.20	back_end_ratio	0.06
gse_eligible_flag_U	0.18	property_type_U	0.04
occupancy_type_2	0.17	property_type_M	0.04
statecode_WV	0.17	property_type_4	0.03
channel_2	0.14	gse_eligible_flag_0	0.02
documentation_type_1	0.12	payment_frequency_2	0.00
coapplicant_present	0.11	channel_1	0.00
property_type_1	0.11	payment_frequency_U	0.00
statecode_OR	0.10	payment_frequency_nan	0.00
gse_eligible_flag_1	0.07	documentation_type_1	0.00
statecode_GA	0.04	property_type_7	0.00
io_term	0.04	channel_nan	0.00

Variable contributions for nonlinear models for pricing for LA County

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
gse_eligible_flag_1	23.94	channel_D	24.29
gse_eligible_flag_0	17.60	gse_eligible_flag_0	9.63
channel_D	16.34	occupancy_type_1	8.84
channel_2	7.66	occupancy_type_3	8.10
product_type_category_A	5.87	fico	6.91
fico	5.77	product_type_category_A	5.97
occupancy_type_3	4.19	product_type_category_F	4.81
occupancy_type_1	3.95	property_type_1	4.59
original_ltv	3.06	number_of_units	4.43
payment_frequency_4	2.24	property_type_U	3.58
number_of_units	1.89	original_ltv	2.93
product_type_category_F	1.79	channel_2	2.44
property_type_2	1.68	property_type_2	1.88
appraised_value	1.45	occupancy_type_2	1.75
property_type_U	0.63	back_end_ratio	1.56
documentation_type_U	0.51	channel_1	1.44
documentation_type_1	0.48	documentation_type_U	1.12
back_end_ratio	0.42	io_flag_N	1.10
property_type_1	0.40	payment_frequency_U	1.01
io_flag_U	0.05	coapplicant_present	0.97
property_type_M	0.05	appraised_value	0.92
combined_ltv_at_origination	0.03	channel_9	0.69
documentation_type_2	0.00	gse_eligible_flag_2	0.59
documentation_type_nan	0.00	payment_frequency_4	0.14
io_flag_N	0.00	property_type_4	0.09
gse_eligible_flag_U	0.00	property_type_6	0.06
io_flag_Y	0.00	combined_ltv_at_origination	0.05

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
io_flag_nan	0.00	channel_3	0.03
gse_eligible_flag_2	0.00	io_term	0.03
channel_U	0.00	documentation_type_2	0.01
product_type_category_U	0.00	documentation_type_1	0.01
product_type_category_nan	0.00	gse_eligible_flag_U	0.01
io_term	0.00	gse_eligible_flag_1	0.00
channel_nan	0.00	product_type_category_nan	0.00
channel_3	0.00	product_type_category_U	0.00
coapplicant_present	0.00	property_type_3	0.00
statecode_nan	0.00	statecode_CA	0.00
property_type_4	0.00	statecode_nan	0.00
property_type_5	0.00	io_flag_nan	0.00
property_type_6	0.00	io_flag_Y	0.00
property_type_7	0.00	io_flag_U	0.00
property_type_Z	0.00	occupancy_type_U	0.00
property_type_nan	0.00	documentation_type_nan	0.00
occupancy_type_2	0.00	occupancy_type_nan	0.00
occupancy_type_U	0.00	channel_nan	0.00
channel_9	0.00	channel_U	0.00
occupancy_type_nan	0.00	property_type_5	0.00
payment_frequency_2	0.00	property_type_7	0.00

Variable contributions for linear models for pricing for LA County

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
appraised_value	24.64	number_of_units	13.34
gse_eligible_flag_0	18.76	channel_2	12.83
occupancy_type_3	5.89	property_type_4	10.78
gse_eligible_flag_2	5.24	channel_D	10.44
original_ltv	4.82	product_type_category_F	6.58
documentation_type_1	4.58	appraised_value	6.49
product_type_category_A	4.50	back_end_ratio	5.21
io_flag_N	3.77	property_type_1	4.92
gse_eligible_flag_1	3.73	occupancy_type_3	3.88
channel_D	3.70	gse_eligible_flag_0	3.54
occupancy_type_1	2.80	channel_1	3.17
channel_1	2.59	property_type_U	2.73
property_type_U	2.34	property_type_6	2.31
fico	1.45	fico	2.18
channel_9	1.32	occupancy_type_1	1.93
property_type_5	1.16	occupancy_type_2	1.92
channel_2	1.16	product_type_category_A	1.70
payment_frequency_U	1.06	io_term	1.39
gse_eligible_flag_U	0.82	combined_ltv_at_origination	1.17
property_type_6	0.81	documentation_type_U	0.77
payment_frequency_4	0.67	original_ltv	0.74
combined_ltv_at_origination	0.56	property_type_2	0.44
product_type_category_F	0.55	property_type_M	0.41
channel_3	0.52	channel_3	0.38
occupancy_type_2	0.51	documentation_type_1	0.34
property_type_1	0.50	property_type_Z	0.26
occupancy_type_U	0.49	payment_frequency_U	0.13

Unconstrained		Constrained	
Variable	**Contribution (%)**	**Variable**	**Contribution (%)**
property_type_Z	0.33	property_type_5	0.01
property_type_2	0.26	channel_U	0.00
coapplicant_present	0.21	io_flag_nan	0.00
channel_U	0.15	product_type_category_U	0.00
back_end_ratio	0.07	io_flag_Y	0.00
number_of_units	0.03	product_type_category_nan	0.00
documentation_type_2	0.02	gse_eligible_flag_1	0.00
payment_frequency_nan	0.00	io_flag_U	0.00
io_flag_nan	0.00	gse_eligible_flag_2	0.00
io_term	0.00	io_flag_N	0.00
statecode_CA	0.00	gse_eligible_flag_U	0.00
statecode_nan	0.00	documentation_type_nan	0.00
property_type_3	0.00	payment_frequency_4	0.00
product_type_category_nan	0.00	documentation_type_2	0.00
product_type_category_U	0.00	channel_nan	0.00
property_type_4	0.00	channel_9	0.00
property_type_7	0.00	payment_frequency_nan	0.00
io_flag_Y	0.00	payment_frequency_2	0.00
payment_frequency_2	0.00	occupancy_type_nan	0.00
io_flag_U	0.00	occupancy_type_U	0.00
property_type_M	0.00	property_type_nan	0.00

Glossary

+ **Adverse Impact Ratio (AIR):** a measure often used to determine if members of a protected class are experiencing disparate outcomes on the basis of an underwriting decision. The AIR is the ratio between the approval rate of the protected class and the approval rate for the corresponding control class.
+ **Area under the receiver operator curve (AUC):** The Area under the Receiver Operator Curve (AUC) is a measure of the accuracy of a yes/no machine learning score. It has a value between 0 and 1, where a value of one means a universally correct predictor, a value of 0 means a universally wrong predictor, and a value of 0.5 means a useless predictor which is no better than random choice.
+ **Bayesian Improved Surname Geocoding (BISG):** A method for estimating the race and ethnicity or sex of an individual from their full name and address using only publicly available data.
+ **Binary Cross Entropy Loss:** The loss function which is used in logistic regression as well as in building most other yes/no classification machine learning models such as underwriting models.
+ **Boosted forest:** A form of decision tree forest constructed using a boosting algorithm. In this document, that algorithm is Gradient Boosting
+ **Cross Entropy Loss:** The generalization of the binary cross entropy loss function to multi-category optimization such as credit line assignment models.
+ **Decision tree:** A form of machine learning target in which the model output function is computed by making a series of ordered binary tests along the paths through a tree, and where the model output is associated with the sequence of tests has no remaining choices to be made.
+ **Disparate impact:** A form of discrimination in which a facially neutral underwriting or pricing policy has a discriminatory outcome on members of a protected class. Like disparate treatment, disparate impact is forbidden under US anti-discrimination law.
+ **Disparate treatment:** A form of discrimination in which underwriting or pricing policies or standards treat members of a protected class differently from members of a control class.
+ **Home Mortgage Disclosure Act (HMDA):** The Home Mortgage Disclosure Act of 1975 requires that certain institutions (mostly banks and credit unions) report certain data on all mortgage applications they receive, no matter the eventual disposition of those applications. These data are reported in Loan Application Records (LARs), which are then

aggregated and anonymized. The resulting collection of anonymized records is released every year.

+ **Jensen-Shannon (JS) Divergence:** The Jensen-Shannon divergence (JS divergence) is a measure of the difference between two probability distributions. It is non-negative and bounded above and is 0 only if the two distributions are identical. It is a symmetrized and normalized form of the KL divergence. It is used extensively in this paper as a term in a loss function which attempts to ensure that the distributions of the scores of two related groups are identical.
+ **Kullback-Leibler (KL) Divergence:** The Kullback-Leibler divergence (KL divergence) is a measure of the difference between two probability distributions. Like the JS divergence or the PSI, it is always greater than or equal to zero, and zero only if the two distributions are identical. Unlike the JS divergence or the PSI, it is not symmetrical. Unlike the JS divergence, it is not bounded above.\
+ **Loss function:** Most machine learning algorithms work by minimizing the difference between a model's output and a set of target values. This difference is measured using a loss function, usually a non-negative function of the model output and the corresponding target value.
+ **Mean difference (MD):** The difference between the means of two distributions. The MD is often used as a measure of the disparity between the loan pricing between two different demographic classes.\
+ **Mean squared error (MSE):** The average of the squares of the differences between a model's output on a given input and the corresponding target value. The MSE is the loss function most often used for regressions with continuous outputs, such as pricing decisions in lending.
+ **Neural network (NN):** A machine learning model which consists of many 'artificial neurons' that accumulate the outputs of other artificial neurons, multiplying them by a set of constants or 'weights', having one of more 'output neurons' which constitute the model output(s). Neural networks can be trained using a standard algorithm called 'backpropagation'.
+ **One-hot encoding:** A mechanism for transforming categorical inputs before using them as inputs to a machine learning model. In one-hot encoding, the single categorical input is transformed into a set of 0-1 inputs, each of which corresponds to one or more possible values of the input. Each value of the input is transformed into a vector of these new values in which the one which corresponds to the value of the input variable has value 1 and all others have value 0. (One of these is 'hot', hence the name.)
+ **Population Stability Index (PSI):** The Population Stability Index is a symmetrized form of the KL divergence. Like the KL divergence and the JS divergence, the PSI is a measure of the difference between two distributions. Like the KL divergence and the JS divergence, it

is always greater than or equal to zero. Like the JS divergence, but unlike the KL divergence, the PSI is symmetric, non-negative, and zero only if the distributions are identical. Unlike the JS divergence, but like the KL divergence, the PSI is not bounded above. The PSI is often used in model risk analysis, particularly to measure the drift of input values, target values, or model scores over time.

+ **Protected class or protected group:** In US law, discrimination in lending on the basis of race, ethnicity, sex, religion, veteran status, marital status, and several other criteria is forbidden. A protected class or protected group is a group of people who share one of these characteristics and who have experienced discrimination on the basis of that shared characteristic. (For instance, Black or African American mortgage applicants.)
+ **Shapley Additive Explanations (SHAP):** A reinterpretation of Shapley values which unifies many forms of contribution assignment. The original authors of the SHAP papers maintain an open source implementation of the algorithm.
+ **Shapley value:** Shapley values are a credit assignment algorithm derived from cooperative game theory. In machine learning, Shapley values are used to explain the contribution of individual variables to the outputs of a machine learning model.
+ **Standardized mean difference (SMD):** The mean difference between two distributions standardized by the pooled standard deviation of the two distributions. Standardization corrects for the spread of the distributions, reducing the mean differences between distributions which are wide relative to those which are narrow.
+ **TreeSHAP:** An efficient implementation of the SHAP algorithm for decision trees.
+ **XGBoost:** An implementation of gradient boosting which efficiently and quickly produces accurate boosted forests with a wide variety of loss functions and input datasets.

Acknowledgements

Support for this research was provided with funding from Wells Fargo, the National Fair Housing Alliance, and other sources.

EMBARGOED UNTIL APRIL 24, 2024

fairplay NFHA NATIONAL FAIR HOUSING ALLIANCE™