Mitigating Discrimination in Machine Learning Models

Nicholas Schmidt, SolasAI & BLDS, LLC
Matthew Boutte, SolasAI & BLDS, LLC

Introduction

In a series of papers, Discover Financial Services (Discover) explored the use of the Wasserstein Distance – a measurement of the distance between two probability distributions – for detecting, understanding, and mitigating discrimination in predictive algorithms. BLDS, LLC is writing three companion pieces on the use of the Wasserstein Distance for these three use cases which are intended for a broader audience [1]

The first of the BLDS papers, *Measuring Discrimination Under Model Usage Uncertainty*, explains the mechanics of what the Wasserstein Distance is, how it is used to measure disparity, and the advantages it has over existing metrics. The second paper, *Identifying Drivers of Discrimination in Machine Learning Models*, explores how the Wasserstein Distance can be used to determine what is driving disparity in a model. This third and final paper will show how the Wasserstein Distance can be used to mitigate disparity that may be found in a model.

The Wasserstein Distance

As explained in the first two papers, the Wasserstein Distance is a measure of how much the distribution of model predictions for a protected group would need to be changed[2] to match the distribution of model predictions for the associated reference group.[3] Assuming the protected group generally receives less favorable predictions from the model, the larger the Wasserstein Distance, the more disparity is present in the model. The goal of this paper is to describe and explore techniques used by Discover to utilize the Wasserstein Distance to change input data for a model with the goal of reducing disparities.

[1] Wasserstein-based fairness interpretability framework for machine learning models (https://link.springer.com/article/10.1007/s10994-022-06213-9). Mutual information-based group explainers with coalition structure for machine learning model explanations (https://arxiv.org/abs/2102.10878). Model-agnostic bias mitigation methods with regressor distribution controls for Wasserstein-based fairness metrics (https://arxiv.org/abs/2111.11259).

[2] More formally, this is known as “work” – how much of the distribution must be shifted multiplied by how far it must be shifted.

[3] A protected group is a group which has historically been subject to discrimination along a prohibited basis, such as sex, ethnicity, or age. Examples include women, racial minorities, and older people. A reference group is the group that a protected group is compared to when testing for discrimination along a prohibited basis. For example, women are compared with men, minorities with the non-minority group, and those who are older versus those who are younger.

The first paper demonstrates how applying the Wasserstein Distance to model disparity is an innovative approach that accounts for any possible use of a model's predictions, regardless of what threshold values are selected. This is a powerful innovation because, in practice, it is frequently the case that (1) how a model will be used is unknown, (2) the model will have multiple use cases, and (3) how a model is used will change over time. This presents a fundamental problem to traditional approaches to testing model disparity since this testing requires knowledge of precisely how the model will be used. Applying the Wasserstein Distance is an elegant solution to this problem by accounting for all possible use cases.

The second paper combines these insights about the Wasserstein Distance with Shapley or Owen values – tools that measure how each feature in a model contributes to an individual model prediction – to gain insight into which feature or features are driving overall model disparity. The total Wasserstein Distance applied to the Shapley or Owen values for any given feature can then be decomposed into Positive Bias (how much the protected group is harmed), Negative Bias (how much the protected group is benefited), and Net Bias (positive bias minus negative bias).

We now turn to how these insights can be leveraged to modify a model to reduce the Wasserstein Distance and reduce model disparity.

How Models Work

To understand Discover's technique for reducing disparity based on the Wasserstein Distance, it is necessary to have a rudimentary understanding of how models work. Models for applications in consumer credit nearly always fit into one of two classes of models: traditional regression models and tree-based machine learning models. We will explain each in turn.

Regression Models

At their core, regression models multiply each variable (or "feature" in the parlance of machine learning) by different constants (or coefficients), sum each of these values, and add a final constant term. Additional mathematical transformations may then be applied, but understanding those additional steps is unnecessary for the purposes of this paper.

$$p = a_1x_1 + a_2x_2 + a_3x_3 + b_0$$

Here, p is the model prediction, each a value is a constant (or coefficient), each x value is a variable (or feature), and b_0 is the constant term. In the following model, one would need to supply each of the x values in order to compute a prediction.

$$p = 2 \cdot x_1 + 4 \cdot x_2 + 5 \cdot x_3 + 1$$

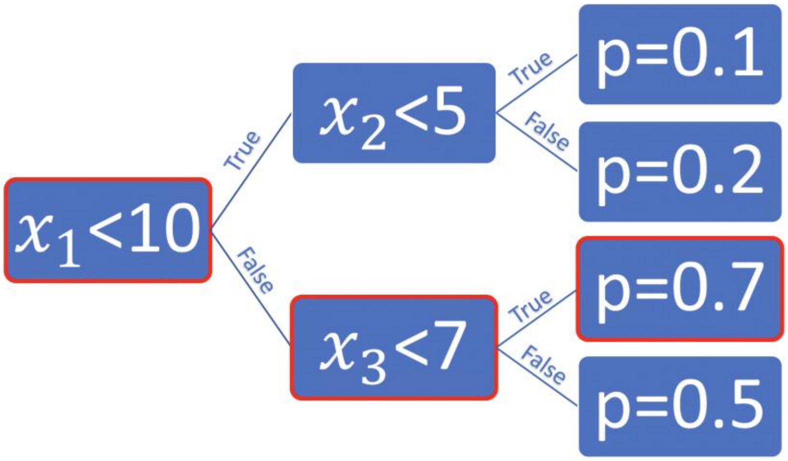
With this understanding of how regression models work, it is easy to see how changing the value of a feature would flow through to a change in the final model output. As shown below, if the x_3 feature was changed from 1 to 1.1, this would increase the prediction by 0.5. The model itself did not change; there was merely a change to an input value that resulted in a change to the model output.

$$2 \cdot (-1) + 4 \cdot (0.5) + 5 \cdot (1) + 1 = 6$$
$$2 \cdot (-1) + 4 \cdot (0.5) + 5 \cdot (1.1) + 1 = 6.5$$

Tree-Based Models

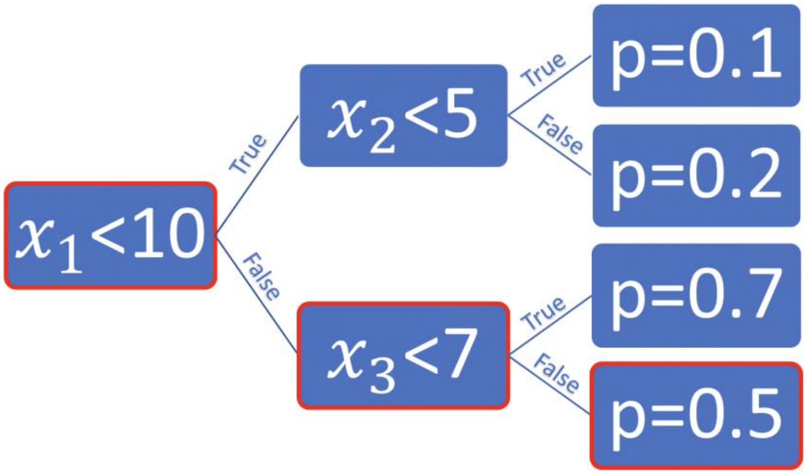
Tree-based models build one or more decision trees that are combined to create predictions of some outcome. As shown in the graphic below, the tree starts with a single node where a variable sends each individual down one of two possible branches. This variable will have a cut point to determine which branch each individual goes down: anyone with a value less than the cut point will go down the upper branch and anyone with a value greater than the cut point will go down the lower branch. This process is then repeated some number of times. Each branch will lead to a new node that has its own associated variable and cut point, which is used to determine which subsequent branch to send the individual down.

At the end of each branch is a terminus known as a "leaf." Each individual traverses the tree and ends up at a single leaf based on their feature values and the cut points used throughout the tree. Each leaf has an associated score or prediction; all the individuals who end up in a particular leaf receive that leaf's associated score or prediction.



The diagram above shows a simple tree-based model and how an individual traversed the tree (as indicated by the red highlights). Suppose this individual has the values $x_1 = 12;\ x_2 = 1;\ x_3 = 6$. The left-most node has an evaluation criteria of $x_1 < 10$ and the individual has a value of 12 for x_1, so this node evaluates to 'False' and the individual is sent down the lower branch. This node has an evaluation criteria of $x_3 < 7$ and the individual has a value of 6 for x_3, so this node evaluates to 'True' and the individual is sent down the upper branch. This results in the individual reaching a leaf with a value of $p = 0.7$, which represents the model's prediction for this individual.

If an individual's value for a particular feature is changed so that it flips from one side of a cut point to the other side, then that individual may proceed down a different branch and end up in a different leaf and thereby receive a different score or prediction.



Here, the tree-based model is the same as before, but we have changed the x_3 value for the individual so that $x_1 = 12;\ x_2 = 1;\ x_3 = 8$ (instead of $x_3 = 6$). The left-most node is evaluated the same way as before because the individual's x_1 value has not changed. However, the next node ($x_3 < 7$) will now be evaluated to 'False' because the individual's value for x_3 is now 8. This results in the individual being sent down the lower branch and ending at a leaf with a value of $p = 0.5$.

Bias Mitigation

With this understanding of how models work and in light of Discover's Wasserstein Distance framework, a data transformation technique can be used to reduce model disparity. At a high level, all that is required is to make slight adjustments to a feature (or features) to minimize the differences between model scores for protected and reference groups. This can be done in practice by decreasing the Wasserstein Distance between the relevant protected groups and associated reference groups so that the differences are smaller than they would be without the adjustments. We will walk through this process in the context of the simple credit model developed in the second paper in this series (*Identifying Drivers of Discrimination in Machine Learning Models*).

The model was trained on seven credit-related features and resulted in a net Wasserstein Distance bias of 6.28% between the Majority and Minority groups. That is, the model produced predictions that, on average, favored the Majority group by 6.28 percentage points over the Minority group. When the Wasserstein metric was then applied to the Shapley contributions for each feature, it became clear that virtually all of the bias was being driven by the Mortgage feature. Specifically, the Mortgage feature contributed 5.98% to the overall 6.28% net bias. Furthermore, the Mortgage bias was composed entirely of Positive (i.e., unfavorable) Bias, indicating that the Mortgage feature only harmed the Minority group. In other words, throughout the entire distribution of Mortgage values, there was nowhere where the Minority group benefited due to the inclusion of the mortgage feature.

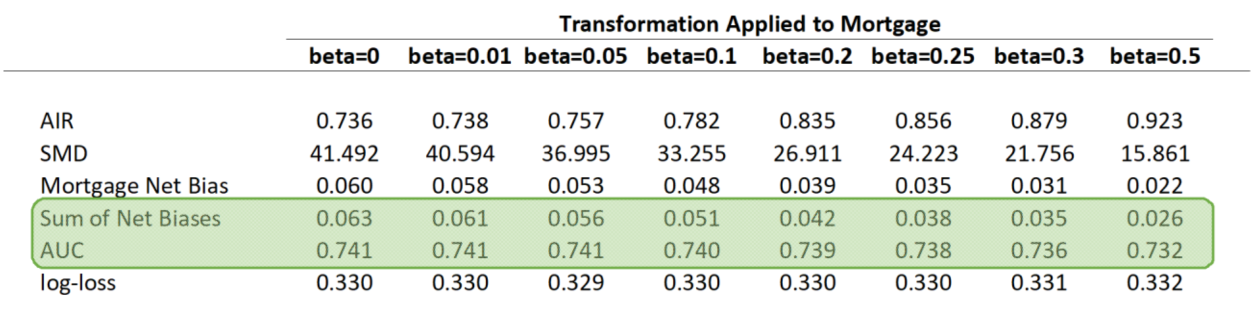
Based on summary statistics, we also know that the Minority group has an average Mortgage balance of $171,649, while the Majority group has an average Mortgage balance of $294,438. Because of how the model was designed, we know that higher Mortgage values can only help an applicant.[4] As a result, it is unsurprising that the Mortgage feature substantially contributes to model bias: the model has been trained to generate more favorable predictions for higher Mortgage balances and the Majority group has higher Mortgage balances on average, so they are receiving more favorable outcomes on average.

Now that we have identified where the disparity in the model is coming from and why it occurs, we can devise a transformation to the Mortgage feature that may ameliorate the bias. To mitigate the bias caused by the Mortgage variable, we must develop a transformation that systematically favors the Minority group. Our knowledge of the bias explanations and summary statistics informs this process. We know the Minority group has lower Mortgage balances on average, so a transformation that increases lower Mortgage balances and decreases higher Mortgage balances is likely to favor the Minority group systematically.

For this paper, we use the transformation $\beta(x - \bar{x}) + \bar{x}$ where $\bar{x}$ is the mean of a given variable, x represents the value for that variable, and β is a positive multiplier that controls the magnitude of the transformation (the smaller the value, the smaller the effect of the transformation). This transformation has the effect of pulling all values toward the mean value. The further from the mean, the greater this “pull” will be, while values near the mean will have a smaller change. Thus, members of the Minority group, who tend to have Mortgage values below the mean, will be pulled up toward the mean. Likewise, members of the Majority group, who tend to have Mortgage values above the mean, will be pulled down towards the mean. Based on the model’s

[4] This is because, as explained in Identifying Drivers of Discrimination in Machine Learning Models paper, we utilized monotonic constraints when building this model. Monotonic constraints are a technique used in machine learning by which a modeler can enforce either a positive or a negative relationship between a variable and what is being predicted. Since we imposed negative monotonic constraints on the mortgage feature, all else equal, people with higher mortgage values would have received lower model scores (i.e., their probability of default would have been lower).

monotonic constraints[5], this should disproportionately improve predictions for the Minority group and worsen predictions for the Majority group.



	Transformation Applied to Mortgage							
	beta=0	beta=0.01	beta=0.05	beta=0.1	beta=0.2	beta=0.25	beta=0.3	beta=0.5
AIR	0.736	0.738	0.757	0.782	0.835	0.856	0.879	0.923
SMD	41.492	40.594	36.995	33.255	26.911	24.223	21.756	15.861
Mortgage Net Bias	0.060	0.058	0.053	0.048	0.039	0.035	0.031	0.022
Sum of Net Biases	0.063	0.061	0.056	0.051	0.042	0.038	0.035	0.026
AUC	0.741	0.741	0.741	0.740	0.739	0.738	0.736	0.732
log-loss	0.330	0.330	0.329	0.330	0.330	0.330	0.331	0.332

As explained above, the beta (β) value controls the magnitude of the transformation. Thus, a beta value of 0 represents the original model without any transformation. The table reports several fairness metrics[6] and quality metrics:

- Adverse Impact Ratio (AIR): This is a commonly used measure of disparate impact in regulatory and legal settings. It is equal to the ratio of the percent of the minority group that receives the favorable outcome over the percent of the majority group that receives the favorable outcome. An AIR of 1 means the two groups receive the favorable outcome at the same rate; an AIR of less than 1 means the minority group receives the favorable outcome less frequently than the majority group. Here, we assume that 75 percent of the population receives the favorable outcome.
- Standardized Mean Difference (SMD): This is another measure of disparate impact that is commonly used in regulatory and legal settings. It equals the number of standard deviations between the average minority score and the average majority score times 100. An SMD of 0 indicates that the average scores for both groups are the same. Larger SMD values indicate more disparity (that is, the minority group's average scores are unfavorably higher than the majority group's average score).
- Mortgage Net Bias: The net Wasserstein bias contributed by the mortgage feature in the model. As previously discussed, the Mortgage feature is driving nearly all of the disparity in the model, and we are transforming this feature, so we focus on the net bias of this feature.
- Sum of Net Biases: The sum of net biases for all features in the model. This is the disparity metric we focus on.
- AUC: A measure of model quality. It can be interpreted as the probability of correctly ranking the risk of any pair of one applicant from the default population and one applicant from the non-default population. Thus, an AUC of 1 would mean that the model could always correctly identify which of the two applications was higher risk, and an AUC

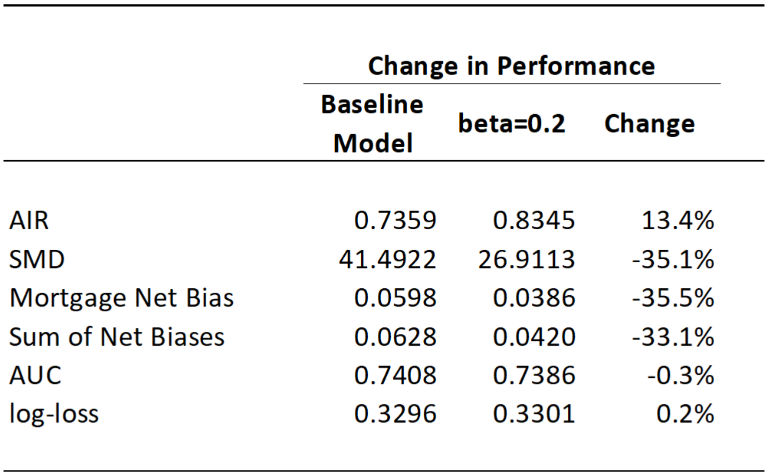
[5] The monotonic constraint for the Mortgage feature means that a higher Mortgage value can only lead to a lower score.

[6] For more information about the two disparity metrics, see https://www.mdpi.com/2078-2489/11/3/137.

of 0.5 would mean that the model was only as good as a coin toss at deciding which application was a higher risk.

- Log-Loss: A measure of model quality. It is difficult to interpret conceptually but it measures how far predictions are from true values (i.e., default or non-default). A smaller log-loss means higher quality model.

The table reports each of these metrics at different beta levels. As can be seen, increasing the beta value (that is, increasing the magnitude of the transformation), results in a significant reduction in disparity across all the metrics while only resulting in relatively small degradations in model quality (a higher AUC value indicates a better performing model while a lower log-loss indicates a better performing model). Choosing beta=0.2 as an example, below, we show how much each metric changes relative to the baseline model.



	Change in Performance		
	Baseline Model	**beta=0.2**	**Change**
AIR	0.7359	0.8345	13.4%
SMD	41.4922	26.9113	-35.1%
Mortgage Net Bias	0.0598	0.0386	-35.5%
Sum of Net Biases	0.0628	0.0420	-33.1%
AUC	0.7408	0.7386	-0.3%
log-loss	0.3296	0.3301	0.2%

We have thus identified a transformation that results in a fairer model usage that performs comparably to the original model usage. All of this was achieved with Discover's Wasserstein framework.[7]

Wasserstein-based Bias Mitigation in the Regulatory Context

The existing legal and regulatory context requires credit models to be tested for disparity and for steps to be taken to mitigate any unacceptable levels of disparity that may be found.[8] However,

[7] There is one additional step in the process of rescoring the applicants after transforming the Mortgage feature values. Because the Mortgage feature no longer reflects true values, the model predictions may not accurately reflect probabilities of default. This is rectified by calibrating the model predictions. Here, this was done with Platt scaling (also known as Platt calibration), where a logistic regression of the model predictions using the transformed Mortgage feature is fit onto the true outcomes in the training dataset.

[8] https://ncrc.org/cfpb-puts-lenders-fintechs-on-notice-their-models-must-search-for-less-discriminatory-alternatives-or-face-fair-lending-non-compliance-risk/

including protected demographic data in the model itself is prohibited – it can only be used for testing and mitigation purposes.

In practice, this process generally looks like the following. Modelers develop a credit model without any access to demographic data. During this stage, modelers focus exclusively on maximizing model performance. When modelers are satisfied with their results, the model is submitted for a fairness review. Demographic data is then appended to the model development data to calculate appropriate fairness metrics. Several such metrics are available, with varying degrees of acceptance and adoption. Typically, the use case determines which metric or metrics are most appropriate in a particular circumstance. Once the fairness metric is calculated, an institution's pre-existing policies are applied to determine whether any unfairness that may have been found rises to the unacceptable level (these are referred to as "practically significant" disparities). If an unacceptable level of fairness is found, the modelers search for a less discriminatory alternative model (LDA). Various techniques exist for conducting this search. The two most common techniques are to swap variables in and out of the model and to adjust the model's hyperparameters.[9] Modelers will have access to demographic data to inform their decisions around swapping variables and adjusting hyperparameters and to evaluate each alternative model's fairness. However, once again, the demographic data will not be included in the model itself. As a practical matter, there is generally a tradeoff between model quality and model fairness, so any LDAs will likely have somewhat lower performance. Therefore, the final stage of this process is to weigh the tradeoffs between model performance and fairness and to make a business decision about which model to adopt.

Discover's Wasserstein-based methodology fits squarely within this framework. It merely changes the tools used during two stages of the process that already involve discretion in choosing between existing tools.

First, Discover's methodology uses the Wasserstein Distance as the fairness metric when evaluating a model's fairness. As noted above, there are already several fairness metrics that are used in various contexts. The first paper in this series, *Measuring Discrimination Under Model Usage Uncertainty*, explored and explained how the Wasserstein Distance is not only an acceptable fairness metric but an innovative improvement in measuring model fairness – particularly when it is unknown precisely how a model will be used. Thus, the decision to use the Wasserstein Distance as the fairness metric during this process is a reasonable one.

Second, Discover uses variable transformations to search for LDAs. Although not the most commonly used tool for searching for LDAs, it is a recognized technique and has been the

[9] Hyperparameters are frequently described as the "knobs and dials" of a machine learning algorithm. They control how the algorithm will operate and be structured. For example, in a tree based model, the number of trees and the number of branches per tree are both examples of hyperparameters. It is common to "tune" these hyperparameters with the goal of finding the hyperparameters that produce the best performing model. It is possible to tune the hyperparameters with an eye to both the quality and the fairness of the model.

subject of research.[10] Like the existing techniques for searching for LDAs, Discover leverages demographic-based metrics (namely, the total, positive, negative, and net bias for each feature) to inform the development of transformations that may reduce model bias.

As was shown here, it is possible to develop transformations that significantly reduce model disparity – whether measured by the Wasserstein net bias or more traditional metrics – while preserving model quality. This entire process was guided by Discover's Wasserstein framework for measuring, explaining, and mitigating model disparity. The writers were pleasantly surprised to see how effective the methodology was at identifying the problem and suggesting a solution that yielded strong results in a way that complies with the existing understanding of disparate impact analysis and the search for LDAs. While the traditional approaches to searching for LDAs highlighted above can require searching over a vast and computationally intensive number of potential alternative models, Discover's process yielded a clear way to generate LDAs, which, notably, does not require training a new model. Importantly, this process was agnostic to the particular model usage; the LDAs that were generated would likely generate fairer results for any possible model usage.

Conclusion

In developing their Wasserstein-based methodology, Discover has put forward a powerful and innovative approach for detecting, understanding, and mitigating model bias. This methodology has the potential to improve real-world outcomes by better detecting and understanding bias. It is also a significant improvement on many existing techniques since it can account for all possible model use cases – a significant limitation of existing approaches.

In addition to these improvements, Discover closely adheres to the existing regulatory framework and legal paradigm for addressing bias in credit models. This is a significant achievement and an indication of Discover's conservative approach in light of more radical methodologies that have been developed in recent years, such as using demographic data in the training of a model in order to reduce disparity.

[10] See, e.g., https://arxiv.org/pdf/1908.09635.pdf