Fairness Constraints in Loan Default Prediction

Jodie Bhattacharya
Stanford University
jodieab@stanford.edu

Andrew Semenza
Stanford University
asemenza@stanford.edu

Arthur Lee
Stanford University
arthurlh@stanford.edu

***Abstract*—We study the problem of algorithmic fairness constraints in a consumer loan repayment prediction problem. We attempt to study the fairness of the allocation induced by a naive allocation, and define a novel way to measure the welfare of a loan prediction system. We then explore several algorithmic methods to constrain allocations to promote equity goals.**

I. Introduction

Algorithmic fairness desiderata have been defined in the literature in a number of ways. Two common definitions are:

1) **Statistical Parity**, which says that if Y is an outcome of interest, and $S \subset V$ is a protected subset of individuals V, then the algorithm must satisfy $P(Y|S) = P(Y|S^c)$ (i.e. both good and bad outcomes must be equally likely for majority and minority groups).
2) **Individual Fairness**, defined by Dwork et al [1]. Roughly speaking, Individual Fairness means individuals with similar task-specific characteristics ought to receive similar outcomes. A mapping from individuals to outcomes that satisfies this property is called *Lipschitz*. Formally, Individual Fairness is defined as follows: Let V be a set of individuals receiving treatment and A be the set of outcomes. Let $d : V \times V \to \mathbb{R}$ be a metric that captures 'distance' between individuals. Let $M : V \to \Delta(A)$ be a mapping that assigns each individual from V a probability distribution over the outcomes. Let D be a statistical measure that captures the similarity of two distributions. A mapping satisfies the (D, d) - Lipschitz property if $\forall x, y \in V$, $D(M(x), M(y)) \leq Kd(x, y)$. An *Individually Fair* (IF) classifier is one that finds a mapping from individuals to distributions over outcomes that minimizes expected loss subject to the Lipschitz condition.

We study these two fairness definitions in the context of a real-world classification task. We study how naive classifiers may violate these constraints. We present a novel definition of the welfare of a loan allocation, grounded in economic theory. We then explore several pre-processing and in-processing approaches to impose fairness constraints. Finally, we study how these fairness metrics may trade-off with each other, and in turn with borrower and lender welfare. The input to our algorithm is data from HomeCredit, an international non-bank financial institution that specializes in installment loans to low-credit consumers. This data contains individual demographics and loan outcomes. We study fairness according to protected categories as defined by the United States Equal Credit Opportunity Act (ECOA).

II. Related Work

There is an existing body of research that looks into the relationship between different fairness measures. Kleinberg, Mullainathan and Raghavan proved an impossibility theorem [2]: except in highly degenerate cases, there is no predictor that simultaneously satisfies all three of statistical parity, equalized odds (same rate of false positive/negative for protected and unprotected groups) and predictive parity. Of course, this does not rule out predictors that reduce each of the three metrics.

Our study is related to the work of Heidari et al [3] who use an experiment on the COMPAS recidivism dataset and a welfare function that encodes the loss associated with an incorrect outcome to examine the tension between welfare and individual fairness constraints (both statistical parity and IF). They show experimental results that enforcing a higher guarantee of "welfare" is associated with non-monotone (decreasing, then increasing) violation of IF constraints. We build on this work by examining a more natural setting where economic welfare loss (lost revenue) is more easily quantified. Many methods of rectifying algorithmic unfairness have also been described in the literature – for example Woodworth et al [4] and Feldman et al [5]. However, most of these approaches rely on strong distributional assumptions on the data or integrating nonconvex constraints into a ML training pipeline. In this paper, we consider two viewpoints on fairness constraints that we feel are applicable – the GridSearch pre-processing algorothm of Agarwal et al [6] and an in-processing model of Fukuchi et al [7].

Finally, our study is situated within a body of empirical literature in economics that studies loan fairness as well as contemporary concerns over relative levels of bias in traditional and algorithmic approaches. Recent pieces in the Harvard Business Review [8] as well as the New York Times [9] suggest the promise of ML methods despite their current insufficiencies in resolving fairness problems. Dobbie et al. finds significant bias against immigrants and older applicants and argue that this bias derives from misalignments in the firm's profit incentive structure [10]. They go on to suggest ML approaches could help ameliorate these biases.

III. Dataset and Features

HomeCredit is a company that provides installment lending to people with poor credit history. In 2017, they made anonymized data available on Kaggle [11] which includes individual demographics and the loan outcomes (i.e. default, no default). This dataset is unique as most financial institutions

are bound by legislation to exclude protected characteristics (e.g. race, age, gender identity, marital status) from being part of financial decision-making. However, HomeCredit was not interested in implementing the outcome of the Kaggle competition, but rather was interested in the methodology deployed by the winners. Hence this sensitive information was included in the dataset. This allows us to compare the outcomes of classification algorithms across different groups, e.g. does constraining by fairness reduce the inequality of outcomes between groups?

In order to preprocess the data, we merge together 7 relevant datasets from Kaggle by applicant ID:

- **application_{train/test}.csv** – 307,511 and 48,744 obs, respectively, with info about the loan and loan applicant at application time
- **bureau.csv** and **bureau_balance.csv** – 1,716,428 and 27,299,925 obs, respectively, with application data from previous loans reported to the Credit Bureau, monthly balance of credits in Credit Bureau, and behavioral data
- **previous_application.csv** – 1,670,214 obs, previous applications to Home Credit with previous loan parameters and client info
- **POS_CASH_balance.csv** – 10,001,358 obs, monthly balance of previous loans in Home Credit
- **instalments_payments.csv** – 13,605,401 obs, past payment data for previous credits to Home Credit
- **credit_card_balance.csv** – 3,840,312 obs, monthly balance of previous credit card loans

The general preprocessing philosophy was to group by applicant ID and take min, max, mean, sum, and counts, as pertained to each variable. For some datasets, we additionally grouped by important features such as Credit Active/Closed and Loan Approved/Denied and performed summary metrics. We also created new metrics by dividing or subtracting variables from each other in order to get percents and other important rates. We used one hot encoding for categorical variables and set anomalous outliers to missing. For logistic regression and neural network models, we discarded rows with ≥ 0.5 missing values and imputed the average for missing values. We also removed columns that were found to have high multicollinearity (via a correlation matrix).

The final training dataset has 307,511 observations and the final test dataset has 48,744 observations, and there are 649 variables.[1] Since Kaggle only scores the ROC AUC for the final test dataset (without providing access to the ground truth target variables), we split the training dataset 80-20 into train and test, in order to evaluate accuracy further. The metrics for this are referred to as "Self Split AUC."

We identify three main protected groups in our cleaned data. These are based off categories protected by the United States Equal Credit Opportunity Act (ECOA), which prevents discrimination on the basis of "race, sex, age, national origin, or marital status, or because one receives public assistance" [12]. In our cleaned dataset, the proxy variables capturing protected features are gender, age, and marital status. Before evaluating fairness, it is important to observe the underlying distributions for different protected groups [13]. Histograms are available for gender (1a), age (1b), and marital status (1c). Note that the underlying distributions between groups are quite different, which can be attributable to a number of factors, including noisy data and differences in behavior between groups. Regardless, we are still interested in making our model as fair as possible for the groups (in some settings, for example, in "affirmative action", governments may be interested in enforcing fair outcomes even when the underlying distribution of test scores, etc. may not be fair).

[1]For logistic regression, this was reduced to 466 variables.

IV. Methods

A. Supervised Learning

We used supervised learning methods logreg, tensorflow, pytorch, and xgboost with a subset of HomeCredit's features to predict loan default. Logistic regression is one of the most common techniques for classification, and it has an objective function of the form $h_\theta(x) = 1/(1+\exp(-\theta^T x))$. xgboost is a widely-used gradient boosting decision tree algorithm, essentially an iterated application of decision trees, where new models are added to correct the errors made by previous model (i.e. to fit a hypothesis to the residuals of previous models). The objective function we choose for xgboost is the binary log-loss function. Tensorflow and Pytorch train a neural network, which, generally speaking, is defined by $\forall j \in [1,\ldots,m], \quad z_j = w_j^{[1]^\top} x + b_j^{[1]}$ where $w_j^{[1]} \in \mathbb{R}^d, b_j^{[1]} \in \mathbb{R}$ with m hidden units and d dimensional input $x \in \mathbb{R}^d$. We use the ReLU activation function, which is defined by $ReLU(t) = \max\{t, 0\}$.

B. Fairness Metrics

We define the following fairness metrics to score and constrain our models:

- **Statistical Parity Difference** ("Demographic Parity Difference") is defined as

$$Pr(\hat{y}_i = 1, i \in A) - Pr(\hat{y}_i = 1, i \notin A) \qquad (1)$$

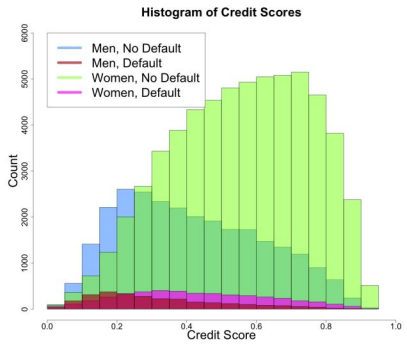
  for individual i and protected group A. The ideal statistical parity difference for a fair model is 0.
- **IF Constraint Violation** measures the average violation of Dwork et al's constraints [1]. This is definition is inspired by an approach of Heidari et al [3]. The data gives us 3 normalised credit scores for each consumer ID, each from a different external credit agency. Let $v(x) \in \mathbb{R}^3$ be the vector of these scores for borrower $x \in V$ (V is the set of borrowers). Let M_x correspond to the probability distribution we assign to individual x. We define

$$d(x,y) = \sqrt{\sum_{i=1}^{3} (v(x)_i - v(y)_i)^2}$$

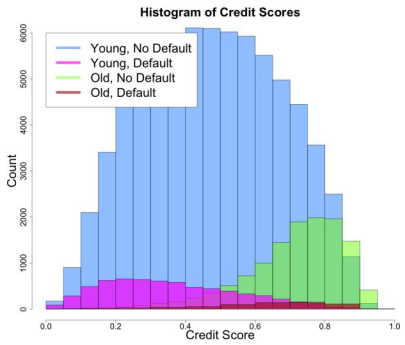
$$D(M_x, M_y) = |M_x - M_y|$$

Fig. 1: Distribution of Credit Scores across Default Status and Protected Groups

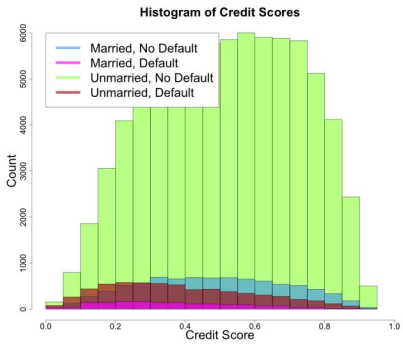
(a) Gender



(b) Age



(c) Marital Status



And define the *average violation of IF constraints* as

$$\frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \max\{0, D(M_i, M_j) - d'(i,j)\}$$

Where $d'(x,y)$ is a rescaling of the norm to be between $[0,1]$. In our case, since n is large, we take a random sample of the data of size 1,000 in order to calculate this metric. The ideal number of violations for a fair model is close to 0, because that means that similar people received the same outcomes, according to our similarity metric.

C. Pre-Processing Approaches to Fairness Constraints

In general, constraints can be implemented either as part of the training pipeline (in-processing, for example by imposing convex constraints on the loss function) or at the pre-processing stage. In this paper we use the Grid Search model of Agarwal et al [6] to pre-process the data to optimize for demographic parity. We choose this method because it is agnostic to the classifier used (we try logreg and random forests) and because it is easily implemented.[2] While the precise details of the algorithm are omitted, the high-level idea is as follows: our task is to learn a hypothesis $h \in \mathcal{H}$ for some data (X, A, Y), where A encodes a protected attribute. We can rewrite demographic parity as a set of linear constraints $\boldsymbol{M}\mu(h) \leq c$, where $\boldsymbol{M}$ and c are linear constraints, and $\mu(h)$ is a vector of conditional moments of form $\mu_j(h) = \mathbb{E}[g_j(X, A, Y, h(X)) \mid \mathcal{E}_j]$, where g_j depends on h but $\mathcal{E}$ cannot depend on h. Now we consider minimizing the loss $\min_{Q \in \Delta(\mathcal{H})} \text{err}(h)$ subject to $\boldsymbol{M}\mu(h) \leq c$ (using a *randomized* classifier Q which makes a prediction by sampling a classifier h and then using h to make a prediction). This randomized classification problem can be reduced to a sequence of weighted classification problems (i.e. iteratively calling the same black-box model and feeding it re-weighted data), whose solutions yield a randomized classifier with the lowest error subject to demographic parity. In particular, the fair regression problem can be rewritten as a Lagrangian $L(Q, \lambda) = \overline{\text{err}}(Q) + \boldsymbol{\lambda}^{\boldsymbol{T}}(\boldsymbol{M}\mu(\hat{h}) - \hat{c})$. The "Grid Search" model considers a grid of possible values for $\boldsymbol{\lambda}$, then re-weights the data accordingly and outputs the fitted models. The user can then select the value of $\boldsymbol{\lambda}$ that induces the best desired tradeoff between accuracy and fairness. In this paper, we chose to use the criteria of lowest demographic parity loss, subject to performance (AUC) of > 0.65.

[2]We made use of the open-source FairLearn package to simplify our analysis.

D. In-Processing Approaches to Fairness Constraints

We considered the problem of training a machine learning classifier that incorporates fairness constraints as part of its loss minimization pipeline. However, many of the powerful tools (e.g. xgboost) we used to on the unconstrained task were ill-suited for adaptation to include additional constraints. To explore one possible use case, we adapted the methods used by FairTorch, a package developed by Fukuchi et al [7], an open source package which imposes fairness constraints on Pytorch networks through the addition of a penalty term. We implemented a ReLU network with 2 hidden layers on Pytorch, and then included a custom penalty term on the BCE-CrossEntropy loss to penalize deviations from Demographic Parity on the gender variable *CODE_GENDER* (see equation 1).

E. Welfare

In order to measure welfare loss by algorithm, we rely on techniques from economic theory. In particular, we are inspired by Karlan et al [14], where they estimate the impacts of expanded credit access by measuring economic self-sufficiency and consumption for customers. They measure welfare for the credit company by taking into account payments and profits, but since we do not have information on interest rates or actual payments, we simply use the loan annuity amount. Note that $\hat{y} = 1$ means predicting that the customer will default on their loan. As a result, we have 3 welfare metrics:

- **Company welfare loss**: False negatives where α is a percentage of the annuity, the profit that the company makes from interest. In this case, since we are only interested in comparing models, we set $\alpha = 1$.

$$\sum_i \mathbb{1}(\hat{y}^{(i)} = 0 \wedge y^{(i)} = 1) \cdot \alpha \cdot AMT\_ANNUITY$$

- **Customer welfare loss**: False positives

$$\sum_i \mathbb{1}(\hat{y}^{(i)} = 1 \wedge y^{(i)} = 0) \cdot AMT\_GOODS\_PRICE$$

- **Relative customer impact**: Measures how much customers are disproportionately harmed by not receiving the loan, relative to their income

$$\sum_i \mathbb{1}(\hat{y}^{(i)} = 1 \wedge y^{(i)} = 0) \cdot \frac{AMT\_GOODS\_PRICE}{AMT\_INCOME}$$

V. Experiments/Results/Discussion

We first present the results of the unconstrained prediction problem. Our goal here is not to achieve maximal scoring (for context, the highest Kaggle leaderboard score is 0.805) but to create a fairly good baseline model that approximates a "real-world" setting where fairness may be relevant. Our baseline models even surpassed other open source projects working with this data [15]. We choose to display AUC as a score of model performance. Accuracy was judged to be less important (the data was overwhelmingly 0's, so even a classifier that output 0 for everything would have accuracy of > 0.91).

In table I, we find that xgboost without fairness constraints has a Kaggle AUC score of 0.77423. It performed well due to the high-dimensional nature of the data, perhaps by discarding redundant attributes. We include 58 rounds of training, which was found to have the maximal AUC using cross-validation with 3 folds. The ROC for the cross-validation is in Fig 2b. For xgboost, we left missing values in the model. We see from figure 2a that the most important predictors are EXT_SOURCE 1, 2, and 3, which represents normalized credit scores from different external credit agencies. This makes sense, although we note that credit companies such as HomeCredit set themselves apart by considering unconventional variables in the hopes of making lending more accessible, regardless of credit score. As we see from the histogram of the data distribution (2a), other variables are indeed helpful indicators, since credit scores do not tell the whole story.

For logreg, we used iteratively reweighted least squares to fit the model, since this led to the highest AUC. We also removed multicollinear columns that were found to result in unstable coefficient estimates. Due to the highly structured nature of the data, logreg performed well – it had only a slightly smaller AUC than xgboost, and with lower welfare loss. When we compare the unconstrained logreg to the constrained version (with Grid Search preprocessing), we see that the AUC is a fair bit higher, but the constrained version leads to better fairness scores as well as lower welfare loss for both consumers and firms.

For the neural networks, we chose a setup with 2 hidden layers with ReLU activation functions. We used a dropout layer between both hidden layers and before the output layer to prevent overfitting. The optimal number of epochs was chosen by plotting validation AUC, and choosing a point where validation AUC started to decrease (around 20 epochs). Tensorflow performed similarly to xgboost and logreg, though Pytorch had a lower AUC, potentially because of different initializations and backpropogation architecture. The baseline model of Pytorch achieved an AUC of 0.67884 on test data[3] and the constrained model achieved 0.67580. While parity for gender appears to have improved, it seems to have "overcorrected" in the positive direction, resulting in the privileging of protected categories. Moreover, the welfare loss to consumers and loss of IF constraints increased substantially. Therefore, we judged that in this setting a more sophisticated in-processing approach is required.

Overall, for the baseline results, we see that parity for each protected group is already quite low, with each metric being very close to 0. This makes sense because our dataset is unbalanced – hence, models constrained by demographic parity would not have as much of an effect as it might in other settings.

It appears that models that perform similarly with respect to AUC differ with respect to the level of welfare loss. This is to be expected, since the variables *AMT_ GOODS_PRICE* and *AMT_ANNUITY* are not correlated that strongly with the loan default outcome, so the models may have randomly chosen some consumers as false negatives/positives over others in edge cases. It is noteworthy that the pre-processing method of Agarwal et al [6] (which reduces demographic parity loss) results in substantially higher consumer welfare compared to other models – whether this is an artifact of the dataset or whether it stems from a deeper statistical fact should be further studied.

VI. Conclusion/Future Work

As discussed, constrained optimization for general black-box machine-learning problems is a difficult task. Future work could explore an implementation of Dwork et al's IF constraints. In the terminology of section IV, we would be interested in the optimization problem

$$\begin{aligned} \min_{\{M_x\}_{x \in V}} \quad & \mathbb{E}_{x \sim V} \mathbb{E}_{a \sim M_x} L(x, a) \\ \text{subject to} \quad & \forall x, y \in V : \quad D(M_x, M_y) \leq d(x, y) \\ & \forall x \in V : M_x \in \Delta(A) \end{aligned}$$

We were unable to implement this on the HomeCredit dataset because the number of constraints $O(n^2)$ where $n \approx 400,000$ was too large. However, future work on smaller datasets or using clever optimizations could tackle this question.

We also identified two promising future directions relating to fairness constraints and included code for partial implementations in the Github repository. One such avenue is using the TFCO package to constrain TensorFlow models for fairness. TFCO supports the optimization of inequality-constrained problems where constraints and objectives are represented as Tensorflow Tensor objects. We could use this

[3]The current FairTorch code requires calibration on labelled data, so we were unable to implement a version that could predict on Kaggle.

TABLE I: Supervised Learning (Baseline) Results

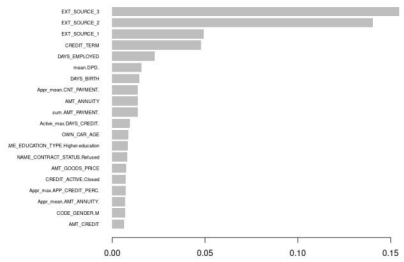
Model	Kaggle (Public) AUC	Self Split AUC Test (Train)	IF Constraint Violation	Parity (Age)	Parity (Gender)	Parity (Marital Status)
xgboost	0.77423	0.7692159 (0.8610486)	0.00234541	-0.0081221	-0.006536	-0.003332
logreg	0.76470	0.76688 (0.77421)	0.0019965	-0.0038910	-0.00442930	-0.0019692
Random Forest	0.6992	0.718303 (1.0)	0.0012657	0.00008481	0.00000884	-0.00007005
Neural Net (Tensorflow)	0.75593	0.762576 (0.78413)	0.00358499	0.001652	-0.000117	-0.0002779
Neural Net (Pytorch)	N/A	0.67884 (0.676937)	0.0001795	-0.0001435	-0.0001784	-0.0002219

TABLE II: Constrained Model Results

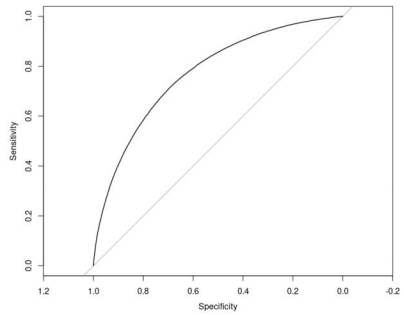
Model	Train Split AUC Test (Train)	IF Constraint Violation	Parity (Age)	Parity (Gender)	Parity (Marital Status)
logreg	0.66537 (0.67179)	0.00037964	0.00045788	-0.00024062	0.000355479
Pytorch	0.67580 (0.66784)	0.002816	-0.0007154	0.0003479	0.0007304
Random Forest	0.722595 (0.74356)	0.000188063	0.000029987	0.000093199	-0.00008617

TABLE III: Welfare by Model

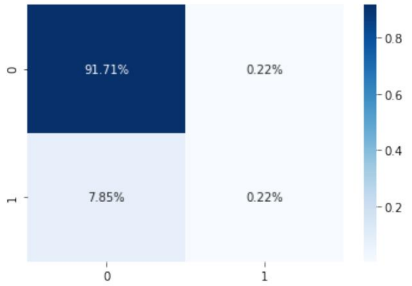
Model	Company Welfare Loss	Customer Welfare Loss	Relative Customer Impact
xgboost	$126,510,223	$111,279,708	823.03
logreg	$95,700,474	$49,500,000	363.547
Tensorflow	$97,136,748	$248,380,911	1659.15
Pytorch	$96,943,023	$7,182,000	42.124
Random Forest	$98,296,042	$3,433,500.0	22.416
Pytorch (constrained)	$98,174,524	$63,823,500	426.51
Logreg (constrained)	$98,318,650	$8,136,000	63.122
Random Forest (constrained)	$101,644,065	$360,000	2.1667



(a) Xgboost Variable Importance



(b) Xgboost ROC on Cross-Validation



(c) Confusion Matrix for Unconstrained Logistic Regression

package to implement fairness constraints (e.g. demographic parity), even in the nonconvex case.

The other potential avenue is the use of "black-box" fairness optimizers, such as AIFairness360 [16]. This package supports pre-, in-, and post-processing methods for more fair algorithms such as Adversarial Debiasing, Meta Fair Classifiers, and Gerry Fair Classifiers. In our partial implementations, these classifiers only outputted binary predictions, rather than probabilities, making them difficult to evaluate. Further, the fairness constraints and unbalanced dataset led to the model predicting almost entirely 0s, with $< 0.0001\%$ of 1s. This leads us to believe that the in-processing algorithms need a better way to support unbalanced datasets and that these algorithms may not be the best fit for our particular classification problem.

While our data does not allow us to make comparisons with traditional (i.e., non-ML) methods of predicting defaults and disbursing loans along our key metrics (fairness and welfare), it is encouraging that our ML models were able to achieve such low levels of parity, even when the baseline distributions of defaults differed substantially across protected groups (Figure 1). These results are particularly striking when set in contrast with previous work, particularly in economics, finding substantial biases in traditional means of disbursing loans. Though we were able to decrease the already low levels of dis-parity under constrained models, we did generally see losses in accuracy, implying real tradeoff between system-wide efficiency (total) welfare, and the fairness of welfare loss between protected groups. These results suggest that firms do have incentives irrespective of loanees to not only implement ML models that are on the whole more accurate and more fair than older methods, but also new models that can improve fairness on the consumer side with little loss on the firm side.

VII. Contributions

All project aspects were equally distributed among members. Jodie did data cleaning, non-constrained classification with xgboost, metric calculation, and fairness method exploration, Arthur did classification with logreg and neural networks, and constrained classification, and Andrew did exploratory data analysis, method investigation, and some method implementation over subgroups.

References

[1] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," 2011.

[2] J. Kleinberg, S. Mullainathan, and M. Raghavan, "Inherent tradeoffs in the fair determination of risk scores," *Proceedings of Innovations in Theoretical Computer Science*, 2016.

[3] H. Heidari, C. Ferrari, K. P. Gummadi, and A. Krause, "Fairness behind a veil of ignorance: A welfare analysis for automated decision making," *Thirty-second Conference on Neural Information Processing Systems*, 2018.

[4] B. E. Woodworth, Gunasekar, M. I. S., Ohannessian, and N. Srebro, "Learning non-discriminatory predictors." *Proceedings of the 30th Conference on Learning Theory (COLT)*, p. 1920–1953.

[5] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian, "Certifying and removing disparate impact." *Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

[6] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. M. Wallach, "A reductions approach to fair classification," *CoRR*, vol. abs/1803.02453, 2018. [Online]. Available: http://arxiv.org/abs/1803.02453

[7] A. Fukuchi, Y. Yabe, and M. Sode, "Fairtorch," https://github.com/wbawakate/fairtorch, 2021.

[8] S. Towson, "Ai can make bank loans more fair," *Harvard Business Review*, 2020. [Online]. Available: https://hbr.org/2020/11/ai-can-make-bank-loans-more-fair

[9] J. Miller, "Is an algorithm less racist than a loan officer?" *The New York Times*, 2020. [Online]. Available: https://www.nytimes.com/2020/09/18/business/digital-mortgages.html

[10] W. Dobbie, A. Liberman, D. Paravisini, and V. Pathania, "Measuring bias in consumer lending," *NBER Working Paper*, 2020.

[11] H. Credit, "Home credit default risk," https://www.kaggle.com/c/home-credit-default-risk/data, accessed: 2021-05-02.

[12] F. T. Commission, "Equal opportunity rights," https://www.consumer.ftc.gov/articles/0347-your-equal-credit-opportunity-rights, January 2013, (Accessed on 06/02/2021).

[13] S. Prince, "Bias and fairness in ai," https://www.borealisai.com/en/blog/tutorial1-bias-and-fairness-ai/, August 2019, (Accessed on 06/02/2021).

[14] D. Karlan and J. Zinman, "Expanding Credit Access: Using Randomized Supply Decisions to Estimate the Impacts," *Review of Financial Studies*, vol. 23, no. 1, pp. 433–464, Jan. 2010. [Online]. Available: https://academic.oup.com/rfs/article-lookup/doi/10.1093/rfs/hhp092

[15] Y. Liang, X. Jin, and Z. Wang, "Loanliness: Predicting Loan Repayment Ability by Using Machine Learning Methods," December 2019. [Online]. Available: http://cs229.stanford.edu/proj2019aut/data/assignment_308832_raw/26644913.pdf

[16] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias," Oct. 2018. [Online]. Available: https://arxiv.org/abs/1810.01943

[17] S. Bird, M. Dudík, R. Edgar, B. Horn, R. Lutz, V. Milan, M. Sameki, H. Wallach, and K. Walker, "Fairlearn: A toolkit for assessing and improving fairness in AI," Microsoft, Tech. Rep. MSR-TR-2020-32, May 2020. [Online]. Available: https://www.microsoft.com/en-us/research/publication/fairlearn-a-toolkit-for-assessing-and-improving-fairness-in-ai/