Patterns of Fairness in Machine Learning

Praveen Nair, Daniel Tong, and Anne Xu *

Halıcıoğlu Data Science Institute, University of California, San Diego

March 8, 2022

Abstract

Machine learning tools are increasingly used for decision-making in contexts that have crucial ramifications. However, a growing body of research has established that machine learning models are not immune to bias, especially on protected characteristics. This had led to efforts to create mathematical definitions of fairness that could be used to estimate whether, given a prediction task and a certain protected attribute, an algorithm is being fair to members of all classes. But just like how philosophical definitions of fairness can vary widely, mathematical definitions of fairness vary as well, and fairness conditions can in fact be mutually exclusive. In addition, the choice of model to use to optimize fairness is also a difficult decision we have little intuition for. Consequently, our capstone project centers around an empirical analysis for studying the relationships between machine learning models, datasets, and various fairness metrics. We produce a 3-dimensional matrix of the performance of a certain machine learning model, for a certain definition of fairness, for a certain given dataset. Using this matrix on a sample of 8 datasets, 7 classification models, and 9 fairness metrics, we discover empirical relationships between model type and performance on specific metrics, in addition to correlations between metric values across different dataset-model pairs. We also offer a website and command-line interface for users to perform this experimentation on their own datasets.

1 Introduction

Today, machine learning tools are becoming increasingly prevalent and are being used for decision-making in contexts that have crucial ramifications. However, the algorithms that are used often are suspect to many biases that may be obscure and difficult to isolate, especially within complex models, and it is often hard to determine if they are fair or not in the intuitive sense. Due to this issue, there have naturally been efforts to create mathematical definitions of fairness that could be used to estimate whether, given a prediction task and a certain protected attribute, an algorithm is being fair to members of all classes of the attribute.

However, just like how philosophical definitions of fairness can vary widely, mathematical definitions of fairness are not always agreed upon. Furthermore, different definitions of fairness can often be mutually exclusive in most real-world data, meaning that anyone trying to judge the fairness of a model must decide the specific metric that is most relevant to their task – a decision which is not only difficult, but also laden with the influence of values and politics.

What makes that decision even more difficult is that we know very little of how different fairness definitions relate to different models. Previous research has shown how some machine learning models lend themselves better than others in certain data contexts - such as neural networks being especially appropriate for unstructured data - where there is just as much work involved in discovering features from the data as there is in optimization. But we lack this same understanding with fairness.

Our capstone project centers around furthering research done in the area of fairness in machine learning, and how it relates to specific data and models. We have produced a 3-dimensional matrix displaying the performances of different combinations of models and datasets, evaluated on different fairness metrics. Upon investigating these results, we aim to answer several potential questions about the relationships that exist between these 3 dimensions as well as within them.

*Supervised by Professor David Danks, Professor of Data Science & Philosophy, University of California, San Diego

For example, we can analyze whether some models are able to perform more fairly for certain datasets, in the same way some models are more accurate than others for certain data. We can explore whether certain models, over the range of datasets, tend to be fairer than others using some definitions of fairness. By looking at correlations between the results for different fairness metrics, we can also find an empirical grouping of which fairness metrics tend to correlate with each other, and whether these correlations align with what we might expect given the philosophical definition of fairness underlying each metric.

Though the analysis done on our end can be insightful and potentially indicative of general patterns, we understand that our findings are heavily based on the specific data we have chosen to create our matrix with. Therefore, we have also created a website that allows users to input their own preprocessed datasets and evaluate them on our chosen models and fairness metrics, so they can apply the same kind of analysis to data more relevant to their own interests or tasks. Our code will also be publicly available for those who may want to run the project with their own specifications.

2 Methods

In our implementation of this project, we use 8 datasets, 7 machine learning models, and 9 fairness metrics, for a total of 504 unique metric values. The specifics are described below.

2.1 Data Acquisition

In order to use datasets that are well-suited for analysis of fairness, as well as relatively clean and well-formatted, we chose to use previously published datasets from the machine learning fairness literature. These datasets are usually attached or linked to the papers they come from, and are cited below in Table 1.

2.2 Data Preparation

Before passing the data into the main pipeline for analysis, we first preprocess the dataset from the initial form found at the data source into a standardized .csv file. This preprocessing includes one-hot encoding of categorical columns, conversion of labels to a numerical 0/1 format, imputation of missing values, and other common dataset preprocessing techniques. In order to encode metadata about the dataset that cannot be included in the CSV file itself, each preprocessing script also outputs a JSON config file with information that is manually entered such as the dataset name, the type of prediction, and most importantly, which columns contain the features, the sensitive groups, and the labels of interest.

2.3 Analysis Pipeline

Once the datasets have been processed, they are then passed into the main Python file for the analysis pipeline. Fundamentally, the analysis is a 3-layer nested loop, with every combination of dataset, model, and metric being run. Firstly, the program collects the dataset CSV file and the config JSON. Using the information from the config file, it selects the corresponding X, y, and group columns, and does a 75/25 train-test split to generate train and test partitions for each dataset. For each model, the script applies the model by training it on the train partitions, and outputs predictions on the test partitions. Finally, these predictions are passed in along with all previous data into the function for each metric, where a real-valued output is returned as the metric value. This value is recorded in a 3-dimensional Python dictionary, which after the full loop is complete, is converted into a Pandas dataframe for legibility.

The models used in this project are all contained in the scikit-learn package. They are: logistic regression, Decision Tree, Random Forest, Multilayer Perceptron, Support Vector Machine (SVM), k-Nearest Neighbors, and Naive Bayes classification. In each model, we used the default hyperparameters in scikit-learn. These models were chosen because they span a wide range of types of learning, from regression, to tree-based learning, to a basic neural network; they were also chosen because of their relative popularity.

Dataset	Domain	Description	Sensitive feature(s)	Shape (before preprocessing)	Citation	Dataset Link
Credit Card Clients	Finance	Predicts whether customers will default on payments	Gender	30000 rows, 24 attributes	[11]	Link
Obermeyer Health	Healthcare	Synthetic dataset of health data used for referral to future care	Race	48724 rows, 148 attributes	[5]	Link
Adult Census	Economics	Predicts whether income exceeds $50k/year based on census data	Race	32561 rows, 15 attributes	[3]	Link
Bank Marketing	Finance	Predicts whether a client will make a deposit subscription	Marital status, Age	45,211 rows, 17 attributes	[4]	Link, Paper discussing cleaning [7]
Law School	Law	Predict whether a candidate would pass the bar exam on first try	Race, Gender	18692 rows, 12 attributes	[10]	Link, Paper discussing cleaning [7]
Diabetes Patient Readmission	Healthcare	Predict whether a diabetic patient would be readmitted to the hospital	Gender	19136 rows, 55 attributes	[9]	Link
Communities and Crime	Criminal Justice	Predict whether a city/town will have high crime	Race	1996 rows, 92 attributes	[8]	Link
Student Performance	Education	Predict whether a student's final grade will be above 12	Gender	649 rows, 33 attributes	[2]	Link

Table 1: Table of Current Datasets Chosen

The metrics used are derived both from the scikit-learn [6] and fairlearn [1] packages. The first two metrics are overall classification metrics agnostic of protected groups: overall accuracy, and overall Brier score, which measures calibration. The rest measure balance between classes in certain classification metrics: false positive rate, F1 score, recall, accuracy, Brier score, demographic parity, and equalized odds, in most cases taking the range of values over all protected group instances. (e.g. if the protected class is sex, the false positive rate metric takes the false positive rate of a model for men, and the rate for women, then finds the absolute difference. Therefore, 0 is ideal, and 1 is the worst possible result.)

3 Results

3.1 Overview

Our project runs machine learning algorithms over a group of datasets, and evaluates these models' performance on a group of metrics. We can then slice the resulting matrix in several interesting ways to study the relationships between models and metrics, between models and datasets, and between metrics. Of course, metric values will vary in scale and value between different datasets due to differences in their underlying structures. Therefore, when aggregating metric values from different datasets, we use the rank of the model performance on the metric for that dataset. These ranks also adjust for whether a larger or smaller number is better for the specific metric – for example, a higher overall accuracy is better, while a lower range of accuracies between groups is ideal. Raw values for the entire dataset are available on our GitHub repository.

3.2 Analyzing Relative Results

3.2.1 Model Performance per Metric

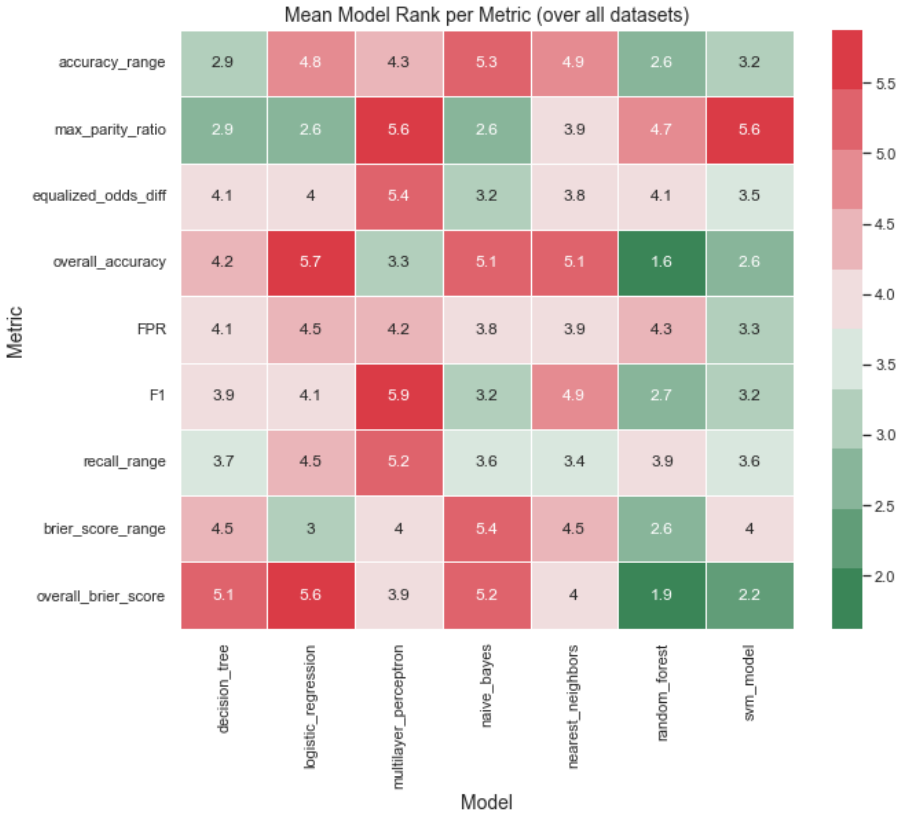

Figure 1: Mean rank of model across all datasets, for each metric.

The above chart displays the mean rank of a model on a particular metric across all datasets. For example, the logistic regression model had on average the third-best Brier score range, meaning that the quality of calibration of the model varied between protected classes the third-least in the logistic regression model when compared to other models.

There are certain observations we can make from the above heatmap about the relationship between models and metrics. The Random Forest model performed the best on both overall metrics, overall Brier score and overall accuracy, and also was the best at balancing accuracy and Brier score between classes, as evidenced by its ranks on Brier score range and overall accuracy. Meanwhile, the Support Vector Machine model does well across a broad spectrum of fairness metrics, indicating that it might be a model better-suited for fairness than something like a decision tree or logistic regression, which scored poorly across fairness metrics. We also observe that the models that tend to be the most accurate tend to do the worst on demographic parity, which makes sense, because demographic parity is the sole metric that does not take into account whether predictions have been made correctly.

3.2.2 Correlation Between Metrics

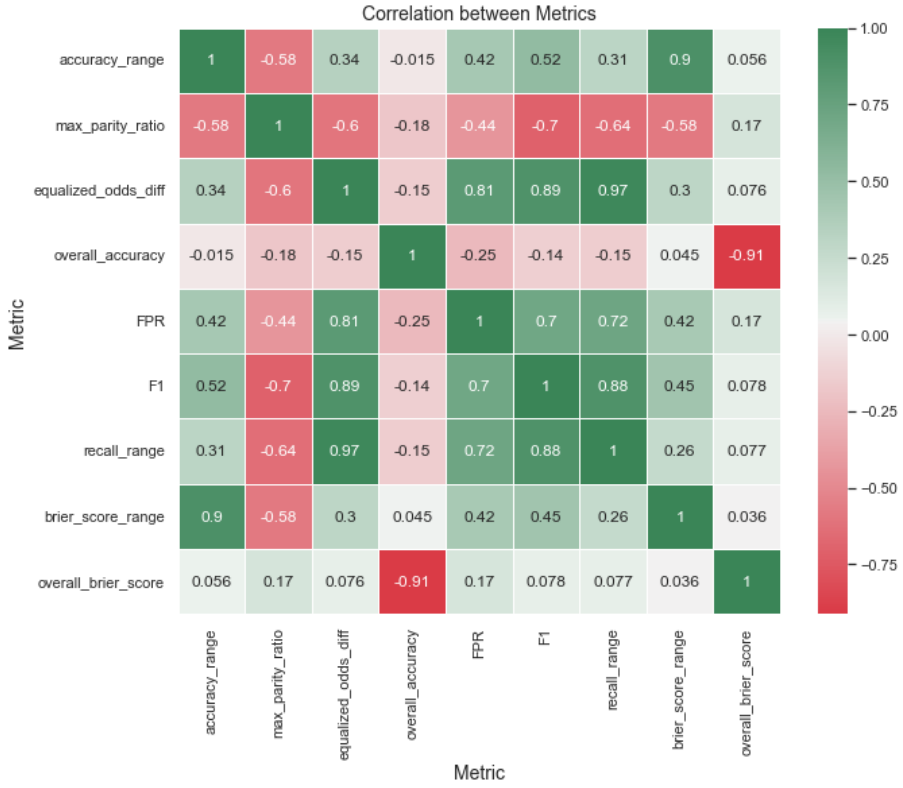

Figure 2: Correlation heatmap between raw metric values.

The above figure shows the correlation between (raw, unranked) metric values overall all datasets and models. First, we notice that demographic parity is negatively correlated with all other group fairness metrics, likely because, again, demographic parity does not take into account whether predictions have been made correctly. More interestingly, we also observe that overall accuracy is also negatively correlated with other metrics of fairness, implying a tradeoff in the accuracy of a model and how well it performs on fairness metrics. In general, however, we do find a positive correlation between most fairness metrics based on comparing different subsets of results to the true labels, such as false positive rate balance, equalized odds, and recall. We also find a very strong negative correlation between accuracy and Brier score (calibration). This may seem counterintuitive as Brier score is generally viewed as just an alternative way of measuring accuracy, but it may be because Brier score punishes being too confident in a prediction even when correct, while accuracy incentivizes predicting the correct class without consideration of exactly how confident it is.

Finally, we also observe that both the overall accuracy and Brier score are basically uncorrelated with the ranges of these metrics between groups, indicating that simply maximizing the metric performance on the entire dataset does not necessarily mean it will improve the degree to which it is equal for every demographic group.

3.2.3 Models and Datasets

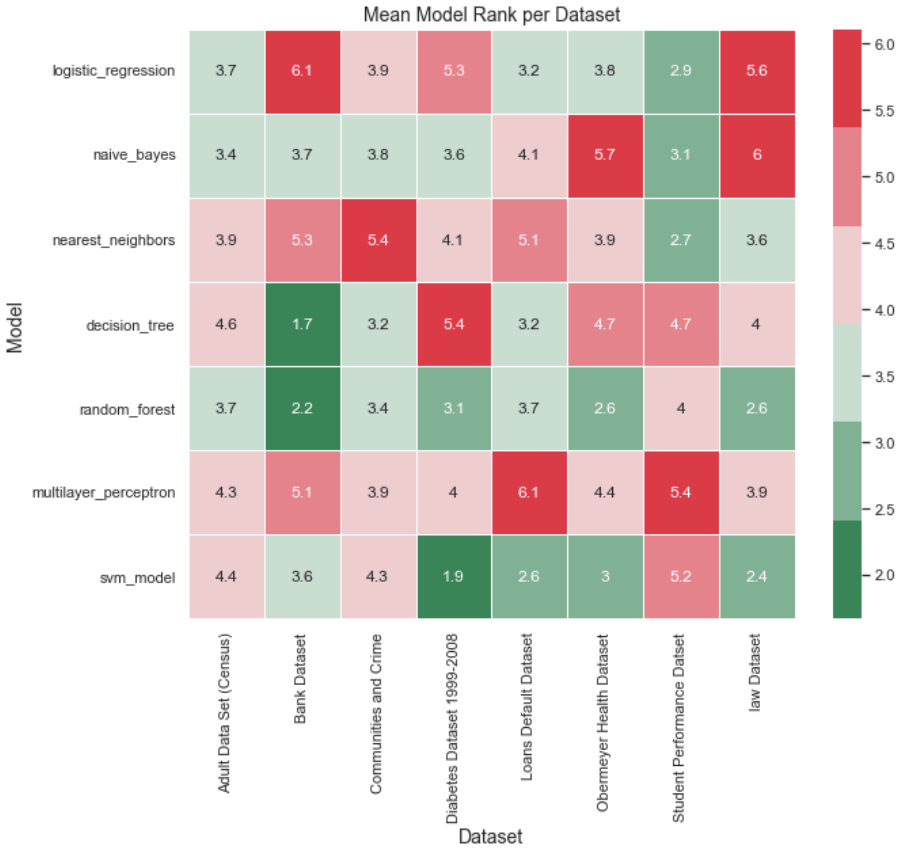

Figure 3: Mean model rank across all metrics, for each dataset.

This figure displays the mean rank of each model (across every fairness metric) for each dataset. While we might not be able to immediately ascertain why certain models perform better in different types of data, we still notice that the relative performance of models are not constant between datasets, and the best model for one dataset might not necessarily be the best model for another dataset. For example, we see the Random Forest model performs well across a variety of datasets, which is perhaps predictable given its reputation as a good "black-box" model that can be universally applied with little feature engineering.

3.3 User Extensibility

3.3.1 Website

In an effort to make our research more reproducible and replicable, we took further actions to make sure that we were transparent about our work and allow for others to use it. In addition to performing our own analysis, we also have created ways for users to perform similar experiments on their own data. This can be done in two ways.

The first is by going to the static website, where users may be informed about an overview of the project that features an explanation of the models and metrics for users who might not be familiar with them. The demo section of the page takes users to the webapp, which allows users to upload their own CSV file, cleaned to the same specifications as described above and with the last column

representing the target label. Users are then able to select what models they would like to run on their data, as well as what metrics. Upon submission, the webapp runs the pythonic code and returns an HTML table displaying the results.

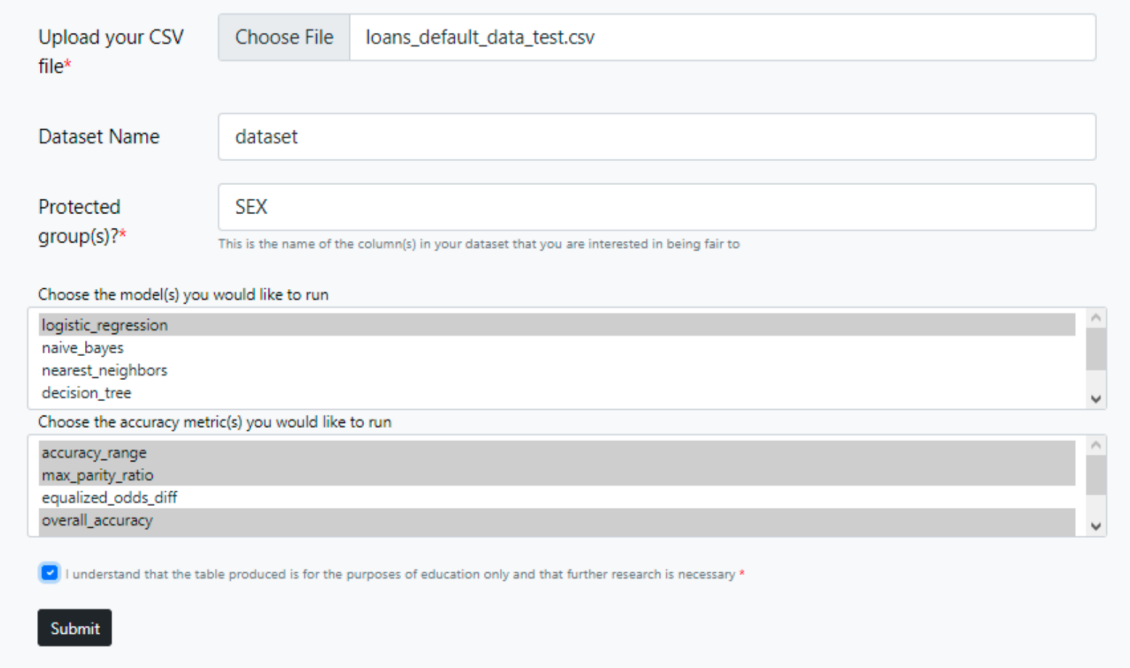

Figure 4: Screenshot using an example CSV with options inputted in the webapp

		accuracy_range	max_parity_ratio	overall_accuracy	FPR	recall_range	brier_score_range
dataset	**logistic_regression**	0.434131	0.588787	0.697333	0.209705	0.356164	0.077604

Figure 5: Resulting table displayed on the webpage

3.3.2 Github Repository

Although the webapp provides a easily accessible way for a curious user who does not have coding experience to quickly get results, due to the Python code being hosted on Flask and through Google Cloud servers, errors may be more obscure and performance may be slow for the user, and they may want to seek further modifications themselves. Users who want to build upon our work or have more experience with code have an option to delve deeper by simply cloning our own GitHub repository, where all users have to do to add datasets to our existing analysis is to place them into the cleaned dataset folder, with the proper cleaning performed and with the JSON config file filled out correctly. Then, users may run the main Python script at the top level of the directory to run the analysis. Instructions for users explaining these steps are located in the README file of the repository. More enterprising users could also add new models and metrics to our pipeline, which is also quite simple to do.

4 Conclusion

4.1 Overview

In this project, we have developed a framework for empirically analyzing the relationships between machine learning models and fairness metrics on a wide variety of datasets. While much more work

needs to be done to understand the ways in which model and metric choice affect how we determine whether an application of machine learning is fair, our broad survey of a handful of models, metrics, and datasets has nevertheless produced some interesting insights. These include which models tend to perform better on fairness metrics, empirical evidence of fairness-accuracy tradeoffs, and how nominally aligned metrics can disagree, such as in the example of accuracy and calibration.

Our project also offers opportunities for users to extend our analysis with new data, and if they are willing to edit our code, they can also add new models and metrics to our analysis. We do this through both a user-friendly website and a GitHub repository with instructions for new analysis. This analysis can be useful by offering a way to quickly evaluate how different machine learning models perform with regards to fairness on new data, and for student learning machine learning, it can be used to learn more about the models, metrics, and datasets at play. Our methods could also be used to evaluate new models or metrics when they are developed: if a researcher develops new machine learning metrics, they could use our analysis pipeline to determine how it correlates with existing metrics, and which machine learning models optimize it. If a new machine learning model for classification is developed, our pipeline could be used to determine how it stacks up against popular existing models across a variety of fairness metrics.

4.2 Limitations

There are some aspects to our project that limit the degree to which we can generalize its results. For example, we used default parameters when training our models for the sake of generalizing the pipeline and decreasing computation time. This is not always the best decision in machine learning, and some of the models might produce slightly different results if parameters were more specifically tuned. Another limitation of our study, naturally, is the limited number of models, metrics, and datasets we used, which could have been larger to generalize to more models and contexts of data.

4.3 Concerns

Another possible concern is whether our project might be used for the reverse-engineering of fairness. As we've discussed, machine learning metrics can differ greatly, if not be mutually exclusive, and therefore, different metrics can present a very different picture of a model's performance. With an increased amount of focus on fairness in recent years, it is possible that anyone developing machine learning algorithms will want to make their models seem more fair than they actually are by cherry-picking a metric that most aids that argument.

We understand that due to the potential for harm or misuse since our topic is based on fairness in machine learning and there may be researchers who would like to use the tool for other purposes, we tried to incorporate a disclaimer as a checkbox to remind users that the form is for educational purposes only. In making this a necessary disclaimer for the user to check before the program will run as well as making users choose what metrics they want to focus on before running the program, we hope that these user design choices will encourage users to approach this from a more educational point of view. While we will never be able to entirely prevent malicious uses of our project, we can at least try to ensure that any well-meaning students or researchers do not misinterpret its results.

4.4 Further Directions for Research

The most clear extension of our project would be to continue to add models, metrics, and datasets to the 3-D matrix, which would theoretically increase how generalizable the results would be to more machine learning contexts. On top of this, there is a great deal of theoretical research still to be done in machine learning fairness to grasp the relationships between fairness metrics, such as using real analysis to study the degree to which different metrics might overlap, which could help to determine how easy it is to perform something akin to p-hacking for fairness, as mentioned earlier. Another direction given more time would be to delve deeper into the specific datasets that already are in the matrix and seek to explain why a certain model works better in terms of metric performance rather than another. This would most likely involve the nuanced differences within a specific dataset and how the industry works, where background knowledge pertaining to the specific industry would be most useful in gleaning explanations.

There are also plenty of directions to expand when it comes to new types of prediction. Especially before the last couple of years, most of the work done on machine learning fairness dealt with binary classification tasks, which do cover a great deal of real-world examples; since the datasets and metrics we used are drawn from previous research, we also chose to focus on binary predictions. But machine learning is used in many other contexts, such as regression, multiclass prediction, text translation, image classification, and much more. Therefore, there exists a need to convey our philosophical understandings of fairness into new metrics that can evaluate all of these more complex types of prediction, and similar experiments to our project could be conducted using these new metrics to understand their properties and how they relate to other metrics and models.

References

[1] Sarah Bird, Miro Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, and Kathleen Walker. Fairlearn: A toolkit for assessing and improving fairness in ai. Technical Report MSR-TR-2020-32, Microsoft, May 2020.

[2] Paulo Cortez and Alice Maria Gonçalves Silva. Using data mining to predict secondary school student performance. 2008.

[3] Ronny Kohavi and Barry Becker. Adult data set.

[4] Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014.

[5] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

[6] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[7] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, and Eirini Ntoutsi. A survey on datasets for fairness-aware machine learning. *arXiv preprint arXiv:2110.00530*, 2021.

[8] Michael Redmond and Alok Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, 141(3):660–678, 2002.

[9] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international*, 2014, 2014.

[10] Linda F Wightman. Lsac national longitudinal bar passage study. lsac research report series. 1998.

[11] I-Cheng Yeh and Che-hui Lien. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert systems with applications*, 36(2):2473–2480, 2009.