Final Project Outline

EECS 298 Group 3: Adhav Rajesh, Garv Patel, Michael Peng, Tom Sherman
Winter 2024

Our Project

Three statistical metrics measure the fairness of a Machine Learning model: Independence (equal acceptance rate by group), Separation (error rate parity by group), and Sufficiency (calibration by group). The goal of our project is to visualize the tradeoffs between these three machine learning fairness metrics while predicting a total person's earnings in the Public Use Microdata Sample (PUMS) published by the US Census for the state of Michigan.

The questions we will answer with our project are the following:

1. How does prioritizing different fairness metrics impact the predictions of total earnings in the US Census PUMS dataset?
2. Which visualizations best demonstrate the tradeoffs between independence, separation, and sufficiency in mitigating bias in machine learning models?
3. What insights can be gained from these visualizations to help stakeholders make informed decisions about fairness in predictive modeling?

Through answering these questions, we hope to provide a comprehensive view of what fairness metrics are, how they impact the results of machine learning models, and provide recommendations on how to use this information to best empower engineers to create both accurate and fair models.

Background

In our project focusing on the impact of fairness on a machine learning algorithm, we explore a sociotechnical system lying at the intersection of economics, machine learning, and equity. The sociotechnical system involves stakeholders like social researchers, data scientists, policy makers, and the general public. The core technical component of this is a machine learning algorithm developed to predict people's income based on census data. These sorts of algorithms then can be used to affect many different areas, such as by financial institutions to determine lending, by economic researchers to investigate poverty, and by policy makers to make informed decisions. Bias in this algorithm can reinforce and perpetuate existing socioeconomic inequalities, which can manifest in unfair predictions that disadvantage a certain group based on a protected attribute such as gender, age, or race. Increasing fairness through metrics like independence, separation, and sufficiency often comes at a trade-off with model accuracy and validity. This harm of perpetuating biases and inequities is particularly relevant to our classes' examination of how fairness can be properly implemented and interpreted in

sociotechnical systems. Through visualizing the trade-offs between fairness metrics and model performance, we aim to deepen our understanding of fairness's impact on machine learning algorithms in this context of income prediction, informing stakeholders' decision-making.

Previous Work

Throughout the history of machine learning, ensuring fairness across socioeconomic groups has been a persistent challenge that the field continually strives to address. Many harms have been linked to the adoption of ML models that can withhold resources or opportunities from minorities or those of lower socioeconomic status. For example, the COMPAS model we have explored in class has been found to exhibit a racial bias against African American defendants by classifying their risk for future crimes to be higher [1]. ML models have also systematically rejected loans to women and non-Caucasian people. These harms have immediate harmful effects, and in the long term can enforce and create systemic issues for these groups [2]. Work has been done in the field to explain the reasons for why these harms within machine learning models arise. Oftentimes, correlations emerge in datasets through proxies, which are variables that correlate heavily to sensitive characteristics like race or gender. If these factors are built into a model, it will start to express harms that have been shown throughout history [3]. One of the methods used to mitigate the effect of proxy variables on a model is “to utilize graph and network based methods for discovering proxies with respect to anonymity criteria of specific notions of fairness” [4]. Moreover, there are additional methods available, such as the Folktables project, which aims to assess the impact of sensitive traits on different metrics.This project uses the PUMS dataset to understand the complexities and consequences of the ML model data. It contains a variety of visualizations that creators of ML models can use to decide which factors to include in their datasets [5]. Our project will use some of these methods to mitigate the harms of machine learning algorithms, but we will focus on prioritization of fairness metrics to outline which one may cause the least harm.

Motivation

Our project dives into the critical role of fairness in machine learning algorithms and AI by examining the effects of applying fairness metrics to a machine learning model that predicts income using the PUMS dataset. An unfair algorithm and an inaccurate algorithm can perpetuate and reinforce stereotypes and current social inequalities. It is important that we better understand the interplay between incorporating fairness into models and its effect on accuracy. Through an analysis of the independence, separation, and sufficiency metrics we bridge the gap between theoretical concepts on fairness and real practical socioeconomic impacts. By leveraging concepts from the course such as statistical fairness, independence, sufficiency, and separation, we explore how the metrics influence the performance and fairness trade-offs of the income prediction model.

Methodology

Acquire Relevant PUMS Data

The first thing we must do is select and download the subset of PUMS data we will use for the project. At the moment, we plan on limiting the scope of our data to PUMS data from the state of Michigan and the following nine variables with their respective codes and descriptions in the table below:

Sex (SEX)	Described as either male or female
Age (AGEP)	Age in years
Recoded detailed race code (RAC1P)	Race as described by one of nine categories (ex. White alone, Black alone, Asian alone, two or more races, etc.)
Educational attainment (SCHL)	Measured by grade of school completed or highest degree attained
Quarter of birth (QTRBIR)	Determined by birth month and the following quarters: I. Quarter 1 - January through March II. Quarter 2 - April through June III. Quarter 3 - July through September IV. Quarter 4 - October through December
Class of worker (COW)	A representation of a worker's employer (ex. Employee of a private for-profit, federal government employee, etc.)
Means of transportation to work (JWTRNS)	A workers method of transportation to work (ex. Car, bus, walked, worked from home, etc.)
Total person's earnings (PERNP)	A person's total earnings including negative earnings.
PUMS person weight (PWGTP)	The PUMS dataset is a weighted sample. This column represents the corresponding weights to be assigned to each row of the dataset in a range of 1-9999.

To download this data, we will go to the Census Data webpage (https://data.census.gov/mdat/#/), select the above variables, select Michigan as our desired location, and then download the dataset as a csv file.

Luckily, as the dataset is anonymized and weighted, there is no data cleaning we will need to do. However, we will filter our data to only include individuals of age 18 or older.

Create Machine Learning Models

For our project, we will use multiple linear regression to predict a person's earnings. We chose to use multiple linear regression because it is simple for stakeholders to understand, easy to implement, explainable, and easy to modify for our use case given the differentiability of the loss function with respect to each model parameter. We plan to integrate fairness metrics into our loss function and use Stochastic Gradient Descent to optimize the multiple linear regression model.

For this project, we will create five distinct models:

Model 1 - No Fairness

In this model, we will use all of the variables (including sensitive variables) to predict a person's total income.

Model 2 - Independence

In this model, we will select a single sensitive attribute (sex, age, or race) and ensure demographic parity across that attribute.

Model 3 - Separation

In this model, we will use the same single sensitive attribute in model 2 and ensure equalized error rates across that attribute.

Model 4 - Sufficiency

In this model, we will use the same single sensitive attribute in model 2 and ensure sufficiency of that attribute.

Model 5 - Optimization

In this model, we will construct the loss function using a weighted combination of all three fairness metrics. We will search a wide variety of weights and assess the fairness implications of each tradeoff.

Visualizations

There are a variety of visualizations we could use to best showcase the similarities and differences between our five models. Although to an extent we will have to play this by ear depending on our results, below are some ideas of visualizations we might find useful.

Model Performance Comparison

In a table, this visual shows the differences in metrics such as accuracy and precision between various models. This type of visualization could be helpful in showing the performance differences between our five models.

Variable Importance Plot

This visualization shows the respective importance of each independent variable on the prediction of the dependent variable. This type of plot could be done as either a bar graph or heatmap. This plot helps explain our model's decisions.

Partial Regression Plot

These plots show the relationship between the dependent variable and a particular independent variable. These could be used to show the independence fairness metric.

Insights

Although we don't know what our exact insights about fairness in machine learning models will be yet, it will be important to display this information succinctly, accompanied by the visualizations discussed above, and with a clear claim presented to stakeholders about how to best incorporate fairness into machine learning models.

Minimum Working Product

The riskiest part of our project is the technical implementation of a custom Machine Learning model that is capable of optimizing for our fairness metrics. Therefore, our minimum working example implements a custom multiple linear regression model that has a *separation* measurement in its loss term. Through Small-Batch Stochastic Gradient Descent, this model optimizes its coefficients against the sum of squared residuals plus a multiple of the separation measurement. In this process, the separation term is calculated by bucketing income by increments of $10,000 (for example, $45,130 would be bucketed into "$40,000 to $50,000") and computing the mean of differences in probability between $\mathrm{P}[\mathrm{H}(X) \mid \mathrm{Y}, \mathrm{A}{=}a]$ and $\mathrm{P}[\mathrm{H}(X) \mid \mathrm{Y}, \mathrm{A}{=}b]$, where A is gender, a is male, and b is female. The full working example is attached as `mlr.py`.

We tested our minimum working example on the first 10,000 rows of our downloaded dataset (to save training time) and compared both its sum of squared residuals and separation measurement against (1) a typical Linear Regression model and (2) a conventional neural network. The results from these tests are attached in `custom_mlr.pdf` and summarized as follows:

Model	Sum of squared residuals	Separation score
Our model	1469070700	**0.00449**
Linear regression	**922083555**	0.0128
Neural network[1]	1269701330	0.00698

The effect of our addition of the separation score to our model's loss function is apparent. Our model sacrifices overall accuracy (sum of squared residuals) in favor of fairness (lower separation score).

For our next steps, we are interested in speeding up the training process of our model. Currently, it uses mini-batch stochastic gradient descent, and it computes the separation part of its loss function using the mini-batch. To improve upon this, we may split the least-squares and minimize-separation optimization steps and perform them in an alternating manner. We'll also investigate ways to JIT-compile the separation computation step using JAX and off-load it onto GPUs. Once we have achieved scalability with this training process, we will add *independence* and *sufficiency* to the model's loss function and conduct more experiments.

[1] Shape: (52, 80, 1); ReLU activation function.

Citations

[1] Aria, Khademi., Vasant, Honavar. (2018). Algorithmic Bias in Recidivism Prediction: A Causal Perspective. 13839-13840.
[2] Borrellas, P., & Unceta, I. (2021). The Challenges of Machine Learning and Their Economic Implications. Entropy (Basel, Switzerland), 23(3), 275. https://doi.org/10.3390/e23030275
[3] Wang, S. (2021, March 11). *Practice ai responsibly with proxy variable detection*. Medium. https://medium.com/bcggamma/practice-ai-responsibly-with-proxy-variable-detection-42c2156ad986
[4] Caton, S., & Haas, C. (2023). Fairness in machine learning: A survey. *ACM Computing Surveys*. https://doi.org/10.1145/3616865
[5] Hägele, A. (n.d.). Visualizing folktables. https://haeggee.github.io/posts/folktables