Michael Hollins: 02553236

0. Introduction

0.1. Data

This report examines the trade off between accuracy and fairness in machine learning models by using the American Community Survey (ACS) dataset.[1] We split the data into train and test datasets with proportions 0.7 and 0.3 respectively. We further split the train dataset into a train-train and train-val dataset with proportions 0.8 and 0.2 respectively, five times, in order to provide a more accurate estimate of the true underlying mean performance of the model and make our estimates less noisy.

0.2. Model

Our task is to predict whether someone is employed based on several features such as age, educational attainment, marital status, ancestry record and so on.[2] More formally, our target label of employment y_i takes values in the set $\{0, 1\}$ for data point i, and our vector of input features is given by X_i. Then, our logistic regression predicts the probability of employment $P(y_i = 1|X_i)$ as:

$$\hat{p}(X_i) = \frac{1}{1 + \exp(-X_i w - w_0)} \tag{1}$$

To find the optimal weights w, we use a loss function which measures the distance between the true and predicted label. In SCIKIT-LEARN, this is implemented as an optimisation problem minimising the loss function:

$$\min_w \sum_{i=1}^{n} s_i(-y_i log(\hat{p}(X_i)) - (1-y_i) log(1-\hat{p}(X_i))) + \frac{1}{C} r(w) \tag{2}$$

s_i refers to the weights assigned to the specific training example (the vector **s** is formed by element-wise multiplication of the class weights and sample weights).[3] For now, these are all set to one, but in Section 2 we consider how changing these impacts our model.

The final term in equation 2, $r(w)$, is known as the regularisation term and it ensures that our model doesn't overfit to the training data. It does this by taking the ℓ_2 norm $= \frac{1}{2} w^T w$ to penalise large absolute weight values. Crucially, the hyperparameter C determines the strength of the regularisation. Low values of C imply stronger regularisation, and vice-versa, and this is the hyperparameter that we will go on to vary in the Sections 2, 3, and 4.

[1] See [1] for more details on the dataset.

[2] Specifically, we want to predict that they are a "civilian employed, at work", as defined in the documentation here. In Section 4 we modify this definition.

[3] See [4] for more details on SCIKIT-LEARN.

0.3. Results

For reference, the results for the models are summarised in Table 1. Models are either standard with weight $s = 1$, or fairness-aware by being reweighed as described in Section 2. Values of the hyperparameter C are chosen according to the model criterion, and finally, accuracy and fairness are reported on the test set.

Section	Model	Criterion	C value	Accuracy	Fairness
1.1	Standard	Highest accuracy	10^{-3}	0.7534	-0.6625
1.2	Standard	Best fairness	10	0.7525	-0.6223
2.1	Fairness-aware	Highest accuracy	10	0.7198	0.0023
2.2	Fairness-aware	Best fairness	10	0.7198	0.0023
3.1	Standard	Highest AF score	10	0.7525	-0.6623
3.2	Fairness-aware	Highest AF score	10^{-3}	0.7206	-0.0232

Table 1. A summary of model performance

1. Generalisation, accuracy and fairness

1.1. Model selection with highest accuracy

We begin by examining how changing our regularisation through values of C impacts accuracy, defined as:

$$\text{accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{positives} + \text{negatives}} \tag{3}$$

Figure 1 plots values of C in powers of 10^{-5} to 10^{6} against model accuracy. The boxplot provides a summary of results across the five train-train/train-val data splits mentioned in Section 0.1. Each whisker shows the maximum and minimum values across the splits, the box the interquartile range, and the line within the box shows the median value. This is overlaid with a red dot showing the mean, which we use for model selection.

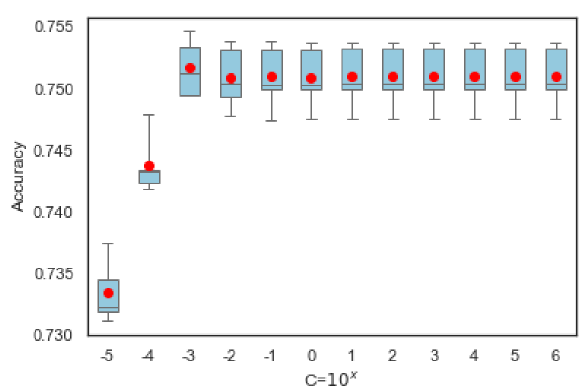

Figure 1. Accuracy of model varying over parameter C. Box whiskers denote highest and lowest results over the 5 sets of train-train/train-val data splits. The red dot is the **mean** result.

From Figure 1 we see that increasing the default strength of regularisation (i.e. reducing C) from 1 improves accuracy, but this is only up to a point, as very strong regularisation leads to steep declines in accuracy.

As $C = 10^{-3}$ produces the best average result, we use a logistic regression with the hyperparameter C set accordingly. The final accuracy of this model is 0.7534, i.e. by selecting the regularisation parameter C to 10^{-3}, our trained model correctly predicts employment status for an individual (y_i) given a set of features (X_i) 75% of the time.

Having explored accuracy, we now turn to the fairness of our model. The fairness metric we use throughout this project is equality of opportunity. As set out in Hardt et al. (2016) [2], in the case of binary variables, for the outcome $y = 1$, equality of opportunity implies that the prediction $\hat{Y}$ has equal true positive rates across two demographics $A = 0$ and $A = 1$.

In our context, equality of opportunity would imply that our model correctly predicts that a person is employed in equal share, regardless of their disability status. In other words, being disabled would have no correlation with the model's prediction, given the person is employed. As implemented in AIF360, this is calculated as the true positive rate difference, where a true positive rate is simply the ratio of true positives (actually employed) to positive examples in the data.[4] With having no disability being the privileged position, then the equality of opportunity, otherwise known as the true positive rate difference, is:

$$\mathrm{TPR}_{D=unprivileged} - \mathrm{TPR}_{D=privileged} \tag{4}$$

The closer the score is to zero, the better the equality of opportunity displayed by the model. While $C = 10^{-3}$ provides the best accuracy, the fairness metric on the test dataset is -0.6625. This means that our accuracy-maximising model does a much worse job at correctly predicting that someone is employed when they have a disability, compared to someone that doesn't.

1.2. Model selection with best fairness

To try to address this, we repeat the exercise in the Section 1.1 but instead select a C value which maximises our fairness metric.

In contrast to the previous case, here the best model (lowest true positive rate difference) has $C = 10$, i.e. a weak regularisation. Using this hyperparameter C on the test set, we derive an equality of opportunity of -0.6223 and an accuracy of 0.7525.

These experiments yield two important results. Firstly, there is a trade-off between accuracy and fairness when hyperparameter tuning: optimising over one leads to worse performance in the other. Second, hyperparameter tuning in the case of our logistic regression results in only modest shifts in accuracy and fairness. While our model has respectable predictive accuracy around 75%, it also consistently low equality of opportunity.

[4]See here for full implementation code.

2. Accuracy and fairness following fairness-aware preprocessing

To improve equality of opportunity, we can use pre-processing techniques which aim to remove discrimination before a classifier is learned. To that end we employ the reweighing method of Kamiran and Calders (2012) [3]. This reweights the training data in each (group label) combination differently. For each data point with $Y = 1$ and sensitive characteristic $A = 0$, the weight is:

$$\mathrm{W}_{Y=1,A=0} = \frac{P(Y=1)P(A=0)}{P(Y=1, A=0)} \tag{5}$$

Intuitively, we desire that the probability of employment $P(Y = 1)$ is statistically independent of being disabled $P(A = 0)$. If this were the case, then $P(Y = 1)P(A = 0) = P(Y = 1, A = 0)$ and our weights would be one as we had previously. However, this is not what we observe, so by dividing each observation by its *observed* joint probability we can recover the *unbiased*, independent probability we desire. By reweighing, we set the weights denoted s_i in equation 2 to the following:

	Disabled	Not disabled
Employed	2.834	0.885
Not employed	1.170	0.581

Table 2. Preprocessing weights on train dataset

With these pre-processed weights applied ahead of training, we now repeat the exercises performed in Section 1.

2.1. Fairness-aware highest accuracy

With the training data reweighted, the highest average accuracy is achieved when $C = 10$. On the test dataset, we report an accuracy of 0.720 and equality of opportunity of 0.002. Comparing with our results from Section 1, for a loss of accuracy of around 3% we have effectively eliminated discrimination from our model predictions.

2.2. Fairness-aware best fairness

We next select the best fairness-aware model with the highest average fairness score. This results in an identical solution to those in subsection 2.1. This makes intuitive sense when we consider that the very purpose of reweighing is, in the words of Kamiran and Calders (2011), to allow a model to optimise accuracy but without having discrimination in its predictions on test data [3]. Therefore, there is no longer any trade-off between accuracy and fairness: we simply reweigh our model to ensure fairness, and subject to this constraint we select C to optimise accuracy.

3. Accounting for both accuracy and fairness

In this section we introduce a model selection criteria that values both accuracy and fairness. We construct new weights $s'_i = \alpha s_i + (1 - \alpha)$ where s_i are the fairness-aware reweights and $\alpha \in 0, 1$. At the extremes, $\alpha = 0$ represents our standard model, $\alpha = 1$ represents our fairness-aware model, and a value in between will be a compromise between them. For example, if one cared equally between accuracy and fairness one could set $\alpha = 0.5$. To assess the models, we can use a consolidated model score, $AFscore = accuracy + (1 - |fairness|)$. A perfectly accurate and fair model would achieve a score of two; a perfectly inaccurate and unfair model would achieve a score of zero. Figure 2 shows how different values of α and C impact a model's AF score. The blue dots correspond to our standard model with weights of one, and the purple dots show the fairness-aware model.

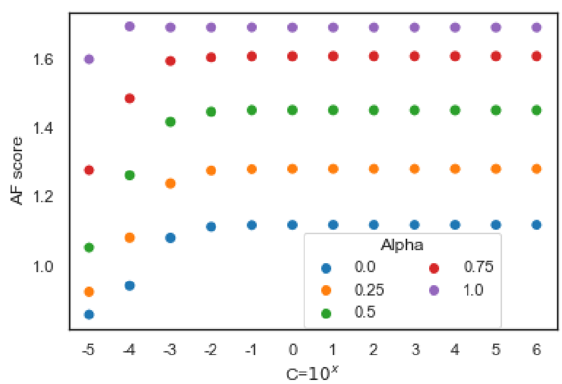

Figure 2. Consolidated AF score of models ranging from standard (alpha=0) to fairness-aware (alpha=1), averaged over five sets of train-train/train-val.

Using the criterion of the highest AF score, for the standard model ($\alpha = 0$), the hyperparameter giving the best average is $C = 10$. With this parameter, the final accuracy on the test set is 0.7525 and fairness of -0.6223, identical to the results in Section 1.2. This makes sense as from Figure 2, we see that the more fairness-aware-weighted the model (higher α), the higher the consolidated AF score. This is because large gains in fairness come with small costs to accuracy, and so a score that combines the two linearly will favour fairness-maximising models.

The best fairness-aware model for the AF-score is when $C = 10^{-4}$, scoring $AF = 1.6975$. Test accuracy with $C = 10^{-4}$ is 0.721 and fairness is -0.023, only very marginally different from the results in Section 2.2.

4. Going further: reconsidering employment

Throughout this report we have seen that higher accuracy often come at the expense of equality of opportunity, and vice versa. In this Section, we show that in some circumstances both can be improved at the same time. Until now we have considered an arguably narrow definition of employment as "civilian employed, at work". [5] If we expand our definition of employment to include other categories such as "civilian employed, with a job but not at work", "armed forces, at work", "armed forces, with a job but not at work", then we increase our sample of employed from around 88,000 to over 90,000.

Table 3 shows the changes to the results that follow from this slight redefinition of employment, y, using a more complete set of values from the data dictionary. We re-run the experiments from Sections 1 and 2, indicating improvements in green, and worsenings in red.

Section	Model	Criterion	C value	Accuracy	Fairness
1.1	Standard	Highest accuracy	0.01	0.008	0.048
1.2	Standard	Best fairness	1	0.008	0.014
2.1	Fairness-aware	Highest accuracy	0.001	0.009	0.008
2.2	Fairness-aware	Best fairness	1	0.008	-0.009

Table 3. Changes to results with respect to Table 1 following a redefinition of employment. Change signs are new values - old values. Improvements are indicated in green, worsenings in red.

In the case of the standard logistic regression model, there is a strict improvement to both accuracy and equality of opportunity. While there is some deterioration in equality of opportunity in the fairness-aware models, it remains very close to zero in magnitude.

This shows that model improvements don't only occur through sophisticated preprocessing and calibrating parameters: sometimes it is simply through mundane considerations of feature construction and measurement.

5. Conclusion

This report has shown in Section 1 that there can be times when model accuracy and fairness are in tension. Section 2 showed how using reweighing is an effective way to near-eliminate model discrimination with only a little sacrifice of accuracy. This was further illustrated in Section 3 which showed how models that value both accuracy and fairness tend to select the same hyperparameters and get similar scores to the fairness-aware methods as large gains to fairness only cost a modest level of accuracy. Finally, in Section 4, we took a step back to consider what we actually want to measure, and by expanding our definition of employment, we managed to improve both accuracy and fairness compared to the baseline model.

[5] See the data dictionary here

References

[1] F. Ding, M. Hardt, J. Miller, and L. Schmidt. Retiring adult: New datasets for fair machine learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6478–6490. Curran Associates, Inc., 2021.

[2] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning, 2016.

[3] F. Kamiran and T. Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33:1–33, 2012.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.