HAMLET4Fairness: Enhancing Fairness in AI Pipelines through Human-Centered AutoML and Argumentation

Supplementary Materials

Table 1: Algorithms and number of hyperparameters for each of the steps in HAMLET4Fairness. Algorithm names and hyperparameters (No. Hps) are imported from the scikit-learn Python library.

Step	Algorithm	#Hyperparams
Mitigation	CorrelationRemover	1
	LearnedFairRepresentation	2
Normalization	StandardScaler	2
	MinMaxScaler	0
	RobustScaler	2
	PowerTransformer	0
Discretization	Binarizer	1
	KBinsDiscretizer	3
Feature Eng.	SelectKBest	1
	PCA	1
Rebalancing	NearMiss	1
	SMOTE	1
Classification	KNeighborsClassifier	3
	RandomForestClassifier	7
	MLPClassifier	6

A Multi-objective settings

We consider the problem of optimizing multiple objectives. Specifically, for our use case, we leverage balanced accuracy (Pedregosa et al. 2011) as a quality metric for the performance, and demographic parity ratio and equalised odds ratio (Weerts et al. 2023) for fairness.

Given a binary classification problem, let $\hat{Y}$ represent the predictions made by a specific model on a given dataset, and let Y denote the corresponding ground truth labels. Then, we define:

- **True Positive** (TP) predictions: cases when $\hat{y}_i = 1$ and the corresponding $y_i = 1$, where $\hat{y}_i \in \hat{Y}$ and $y_i \in Y$;
- **True Negative** (TN) predictions: cases when $\hat{y}_i = 0$ and the corresponding $y_i = 0$, where $\hat{y}_i \in \hat{Y}$ and $y_i \in Y$;
- **False Positive** (FP) predictions: cases when $\hat{y}_i = 1$ and the corresponding $y_i = 0$, where $\hat{y}_i \in \hat{Y}$ and $y_i \in Y$;
- **False Negative** (FN) predictions: cases when $\hat{y}_i = 0$ and the corresponding $y_i = 1$, where $\hat{y}_i \in \hat{Y}$ and $y_i \in Y$.

Thus, we can introduce the balanced accuracy as:

$$\text{Balanced Accuracy} = \frac{1}{2}\left(\frac{\text{TP}}{\text{TP}+\text{FN}} + \frac{\text{TN}}{\text{TN}+\text{FP}}\right)$$

As to the fairness metrics, let us now represent the sensitive attribute with $X_s \in \mathbb{X}$ and the set of all possible values of the sensitive attribute with $\mathcal{X}_s$. We define the Demographic Parity Ratio (DMR) and Equalised Odds Ratio (EOR).

$$\text{DMR} = \frac{\min_{x_s \in \mathcal{X}_s} \mathbb{P}(\hat{Y} = 1 \mid X_s = x_s)}{\max_{x_s \in \mathcal{X}_s} \mathbb{P}(\hat{Y} = 1 \mid X_s = x_s)}$$

$$\text{True Positive Ratio} = \frac{\min_{x_s \in \mathcal{X}_s} \mathbb{P}(\hat{Y} = 1 \mid X_s = x_s, Y = 1)}{\max_{x_s \in \mathcal{X}_s} \mathbb{P}(\hat{Y} = 1 \mid X_s = x_s, Y = 1)}$$

$$\text{False Positive Ratio} = \frac{\min_{x_s \in \mathcal{X}_s} \mathbb{P}(\hat{Y} = 1 \mid X_s = x_s, Y = 0)}{\max_{x_s \in \mathcal{X}_s} \mathbb{P}(\hat{Y} = 1 \mid X_s = x_s, Y = 0)}$$

$$\text{EOR} = \min\left(\text{True Positive Ratio}, \text{False Positive Ratio}\right)$$

In particular, in (Weerts et al. 2023), the probability is estimated with the frequencies on the real dataset.

When considering multiple objectives, we cannot definitively decide which solution is the best, as improvements in one objective may lead to degradation in another. We seek a Pareto front, which represents the solutions that have the best trade-offs with respect to the objectives (i.e., the set of non-dominated solutions). A solution is considered non-dominated if there is no other solution that is better in at least one objective, without being worse in another. The Pareto front $P_{\mathcal{D}_{val}}(\mathcal{H})$ for a given set of solutions $\mathcal{H} \subset \mathbb{H}$ evaluated on dataset $\mathcal{D}_{val}$ is defined as:

$$P_{\mathcal{D}_{val}}(\mathcal{H}) = \left\{ H \;\middle|\; \begin{array}{l} H \in \mathcal{H}, \nexists H' \in \mathcal{H} : \\ \quad \forall m \in \{1, \ldots, M\} : \\ \quad\quad \mathcal{M}_m(H', \mathcal{D}_{val}) \geq \mathcal{M}_m(H, \mathcal{D}_{val}), \\ \exists j \in \{1, \ldots, M\} : \\ \quad \mathcal{M}_j(H', \mathcal{D}_{val}) > \mathcal{M}_j(H, \mathcal{D}_{val}) \end{array} \right\}.$$

Given a Pareto front, and the quality metric values of the models within $v = \{(\mathcal{M}_1(H), \ldots, \mathcal{M}_m(H)) : H \in P_{D_{val}}(\mathcal{H})\}$, the hypervolume is defined as:

$$\text{Hypervolume} = \text{Volume}\left(\bigcup_{v_i \in v} \{\boldsymbol{x} \in \mathbb{R}^d \mid v_i \preceq \boldsymbol{x} \preceq \boldsymbol{r}\}\right)$$

where $\boldsymbol{r}$ is the optimal reference point (i.e., best value for each quality metric).

B Search Space

The tests are run on datasets from OpenML (Vanschoren et al. 2013)—a well-known repository for data acquisition and benchmarking. As it provides already-encoded datasets, we do not consider the encoding step. As to Imputation, the Credit-g and COMPAS datasets have no missing values, and in the COMPAS dataset, we drop the few instances with missing values. Besides, we dropped instances belonging to sensitive groups that are extremely under-represented ($< 0.5\%$).

Except for that, we included all the Data Pre-processing steps available in the scikit-learn (Pedregosa et al. 2011) Python library (plus imbalance-learn (Lemaitre, Nogueira, and Aridas 2017) for Rebalancing transformations). The leveraged steps, algorithms per step, and hyperparameters per algorithm are reported in Table 1.

References

Lemaitre, G.; Nogueira, F.; and Aridas, C. K. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* 18:17:1–17:5.

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

Vanschoren, J.; van Rijn, J. N.; Bischl, B.; and Torgo, L. 2013. Openml: networked science in machine learning. *SIGKDD Explor.* 15(2):49–60.

Weerts, H.; Dudík, M.; Edgar, R.; Jalali, A.; Lutz, R.; and Madaio, M. 2023. Fairlearn: Assessing and improving fairness of ai systems.