Fairness in Recruitment Decision Algorithms

Edoardo Saturno, Lorenzo Balzani, Riccardo Marvasi

Ethics in Artificial Intelligence 2023-2024

Abstract. With the increasing use of Artificial Intelligence (AI) in various industries, ensuring fairness in these systems has become a critical theme. This work investigates the biases introduced by hiring algorithms employed by large corporations to prescreen job candidates. These algorithms are increasingly used to streamline the hiring process, making it more efficient and cost-effective. However, despite their advantages, these AI systems can perpetuate and even exacerbate existing biases, leading to unfair discrimination against certain groups. Our research aims to identify specific biases within these prescreening algorithms and propose methods to mitigate them. In our study we will compare traditional ML technique together with more advanced neural network architectures, making use of the *Aequitas WP7 T7.3. Hiring Data For STEM Field. v1.0.* Our aim is to demonstrate the potential presence of unfairness both in the dataset and in the decision-making process of the algorithms, and the possible solutions that can be implemented to correct this behavior. Our findings highlight the importance of transparency, accountability, and continuous monitoring in the deployment of AI systems in hiring, ensuring that they promote equality and diversity in the workplace.

1. Introduction

In the last decade, there has been a significant increase in digitalization across many human activities. One such activity is the online submission of job applications. Large companies can receive hundreds of resumes in a short period of time for a particular job position. Human recruiters often do not have the time to screen all of them accurately and can make mistakes due to the sheer volume of resumes. One possible solution is to use decision-making algorithms to perform a preliminary screening on the bulk of resumes and retain only the most suitable ones for the job position. However, this approach can introduce bias into the evaluation process, as algorithms may incorporate discriminatory beliefs during training [1].

This report aims to analyze various traditional machine learning models and neural networks used for resume screening. The goals are to identify potential biases in each model and to evaluate their performance. In the second part of the report, we will describe the implementation of bias mitigation techniques such as *Reweighting* and *Adversarial* Debiasing [2][3].

2. Data

In this section we will introduce the Aequitas dataset, showing its structure and distribution. We will also present the pre-processing techniques we used.

2.1 Dataset

The chosen dataset is the “AEQUITAS WP7 T7.3. Hiring Data For STEM Field. v1.0” available at the following link.

This dataset contains information about the hiring process conducted by Akkodis for job positions and candidates in the STEM fields. The data have been thoroughly anonymized and include the curricula of candidates and details of the job positions to which they were matched. This dataset was assembled to allow the study of bias in the context of class-imbalanced data, which can favor specific groups based on gender, age, race, or social background, thereby discriminating against selected groups or minorities.

In our analysis, we will also focus on bias originating from the models themselves. We have identified three main reasons that can give rise to bias and discrimination in recruitment algorithms:

- **Inherently Biased Systems:** These systems are biased by design. We can identify this by comparing different models and looking for significant differences in fairness.
- **Systems Trained on Historical Human Judgments:** These systems perpetuate the unfairness and prejudice inherent in historical judgments. We can identify this type of bias by directly analyzing the dataset.
- **Systems Using Discriminatory Proxies:** These systems pursue a non-discriminatory objective using a discriminatory proxy. We can detect this bias using explainable AI frameworks like SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations).

The dataset consists of 40 columns and 21,377 entries and it have been carefully anonymized. In particular:

- The name (and surname) of each candidate has been replaced with a hash code.

- Names of previous companies where the candidate worked have been removed. This process involved the Job Description, Recruitment Request, and Last Role fields.

Following is a brief description of each field:

- **ID**: Unique identifier for the candidate. Each 'ID' is unique for a given name.
- **Candidate State**: Status of the candidate's application. In particular, there are different states:
    - Hired: The candidate has been offered and has accepted the job position.
    - Vivier: It refers to the candidate being inserted in a reserve list. Candidates in this state are considered for future opportunities.
    - QM: Stands for "Qualification Meeting". The candidate is waiting for a more technical interview.
    - In selection: The candidate is currently being considered for the job and is undergoing the selection process.
    - First contact: Initial communication has been made with the candidate, indicating they have just applied.
    - Economic proposal: An offer has been made to the candidate.
    - Imported: The candidate's data has been imported into the system from an external source or database.
- **Age Range**: Range of age for the candidate.
- **Citizenship**: Country of citizenship for the candidate.
- **Residence**: Current place of residence for the candidate.
- **Sex**: Gender identification of the candidate (Male/Female).
- **Protected category**: Indicates if the candidate falls into a protected category. There are two articles: Article 1 and Article 18.
- **TAG**: Keywords used by recruiters.
- **Study area**: Field of study or academic discipline.
- **Study Title**: Academic degree or title obtained.
- **Years Experience**: Number of years of professional experience.
- **Sector**: Industry or sector in which the candidate has experience.
- **Last Role**: Candidate's most recent job role.
- **Year of insertion**: Year when the candidate's information was entered into the portal (database).
- **Year of Recruitment**: Year in which the candidate was hired.
- **Recruitment Request**: Represents the application request for a candidacy.
- **Assumption Headquarters**: Headquarters location associated with the hiring assumption.
- **Job Family Hiring**: Job family or category for the hiring position.
- **Job Title Hiring**: Specific job title for the hiring position.

- **Event_type_val**: Specifies the stage of the recruitment process for the candidate.
- **Event_feedback**: Feedback received from an event.
- **Linked_search_key**: Indicates the number of searches conducted for a job position (e.g., RS23.0594, “594” represents the 594th search conducted for that position).
- **Overall**: Overall assessment. Interview score.
- **Job Description**: Description of the job role.
- **Candidate Profile**: Profile information for the candidate.
- **Years Experience**: Additional field for specifying years of experience.
- **Minimum Ral (Gross Annual Salary)**: Minimum expected gross annual salary.
- **Ral (Gross Annual Salary) Maximum**: Maximum expected gross annual salary.
- **Study Level**: Level of study.
- **Study Area**: Additional field for specifying the academic field of study.
- **Akkodis headquarters**: Headquarters location for Akkodis.
- **Current Ral (Gross Annual Salary)**: Current or existing salary.
- **Expected Ral (Gross Annual Salary)**: Expected salary.
- **Technical Skills**: Skills related to technical or specialized expertise.
- **Standing/Position**: Standing or position within the organization.
- **Communication**: Communication skills rated from 1 to 4.
- **Maturity**: Level of maturity rated from 1 to 4.
- **Dynamism**: Level of dynamism rated from 1 to 4.
- **Mobility**: Mobility rated from 1 to 4.
- **English**: Proficiency in the English language rated from 1 to 4.

2.2 Hiring Process

The selection process at *Akkodis* starts either when a client specifies a need or when a Business Manager identifies an opportunity. The process begins with analyzing the required professional roles. If the roles are not available internally, the search phase uses major search engines, professional social networks, the *Akkodis* website, and internal referrals to find candidates.

Initial telephone interviews are conducted to confirm candidates' availability, followed by detailed HR interviews to assess their background and aspirations. Candidates then undergo a technical interview by an internal expert or a qualified resource at the client's site. Successful candidates may then have a Qualification Meeting (QM) at the client's site, which serves as a second technical evaluation by the client, though sometimes only one of these interviews is conducted.

The client ultimately decides on the suitability of the consultant. In turnkey projects, only the technical interview is conducted, skipping the QM. After these evaluations, the onboarding phase begins, involving negotiation and administrative procedures to prepare the new hire's entry into the company.

2.3 Preliminary Inspection and Preprocessing

- **Download and Analyze the Dataset**: Begin by obtaining the dataset and conducting an initial analysis to understand its structure, distribution, and the presence of any anomalies or outliers. We check the distribution of sensitive features namely: Sex, Citizenship, Age and Protected Category. The results can be seen in Figure 1, 2, 3 and 4.
- **Drop Unnecessary Columns**: Identify and remove columns that do not contribute meaningful information to the analysis of our specific goal, ensuring the dataset is streamlined for subsequent processing.
- **Handle Missing Values**: For each column with missing values (*NaNs*), apply appropriate techniques to fill these gaps. In particular, for numeric columns we substituted with the average of the column, for categorical ones with a specific string.
- **Reduce Label Variability and Encode Features**: For categorical columns with a high number of unique values, we reduced label variability by consolidating infrequent categories. For example, we aggregated citizenships into “European” and “Non-European,” and grouped ages into “Young” and “Senior.” This approach helps avoid situations with many infrequent features, making it easier to study potential fairness and discrimination. We also encoded all categorical features into numerical formats using techniques as one-hot encoding and label encoding.
- **Encode Target Column**: We created the target column artificially. We first remove rows with Status label equal to “*In selection”, “First Contact” and “Imported*” because it was not clear if the candidates were a good fit for recruiters. Secondly, we encoded as “*Hired*” the candidates that were hired and those who received an economic proposal. All other candidates were considered as “*Not* Hired”. We remind you to section 2.1 for a deeper explanation of the feature. After this processing the distribution of labels for the target column can be observed in Figure 5.

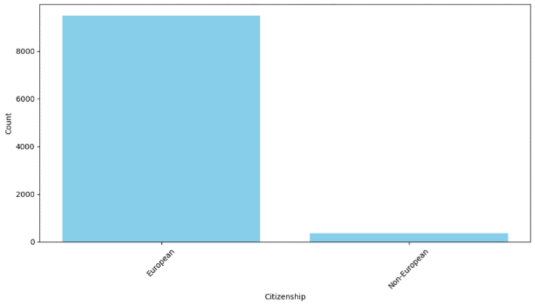

Figure 3 - Distribution of Citizenship

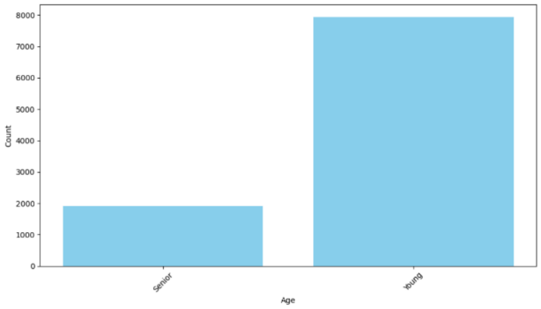

Figure 2 - Distribution of Age

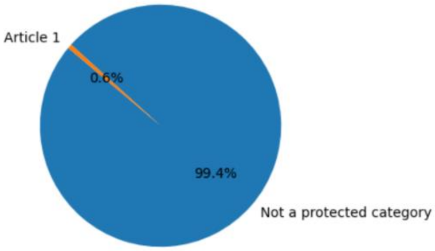

Figure 1 - Distribution of "Protected Category"

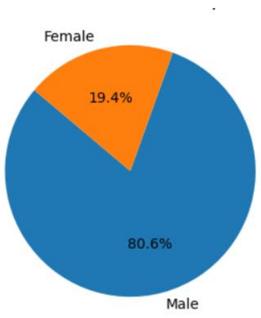

Figure 4 - Distribution of Gender

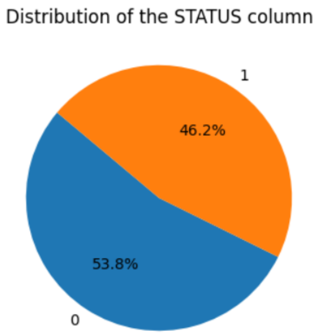

Figure 5 - Distribution of the target column

3 Models and Hyperparameters

In this section, we will present the various types of models we used for our work and the specific choice of hyperparameters. We distinguish three sets of models, depending on the specific goal of the evaluation.

The first set of models comprises: Decision Tree, XGBoost, Linear Regressor, Naïve Bayes, K-Nearest Neighbors (KNN), and Neural Networks. For the NN model, we used seven different seeds to initialize seven distinct neural networks, ensuring the robustness of our results. The neural network structure consists of five fully connected layers, with the final layer comprising a single neuron with a sigmoid activation function to output prediction probabilities. We consistently used a batch size of 64 and trained for 15 epochs for all networks, utilizing the Adam optimizer to automatically manage the learning rate. This first set of models serves to define a baseline, collecting the results of the model on the dataset without any further adjustment.

The second set of models consists of the same ones as the first but trained on a different training set, which we will describe in the reweighting section. This second set allowed us to perform a comparison after the reweighting, in order to see the changes in terms of classical metrics and fairness techniques.

The third set of models includes only neural networks. Specifically, this set comprises the original seven neural networks plus four smaller networks, referred to as adversary nets. These adversary nets consist of four layers, with the final layer containing a single neuron. This time both the main NN and the adversary nets use a "*ReLU"* activation function in the last layer instead of a sigmoid because we require logits, not predictions.

4 Evaluation

In this section we will present the metrics with which we evaluated the models. In the following subsections we will give further insights on the explainable AI frameworks we used and on the fairness metrics we implemented.

For each training scenario, we evaluated the corresponding set of models using three distinct methods:

1. **Standard Metrics**: We assessed precision, recall, ROC AUC score, F1 score, and accuracy to provide a quantitative comparison between models.
2. **Graphical Explainability Methods**: We utilized LIME and SHAP to examine the inner workings of each model, specifically to determine whether predictions were influenced by sensitive attributes [4][5].
3. **Customized Fairness Metrics**: We employed *Demographic Parity* and *Equalized Odds*, which yielded Boolean values for each sensitive feature across models. This allowed for a deeper introspection into the bias present in each model [6][7].

4.1 LIME and SHAP

These methods approach the problem differently:

- **LIME (Local Interpretable Model-agnostic Explanations)** modifies a single data sample by tweaking the feature values and observing the resulting impact on the output. These perturbations aim to understand the hyperspace configurations in the vicinity of the data point. The perturbed points are then used to train a local surrogate model that is inherently interpretable.
- **SHAP (SHapley Additive exPlanations)** is based on the approximate computation of Shapley values, a metric from game theory used to evaluate the contributions of each player. In the context of machine learning, each feature is treated as a player, and SHAP values are computed by averaging the marginal contributions of each feature across all possible permutations. We have included example visuals to illustrate these methods.

LIME can be used for analyzing single instances, while SHAP can be applied to both multiple and single instances. In our study, we used LIME only in the initial "Naïve" training phase of the models to gain a complementary understanding of feature importance. In all other cases, we used SHAP because it allows us to focus on all data points simultaneously, examining the distribution of feature importance across each example.

4.2 Customized Fairness Metrics

We used two different fairness metrics implemented by us for detecting Bias. Namely they were:

- **Demographic parity:** Demographic parity is a principle in machine learning aimed at ensuring that predictive models do not discriminate against individuals based on their membership in certain demographic groups, such as race, gender, or age. The goal is to achieve equal treatment across different demographic categories, thus promoting fairness and reducing bias in decision-making processes. Demographic parity requires that the proportion of positive outcomes predicted by the model should be equal across all demographic groups. In other words, if there are two groups (e.g., males and females), demographic parity means that the model should predict positive outcomes (such as loan approvals or job offers) for each group at the same rate, regardless of their demographic attributes. In our implementation a difference higher than 15% lead to trigger a False statement for the current model on the current sensitive attribute.

- **Equalized odds:** Equalized Odds is a fairness criterion in machine learning designed to ensure that predictive models make decisions fairly across different demographic groups. It focuses on minimizing disparate impact by requiring that model predictions are equally accurate regardless of sensitive attributes like race or gender. This principle aims to uphold fairness by balancing predictive performance across all groups without favoring or disadvantaging any demographic. Equalized Odds requires that for each sensitive attribute category (e.g., male and female), the model's predictions should have equal rates of true positives, false positives, true negatives, and false negatives. In simpler terms, the model should make equally accurate predictions for each group, ensuring that errors (both false positives and false negatives) are distributed evenly among all demographic categories. In our implementation a difference higher than 30% on TPR or FPR lead to trigger a False statement for the current model on the current sensitive attribute.

5 Results

In this section we will discuss the results of the three sets of models.

5.1 Standard Training

We begin by discussing the first strategy, which employs both classical models and neural networks. Specifically, we consider 7 differently initialized neural networks. We observed that:

- Firstly, after a certain number of epochs, neural networks (NN) demonstrate relatively good performance compared to classical models. This suggests that NN models have the potential to perform as well as traditional models and can be effectively utilized for similar tasks however this at cost of a higher inference time. More than that we have that there is no significant difference in Bias from the model construction themselves, so regarding the ones we used we have not noticed some unfairness due to construction between them. Among the models evaluated, only XGBoost, k-NN, and decision trees consistently outperformed the NNs as we can see from the result of Table 1.

Table 1 – Models Performance

Model	Accuracy	Precision	Recall	F1-score	ROC AUC
Linear Regression	0.727	0.913	0.470	0.620	0.715
Decision Tree	0.782	0.783	0.748	0.765	0.780
Naïve Bayes	0.779	1.000	0.534	0.696	0.767
XGBoost	0.816	0.900	0.690	0.781	0.810
kNN	0.789	0.820	0.711	0.762	0.785
NN (average)	0.767	0.804	0.679	0.735	0.763

- Secondly, the performance of all models regarding fairness metrics is notably poor. In almost every case for both demographic parity (DE) and equalized odds (EO), the differences in distribution between groups are significant. This indicates a substantial problem linked to bias in the distribution of positive outcomes across different groups. We can see results in Table 2a and 2b.

Table 2.a – Demographic Parity

	Sex	Age	Citizenship	Prot_Cat
LR	F	F	T	T
DT	T	T	T	T
NB	F	T	T	F
XGB	F	T	T	F
kNN	T	T	T	T
NN1	F	F	T	T
NN2	F	F	T	F
NN3	F	T	T	T
NN4	T	T	T	F
NN5	F	F	T	T
NN6	F	T	T	F
NN7	T	F	T	T

Table 2.b – Equalized Odds

	Sex	**Age**	**Citizenship**	**Prot_Cat**
LR	F	F	F	T
DT	F	F	T	T
NB	F	F	F	F
XGB	F	F	T	F
kNN	F	F	T	T
NN1	F	T	T	F
NN2	F	F	T	F
NN3	F	F	T	F
NN4	F	F	T	F
NN5	F	T	T	T
NN6	F	F	T	F
NN7	T	F	T	F

- Another possible type of bias, as we explained in the introduction, is not related to distribution but rather to the reliance on specific features for making predictions as we can see in Figure 6. While output distribution differences indicate explicit bias, reliance on features indicates implicit bias. We investigated this using feature importance measures provided by SHAP and LIME. SHAP summary plots reveal that the importance of the four sensitive features (sex, age, citizenship, and protected category) is quite low. This is positive, as models should not base their predictions heavily on these sensitive attributes. This desirable behavior however is stronger in neural networks while other models, like Linear Regressor use sensitive feature as stronger discriminant as we can see from the fact that age and sex are of higher feature importance.

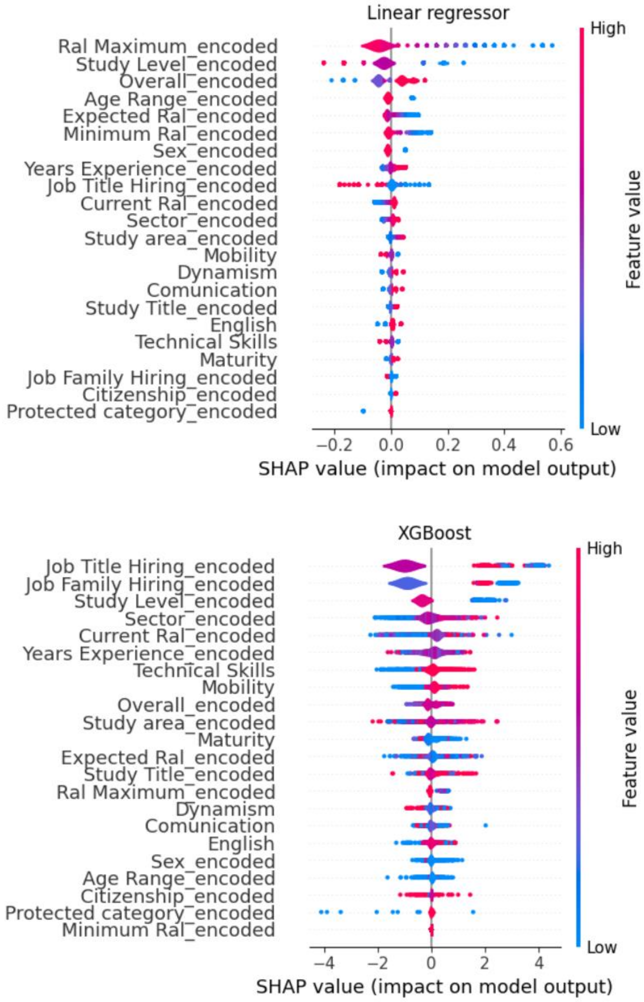

Figure 6 – SHAP values on Linear Regressor

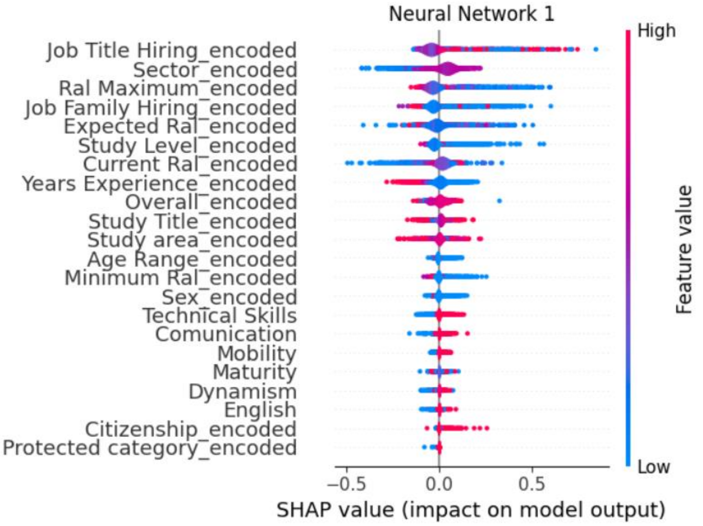

figure 7 – SHAP values on NN1

Overall, the first approach demonstrates that neural networks are a viable option for modeling tasks and that classical ML algorithms are a suitable option as well. These models do not heavily rely on sensitive features for their predictions. However, they fail to ensure similar distributions among groups, indicating that bias arises from other sources. One potential source is the skewed distribution of data in the dataset, as we have seen in section 2.3.

5.2 Reweighting

In machine learning, reweighting techniques are used to mitigate bias in predictive models, ensuring fairer and more equitable outcomes. Bias can arise when certain groups in a dataset are underrepresented or overrepresented, leading to skewed predictions that favor or disadvantage specific groups. Reweighting involves assigning a specific weight to each element belonging to a particular “sensitive group.” For example, in our study, the sensitive features include categories such as sex, age, citizenship, and protected status, resulting in groups like “male-young-European-non protected,” “female-young-European-non protected,” and so on. Each group receives a weight based on the inverse of its frequency, and during the reweighting process, elements belonging to minority groups are duplicated. This approach helps transform a dataset from one where some groups are minimally represented to a more balanced situation.

We explored various types of reweighting techniques. Initially, we increased the number of underrepresented groups to equalize them with the more frequent ones. This method allowed us to retain all elements of the original dataset. However, due to the dataset's size, the large number of groups, and the extreme underrepresentation of some groups, this approach led to an excessively large, reweighted dataset. Consequently, we adopted a reweighting strategy involving replication and substitution: we reduced the number of elements from more frequent groups and duplicated elements from underrepresented groups. This approach achieved a more balanced dataset without excessively increasing its size. To better understand the effects of the reweighting on the dataset, we invite you to refer to Table 3 and Table 4.

Table 3: Group distribution before reweighting

Sex	Age	Citizenship	Protected Category	Number of Elements	Percentage
Male	Young	European	No	**5980**	**60.7%**
Male	Senior	European	No	**1612**	**16.4%**
Female	Young	European	No	**1597**	**16.2%**
Male	Young	Non-Euro	No	**292**	**3.0%**
Female	Senior	European	No	**247**	**2.5%**
Female	Young	Non-Euro	No	**35**	**<1%**
Male	Senior	Non-Euro	No	**27**	**<1%**
Male	Young	European	Yes	**19**	**<1%**
Male	Senior	European	Yes	**16**	**<1%**
Female	Young	European	Yes	**14**	**<1%**
Female	Senior	European	Yes	**9**	**<1%**
Female	Senior	Non-Euro	No	**9**	**<1%**

Table 4 - Distribution of groups after reweighting

Sex	Age	Citizenship	Protected Category	Number of Elements	
Female	Young	European	Yes	**844**	**8.6%**
Male	Senior	European	No	**844**	**8.6%**
Female	Senior	European	Yes	**840**	**8.5%**
Female	Senior	European	No	**833**	**8.5%**
Female	Young	Non-Euro	No	**833**	**8.5%**
Male	Young	Non-Euro	No	**823**	**8.3%**
Female	Senior	Non-Euro	No	**821**	**8.3%**
Female	Young	European	No	**820**	**8.3%**
Male	Senior	Non-Euro	No	**819**	**8.3%**
Male	Senior	European	Yes	**806**	**8.2%**

Male	Young	European	Yes	**787**	**8.0%**
Male	Young	European	No	**787**	**8.0%**

The primary advantage of this reweighting process is its simplicity and speed, which allows for straightforward correction of dataset inequalities. However, its simplicity also means it is not effective in cases where a group is advantaged or disadvantaged not only by its size but also by its hiring percentage. For example, if "male-European-young-non protected" individuals have a hiring rate of 65% in the original dataset—significantly higher than the average—reducing their sample size from 5980 to 787 does not change this percentage due to random replacement. Consequently, models trained on the re-sampled dataset will still learn that 65% of "male-European-young-non protected" individuals are hired, perpetuating their advantage over other groups. This method fails to address intrinsic discrimination present in the dataset, stemming from biases of the human recruiters who generated the data.

However, reweighting is effective when addressing disparities in group sizes, particularly when two groups have similar hiring rates but vastly different representation. For instance, if there are 800 male applicants and 200 female applicants with similar hiring rates, the model might learn a bias simply because the number of hired males exceeds the number of hired females. In such cases, reweighting can be very effective in balancing the dataset and mitigating bias.

All the machine learning models, and the neural network were trained on this reweighted dataset and then tested on the same test set used previously. The results in terms of classical metrics are showed in table 5. It is possible to assess that they remained pretty much untouched.

Table 5 - Metrics of the models after reweighting

Model	Accuracy	Precision	Recall	F1-score	ROC AUC
Linear Regression	0.678	0.762	0.468	0.580	0.668
Decision Tree	0.809	0.812	0.778	0.795	0.808
Naïve Bayes	0.779	1.000	0.534	0.696	0.767
XGBoost	0.809	0.844	0.734	0.785	0.806
kNN	0.759	0.783	0.681	0.728	0.755
NN (average)	0.751	0.768	0.699	0.729	0.748

These results are encouraging. If the models perform similarly on the reweighted dataset as they did on the original dataset, it suggests that the sensitive features were not crucial for the final prediction.

Anyway, by looking at the results of the fairness techniques showed in table 6 and table 7, it is possible to see that reweighting the dataset did not solve the disparities in hiring percentages within classes (in the parenthesis the results before reweighting).

Table 6 - Democratic Parity

	Sex	**Age**	**Citizenship**	**Prot_Cat**
LR	False	True (F)	False (T)	False (T)
DT	True	True	True	True (F)
NB	False	True	True	False
XGB	True (F)	False (T)	True	True (F)
kNN	False	False (T)	True	True
NN1	False	True (F)	True	True (F)
NN2	False	True	True	True
NN3	True (F)	True (F)	True	True (F)
NN4	False	True	True	True (F)
NN5	False (T)	True	True	True (F)
NN6	False (T)	True (F)	True	False (T)
NN7	False (T)	True	True	True

Table 7 - Equalized Odds

	Sex	**Age**	**Citizenship**	**Prot_Cat**
LR	False	True	False	False (T)
DT	False	True	False (T)	False
NB	False	True	False	False
XGB	False	False (T)	False (T)	False
kNN	False	False (T)	False	False (T)
NN1	False	True	True	False
NN2	False	True	True (F)	False
NN3	True (F)	True	False (T)	False
NN4	False	True	True	False
NN5	False	True	True	False
NN6	False	True	True	False (T)
NN7	False (T)	True	False (T)	False

Results show some inconsistencies. The differences in results between the neural networks are also influenced by the fact that they were redefined before re-training, and as we know, different random seeds can lead to significantly different outcomes. Therefore, to evaluate the performance of reweighting for the neural networks, we need to consider tendencies. Specifically, we observed 17 "True" results for democratic parity before reweighting and 21 after, while there were 15 "True" results for equalized odds before reweighting and 13 after. Even for the machine learning models, reweighting does not emerge as a clear solution for addressing discrimination.

For some specific categories, like sex performances are generally bad: reweighting fails to eliminate the unfairness in the groups. To understand these results better, it's important to revisit the principles of Equalized Odds and Demographic Parity. Reweighting does not fundamentally alter the percentages of positives within groups, so it cannot be deemed effective by techniques that rely on these percentages within each group. Specifically, in the dataset, the percentages of hires within sex category are shown in Table 8.

Table 8

Sex	% of Hired
Male	44.3%
Female	54.3%

This misalignment between the percentages persists even in the reweighted dataset. As a result, plain models trained on such a dataset will likely maintain similar percentages in their predictions, leading to "False" outputs for both fairness techniques.

We also observed SHAP plots after reweighting (Figure 8.a and 8.b) and noticed no significant differences with respect to the standard training. These again can be explained by the fact that reweighting does not alter the feature importance for the models.

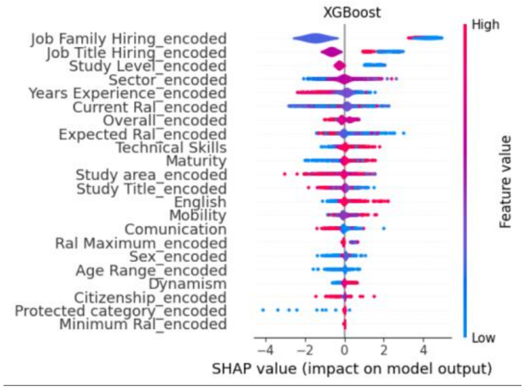

Figure 8.a

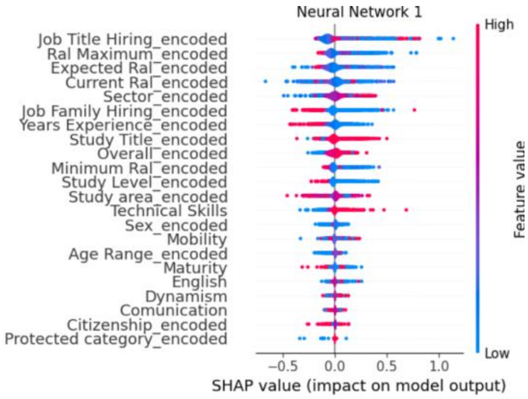

figure 8.b

5.3 Adversarial Debiasing

Adversarial debiasing is an advanced technique in machine learning designed to reduce bias in predictive models, ensuring that their decisions are fair and not influenced by sensitive attributes such as race, gender, or age. This method borrows the concept of adversarial training from generative adversarial networks (GANs) to create a dual-training framework that promotes fairness while maintaining model accuracy. In adversarial debiasing, there are two key components: the predictor model and the adversary model. The predictor model is the primary model that aims to make accurate predictions based on input data. The adversary model, on the other hand, is designed to predict the sensitive attribute from the output of the predictor model. Its purpose is to identify any biases the predictor model might have.

The training process involves three main steps:

1. **Initial Training**: First, the predictor model is trained on the input data to make accurate predictions for the main task. At this stage, the model learns patterns and relationships within the data without considering biases.
2. **Adversarial Training**: Simultaneously, the adversary model is trained to predict the sensitive attribute from the predictor's output. This model's goal is to be as effective as possible in identifying the sensitive attribute, revealing any biases present in the predictor model.
3. **Objective Adjustment**: The predictor model's training objective is then adjusted to minimize the adversary's ability to accurately predict the sensitive attribute. This is achieved by incorporating an additional term in the loss function of the predictor model that penalizes it if the adversary successfully predicts the sensitive attribute. Essentially, the predictor is encouraged to make predictions that are independent of the sensitive attribute, thereby reducing bias. The combined loss is obtained as the difference of the main neural network loss (main loss) and the adversary loss multiplied for a coefficient α which manage the strength of debiasing.

The primary goals of adversarial debiasing are twofold: to maintain high predictive performance on the main task and to ensure that the model's predictions are not unfairly influenced by sensitive attributes. This method aims to achieve a balance between accuracy and fairness, producing models that make equitable decisions.

Theoretically, we could consider debiasing multiple sensitive features during training by adding a series of loss terms. However, this approach yielded poor results because the models could not distinguish which sensitive feature debiasing was improving, leading to worse performance in both quantitative metrics and unfairness metrics. Instead, we chose to enable the model to debias one feature at a time by adjusting a parameter that determines which sensitive feature will be debiased. We then analyzed the effect of debiasing for each feature independently. From the results obtained we can observe that:

- The performance is slightly lower compared to the first approach. This decrease is due to the optimization being guided towards equalizing prediction percentages between groups, rather than solely learning from the distribution in the dataset. Unlike the reweighting approach, this method necessitates a trade-off between performance and the equalization of predictions.

Table 9 Adversarial Performance mean across the 4 feature debiasing

Model	Accuracy	Precision	Recall	F1-score	ROC AUC
NN1	0.770	0.965	0.539	0.690	0.759
NN2	0.773	0.963	0.546	0.695	0.761
NN3	0.778	1.000	0.533	0.695	0.766
NN4	0.776	0.979	0.540	0. 696	0.764
NN5	0.760	0.925	0.565	0.692	0.750
NN6	0.779	1.000	0.534	0.696	0.767
NN7	0.779	0.999	0.535	0.697	0.767

- Regarding fairness metrics, we observe significant improvement for most of them, specifically for Sex and Age. However, there is no improvement for the Protected category. This discrepancy is due to the nature of adversarial debiasing, which aims to reduce the feature importance of sensitive attributes. For Sex and Age, SHAP plots indicate that, while these features are not the top correlated, they are still relevant. Thus, their influence can be effectively reduced through debiasing. Conversely, for the Protected category, SHAP plots show it is the least correlated feature with the models. Therefore, debiasing is ineffective in this case because the feature is already uncorrelated with the model's predictions, leaving nothing to be debiased.

Table 10 Demographic Parity (Sex Debiased with α = 0.4)

	Sex	Age	Citizenship	Prot_Cat
NN1	T	T	T	T
NN2	T	T	T	F
NN3	T	T	T	F

NN4	T	T	T	F
NN5	T	T	T	F
NN6	T	T	T	F
NN7	F	T	T	F

Table 11 Demographic Parity (Age Debiased with α = 0.4)

	Sex	**Age**	**Citizenship**	**Prot_Cat**
NN1	T	T	T	False
NN2	T	T	T	False
NN3	T	T	T	False
NN4	T	T	T	T
NN5	False	T	T	T
NN6	T	T	T	False
NN7	T	T	T	False

Table 12 Demographic Parity (Citizenship Debiased with α = 0.4)

	Sex	**Age**	**Citizenship**	**Prot_Cat**
NN1	T	T	T	False
NN2	T	T	T	False
NN3	T	T	T	False
NN4	T	T	T	False
NN5	T	T	T	False
NN6	T	T	T	False
NN7	T	T	T	False

Table 13 Demographic Parity (Protected Category Debiased with α = 0.4)

	Sex	**Age**	**Citizenship**	**Prot_Cat**
NN1	False	T	T	False
NN2	T	T	T	False
NN3	T	T	T	False
NN4	T	T	T	False
NN5	T	T	T	False
NN6	T	T	T	False
NN7	T	T	T	False

- Analysis using SHAP graphs shows similar results to the standard training. The importance of sensitive features remains low, indicating that the models do not heavily rely on these features for their predictions.

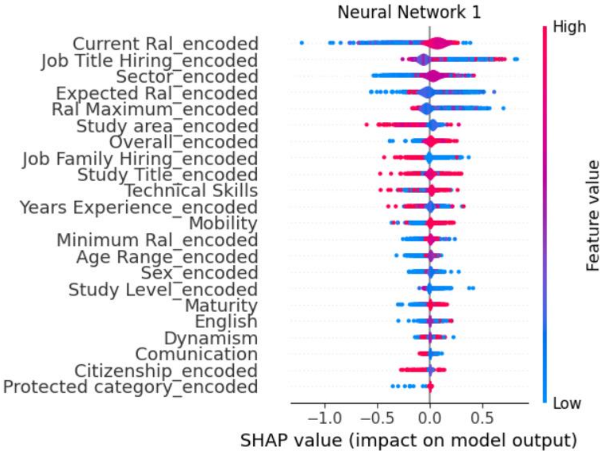

Figure 9

6 Conclusions

Regarding the first type of bias, related to model construction, we observed that models do not differ significantly. Thus, using one model is equivalent to using another, and the only discriminant in this case would be choosing the best-performing one.

The second type of bias, stemming from imbalanced dataset distributions, strongly influences all the models. This is evident from the fairness metrics, which show a significant difference in positive predictions between groups. To counteract this, we employed two debiasing techniques: reweighting and adversarial debiasing. The reweighting approach did not achieve significant progress because it operates at the frequency level rather than adjusting the positive prediction distribution between groups. Conversely, adversarial debiasing, though more complex and only applicable to neural networks, yielded better results, but only for sensitive features correlated with the output.

The third type of bias, related to the use of sensitive features for making predictions, was quite low for the best-performing models, such as neural networks, where sensitive features had very low feature importance. However, for other models, like linear regressor some sensitive feature could be deemed more important by the model for the predictions.

7 References

1. link
2. https://machinelearning.apple.com/research/learning-to-reweight-data
3. https://machinelearning.apple.com/research/learning-to-reweight-data
4. https://shap.readthedocs.io/en/latest/
5. https://c3.ai/glossary/data-science/lime-local-interpretable-model-agnostic-explanations/
6. https://en.wikipedia.org/wiki/Equalized_odds
7. https://fairlearn.org/main/user_guide/assessment/common_fairness_metrics.html