Simulating Bias for Mitigation Models and Fairness

1st Ricardo Inácio
Faculdade de Engenharia
Universidade do Porto
Porto, Portugal
up202302742@up.pt

2nd Tomás Maciel
Faculdade de Engenharia
Universidade do Porto
Porto, Portugal
up202006845@up.pt

3rd Carlota Silva
Faculdade de Engenharia
Universidade do Porto
Porto, Portugal
up201908057@up.pt

***Abstract*—Predictive models often showcase and reinforce biases present in the data they were trained on, providing unfair assessments and prejudicial decisions, which could lead to real world consequences, further harming the already disadvantaged groups. In such cases, mitigation methods are critical to ensure that, regardless of the perceived prejudice, model outcomes are corrected to be fair and balanced. By simulating an environment that depicts socio-economical activities within a biased society, considering two groups: privileged (*A*) and underprivileged (*B*), where agents work, carry out transactions, and procreate, we can then generate synthetic data that conveys such inherent disparities. Afterwards, a set of classifiers is trained on such data, to ultimately evaluate through appropriate metrics that reveal how the outcomes regarding each group, are impacted by skewed estimates, given that aggregate metrics fall behind in biased environments. We demonstrate how biased data contributes to issues such as disparate misclassification rates and unequal opportunity. We also employ fairness intervention methods of variable complexity, such as cost-sensitive class weighting, and Exponentiated Gradient mitigation using a Demographic Parity constraint, in an effort to mitigate the related problems. We aim to obtain classifiers that, even if less accurate, ascribe positive predictions to entities regardless of group belonging. Results show that, although it is possible to attenuate the impacts of bias, there is a trade-off on the scope of the fairness methods, effectiveness of the mitigation on each group, and overall performance.**

Keywords: modelling, simulation, agent-based systems, bias.

I. Introduction

The use of machine learning approaches is currently crucial in several domains, such as medical [1], or financial [2], to ensure a proper data-driven decision-making process. By leveraging the patterns present in data, such methods are able to characterise novel instances, leading to data-driven decisions that can be validated with greater certainty. Classifiers, such as ensemble methods [3], are particularly beneficial in this regard, being able to harness past observations in order to distinguish new instances with high reliability, in diverse applications such as fraud detection or automatic candidate selection, leveraging predictions from multiple simple methods.

Nonetheless, the performance of such models can be hampered by several factors, mainly related with the data used in training. In binary classification tasks, if the target distribution is skewed, predictions regarding the minority class might not be as reliable as the ones for the majority [4]. Also, if societal bias are concealed in the data, not only will they be apparent in the predictions, but also further amplified [5].

To replicate the preceding phenomena, we developed two *Agent-Based Models* (ABM) using the `MESA Python` framework [6], and the `Netlogo` modelling environment [7]. Both models capture the socio-economical discrepancies between two groups, regarding wealth-acquisition perspectives, and how a system can be affected by those disparities. Both the privileged (*A*) and unprivileged (*B*) groups partake in the same set of activities, while having different probabilities associated to the occurrence of events. We showcase different scenarios by manipulating variables and attributes throughout the simulations. We generate and collect the consequent data to train a set of three classifiers. The goal is to demonstrate how biased data exacerbates issues such as disparate misclassification rates and unequal opportunity, by analysing model performance. Finally, fairness intervention methods are used in an effort to mitigate the problems that arise with the application of biased data. Different approaches of varying complexity are implemented, such as a simpler cost-sensitive class weighting, in which different weights are specified for members of different classes, as a way to push the focus of the classifier into a particular one, and Exponentiated Gradient mitigation using a Demographic Parity constraint, via the `fairlearn` [8] `Python` package, to assess the trade-off in fairness, bias mitigation, and predictive performance. In the evaluation process, the significance of successfully mitigating unjust classifications across groups is assessed, by verifying if the selection rate, which measures the probability instances have to be assigned the positive class, is balanced across groups, and within reasonable limits.

II. Literature Review

This section outlines the background underlying the present work, regarding agent-based models, and how they have been applied across the literature, in conjunction with machine learning methods. Also, we examine how bias can be embedded into datasets, and how it affects predictions of machine learning models. Finally, a review is conducted on how the effects of bias can be mitigated.

A. Agent-based Models

Agent-based models (ABS) have been used for several decades to clearly represent and interpret complex interactions in environments composed of multiple entities, allowing for a more nuanced analysis of the structures and dynamics it

simulates [9]. Within them, model variables allow for the specification of concrete interaction scenarios, that can be explored to understand systemic activities. Thus, behaviours can be interpreted from generalised patterns [10].

The fundamental element in such systems is the agent. These contain a set of attributes, whose values determine the individual characteristics of each. Such qualities can be equal amongst all, related to a specific group, or unique. Some attributes may undergo shifts as the scenarios unfold, while others remain constant. Agents also embody a set of behaviours, that define the interactions they engage in, with other agents, or the environment. At any given point, the set of specific attribute values an entity bears defines the momentary state [10], which can more accurately describe the entity.

This work focuses on models of structural differentiation, which illustrate how individuals amid an environment engage in basic interactions with each other, forming dense networks, resulting in emergent patterns that reveal social phenomena [11]. These are ideal to showcase disparities within societies and cultures.

Research on ABS has been conducted for several decades, with its origins often attributed to the works of John von Neumann and Stanislaw Ulam, in *cellular automata* and self-replicating systems. In these, the notion of having individual entities arranged in a grid-like matrix, while engaging in local interactions was introduced [12]. This led to their use in interpreting complex observed patterns. In the context of social sciences, where such methods can be used to explain emergent social phenomena, the foundational work by Schelling, regarding a shared-space segregation model, serves as the basis for the conceptual framework behind this research. In it, moving agents that showcase a slight bias towards others of the same coup eventually lead to segregated neighbourhoods [13]. The possibility to demonstrate societal disparities based on biased events, is tightly related with the scope of this work, of showcasing how unequal wealth acquisition leads to unfair representations on generated data.

B. Machine Learning for Agent-based Models

Despite the fact that ABS are composed by simple entities that perform basic actions, the sum of all interactions may lead to complex emergent phenomena that were not expressed individually [14]. Given this unbounded complexity, it can be challenging to interpret the results from such simulations, via direct analysis. Thus, machine learning (ML) techniques can be employed to obtain a better grasp of these behaviours, by analysing patterns found in the learned features. In [15], the authors build surrogate models, which try to approximate the input-output causality of the complex system, to thoroughly understand it. However, they faced difficulties in estimating the parameter space of the ABS, given its intricate structure.

Accordingly, to potentially ease the learning task, we collect the outcomes of the executed simulations, and use the resulting dataset to train a predictive model. The method classifies each entity into one of the two classes (privileged or not), based on the learned features, and the group-wise performance provides insights concerning bias in the system.

C. Bias in Machine Learning

Predictions made by ML models are crucial in the current landscape of various domains regarding informed decision-making. Leveraging past data, these algorithms are able to capture relevant patterns to perform forecasts with significant certainty. However, although global performance metrics may indicate an overall strong effectiveness, when evaluated at the group level, a skew towards one may emerge. Given the advent of big data, biases present in societal life, will inevitably end up embedded in the harvested data. Consequently, models trained on said data, may reinforce the encoded prejudices, which in turn foster avenues to new ones [16].

Bias can be expressed in several manners in the data. Mainly, it arises via sensitive features, which are characteristics of individuals that may strongly indicate belonging to groups prone to prejudices (e.g., sex, ethnicity, background, religion, or sexuality). Sometimes, such features might not be directly present, but inferred correlations may serve as proxies for discrimination [17] (e.g., address or income). It can also be the case where the collected data is not well-representative of all groups, leading to poor predictive performance on the minorities. This leads to various categories of unfairness, such as selection, reporting, and detection bias [16], which also further aggravate the previously mentioned prejudices.

D. Fairness and Bias Mitigation

Given the aforementioned problems, several methods have been developed to mitigate the effects of bias in the predictions of ML models. In this work, we focus on binary classification tasks. Mitigation methods can be grouped based on the step of the ML pipeline the processing is conducted. Pre-processing methods, which we employ in this paper, try to balance the distribution of the data before the training phase, thus being a model-agnostic approach. One way is by resampling the data, either by augmenting the minority class, reducing the majority, or both. Also, assigning different weights to instances, based on class membership is another agnostic method, which benefits from not losing information compared with certain resampling approaches. Although straightforward, these methods lack on controllability regarding the mitigation outcomes [16]. As for evaluation, some metrics have been particularly designed to appraise bias, such as the equal opportunity difference (EOD), which measures the difference between the largest and smallest group-level true positive rate, across all sensitive features [18].

E. Cost-Sensitive Learning

In several classification tasks, the cost of misclassifying instances of one class can be far greater than the cost of doing it for the remaining. Examples of such are fairly common in the healthcare domain, where failing to identify a potentially fatal disease could be catastrophic (false negative), whereas falsely identifying it in a healthy patient (false positive), although

distressing, would not result in a potentially avoidable loss of a life [19]. There are different schemes applicable in this modality, based on the concept of cost-sensitive learning [20]. Practically, the underlying loss function is altered so that the weights are incorporated, in a way that higher relevance is assigned to specific instances, thus amplifying the effects of misclassification [21], or reducing the effects of under-representation, relative to frequency. This can be done either statically, in which weights are fixed during training, at the instance or class levels, or dynamically, based on performance estimations or changes in data distribution.

III. Methodology

In this section we lay out the core elements of this investigation, describing the entities that participate in the simulations, alongside their common and class-specific attributes. Also detailed are the model variables, that specify which scenario we are interested in simulating at a given time. Furthermore, we specify which tools we use, and how we employ them to simulate bias in the exchanges within a society. In the end, we evaluate the developed models, and employ mitigation techniques to attenuate the effects of bias.

A. Project Goals

Our main goal is to model economic disparities, simulating the dynamics of wealth accumulation in a society with sustained bias. We aim to understand and evaluate bias propagation, specifically the process by which initial disparities in wealth and several affected factors persist over time, resulting in ongoing financial discrepancies, which are then reflected in collected data. Also, we seek to showcase that training classifiers on this data, results in skewed predictions and unfair assessments, which could further affect the already prejudiced class [22], even if social disparities get eventually rectified. We expect that although global metrics may showcase strong performance, group-level evaluation may show signs of unevenness. We employ different classification methods, namely `LightGBM` [23], `XGBoost` [24], and `RandomForest` [25], to try to predict the class of individuals based on different attributes related with opportunities in wealth accumulation. In other words, we show that, although model training is properly conducted, the predictions still favour the advantaged class. We then explore different fairness metrics that are prevalent in the literature, such as the *Equal Opportunity Difference* [26], to evaluate the level of bias in predictions, and to identify potential solutions to mitigate the perpetuation of harm. We aim, in the end, to employ several methods of varying complexity, in order to balance the *selection rate* across groups, which indicates the proportion of elements from each class that are able to be fairly (positively) classified.

B. Entities and Attributes

It was defined that our simulated society would be constituted of two classes of entities: the class of privileged individuals, denoted by `A`, and the class of unprivileged individuals, denoted by `B`. In each class, the members are endowed with a wide range of *attributes*, which were specifically chosen to increase the level of detail and complexity of the simulation, while also defining the data features to be generated, so that we can train a set of classifiers.

Some attributes are intrinsic to the entities themselves, and independent of class belonging, such as sex or age. The values of such attributes can be updated at each step, or at specific events, to accurately depict the evolution of the dynamics within the modelled society. Others, however, are dependent on the class of the entity, and it is mainly through these differences that we can model societal bias. One such example is the "*wealth accumulation rate*", which represents the maximum quantity of wealth each entity can receive at each step. At the start, and by default, the value of this attribute for class *A* is two times the value defined for class *B*. It should be noted that, regardless of class, female entities (*sex* = *F*) always receive $2/3$ of their male (sex = M) counterparts, in order to also depict intra-class discrepancies, which are usually missed by standard training procedures [27]. Another example is the "*opportunities*" attribute, representing access to favourable resources and conditions that accelerate wealth accumulation, which boosts it by $1.5\times$. There are other class-based attributes present, such as "*diseases*" or "*job-loss*" probabilities, which assign greater likelihood of such adverse events to occur for class *B*. The remaining attributes, although not directly related to class, are indirectly affected by it, such as the "*number of possible children*", or "*reproduction chance*", which increases in the event of a house or car acquisition. These are, in turn, more likely to occur to class *A*, since they are naturally able to accumulate more wealth. A succinct description of these can be encountered in Annex I.

C. Variables of the System

These specify the properties of the *model* itself, which in turn affect every entity in the simulation. These can be specified before starting the execution of any scenario, and some will get updated as steps advance or events take place. Certain factors remain unchanged during the entire run. More details can be found in Table II.

The default values for any simulation are the following: 150 agents in each group; the life expectancy in group *A* is 90 years, and in group *B* is 80; the wealth growth rate for group *A* is 0.6 and for group *B* is 0.3, the tax rate is 0.10 and the number of steps 100.

By altering these values, we can specify distinct scenarios we wish to simulate, in order to evaluate how they affect the data collection process, and how bias gets embedded into it. This allows us to gather insights regarding the most sensitive features that may affect results [28], so that we are able to successfully employ mitigation techniques.

D. Simulation Tools

To build the simulation environments used in the experiments, we employed both the `MESA` and `Netlogo` agent-based modelling (ABM) frameworks [6], [7]. In both, we resort to a *grid* where the realizations of entities, are represented

by pictograms, in which the colour denotes class, being red class *B* and blue class *A*, the shape denotes sex (M or F), and size denotes current wealth.

For the model training and evaluation tasks, we decided to proceed with using the *Python* programming language, as it is the norm for machine learning approaches [29]. A pipeline of three distinct classifiers was created, in order to showcase how the hypothesis is relevant regardless of algorithms, and that the bias is inherent to the data itself. The chosen methods were `XGBClassifier` [24], from the `xgboost` package, a gradient-boosting implementation that leverages decision trees, `RandomForestClassifier` [25], from the `scikit-learn` package, which builds multiple decision trees and merges their predictions to arrive at accurate estimations, and `LGBMClassifier` [23], from the `lightgbm` package, another gradient-boosting algorithm.

To ensure consistency across all methods, and to facilitate the modelling task, we apply standard scaling to the train and test features, using the `StandardScaler` method from the `scikit-learn` package. We leverage the `RandomizedSearchCV`, also from `scikit-learn`, to tune the parameters of each classifier, targeting optimal performance. Finally, if specified at the start, the models can employ *Isotonic Regression*, based on the `CalibratedClassifierCV` method, in order to calibrate the probabilities of the predictions, which is particularly significant in class-weighting settings, commonly paired with imbalanced datasets [30].

E. Data Requirements

Since we are generating all the needed data during the execution of the simulations, no external resources are needed to fully carry out the experiments. In the `MESA` framework, custom agent attributes are defined via the `mesa.Agent` class, which gets updated in every step and at specific events. Model variables are defined on the `mesa.Model` class, which are set at the start of the simulation. These values are tracked during each execution, via the `mesa.DataCollector` interface, and can at any point in time be stored in a `Pandas Dataframe`. In `Netlogo`, agents are defined as "*turtles*", which are easily accessed and manipulated via the "*ask*" and "*set*" commands. Then, the data originated at each step can be saved to a file via the "*file-print*" command, where the value associated with each defined attribute is captured.

From the generated data, we perform feature engineering before fitting the models, to ensure a reliable and correct training procedure. The employed features are the following: "Wealth", "Career Years", "Sex", "Job", "Has Diseases", "Has Car", "Has House", "Child Possibility", "Personal Luxuries" and "Health Cost". The target in the classification task is the "Group" label (originally *A* and *B*, converted to binary, 1 and 0 respectively), to ensure compatibility with all models. We employ both methods, as to assure that findings are independent of the data generating processes, given that although similar, the specificities of each inevitably lead to distinctive underlying patterns in the datasets.

IV. Operation Policies

In this section we detail the several simulation scenarios carried out to verify our hypothesis: data collected on the activities of a biased society, produces data with these biases embedded, which then skews the results of classifiers, even if the training and deployment process is correctly undertaken.

The base scenario, which considers the default values of the variables of the system, is a simple balanced binary classification problem (both classes are equally or nearly equally represented), where the privileged class benefits from diverse probabilistic boosts that allow the related entities to reach far superior wealth accumulation. These can be associated with opportunities, job loss, and even the base wealth accumulation rate. In this scenario, we simulate societal advances by, at some point (e.g., $3/5$ of the steps), equalling the wealth accumulation rate between the two classes. However, the remaining class-inherent biases and disparities are still present. Other scenarios are further developed based on this foundation, such as starting with a "semi-levelled field", where wealth accumulation rate is equal, but the class-inherent bias is kept unchanged. We aim to assess how impactful these are to the outcomes, and how can some characteristics, which are intrinsic to individuals, can be detrimental to their livelihoods. This implicitly conveys how crucial directly addressing these imbalances is, as superficial changes may not be enough to "level the field". It can be assumed, unless stated otherwise, that in each experiment, attributes of entities are not manually altered, and only the variables of the system.

Subsequently, we developed a variation for each of the previous scenarios, where we address the imbalanced domain learning problem, where one class, which is the most important for a given problem, is underrepresented in the data [31]. In these cases, we show how even if the data conveys just a small amount of bias (e.g., nearly equal opportunities), it can be severely amplified if the hampered class consists of fewer examples in the set. We apply the same process to the privileged class, to assess if the class-inherent boosts help in mitigating the imbalanced problem. In summary, the conducted operation policies were the following:

1: 150 members on each group, balancing the wealth accumulation rate (0.6), at $3/5$ of the maximum steps;
2: 150 members on each group, starting with a balanced wealth accumulation rate (0.6);
3: 200 members from group *A* with wealth accumulation rate of 0.6 vs. 100 members from group *B* with wealth accumulation rate of 0.3
4: 100 members from group *A* with wealth accumulation rate of 0.6 vs. *200* members from group *B* with wealth accumulation rate of 0.3
5: 200 members from group *A* vs. 100 members from group *B* with a balanced wealth accumulation rate (0.6);
6: 100 members from group *A* vs. 200 members from group *B* with a balanced wealth accumulation rate (0.6);

The experimental results of are present in Appendix V to XL.

V. Bias Introduction Process

We introduce bias in the previously presented simulations via two processes, which could be combined for maximal reinforcement. The first, consists of granting a probabilistic boost to the entities belonging to class *A*, in which for every positive event (wealth acquisition, being born with opportunities), the likelihood of occurrence is greater. More specifically, the *opportunities* attribute, which grants a boost of $1.5\times$ to the wealth growth rate, is assigned to entities of class *A* with a probability of 80%. Conversely, for members of class *B*, the likelihood of these events is minor, such as being assigned the *opportunities* attribute with a probability of 30%. Also, for them, the probability of negative events (such as job loss, contracting diseases) is higher. Furthermore, as some events that can benefit entities take place based on the momentary wealth level of each, this indirectly poses as an unfair policy for class *B*, since their members acquire wealth at a slower rate, effectively accumulating less.

The second bias induction process is simply based on the imbalanced domain problem, whereby setting a lower amount for the instances of one class, the employed binary classifier will be more favourable to the other [31]. This is then reflected on the predictions, which if evaluated, may show disproportionate scores for each class, assigning lower results to the minority, such as in *recall* (correctly identifying minority entities among the full minority class), for example. This imbalance is directly introduced in some scenarios, however, it should be noted that in most experiments, since the life expectancy for class *B* is lower, and the probability for reproduction for class *A* is higher, by the end of each run the number of entities on the first class may be noticeably inferior to the latter, impacting the results similarly.

Thus, by combining both methods, and altering the variables in order to assign different weights to each, we can simulate diverse situations that exhibit bias with varying magnitudes.

VI. Experimental Results

We carried out each of the previously mentioned policies, to interpret how bias manifests in the predictions. A pipeline including three different classifiers was employed to ensure the results were independent of algorithm selection. Furthermore, each is implemented in both of the selected ABS frameworks, to assess if the broad phenomenon is found across both, validating that this behaviour is inherent to the operations, and not the used tools. Although it is expected that results significantly diverge, given the differences of the underlying mechanisms, the overall patterns should still be somewhat parallel. Generally, `Netlogo` led to stricter learning tasks, when compared to `MESA`, mainly related to smaller discrepancies in target distribution, allowing us to use it for sensitivity analysis, to further characterise the circumstances of bias.

A. Scenario 1

By starting with a skewed wealth accumulation rate, but balancing it at $3/5$ of the simulation, we are able to observe a notable discrepancy across most metrics in `MESA`. *Precision* is slightly higher for class *B*, while *recall* is much lower. This implies that, since the model fails to detect most entities in this class, the few instances it does, are likely to be correct. This arises from the discrepancy in the *support* for each class, even though the experiment started with a balanced target distribution. In turn, the classifiers make use of the additional examples of class *A* to effectively learn the related patterns. In `Netlogo`, metrics are more evenly allocated, but overall, class *B* still boasts lower recall, validating the results. Notably, the class imbalance is much less prominent in this tool, which allows to further stress the class-inherent discrepancies. In this case, it showed that *recall* still falls short in more difficult scenarios. The results are shown in Tables V to X.

B. Scenario 2

As this setup was analogous to the previous, final outcomes did not deviate considerably in `MESA`, having models perform generally better for class *A*. Regarding metrics, *recall* slightly declined for class *B*, and *precision* got closer to the majority class values. Since wealth accumulation is balanced, the models may find it even harder to distinguish between classes, as only the class-inherent discrepancies remain. In a real world scenario, this factor is critical, as it showcases that solving "current problems", does not erase past issues. In `Netlogo`, the outputs are very consistent across classes, again, since only the class-inherent biases are used to distinguish them. Given that this tool leads to remarkably well-balanced data, the simple classifiers could not properly identify inherent differences, reinforcing the importance of balance in classification tasks. The results are shown in Tables XI to XVI.

C. Scenario 3

To further stress imbalance, both the number of entities, and the wealth accumulation rate of class *B* are set to half of the values from the *A*. This results in a set of heavily skewed models. Even if *precision* and *recall* for class *B* seem relatively comparable in `MESA`, to the values from the previous scenario, for *A* we can see an increase to just below total accuracy. Accordingly, it implies that the models are overfitting to class *A*, given the extreme under-representation of *B*, further amplifying neglect. In `Netlogo`, although the learning task was stricter, the disparities were made evident. As the unbalance was reinforced in this case, the classifiers had a harder time capturing relevant patterns from the minority. The irregularities that both bias and imbalance bring about when combined, led all classifiers to overfit on class *A*, exhibiting the importance of balancing the target distribution. The results are shown in Tables XVII to XXII.

D. Scenario 4

The following experiment flips the values from the number of entities assigned in the previous, while keeping wealth accumulation rate unchanged. This aims to stress the relevance of the wealth accumulation rate, while trying to mitigate the impact of the larger amount of offspring from class *A*, thus flipping imbalance. We found that the higher propensity to

reproduce, leads class *A* to consistently end scenarios with a larger support, which further skews target distribution. The results show that, in `MESA`, *precision* between both classes became somewhat balanced. *Recall*, on the hand, is now surprisingly higher for class *B*. This means that, even if the wealth accumulation rate is lower, and class-inherent bias are still present, the target distribution stands as the main source of inequality. In `Netlogo`, the same conclusions were reached, although with more adjacent values across classes. The results are shown in Tables XXIII to XXVIII.

E. Scenario 5

Wealth accumulation rate was fixed to the majority value, in order to independently assess the impact of the number of entities. The entity count of class *A* was set to the double of class *B*. Since the data was now more homogenous regarding bias, target imbalance served as the dominant issue. This lead to more than half of minority entities to be missed by the three classifiers in `MESA`. In `Netlogo`, the outcomes were more positive, although *recall* also achieved the lowest values in this tool, for class *B*, whereas for class *A* it led to near perfect scores. A lower *precision* for class *B* was also consistent in both frameworks. The results show yet more overfitting to class *A*. Results are shown in Tables XXIX to XXXIV.

F. Scenario 6

Wealth accumulation was kept unchanged, balanced, and the number of entities in each class was flipped. Corroborating the previous results, the now majority, class *B*, reached the best results across both frameworks, although once again, less discrepant in `Netlogo`. Interestingly, the class-inherent bias seemed to remain relevant, as it kept introducing some constraint to the predictions. Even with the boost given by the skewed target distribution, the unprivileged class, when in the majority, could not attain the high scores of the privileged one. Conversely, the privileged class *A*, never attained scores as low as class *B*, when in the minority, more pronounced with *MESA* generated data. In `Netlogo`, the effects were also present, as the best scoring class (now *B*), performed roughly $1 - 2\%$ worse than class *A*, in the previous experiment. The results are shown in Tables XXXV to XL.

G. Summary

These experiments showcased that the main factor contributing to bias is imbalance in the distribution of the target feature. As a result, it may lead to cases where the model essentially overlooks entities of the hampered class. Additionally, class-inherent bias, even if just slightly, always contributes to further hinder the predictions regarding class *B*. Notably, this is also the case even when advantages are assigned to its entities as to benefit them, as seen in Scenario VI-F. If the nuances that bias imparts upon data are subtle, simply balancing the distribution may lead to more uniform outcomes. However, in more complicated cases, these biases might still adversely affect the unprivileged class, leading models to inferior efficiency, compared to the privileged class on the same circumstances.

Overarching patterns of bias were identified across most outcomes, even when differing results were observed. While `Netlogo` always led to more homogenous and balanced learning tasks, `MESA` kept producing datasets that leaned towards imbalance in almost all settings. In the end, the insights from both allowed the assertion that target distribution is the most significant factor to account for, while also showcasing the need to compensate for subtle class-inherent discrepancies, as they still affect outcomes, even in balanced scenarios.

VII. Key Performance Indicators

The chosen key performance indicators (KPI) should reflect how bias inherent to the data affects each classifier, providing a precedent for comparison when revisions are introduced. While traditional metrics are concerned with overall performance, fairness metrics put emphasis on inhibiting disparities.

A. Performance Metrics

Regarding the classifier pipeline, standard metrics such as accuracy, precision, recall, f1-score and ROC-AUC were adopted, which acknowledge the ability to predict the correct class based on the provided features. However, when in aggregate, these represent the global performance of a method, which may not be reliable in unfair environments, being mostly based on the majority class estimations. In a per-group analysis for each, discrepancies were made apparent.

For the evaluation of existing fairness, Equal Opportunity Difference (EOD) and Difference in Misclassification Rates (DMR) were employed. EOD measures the difference in true positive rates across both groups (privileged and unprivileged), to ensure that the two yield equally correct predictions. DMR captures the differences in error rates between both classes, which enables us to assess if the models are unfairly misclassifying one group more than the other. Furthermore, selection rate, which measures the probability of entities from a group being assigned the positive class, is a useful way to assess how a model treats distinct classes differently. Consequently, by using it conjointly with Demographic Parity, we can assess if predictions are independent of group membership. Ideally, selection rate across both would be equal, and close to the global value. A description of each can be found in Table III.

VIII. Bias Mitigation Methods

We employed two methods to try to mitigate the effects of bias: a simpler *cost-sensitive learning* approach via *weighting*, where we increasingly assign more emphasis to class *B*, making the classifier "pay more attention" to its entities [32]. The second, more complex, uses the fairness metrics and methods from the `fairlearn` *Python* package [8].

A. Cost-Sensitive Learning

For the sake of brevity, we only use the `LightGBM` model, as we have shown the phenomena replicates across all classifiers. We also employ the same data provided in the multi-classifier pipeline. We have to instantiate a new model in order to change class weighting. This setting is defined in

the `is_unbalance` parameter, preparing the classifier for different weights, and `scale_pos_weight`, in which the specific class weighting scheme is specified.

1) Balanced ratio [1:1]: As shown in the experiments, a bias towards favouring class *A* is explicit. For it, the classifier showcases high precision, recall, and F1-score, but falls short with class *B*, especially in terms of recall. This indicates class *A* is estimated more accurately and consistently. The equal opportunity difference (EOD) and disparate misclassification rate (DMR) indicate that the model is not treating both classes equitably. Since class *A* is the "positive one", the classifier is far better at identifying related instances, in contrast to class *B*. A negative value in DMR, could indicate that the model is disproportionately misclassifying class *B* instances.

2) Favouring class B [2:1]: By favouring class *B*, a clear improvement is evident for its entities, with more true negatives, and less false positives. Interestingly, it performs worse in identifying class *A*, with more false negatives. Regarding recall, a noticeable improvement is seen for class *B*, which means it is better at identifying actual "negatives". Conversely, the precision has declined, indicating a higher false positive rate. The inverse is observed regarding class *A*. Overall, accuracy slightly lowered, but precision improved, meaning its positive predictions are more reliable. In EOD, a slightly lower difference is observed, meaning it is now fairer in identifying both classes. In DMR, a significant adverse effect against class *B* is still shown, although less promintent. Still, both metrics consistently showcase significant bias.

3) Further favouring class B [3:1]: In particular, recall appears oddly balanced across both classes. This setting leads to a proper recognition of class *B*, with a higher number of true negatives. Even so, it performs worse in detecting class *A*, with far more false negatives. This means that it might fail to identify "positive" instances. Recall further increases for class *B*, while precision decreases. Despite having lower overall accuracy, precision has increased, at the cost of a lower recall. This may imply a trade-off between missing positive cases while improving reliability, which is expected, as discussed in [33]. EOD has improved, meaning this setting is fairer in terms of differentiating both classes. The disparity in recall between the two classes is reduced, reflecting improved balance. DMR has also slightly improved, which means the model is not misclassifying instances of class *B* more than class *A*. It should be noted that bias is still significant.

4) Summary: The preceding scenarios show how by means of simple class-weighting, paired with prior knowledge of bias towards a particular group, its tenuous mitigation is attainable. Different ratios were set up to reflect the importance a classifier must place in a certain group, making it evident how difficult properly solving injustices can be with this method, as an improvement in fairness consistently coincided with a degradation of efficacy. A summary of the results can be found on Annex IV.

B. Fairlearn Package

Regarding the second method, using the `fairlearn` package [8], we can look at the effects of bias from distinct dimensions. Simply put, we evaluate how the model performs in the established binary classification task (identifying social class, A or B), grouping entities based on specific values in sensitive attributes. A sensitive attribute, in this context, defines characteristics that require additional attention due to high likelihood of unfair outcomes. In this experiment, we selected the *Has Car* feature, denoting if wealth is at least above 70% of the average wealth level, as it indirectly implies wealth prospects. Nonetheless, an analysis of other features is present in Tables XLI to XLIV. We leveraged the data generated from both the `MESA` and `Netlogo` frameworks, to analyse differences in results, and if the patterns observed in the previous experiments are still present. Once again, we employ only the `LightGBM` model.

1) Mitigation using `MESA` data: Preliminary results indicate that the accuracy is about 71.5% when evaluated on the entire dataset, without differentiating by groups, using the overall `Metric Frame` method. On the other side, the `Metric Frame by Group` evaluates the accuracy score (as it was the selected metric to showcase the general obfuscation of bias) for different groups within this feature, divided into two categories, *False* and *True*, since it is a binary attribute. For the group labelled *False* under *Has Car*, the model has an accuracy score of 66.5%. This indicates a notably lower performance when compared to the overall accuracy, of nearly 5.3%. For the group labelled *True*, the accuracy is 71.8%, which is closer to the overall score and significantly better than the group under *False*. This corroborates the previous results that showcase conventional global metrics, such as accuracy, as not suitable to quantify fair model assessments.

Entities with a positive *Has Car* value, tend to lead the model to better scores across several metrics, as shown by Figure 1, such as accuracy, precision, and false negative rate, which means that the classifier favours this group. Conversely, the (*Has Car* = *False*) group is impacted from a higher false negative rate, suggesting that the model consistently

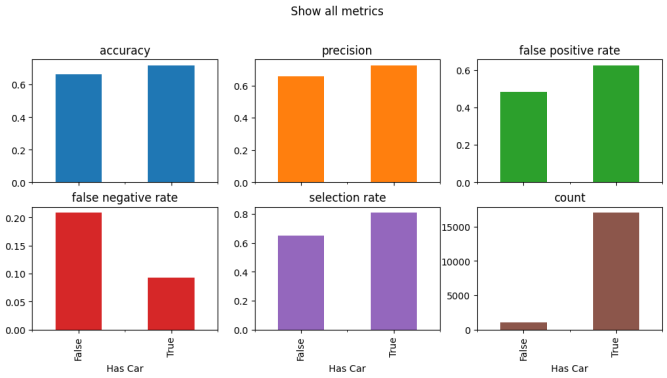

Fig. 1. Discrepancies between model accuracy regarding different groups under the *Has Car* feature, using `MESA`.

misclassifies negative instances. The reason for such inequity likely stems from the fact that there is less data available regarding entities without a car, mostly associated with class *B*, as seen in previous experiments.

Given the unsuitability of the accuracy metric, we decided upon the *selection rate* metric, which represents the proportion of individuals who were predicted as the positive across groups, revealing the tendency the model has to predict differently for each. The goal is to re-fit the same model, now with some constraints imposed, to ensure that the probability of a positive (fair) classification is the same across different groups. This metric is relevant since it avoids issues related to the prevailing imbalance. To that end, we employed, from the same package, the *Demographic Parity* constraint with the *ExponentialGradient* method, which iteratively modifies the predictions to minimise the differences across groups, while optimizing the trade-off around performance and fairness.

By calculating the *selection rate* across both groups, we can evaluate the effect of this constraint. After re-fitting and evaluating, the overall *selection rate* is 0.714, meaning that nearly 71.4% of entities are predicted to be in the positive class, regardless of their "*Has Car*" group. By evaluating group-specific *selection rates*, the percentage difference across groups decreased to just 1.5%, being nearly equal. Hence, we showcase promise in using these methods to successfully mitigate some problems inherent to biased classifiers.

2) Mitigation using `Netlogo` *data:* This experiment highlighted some previously identified patterns when employing data from this framework: both classes are more closely balanced, and discrepancies across metrics are more extreme: or nearly equal, or very distant. Here, the overall accuracy is about 62.0%, which is once again nearly equal to the accuracy of the group classified as *True* under *Has Car*, with 62.4%. The group classified as *False*, attained again a lower score, of 58.8%. The differences across metrics are shown in Annex 2.

By applying the same mitigation technique, the *selection rate* across groups was also successfully balanced, although it led to a lower score of 53.9%, meaning this process applied heavy constraints, to ensure both groups attained proportional levels of positive predictions, given the stricter learning task.

3) Other features: In order to derive more robust conclusions, the same experiment was conducted with other features: *Has Diseases*, in which a *False* value is the positive outcome, based on opportunity-influenced probabilities, and *Job*, which is the central driver for wealth accumulation. The results obtained are somewhat detached from the previously disclosed outcomes, given the contrasting feature nature. However, this makes it possible to stress test these methods, by introducing a varying range of attributes with distinct underlying conditions.

Nonetheless, the end results, which are detailed in Tables XLI to XLIV, all suggested the same conclusion: no matter what were the characteristics of the data, or more specifically, of a given feature, if a discrepancy in attribution of positive predictions is shown by uneven *selection rates*, these methods are successfully able to mitigate it.

IX. Limitations & Future Work

Although each devised ABM successfully highlighted patterns of bias in their respective operations, the simulated society and related wealth-centered activities were not as comprehensive as they ought to be. Even if it would be unfeasible to expect that all the intricacies of a civilization and its routines could be faithfully represented, the employed models still fell short of their potentialities, mainly grounded on a deficit of proficiency with the tools, and short development times. Thus, with a broader time frame, and the experience acquired during this analysis, we should be able to devise a new set of more complex models, which could in a future work simulate the proposed scenarios with higher fidelity. Also, this would facilitate the introduction of increasingly nuanced levels of bias, further stressing how classifiers might capture it, rising the scope and significance of this work.

In the implemented classifier pipeline, well-established preparation, tuning, and calibration procedures were conducted to ensure consistency and rigour. Nonetheless, to expand the scale and significance of the conducted sensitivity analysis, different methods, with distinct characteristics and underlying mechanisms, could also be included. This might be valuable given that in some reported cases, regardless of the small classifier selection, results were quite varied. Hence, incorporating a new set of classifiers, with more comprehensive ABMs, would allow for a closer portrayal of the effects of bias exhibited in real-world environments.

X. Conclusion

In this work, we successfully showcased how bias, present in a given simulated society, can become ingrained in data collected on the carried out activities. Later, these patterns were shown to be reinforced, when such data was applied in the training process of several classification models. By constructing agent-based models, a simplified example of this kind of society was built, and bias was represented by specifically setting environment variables and entities attributes accordingly, which affected differently members of each simulated group: *A*, privileged, and *B*, unprivileged. Afterwards, we verified that such bias was being embedded into training data, leading to skewed predictions, when evaluated in a group-wise manner, which was not clear in aggregated reports.

We developed different scenarios, on two different agent-based modelling frameworks, namely `MESA` and `Netlogo`, to illustrate the concept of societal-bias, focusing on transaction-based events. Subsequently, different bias mitigation techniques were applied, highlighting how methods of varying complexity could alleviate the negative effects of bias. Our findings suggest that simple weighting methods, although somewhat effective, also decrease the overall classifier performance, and inevitably lead to a tradeoff across different metrics, while still affecting distinctively both groups. More complex approaches are able to deal with bias regarding each sensitive feature of the train set individually, which contributes to more interpretable estimations, and balanced selection rates across both classes, thus leading to fairer outcomes.

REFERENCES

[1] L. Adlung, Y. Cohen, U. Mor, and E. Elinav, "Machine learning in clinical decision making," *Med*, vol. 2, no. 6, pp. 642–665, 2021.

[2] S.-A. Ionescu and V. Diaconita, "Transforming financial decision-making: the interplay of ai, cloud computing and advanced data management technologies," *International Journal of Computers Communications & Control*, vol. 18, no. 6, 2023.

[3] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits and systems magazine*, vol. 6, no. 3, pp. 21–45, 2006.

[4] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert systems with applications*, vol. 73, pp. 220–239, 2017.

[5] A. Wang and O. Russakovsky, "Directional bias amplification," in *International Conference on Machine Learning*. PMLR, 2021, pp. 10 882–10 893.

[6] J. Kazil, D. Masad, and A. Crooks, "Utilizing python for agent-based modeling: The mesa framework," in *Social, Cultural, and Behavioral Modeling*, R. Thomson, H. Bisgin, C. Dancy, A. Hyder, and M. Hussain, Eds. Cham: Springer International Publishing, 2020, pp. 308–317.

[7] U. Wilensky, "Netlogo," Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL, http://ccl.northwestern.edu/netlogo/, 1999. [Online]. Available: http://ccl.northwestern.edu/netlogo/

[8] H. Weerts, M. Dudík, R. Edgar, A. Jalali, R. Lutz, and M. Madaio, "Fairlearn: Assessing and improving fairness of ai systems," pp. 1–8, 2023. [Online]. Available: http://jmlr.org/papers/v24/23-0389.html

[9] E. Bonabeau, "Agent-based modeling: Methods and techniques for simulating human systems," *Proceedings of the National Academy of Sciences*, vol. 99, no. suppl_3, pp. 7280–7287, 2002. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.082080899

[10] S. De Marchi and S. E. Page, "Agent-based models," *Annual Review of political science*, vol. 17, no. 1, pp. 1–20, 2014.

[11] M. W. Macy and R. Willer, "From factors to actors: Computational sociology and agent-based modeling," *Annual review of sociology*, vol. 28, no. 1, pp. 143–166, 2002.

[12] C. O. Retzlaff, M. Ziefle, and A. Calero Valdez, "The history of agent-based modeling in the social sciences," in *International Conference on Human-Computer Interaction*. Springer, 2021, pp. 304–319.

[13] T. C. Schelling, "Dynamic models of segregation." *Journal of Mathematical Sociology*, vol. 1, no. 2, pp. 143 – 186, 1971.

[14] E. Ch'Ng, "Model resolution in complex systems simulation: Agent preferences, behavior, dynamics and n-tiered networks," *Simulation*, vol. 89, no. 5, pp. 635–659, 2013.

[15] C. Angione, E. Silverman, and E. Yaneske, "Using machine learning as a surrogate model for agent-based simulations," *Plos one*, vol. 17, no. 2, p. e0263150, 2022.

[16] E. Ntoutsi, P. Fafalios, U. Gadiraju, V. Iosifidis, W. Nejdl, M.-E. Vidal, S. Ruggieri, F. Turini, S. Papadopoulos, E. Krasanakis *et al.*, "Bias in data-driven artificial intelligence systems—an introductory survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 10, no. 3, p. e1356, 2020.

[17] A. Wang and O. Russakovsky, "Overwriting pretrained bias with finetuning data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3957–3968.

[18] M. Hardt, E. Price, and N. Srebro, "Equality of opportunity in supervised learning," *Advances in neural information processing systems*, vol. 29, 2016.

[19] K. V. Ramana, S. Sj, P. Ponsudha, S. Pd, A. V. Sangeetha *et al.*, "Applying cost-sensitive learning methods to improve extremely unbalanced big data problems using random forest," in *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*. IEEE, 2023, pp. 1–7.

[20] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, no. 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.

[21] M. Suri, "Pickle at semeval-2022 task 4: Boosting pre-trained language models with task specific metadata and cost sensitive learning," in *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 2022, pp. 464–472.

[22] C. O'neil, *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown, 2017.

[23] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in neural information processing systems*, vol. 30, 2017.

[24] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[25] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[26] J. Huang, G. Galal, M. Etemadi, and M. Vaidyanathan, "Evaluation and mitigation of racial bias in clinical machine learning models: scoping review," *JMIR Medical Informatics*, vol. 10, no. 5, p. e36388, 2022.

[27] T. Duboudin, E. Dellandréa, C. Abgrall, G. Hénaff, and L. Chen, "Encouraging intra-class diversity through a reverse contrastive loss for single-source domain generalization," in *Proceedings of the IEEE/CVF International conference on computer vision*, 2021, pp. 51–60.

[28] C. Ferrara, G. Sellitto, F. Ferrucci, F. Palomba, and A. De Lucia, "Fairness-aware machine learning engineering: how far are we?" *Empirical software engineering*, vol. 29, no. 1, p. 9, 2024.

[29] S. Raschka, J. Patterson, and C. Nolet, "Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence," *Information*, vol. 11, no. 4, p. 193, 2020.

[30] L. Huang, J. Zhao, B. Zhu, H. Chen, and S. V. Broucke, "An experimental investigation of calibration techniques for imbalanced data," *Ieee Access*, vol. 8, pp. 127 343–127 352, 2020.

[31] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[32] K. M. Ting, "Inducing cost-sensitive trees via instance weighting," in *European symposium on principles of data mining and knowledge discovery*. Springer, 1998, pp. 139–147.

[33] M. Buckland and F. Gey, "The relationship between recall and precision," *Journal of the American society for information science*, vol. 45, no. 1, pp. 12–19, 1994.

APPENDIX

Entity Class	**Description**
A	Enhanced opportunities and wealth growth
B	Fewer opportunities and higher adversity risk
Attribute	**Description**
class	Denotes the entity's social class (A or B)
sex	Defines the gender of the entity (M or F)
age	Specifies the entity's age, updated with each step
career_years	Years spent working, affects wealth accumulation
wealth_acc_rate	Wealth growth factor, higher for Class A and males
opportunities	Conditions for prosperity, boosting wealth by 1.5x
disease_prob	Likelihood of contracting diseases, higher for Class B
job_loss_prob	Chance of job loss, higher for Class B
offspring_count	Attainable child count, influenced by assets
repro_chance	Likelihood of reproduction, increased by assets

TABLE I
ENTITIES AND ATTRIBUTES

Variable	**Description**
num_agents_a	The starting count of agents in class A
num_agents_b	The starting count of agents in class B
group_a_wealth_rate	The starting wealth accumulation factor in class A
group_b_wealth_rate	The starting wealth accumulation factor in class B
age_of_death_a	Specifies the base lifespan for class A agents
age_of_death_b	Specifies the base lifespan for class B agents
taxes_rate	Determines the tax rate applied to each entity
max_steps	Limits the total number of simulation steps

TABLE II
SYSTEM VARIABLES

KPI	Description
Accuracy	Correct predictions over total predictions
Precision	TP out of predicted positives
Recall	TP out of actual positives
F1-Score	Balance of precision and recall
ROC-AUC Score	Overall classifier performance
EOD	Variation in TPR between classes
DMR	Disparity in error rates between classes
Selection Rate	Proportion of favourable predictions for a group
Demographic Parity	Independence of predictions from group membership

TABLE III
KEY PERFORMANCE INDICATORS (KPI) FOR MODEL EVALUATION

Metric	1:1	2:1	3:1
Accuracy	0.86	0.85	0.82
Precision (Class 0)	0.87	0.79	0.69
Precision (Class 1)	0.86	0.88	0.91
Recall (Class 0)	0.67	0.75	0.82
Recall (Class 1)	0.95	0.90	0.82
F1-score (Class 0)	0.76	0.77	0.75
F1-score (Class 1)	0.90	0.89	0.86
ROC AUC	0.92	0.92	0.92
Equal Opportunity Difference	0.95	0.90	0.82
Disparate Misclassification Rate	-1.95	-1.90	-1.82

TABLE IV
IMPACT OF DIFFERENT CLASS WEIGHT RATIOS IN A LIGHTGBM CLASSIFIER

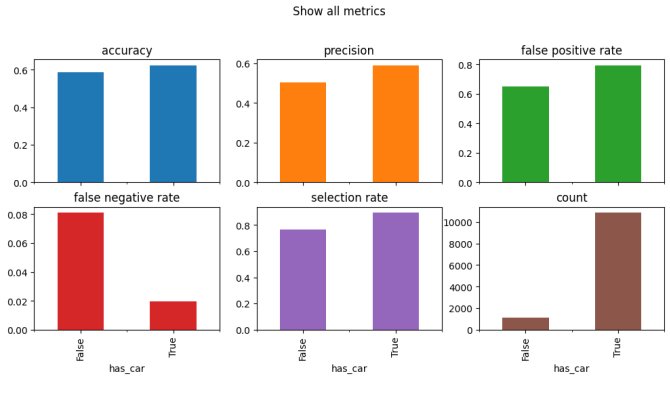

Fig. 2. Discrepancies between model accuracy regarding different groups under the *Has Car* feature, using `Netlogo`.

Class	Precision	Recall	F1-Score	Support
B	0.88	0.67	0.76	7264
A	0.82	0.94	0.88	11855
Accuracy		0.84		
Macro Avg	0.85	0.80	0.82	19119
Weighted Avg	0.84	0.84	0.83	19119

TABLE V
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPERIENCE 1 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.87	0.68	0.76	7264
A	0.83	0.94	0.88	11855
Accuracy		0.84		
Macro Avg	0.85	0.81	0.82	19119
Weighted Avg	0.84	0.84	0.84	19119

TABLE VI
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 1 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.88	0.67	0.76	7264
A	0.82	0.94	0.88	11855
Accuracy		0.84		
Macro Avg	0.85	0.81	0.82	19119
Weighted Avg	0.84	0.84	0.83	19119

TABLE VII
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 1 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.92	0.90	0.91	5751
A	0.91	0.93	0.92	6502
Accuracy		0.91		
Macro Avg	0.91	0.91	0.91	12253
Weighted Avg	0.91	0.91	0.91	12253

TABLE VIII
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPERIENCE 1 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.88	0.88	0.88	5751
A	0.89	0.90	0.90	6502
Accuracy		0.89		
Macro Avg	0.89	0.89	0.89	12253
Weighted Avg	0.89	0.89	0.89	12253

TABLE IX
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 1 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.89	0.88	0.88	5751
A	0.89	0.90	0.90	6502
Accuracy		0.89		
Macro Avg	0.89	0.89	0.89	12253
Weighted Avg	0.89	0.89	0.89	12253

TABLE X
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 1 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.83	0.59	0.69	7557
A	0.78	0.92	0.85	12142
Accuracy		0.80		
Macro Avg	0.81	0.76	0.77	19699
Weighted Avg	0.80	0.80	0.79	19699

TABLE XI
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPERIENCE 2 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.81	0.61	0.70	7557
A	0.79	0.91	0.85	12142
Accuracy		0.80		
Macro Avg	0.80	0.76	0.77	19699
Weighted Avg	0.80	0.80	0.79	19699

TABLE XII
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 2 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.82	0.60	0.69	7557
A	0.79	0.92	0.85	12142
Accuracy		0.79		
Macro Avg	0.80	0.76	0.77	19699
Weighted Avg	0.80	0.79	0.79	19699

TABLE XIII
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 2 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.87	0.87	0.87	6613
A	0.86	0.87	0.86	6352
Accuracy	0.87			
Macro Avg	0.87	0.87	0.87	12965
Weighted Avg	0.87	0.87	0.87	12965

TABLE XIV
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPERIENCE 2 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.85	0.85	0.85	6613
A	0.84	0.84	0.84	6352
Accuracy	0.84			
Macro Avg	0.84	0.84	0.84	12965
Weighted Avg	0.84	0.84	0.84	12965

TABLE XV
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 2 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.84	0.84	0.84	6613
A	0.84	0.83	0.83	6352
Accuracy	0.84			
Macro Avg	0.84	0.84	0.84	12965
Weighted Avg	0.84	0.84	0.84	12965

TABLE XVI
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 2 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.82	0.63	0.72	2741
A	0.96	0.98	0.97	22008
Accuracy	0.94			
Macro Avg	0.89	0.81	0.84	24749
Weighted Avg	0.94	0.94	0.94	24749

TABLE XVII
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPEREINCE 3 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.87	0.60	0.71	2741
A	0.95	0.99	0.97	22008
Accuracy	0.95			
Macro Avg	0.91	0.80	0.84	24749
Weighted Avg	0.94	0.95	0.94	24749

TABLE XVIII
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 3 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.88	0.59	0.71	2741
A	0.95	0.99	0.97	22008
Accuracy	0.95			
Macro Avg	0.92	0.79	0.84	24749
Weighted Avg	0.94	0.95	0.94	24749

TABLE XIX
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 3 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.90	0.89	0.89	3309
A	0.96	0.96	0.96	8885
Accuracy	0.94			
Macro Avg	0.93	0.93	0.93	12194
Weighted Avg	0.94	0.94	0.94	12194

TABLE XX
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPEREINCE 3 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.88	0.83	0.85	3309
A	0.94	0.96	0.95	8885
Accuracy	0.92			
Macro Avg	0.91	0.89	0.90	12194
Weighted Avg	0.92	0.92	0.92	12194

TABLE XXI
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 3 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.87	0.84	0.86	3309
A	0.94	0.95	0.95	8885
Accuracy	0.92			
Macro Avg	0.91	0.90	0.90	12194
Weighted Avg	0.92	0.92	0.92	12194

TABLE XXII
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 3 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.86	0.90	0.88	11358
A	0.81	0.73	0.77	6270
Accuracy	0.84			
Macro Avg	0.83	0.82	0.82	17628
Weighted Avg	0.84	0.84	0.84	17628

TABLE XXIII
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPERIENCE 4 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.83	0.94	0.88	11358
A	0.85	0.65	0.74	6270
Accuracy	0.84			
Macro Avg	0.84	0.80	0.81	17628
Weighted Avg	0.84	0.84	0.83	17628

TABLE XXIV
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 4 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.83	0.95	0.88	11358
A	0.88	0.64	0.74	6270
Accuracy	0.84			
Macro Avg	0.85	0.79	0.81	17628
Weighted Avg	0.84	0.84	0.83	17628

TABLE XXV
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 4 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.96	0.97	0.97	8813
A	0.94	0.91	0.92	3784
Accuracy	0.95			
Macro Avg	0.95	0.94	0.95	11897
Weighted Avg	0.95	0.95	0.95	11897

TABLE XXVI
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPERIENCE 4 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.93	0.97	0.95	8113
A	0.93	0.85	0.89	3784
Accuracy	0.93			
Macro Avg	0.93	0.91	0.92	11897
Weighted Avg	0.93	0.93	0.93	11897

TABLE XXVII
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 4 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.94	0.96	0.95	8113
A	0.92	0.88	0.90	3784
Accuracy	0.94			
Macro Avg	0.93	0.92	0.92	11897
Weighted Avg	0.94	0.94	0.94	11897

TABLE XXVIII
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 4 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.85	0.45	0.59	2991
A	0.93	0.99	0.96	20683
Accuracy	0.92			
Macro Avg	0.89	0.72	0.77	23674
Weighted Avg	0.92	0.92	0.91	23674

TABLE XXIX
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPERIENCE 5 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.83	0.48	0.61	2991
A	0.93	0.99	0.96	20683
Accuracy	0.92			
Macro Avg	0.88	0.73	0.78	23674
Weighted Avg	0.92	0.92	0.91	23674

TABLE XXX
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 5 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.84	0.47	0.60	2991
A	0.93	0.99	0.96	20683
Accuracy	0.92			
Macro Avg	0.88	0.73	0.78	23674
Weighted Avg	0.92	0.92	0.91	23674

TABLE XXXI
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 5 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.87	0.76	0.81	3574
A	0.92	0.96	0.94	9581
Accuracy	0.90			
Macro Avg	0.89	0.86	0.87	13155
Weighted Avg	0.90	0.90	0.90	13155

TABLE XXXII
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPERIENCE 5 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.87	0.70	0.77	3574
A	0.89	0.96	0.93	9581
Accuracy	0.89			
Macro Avg	0.88	0.83	0.85	13155
Weighted Avg	0.89	0.89	0.88	13155

TABLE XXXIII
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 5 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.86	0.67	0.76	3574
A	0.89	0.96	0.92	9581
Accuracy	0.88			
Macro Avg	0.87	0.82	0.84	13155
Weighted Avg	0.88	0.88	0.88	13155

TABLE XXXIV
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 5 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.84	0.94	0.88	13306
A	0.76	0.52	0.62	5061
Accuracy	0.82			
Macro Avg	0.80	0.73	0.75	18367
Weighted Avg	0.82	0.82	0.81	18367

TABLE XXXV
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPERIENCE 6 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.84	0.93	0.89	13306
A	0.76	0.54	0.63	5061
Accuracy	0.83			
Macro Avg	0.80	0.74	0.76	18367
Weighted Avg	0.82	0.83	0.82	18367

TABLE XXXVI
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 6 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.84	0.93	0.88	13306
A	0.75	0.53	0.62	5061
Accuracy	0.82			
Macro Avg	0.80	0.73	0.75	18367
Weighted Avg	0.82	0.82	0.81	18367

TABLE XXXVII
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 6 (MESA)

Class	Precision	Recall	F1-Score	Support
B	0.92	0.94	0.93	8762
A	0.84	0.79	0.81	3645
Accuracy	0.89			
Macro Avg	0.88	0.86	0.87	12407
Weighted Avg	0.89	0.89	0.89	12407

TABLE XXXVIII
RESULTS OF RANDOMFOREST CLASSIFICATION ON EXPERIENCE 6 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.89	0.94	0.91	8762
A	0.82	0.72	0.77	3645
Accuracy	0.87			
Macro Avg	0.86	0.83	0.84	12407
Weighted Avg	0.87	0.87	0.87	12407

TABLE XXXIX
RESULTS OF XGBOOST CLASSIFICATION ON EXPERIENCE 6 (NETLOGO)

Class	Precision	Recall	F1-Score	Support
B	0.88	0.94	0.91	8762
A	0.84	0.70	0.76	3645
Accuracy	0.87			
Macro Avg	0.86	0.82	0.84	12407
Weighted Avg	0.87	0.87	0.87	12407

TABLE XL
RESULTS OF LIGHTGBM CLASSIFICATION ON EXPERIENCE 6 (NETLOGO)

Metric	Value
Overall Accuracy	70.2%
Accuracy by Group	
False	76.6%
True	59.6%
Selection Rate by Group (Pre-Mitigation)	
False	0.90
True	0.70
Overall Selection Rate (Mitigated)	0.69
Selection Rate by Group (Mitigated)	
False	0.69
True	0.68

TABLE XLI

MITIGATION RESULTS ACCORDING TO *selection rate*, REGARDING THE *Has Diesises* FEATURE USING MESA. IN THIS SPECIFIC CASE, THE NEGATIVE VALUE IS COMMONLY ASSOCIATED WITH THE PRIVILEGED GROUP, GIVEN ITS NATURE.

Metric	Value
Overall Accuracy	62.0%
Accuracy by Group	
False	75.5%
True	50.4%
Selection Rate by Group (Pre-Mitigation)	
False	1.00
True	0.80
Overall Selection Rate (Mitigated)	0.45
Selection Rate by Group (Mitigated)	
False	0.45
True	0.44

TABLE XLII

MITIGATION RESULTS ACCORDING TO *selection rate*, REGARDING THE *Has Diesises* FEATURE USING NETLOGO. IN THIS SPECIFIC CASE, THE NEGATIVE VALUE IS COMMONLY ASSOCIATED WITH THE PRIVILEGED GROUP, GIVEN ITS NATURE.

Metric	Value
Overall Accuracy	71.5%
Accuracy by Group	
False	78.3%
True	70.9%
Selection Rate by Group (Pre-Mitigation)	
False	0.30
True	0.90
Overall Selection Rate (Mitigated)	0.72
Selection Rate by Group (Mitigated)	
False	0.71
True	0.72

TABLE XLIII

MITIGATION RESULTS ACCORDING TO *selection rate*, REGARDING THE *Job* FEATURE USING MESA. ALTHOUGH THE GROUP UNDER *False* ATTAINED HIGHER ACCURACY, IT WAS ALMOST NEVER CLASSIFIED POSITIVELY.

Metric	Value
Overall Accuracy	62.0%
Accuracy by Group	
False	62.2%
True	61.2%
Selection Rate by Group (Pre-Mitigation)	
False	0.60
True	1.00
Overall Selection Rate (Mitigated)	0.51
Selection Rate by Group (Mitigated)	
False	0.51
True	0.51

TABLE XLIV

MITIGATION RESULTS ACCORDING TO *selection rate*, REGARDING THE *Job* FEATURE USING NETLOGO. ALTHOUGH ACCURACY WAS BALANCED, SELECTION RATE WAS SIGNIFICANTLY DIVERGENT.