RE-SAMPLING TECHNIQUES TO MITIGATE DISCRIMINATION IN TRAINING DATA

Andrea Alfonsi, Giacomo Melacini
University of Bologna
{andrea.alfonsi2, giacomo.melacini}@studio.unibo.it

Marios Pitsiali
University of Cyprus
pitsiali.marios@ucy.ac.cy

ABSTRACT

With the widespread adoption of AI-powered models in various fields and applications, the issue of fairness and mitigation of bias is becoming increasingly important. The goal of this literature review is to gather some of the most popular solutions proposed in literature which deal with an unbalanced dataset, focusing on pre-processing methods that use re-sampling techniques to mitigate discrimination in training data, also trying to understand the impact re-sampling has on classification accuracy. This survey will first introduce the most basic methods, such as random undersampling and oversampling, it will then continue with the description of some of the more recent and advanced ones and will conclude with the implementation of some of them in practice to explore the effects they have in mitigating bias by experimenting with two datasets which focuse on credit card approval.

1 Introduction

The field of machine learning has made significant strides in recent years, with advances in algorithms and computing power enabling the development of increasingly accurate models. Nowadays, machine learning is influencing every aspects of our lives: from giving suggestions on what movies to watch or songs to listen, to more high stakes scenarios like hiring decisions or credit card approval. There are several advantages of using machine learning applications. For example they allow the automation of repetitive or time-consuming tasks, the predictive capabilities to capture future trends or outcomes, or the personalization of services and recommendations based on individual preferences.

However, with the continuous evolution and the widespread adoption of these techniques, there have been several cases of AI-powered models having issues in terms of bias and discrimination, leading to unfair outcomes and replicating existing societal inequalities [1]. In particular, imbalanced datasets can result in biased models that discriminate against underrepresented groups, leading to unfair outcomes.

Fairness in the context of machine learning refers to the absence of biases or discrimination in the data, algorithms, and outcomes of ML applications. A fair ML system should treat all individuals or groups equally, regardless of their race, gender, age, or other personal characteristics that are not relevant to the task at hand. The concept of fairness in ML is critical because biased or discriminatory outcomes can have significant social and economic consequences. For example, biased hiring algorithms can lead to discrimination against certain groups, while biased credit-scoring algorithms can perpetuate financial inequality. In the next section, further examples will be illustrated. There exist different metrics to measure the fairness of a predictive model. In [2], the authors have provided detailed coverage of the different techniques. To address the issue of algorithmic bias, solutions can be adopted in different parts of the system, and we'll describe the different possibilities in the next section.

This literature review will first introduce the concepts of bias and discrimination on machine learning systems, by also describing real-life examples where the consequences of these problems were observed. It will then continue by describing the main classes of solutions used to solve these problems in the literature, focusing on the pre-processing techniques used to manage fairness in a system and, in particular, it will explore recent research on re-sampling techniques to mitigate unintended bias, also experimenting with the implementation of some of them for two datasets that focus on the credit card approval domain.

2 Bias and Discrimination

Bias and discrimination are closely related concepts. Bias involves having a preference or prejudice towards a particular group or individual, often based on stereotypes or preconceived ideas. Discrimination is the act of treating people unfairly based on these biases. Both of these have long-standing implications in society, often leading to unequal opportunities and unfair treatment of individuals based on their race, gender, age, sexual orientation, or other attributes. These disparities have been observed in various aspects of daily life, such as hiring practices, lending decisions and law enforcement. For instance, studies have shown that job applicants with African American-sounding names are less likely to be called for interviews compared to those with white-sounding names, even when their qualifications are identical [3].

As machine learning models become more prevalent in decision-making processes, there is a growing concern that these biases, can become further entrenched and perpetuate the existing cycle of discrimination if left unaddressed. Predictive policing algorithms have been criticized for perpetuating racial profiling and exacerbating existing biases in the criminal justice system [4]. Another well-known case is Amazon's AI recruitment tool, which was abandoned after it was discovered to be biased against female candidates. The algorithm favored resumes containing words typically associated with male candidates, penalizing resumes with references to women's colleges or containing terms like "women's chess club"[5]. In healthcare, a study revealed that a widely used commercial algorithm assigned lower risk scores to Black patients than to White patients with the same level of health, resulting in unequal access to care management programs[6]. These examples underscore the importance of addressing biases in machine learning models to ensure fair and equitable outcomes across all sectors, which is why exploring techniques to mitigate biases, such as re-sampling techniques, is crucial.

2.1 Problem statement

The task we are handling is known as Discrimination-Aware Classification Problem[7]: a special case of classification where the training data exhibit unlawful discrimination towards sensitive attributes (features of an individual or a group that are protected by law or social norms, such as race, gender, age, religion, or sexual orientation.). The objective is to learn a classifier that optimize accuracy, but does not discriminate in its predictions on test data. In the discrimination-aware classification problem, a dataset with labeled data and one or more sensitive attributes is provided as input. The goal is to develop a classifier that predicts the label without any correlation with the sensitive attribute. The effectiveness of the classifier is evaluated based on its accuracy and discrimination.

2.2 Solutions

As suggested by [2], there are three main steps to exploit in order to mitigate algorithmic bias.

The first step is Bias Detection, which involves various techniques with the goal of identifying any systematic bias. The methods used for Bias Detection are Auditing (a set of techniques based on making cross-systems or within-systems comparisons on data) and Discriminatory Discovery (practices that use statistical metrics to detect unfair treatment by data/algorithms).

The second step is Fairness Management and its techniques can be grouped in three main categories[8]: pre-processing, in-processing, and post-processing. Pre-processing techniques (the ones we will focus on) aim to transform the data before any other steps are taken in order to remove any intrinsic discrimination. This can involve various methods, such as removing or balancing data elements, or adjusting the feature space to make it more neutral. By removing any potential sources of bias at the data level, pre-processing techniques can help create more fair machine learning models. In-processing techniques instead, include methods that modify the learning algorithms to remove discrimination during training. Examples for this category include regularization and optimization techniques [9]. Finally, post-processing techniques are used to try to mitigate any residual bias in the model by using a holdout set that was not involved in the model's training. Some of the post-processing methods include re-labeling of the decision outcome or re-ranking of the retrieved search results[2].

The third and final step is Explainability Management. It comprises techniques that promote transparency and establish trust between the user and the system. Explainability approaches aim to offer transparency of the system, enabling the identification of any bias or fairness issues in the data and model. Explainability Management techniques can be divided into two classes: Model Explainability (which provide a description of the training process for the model) and Outcome Explainability (which provide descriptions of the output that are useful especially if the user of the system in not an expert).

In the next section we'll dive deeper into pre-processing methods used for Fairness Management.

2.3 Pre-processing methods

We will now focus on pre-processing methods, briefly introducing the most popular ones. The most basic strategy, known as fairness through unawareness, is to directly remove the sensitive attributes from the training data. This method has been demonstrated to not always be beneficial[10], as in some cases the inclusion of sensitive characteristics in the data may actually be beneficial to the design of a fair model.

Another pre-processing technique is called massaging[7], and it works by changing the class labels of some instances in the training set from negative to positive and vice versa in order to ensure fairness. Instances are selected to be relabeled using a ranker algorithm. The algorithm identifies the top candidates closest to the decision boundary for promotion or demotion to minimize accuracy loss. The overall distribution of class labels remains the same after the massaging technique is applied.

The two most popular pre-processing practices are re-weighting and re-sampling. Re-weighting involves assigning higher weights to the minority class samples during the training process by a ratio equal to the population proportion over its sampling proportion. This modification ensures that the learning algorithm focuses more on the under-represented samples, compensating for the imbalance in the data. Re-sampling, on the other hand, involves modifying the original dataset by either oversampling the minority class, undersampling the majority class, or a combination of both. While both re-weighting and re-sampling methods yield statistically equivalent results, re-sampling outperforms re-weighting significantly when stochastic gradient methods are in use[11]. According to the study, re-sampling helps to correct sampling bias more effectively than re-weighting. In experiments on three tasks of classification, regression, and off-policy prediction, re-sampling consistently outperformed re-weighting in all three scenarios. They also concluded that re-sampling is numerically more stable and robust compared to re-weighting, making it a preferred choice for addressing biases in undersampled subgroups.

In the next section we will focus on re-sampling methods.

3 Re-sampling methods

With the rising popularity of stochastic gradient methods, exploring re-sampling techniques for mitigating biases in under-sampled subgroups is both timely and valuable for achieving robust, stable results. In the following paragraphs we'll first introduce the most general methods and then we'll continue with the description of some of the most recent ones.

3.1 General methods

A simple undersampling technique, as shown in [12], consists of uniform random undersampling of samples in the majority class. The problem with this technique is that it could potentially lead to a loss of information. An opposite approach to random undersampling is random oversampling, where the minority class is oversampled from a uniform distribution.

Another technique, called Tomek Links Removal, consists in removing pairs of examples that belong to different classes but are each other's nearest neighbours.

Synthetic Minority Oversampling Technique (SMOTE) is a more sophisticated method to oversample the minority class. For each minority sample p, r neighbours are selected from the k nearest neighbours. For each point r, a new synthetic datapoint is created by interpolation of p and r.

Another method used in the context of re-sampling without discrimination is Preferential Sampling [13], which involves selecting individuals that are representative of the population but do not exhibit any discriminatory patterns, rather than using a random sample. In this technique, data objects from the protected group that is close to the decision boundary are considered more vulnerable to having been discriminated against, while those from the unprotected group are highly likely to be favored due to dataset unfairness. Thus, these objects are preferred for sampling. The algorithm then learns a ranker to identify the borderline objects and preferentially samples them to reduce discrimination. This method was able to reduce the discrimination level by maintaining a high accuracy level.

More, Ajinkya [12] investigating the effects of re-sampling techniques in classification demonstrated improved performance on many re-sampling methods, with methods like SMOTE+ENN combined with logistic regression and the BalanceCascade method yielding the best results. To conduct the experiments, synthetic data was generated using the scikit-learn module. They focused on tracking the recall metric for the minority class and the precision metric for the majority class, as these metrics are particularly important when dealing with imbalanced datasets.

Some experiments[14] were done in literature to investigate the impact of re-sampling on classification accuracy, comparing different methods and highlighting key points and difficulties of re-sampling. After experimenting with different configurations of re-sampling methods, classification algorithms and parameters on multiple datasets, results showed that re-sampling improves the classification of imbalanced datasets in most cases if selected properly, but if not, re-sampling may have a negative effect on the quality of the classification. Moreover, there is no universally good choice of how to re-sample a dataset: the best re-sampling method for one dataset can be worse than no re-sampling for another.

In the next section we'll describe some more advanced re-sampling methods.

3.2 Advanced methods

In contrast to conventional approaches that focus on noise filtering of the majority class, Kang et al. [15] propose a novel re-sampling method that applies noise filtering to the minority class along with undersampling. This method, is called the Noise-Filtered Undersampling scheme and incorporates a K-Nearest Neighbors filter prior to the re-sampling process. The experimental results demonstrate that including a KNN filter significantly enhances overall performance.

Salimi et al.[16] proposed a re-sampling technique using the causal model approach called "interventional fairness". This approach treats the problem as a database repair issue where the training data is modified by inserting or removing tuples to remove any causal relationship between the sensitive attributes and the decision variable. This approach does not require knowledge of the underlying causal model as it is based on "intervention", which can be guaranteed even when the causal model is unknown. The only input required for this technique is labeling the attributes in the dataset as "admissible" or "inadmissible", where admissible attributes are allowed to influence the outcome despite having a causal relationship with the sensitive attribute.

Another re-sampling method which showed promising results is Fair-SMOTE[17]. Introduced in 2021, it is based on the postulation that the root causes of bias are the prior decisions that affect what data was selected and the labels assigned to those examples. The core idea is to rebalance data by first dividing the original dataset into subgroups characterised by every possible combination of values for sensitive attribues and classes. Then, the subgroups will be oversampled in order to reach the same cardinality for each one of them. To generate data for a subgroup, the procedure is similar to SMOTE: a sample is randomly choosen from the subgroup, knn is used to find the closest neighboors and then they'll act as parents to create new synthetic data. The procedure will be further explained in the next section, as it will be implemented and tested on two datasets.

FAWOS [18] is a fairness-aware oversampling technique that combines the SMOTE algorithm with a more sophisticated way to create new synthetic datapoints. Each minority datapoint is categorized based on its local neighbourhood with respect to the sensitive attributes, identifying its easiness to be learned as Safe, Borderline, Rare or Outlier. FAWOS introduces parameters to control the probability of datapoints belonging to each type to be used for oversampling. After selecting an existing minority datapoint, a new synthetic one is created by interpolation such as SMOTE. FAWOS computes the number of datapoints to be generated for each combination of sensitive attributes containing at least an unprivileged one. This allows for handling multiple sensitive attributes simultaneously. This method will also be implemented in the experiments section, and its procedure will be described more deeply.

Another similar technique to the previously described Fair-SMOTE is the one presented in [19]. It is based on the SMOTE algorithm but it uses a different way to select the datapoints to be used for oversampling. The algorithm is called Fair Oversampling (FOS) and it is based on the idea that bias in a model is caused by both under-represented classes and features. The objective of FOS is to restore the balance between the classes and protected features such that the number of examples in the majority class equals the number of examples in the minority one. It does this by first drawing samples only from the protected sub-group that requires the least number of samples to reach numerical equivalence. Then, it applies the same oversampling procedure using the protected group requiring the largest number of samples to reach numerical equivalence and the whole minority class. This approach provides an opportunity to increase the representation of privileged minority members, thus facilitating balanced parametric learning.

The work of [20] focuses on the detection of the samples that are most responsible for introducing the unwanted bias in the dataset and on their subsequent removal. This technique is similar to the Tomek Link removal described above. Flagged as problematic are the samples that have similar attributes but different protected attributes and different labels. The authors show that by dropping these samples the model fairness is ensured and the accuracy is improved. In addition to this, hyperparameters are provided to adjust fairness and accuracy as per dataset and business requirements.

4 Experiments

In order to observe the impact of resampling techniques in the fairness of a trained model, we conducted a series of experiments.

This section can be divided into three parts. First we'll introduce and describe the two datasets we decided to work on, which are both unbalanced on sensitive attributes and classes distribution. These datasets were chosen to represent real-world scenarios where fairness issues may arise. Then we will perform pre-processing techniques to them and train two models (logistic regression and SVM) to perform classification and analyze fairness. Finally we'll implement two recently introduced and previously described re-sampling methods (Fair-SMOTE and FAWOS) on both datasets in order to study their effects on the classification and fairness metrics. The implemented code is accessible on Github. [21]

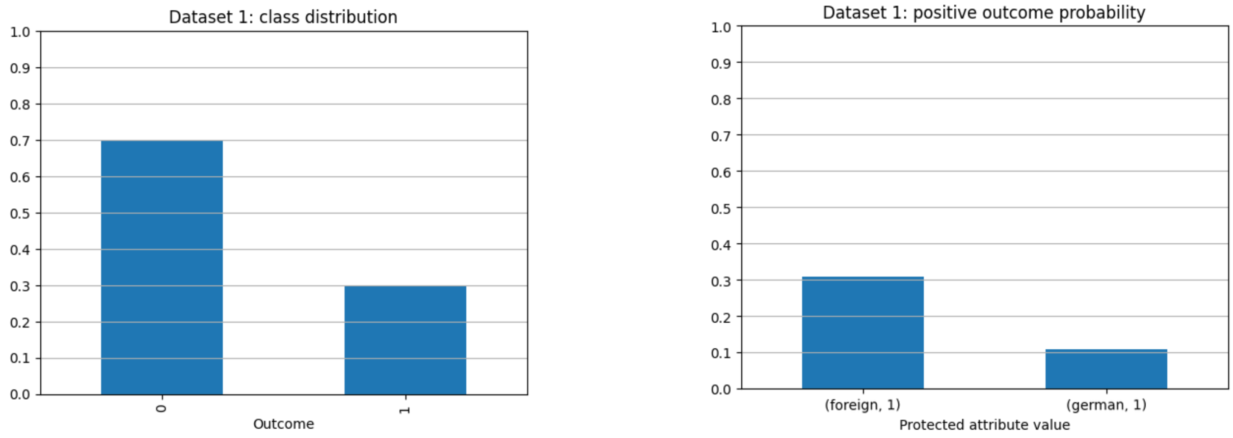
4.1 Datasets

These experiments are performed on two datasets related to the domain of credit card approval. Both datasets present many features, both categorical and numerical, and have sensitive attributes such as age, gender and ethnicity. To each entry there is an associated binary class where 0 represents the negative outcome (bad credit risk / credit card denied) and 1 the positive outcome (good credit risk / credit card approved). Both datasets, as shown later in detail, are unbalanced on some sensitive attributes, i.e. they are biased in favour of people presenting specific attributes (privileged attributes). This means that examples that have present privileged attributes have a higher chance of receiving a positive outcome. The presence of such bias, as explained in the previous sections, leads to the training of models that will be discriminating toward people presenting unprivileged attributes. This would be a problem in every domain, but it is especially in contexts that have legal and social implications such as the one of credit card approval. Let's now examine in greater detail each dataset.

4.1.1 Dataset 1: German Credit

The first dataset used in the experiments is the German Credit Card dataset [22]. It contains 1000 entries where people are classified as having good or bad credit risk. Each entry has 16 features and among them we have *age*, *sex* and *foreign* which will be treated as sensitive attributes. For the purpose of the experiments, we first tried to experiment on the gender attribute, but results showed that the dataset has a decent balance for this attribute: females had 35.2% chance of getting their credit card approved, while males had 27.7%. Because of this, we will instead focus on the attribute *foreign*, which is used to represent the nationality of the customer asking for a credit card.

This is a inherently binary attribute where the possible values that can be taken are *foreign* and *german*. In the dataset we have 37 *german* entries and 963 *foreign* ones. As shown in image 2, the dataset is class imbalanced and the imbalance rate changes when we split the dataset based on the protected attribute. It is possible to observe that entries with *german* attribute have a lower chance of getting good credit card risk (slightly above 10%). On the other hand, entries that have *foreign* attribute have a higher chance of receiving a positive outcome (about 30%).



(a) Class distribution over whole Dataset 1

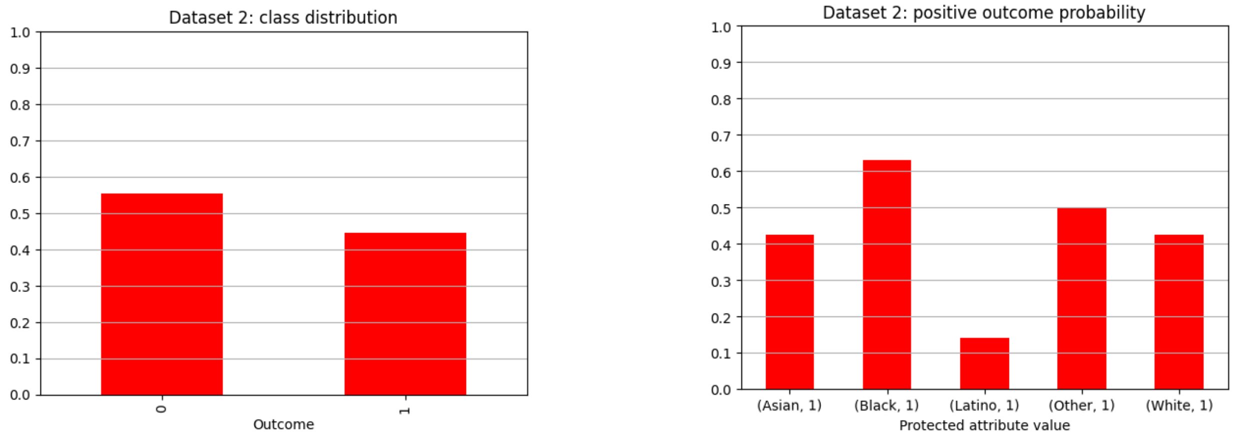
(b) Positive outcome probability by sensitive attribute

Figure 1: Distribution graphs for Dataset 1

4.1.2 Dataset 2: Credit Card Approval

The second dataset used is the Credit Card Approval dataset [23]. This dataset contains 690 entries containing 16 features. Each entry is classified as successful (1) or not (0) in applying for a credit card. The attributes that are considered sensitive in this case are *Gender*, *Age* and *Ethnicity*. Also in this case the experiments showed a decent balance for the gender attribute (46.7% approval ratio for female and 43.5% for male), so we will focus again on one attribute: *Ethnicity*.

This is a categorical attribute, which is different from the *foreign* attribute of the previous dataset (which only had 2 possible values), because its possible values are 5: *Asian*, *Black*, *Latino*, *Other* and *White* for which there are respectively 59, 138, 57, 28, and 408 entries. Also in this case we have that the dataset is imbalanced on the class but, as shown in image 2b, the imbalance rate for credit card approval varies greatly depending on the protected attribute value. We go from people with attribute *Black*, which have a chance of over 60% of a positive outcome, to Latino which have just slightly above 10%.



(a) Class distribution over whole Dataset 2

(b) Positive outcome probability by sensitive attribute

Figure 2: Distribution graphs for Dataset 2

In the next sections, we'll first describe the pre-processing methods applied, then we'll train two classifiers on both datasets and finally we'll compute fairness indicators on the predictions of the models over the test sets in order to see which categories should be treated as privileged and which ones as unprivileged.

4.2 Pre-processing

Before training a classification model we applied some pre-processing to both datasets.

We started by grouping all the columns that need pre-processing into numerical and categorical groups. The first dataset had 7 numerical and 9 categorical features, while the second dataset had 5 numerical and 3 categorical ones. For the sake of the experiments we ignored the feature *ZipCode*, since it wouldn't make sense as numerical and at the same time it would generate too many columns if we were to apply one-hot encoding to it.

We proceeded by focusing on the categorical features and transforming them using one-hot encoding. Categorical data that had only two possible values (i.e. *sex* and *tele* for the second dataset, which has value 'yes' if the customer has a telephone) can be represented by just using one column and therefore we got rid of the second one. The only exception is the sensitive attribute *foreign*, with possible values 'foreign' and 'german', which we decided to keep in two separated columns for coding purposes.

After focusing on the categorical features, we moved on to the numerical ones. We analyzed them and observed that they all had very different mean values, so in order to make them share a common scale we decided to perform normalization. This way all numerical values became between 0 and 1.

We didn't perform other pre-processing methods as we didn't think they were necessary and because we wanted to keep the experiments as simple and easily reproducible in every context.

4.3 Description of the re-sampling techniques applied

As mentioned before, we applied to both dataset two re-sampling techniques: FAWOS and Fair-SMOTE. These techniques are both based on SMOTE, i.e. both are based on the idea of generating new synthetic data by interpolation of existing ones. They differ in the selection of data to be interpolated and in the number of synthetic samples to generate. While FAWOS aims at equalizing the ratio between positive privileged (PP) and negative privileged (NP) with the ratio between positive unprivileged (PU) and negative unprivileged (NU), Fair-SMOTE aims at equalizing the cardinality for every possible combination of sensitive attributes and classes.

Let us consider a dataset D containing:

- S - the sensitive attribute (in our case foreign or ethnicity) containing privileged attributes represented as 1 and unprivileged attributes represented as 0
- CSU - the set of combination of sensitive attributes where each combination contains at least one unprivileged attribute
- CSC - the set of combination of sensitive attribute values and class
- Y - a target class where 1 is the positive class and 0 is the negative class (e.g. receiving credit or not)
- $\hat{Y}$ - the predicted class where 1 is the positive class and 0 is the negative class

The objectives of the two re-sampling techniques can be described as follows:

$$\text{FAWOS: } \frac{P(Y=1 \land S=1)}{P(Y=0 \land S=1)} \approx \frac{P(Y=1 \land S=0)}{P(Y=0 \land S=0)} \equiv \frac{PP}{NP} \approx \frac{PU}{NU} \tag{1}$$

$$\text{Fair-SMOTE: } \forall\, csc \in CSC : |csc| = n \text{ where } n \text{ is the cardinality of the biggest set in } CSC. \tag{2}$$

We proceed by looking in detail the algorithm of these techniques.

4.3.1 FAWOS

For each combination of sensitive attributes CSU_i ($S = 1$) we calculate how many positive synthetic points ($Y = 1$) we need to generate in order to satisfy the equation 1:

$$N_i = \frac{|PP| * |NU_i|}{|NP|} - |PU_i| \tag{3}$$

After computing the value of N_i for each combination of the sensitive attributes, we have to categorize every element P of PU_i based on its k-neighbours. In our case, following the work done in [18], we used $k = 5$. For each neighbour, we consider it a "real" neighbour if it has both the same target class and same sensitive attributes values as e. Finally, we classify P according to the number of "real" neighbours out of k ($k = 5$):

- 5:0 or 4:1 - *Safe* datapoint
- 3:2 or 2:3 - *Borderline* datapoint
- 1:4 - *Rare* datapoint
- 0:5 - *Outlier* datapoint

After labeling the datapoints, FAWOS generates N_i points belonging to PU_i. Each synthetic point is generated by a SMOTE interpolation between a point $P \in PU_i$ and one of its "real" neighbours. The selection of P is a weighted random selection where the probability of being selected depends on its label. These probabilities are defined as S_w, B_w, R_w and O_w, being hyper-parameters of FAWOS. In this set of experiments the weights were defined respectively as 0, 0.6, 0.4 and 0.

Since for Dataset 1 we have the same set of privileged and unprivileged attributes with both classifiers, the results of FAWOS will be the same. On the other hand, with Dataset 2 we will have also different re-sampling results.

In our case, for Dataset 1 and Dataset 2, before applying the re-sampling we had the following ratios:

$$\text{Dataset 1: } 0.44 \overset{?}{=} 0.12 \qquad \text{Dataset 2 LR: } 1.31 \overset{?}{=} 0.73 \qquad \text{Dataset 2 SVM: } 0.91 \overset{?}{=} 0.84 \tag{4}$$

These ratios confirm the fairness insights resulting from the fairness metrics. While both models had fairness issue when trained on Dataset 1, SVM seems not to have a consistent issue when trained on dataset 2. It is possible to notice that even before applying the re-sampling the ratios are very similar.

By applying the procedure described in equation 3 we discovered that the number of points to generate was the following:

- Dataset 1: 8 points having *german* attribute and positive outcome;
- Dataset 2 LR: 40 points having *latino* attribute, 2 points for *other* and 79 points for *white*, all having positive outcome.
- Dataset 2 SVM: 14 points having *white* attribute and positive outcome.

After the generation of the synthetic datapoints we ended up with the following ratios:

$$\text{Dataset 1: } 0.44 \approx 0.44 \qquad \text{Dataset 2 LR: } 1.31 \approx 1.30 \qquad \text{Dataset 2 LR: } 0.91 \approx 0.92 \tag{5}$$

4.3.2 Fair-SMOTE

Fair-SMOTE re-sampling method work by dividing the training data into subgroups characterized by class and sensitive attributes. For example, if both the class and the sensitive attribute are binary, there would be 4 (2*2) subgroups: favorable & privileged, favorable & unprivileged, unfavorable & privileged, unfavorable & unprivileged. These four subgroups would generally have different number of data points.

Fair-SMOTE changes the subgroups by synthetically generating new data points until all of them will have the same cardinality (same size of the subgroups which had the biggest size). The new data point is created by randomly choosing a parent node and by extrapolating between the values seen in two neighboring examples (using **K-nearest neighbor**). This way, Fair-SMOTE takes care of a problem that can rise when mutating data, which is that important associations between variables can be lost. In order to generate data, two hyperparameters, which both lie between 0 and 1, are used:

- *Mutation amount* (f): used to denote the probability the new data point will be different from the parent. We set it at 0.8 like it was proposed in the original paper (meaning it's different 80% of the time);
- *Crossover frequency* (cf): used to denote how much different the new node will be from its parent. The best value obtained in literature was again 0.8.

A different logic is used, based on the type of data. For boolean features, the new values will be randomly chosen between the ones of the parent and the neighboring samples. For numerical features instead, the new value will be generated by taking the parent node value and summing it with the difference of the neighboring nodes multiplied by *f*.

Instead of randomly creating a new data point, what Fair-SMOTE actually does, is generating a new data point which is very close to its parent. Because of this, the new data points will belong to the same distribution.

At the end of the application of Fair-SMOTE, the training data will have equal proportion of both classes and the sensitive attributes.

In the next sections, after describing the two fairness metrics that will be used, we'll start by describing the first set of experiments, consisting in applying two classification models to the datasets, without applying any methods to manage fairness. Then, we'll apply the two previously described re-sampling methods to the same datasets and using the same classifiers in order to study their effects and see if they are able to reduce bias.

4.4 Discriminatory Discovery

As explained in section 2.2, a common method used for Bias Detection is Discriminator Discovery. This method relies on statistical metrics used to detect unfair treatment by data/algorithms. In the case of these experiments, we relied on two metrics for Bias Detection: Demographic Parity and Equality of Opportunity.

Demographic Parity (or Statistical Parity) is obtained when both subjects of the protected and unprotected group have equal probability to be assigned to the positive predicted outcome. [2] In mathematical terms we want the following equality to hold:

$$P(\hat{Y} = 1|S = 1) = P(\hat{Y} = 1|S = 0) \tag{6}$$

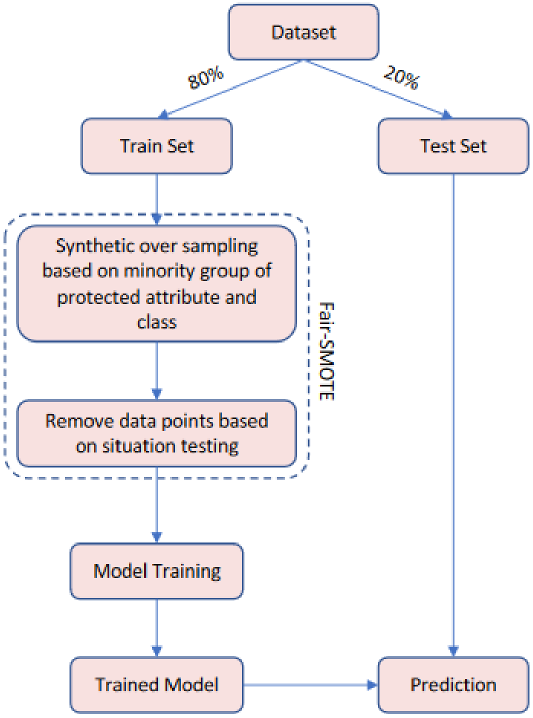

Dataset
80%
20%
Train Set
Test Set
Synthetic over sampling based on minority group of protected attribute and class
Remove data points based on situation testing
Fair-SMOTE
Model Training
Trained Model
Prediction

Figure 3: Block diagram of Fair-SMOTE[17]. It's worth highlighting how this diagram also includes another step, which is to remove data points based on situation testing. We didn't include this step in our experiments as it didn't have a positive impact on the performance.

In practice, we want to minimize the difference in positive rates for each combination of protected attributes.

Equality of Opportunity (or False negative error balance) is satisfied when the protected and unprotected groups have the same false negative rate. [2]

This means that we want the following equation to hold:

$$P(\hat{Y} = 0|S = 1, Y = 1) = P(\hat{Y} = 0|S = 0, Y = 1) \tag{7}$$

As for Demographic Parity, we aim at minimizing the difference of this value for each protected group.

4.5 Classification

The datasets were split into training and test set, using a test size of 25% of the total. Two different classifiers have been used to experiment with. The first classifier we used was a Logistic Regressor. The second one is a Support Vector Machine (SVM).

The metrics achieved by both models can be observed in table 1. These results will be used as a baseline to be compared with the results obtained after fairness re-sampling. The most preferable outcome is to improve the fairness of the models while preserving or improving their performance metrics.

Still, before applying any re-sampling technique, some objective metric shall be used to establish how unfair the models are and which are the discriminated groups. To measure the fairness, Demographic Parity and Equality of Opportunity have been used as per their description in section 4.4. The results for both models and both datasets are displayed in table 1.

It is shown that the logistic regressor trained on Dataset 1 is not fair toward people having *german* attribute, which has negative values for both fairness metrics. This means that *german*, in this specific case, will be treated as unprivileged

Category	DP	EoO	Acc	F1 M
Dataset 1			0.76	0.69
Foreign	0.0075	0.0021		
German	-0.2	-0.0569		
Dataset 2			0.85	0.84
Asian	0.0627	0.0076		
Black	0.1894	0.0189		
Latino	-0.3274	-0.0693		
Other	-0.1988	0.1306		
White	-0.0157	-0.0039		

(a) Logistic Regressor

Category	DP	EoO	Acc	F1 M
Dataset 1			0.75	0.65
Foreign	0.0066	0.0029		
German	-0.176	-0.0769		
Dataset 2			0.58	0.57
Asian	0.1281	-0.1196		
Black	0.0602	0.0682		
Latino	0.3039	-0.1965		
Other	-0.0104	0.0035		
White	-0.0740	0.0184		

(b) SVM

Table 1: Baselines for fairness and performance on both datasets and classifiers. Demographic Parity (DP) and Equality of Opportunity (EoO) are the metrics for fairness. A negative value means biased against while a positive one means biased in favor. Accuracy and F1 Macro score are the metrics used to measure the performance of the model.

attribute while *foreign* as privileged one. As concerns the second dataset, according to the fairness metrics, the categories *Asian* and *Black* will be considered privileged while the other three will be treated as unprivileged.

As for the Logistic Regressor, also the SVM trained on Dataset 1 is biased against people having *german* attribute. This means that also in this case *german* will be the unprivileged group. On the other hand, the results on Dataset 2 are less clear. We can observe in table 1b that the fairness of the model toward a group is opposite based on the metric we are using to measure it. In these cases, we will make a decision about a group being privileged or unprivileged based on the sum of the two metrics. If the sum of Demographic Parity and Equality of Opportunity is positive, then the group will be considered privileged and if the sum is negative the group will be considered unprivileged. In this case we have that *Asian*, *Black* and *Latino* are privileged groups while *Other* and *White* are the unprivileged ones.

With all the fairness and performance baselines set up, we can proceed experimenting with the re-sampling techniques showed in section 4.3. In the next section we'll show and discuss the results obtained.

5 Results

In this section we present the results regarding performance and fairness of the models trained on the re-sampled datasets.

Table 2 shows the accuracy and f1-macro score obtained with the baseline training sets and with the re-sampled ones for both logistic regressor and SVM.

	Accuracy	F1_macro
Dataset 1		
Baseline	**0.76**	**0.69**
FAWOS	**0.76**	0.68
Fair-SMOTE	0.64	0.62
Dataset 2		
Baseline	**0.85**	**0.84**
FAWOS	0.84	**0.84**
Fair-SMOTE	0.69	0.59

(a) Logistic Regressor

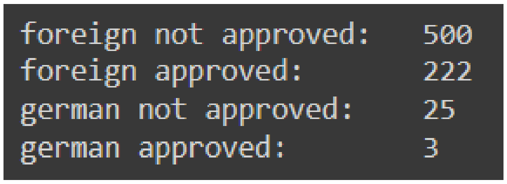
	Accuracy	F1_macro
Dataset 1		
Baseline	0.75	0.65
FAWOS	**0.76**	0.68
Fair-SMOTE	**0.76**	**0.69**
Dataset 2		
Baseline	0.58	0.57
FAWOS	0.60	0.58
Fair-SMOTE	**0.69**	**0.59**

(b) SVM

Table 2: Accuracy and F1 macro score for both classifier on both datasets. Scores are shown for baseline and after applying the re-sampling techniques.

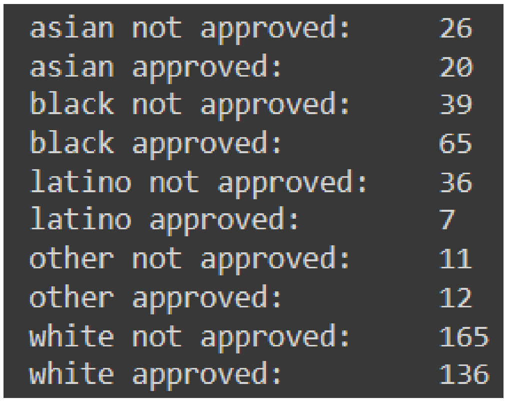
It's interesting to see how FAWOS is the most consistent method, being able to reach or overcome the values of the baseline for both classifiers and datasets. Fair-SMOTE on the other hand, obtains very low scores for the first classifier on both datasets, while for the second one (SVM) it is able to get the best results, with an especially big improvement for the second dataset.

The reason for the better results which Fair-SMOTE obtains in the second dataset with respect to the first dataset, is probably due to the number of samples in each subgroup. In fact, the subgroup representing germans whose credit is approved has only 3 elements, while germans having their credit declined have 25 elements. On the other hand, the subgroup for foreign not having their credit approved has 500 elements. As said before, because of the way Fair-SMOTE works, every subgroup should have the same cardinality after re-sampling, meaning in this case each one should have 500 elements. This means that for *german approved*, Fair-SMOTE will have to generate 497 synthetic data and 475 for *german not approved*. So the two groups will have respectively 99.4% and 95% of synthetic data, which could explain the lower results w.r.t. the second dataset, where the number of samples for each subgroup is much more homogeneous as we can see in image 5.



```
foreign not approved:   500
foreign approved:       222
german not approved:    25
german approved:        3
```

Figure 4: Number of elements in each Fair-SMOTE subgroup for dataset 1



```
asian not approved:     26
asian approved:         20
black not approved:     39
black approved:         65
latino not approved:    36
latino approved:        7
other not approved:     11
other approved:         12
white not approved:     165
white approved:         136
```

Figure 5: Number of elements in each Fair-SMOTE subgroup for dataset 2

On the other hand, FAWOS needs to generate fewer synthetic data in order to reach the equality of the ratios showed in equation 1. This means that the aims for fairness are satisfied without altering deeply the distribution of the dataset. In fact, the number of points that need to be generated for an unprotected group is proportional to the size of the group itself. The fact that the performance metrics are not always improved is consistent with the idea that the original datasets are biased. Improving the fairness is not guarantee of improving the results on a biased test set, but it should lead to fairer models that still perform well in real-world scenarios.

Table 3 shows the results for Demographic Parity (DP) and Equality of Opportunity (EoO) for every combination of classifier, re-sampling method and dataset.

Regarding the results obtained with the logistic regressor trained on Dataset 1, FAWOS was able to improve all the metrics. It is possible to observe that the improvement of the fairness metrics for the german attribute are greater in magnitude w.r.t. the improvements in the foreign attribute. This is explained by the fact that the model had very low values for bias in favour of the privileged group (foreign), while it was much more discriminating against the unprivileged one (german). On the other hand, Fair-SMOTE makes the fairness metrics slightly worse. This could be explained by the significant difference in the number of elements in each subgroup managed by Fair-SMOTE, as explained previously.

The results on Dataset 2 show that, also in this case, FAWOS is able to obtain better results than Fair-SMOTE, as it improves the metrics for more groups. The worst results obtained by FAWOS are related to the people having *Other* attribute. This could be due to the fact that the fairness metrics were conflicting on whether to label the group as privileged or unprivileged. We can also observe that the groups for which we had a bigger issue of fairness in terms of magnitude obtained also the biggest improvements.

Category	FAWOS		Fair-SMOTE	
	DP	EoO	DP	EoO
Dataset 1				
Foreign	0.0056 (0.0019)	0.0013 (0.0008)	0.0138 (-0.0064)	0.0033 (-0.0012)
German	-0.1225 (0.0775)	-0.0277 (0.0292)	-0.3688 (-0.1288)	-0.088 (-0.0311)
Dataset 2				
Asian	0.0338 (0.0289)	0.0191 (-0.0115)	0.0920 (-0.0293)	-0.0582 (*-0.0506*)
Black	0.1605 (0.0289)	0.0304 (-0.0115)	0.0377 (0.1517)	0.1815 (-0.1626)
Latino	-0.2849 (0.0425)	-0.0578 (0.0115)	0.0041 (*0.3233*)	-0.2890 (-0.2197)
Other	-0.2277 (-0.0289)	0.1422 (-0.0116)	0.0612 (0.1376)	-0.0890 (*0.0416*)
White	-0.0072 (0.0085)	-0.0111 (-0.0072)	-0.0265 (-0.0108)	-0.0086 (-0.0047)

(a) Logistic Regressor

Category	FAWOS		Fair-SMOTE	
	DP	EoO	DP	EoO
Dataset 1				
Foreign	0.0032 (0.0034)	0.0064 (-0.0035)	0.2896 (-0.2236)	-0.089 (*-0.0861*)
German	-0.0849 (0.0911)	-0.172 (-0.0951)	-0.0889 (0.0871)	-0.168 (-0.0769)
Dataset 2				
Asian	0.1165 (0.0116)	-0.1080 (0.0116)	0.0978 (0.0303)	-0.0524 (0.0672)
Black	0.0486 (0.0116)	0.0797 (-0.0115)	0.0435 (0.0167)	0.1873 (-0.1191)
Latino	0.2923 (0.0116)	-0.1850 (0.0115)	-0.0615 (*0.2424*)	-0.2832 (-0.0867)
Other	-0.0220 (-0.0116)	0.0150 (-0.0115)	0.067 (*-0.0566*)	-0.0832 (-0.0797)
White	-0.0668 (0.0072)	0.0113 (0.0071)	-0.0207 (0.0533)	-0.0122 (0.0062)

(b) SVM

Table 3: Fairness metrics computed on the models trained on the re-sampled dataset. In between the parentheses it is shown how much the metric got closer to zero after applying the re-sampling methods w.r.t. the baseline. Since 0 is the best possible result, a positive value in between the parenthesis means that the fairness toward that category improved.

Commenting on the SVM trained on Dataset 1, Fair-SMOTE also has bad results. These are related to the same issues that have already been commented. In this case also FAWOS had contrasting results having greatly improved the Demographic Parity but also worsened in the same way the Equality of Opportunity. Regarding dataset 2, FAWOS is able to obtain better results than Fair-SMOTE, with an improvement on seven values out of ten against the six improved by Fair-SMOTE. Also in this case we can notice that the metrics worsened by FAWOS are related to the groups that had the least discrimination problems.

These experiments showed the importance of testing different re-sampling methods based on the characteristics of the datasets and of the classifier we are using.

6 Conclusion

In conclusion, re-sampling techniques have emerged as effective methods for mitigating discrimination and biases included in the training data. In order to better understand how, we explored the definitions of bias and discrimination and presented some examples. There are multiple steps in the process of trying to mitigate algorithmic biases and in this study we specifically focus on the Fairness Management step and the pre-processing methods using re-sampling. There are numerous re-sampling methods which have been published in the last few years, exploring multiple ways of applying re-sampling using different techniques and methodologies. Some commonly used methods are undersampling, oversampling and SMOTE and many advanced techniques use variations of the above methods. As shown by [14], there is no one-size-fits-all solution for bias mitigation and the experiments that we have done have confirmed it. For instance, undersampling techniques may be very effective but require a large enough dataset in order to work as desired.

The experiments were done considering two different datasets regarding credit card approval/denial that contained sensitive attributes. The datasets were unbalanced both w.r.t. the class and the sensitive attributes. We focused on finding biases related to the ethnicity. For each dataset two classifiers were trained and two re-sampling techniques were used: FAWOS and Fair-SMOTE. The goal of these experiments was to measure the effectiveness of these methods in improving the fairness of the models while preserving or increasing their performance.

In terms of accuracy and F1 Macro score, both techniques lead to different results based on the model, confirming that the selection of the model will influence greatly the choice of the re-sampling technique. While Fair-SMOTE has been better when coupled with a SVM model, FAWOS proved to be a better choice when using a Logistic Regressor. More importantly, the choice of the re-sampling technique affects deeply the effectiveness of the fairness improvement of the model. Moreover, this choice should be strictly related to the nature of the dataset. As the experiments showed, Fair-SMOTE is not able to deal with datasets which have a great difference in the magnitude of each combination of protected attributes and class. On the other hand, FAWOS showed to be effective in those cases too. Both techniques have been able to generally improve the fairness metrics when applied to Dataset 2. Another consideration that we can make is that also the notion of privileged and unprivileged groups may differ based on the model we are using. As the experiments with Dataset 2 have shown, these groups changed depending on whether we trained a Logistic Regressor or a SVM.

Of course these experiments do not aim at being exhaustive since there are many factors that can affect the final results: the choice of the dataset, the pre-processing techniques applied on it, the selection of the models and of the choice of the re-sampling technique.

Future work could involve experimenting with all the aforementioned factors. It would be particularly interesting to try some fairness aware under-sampling techniques as well as to train different models on the datasets.

References

[1] Alina Köchling, Shirin Riazy, Marius Claus Wehner, and Katharina Simbeck. Highly Accurate, But Still Discriminatory: A Fairness Evaluation of Algorithmic Video Analysis in the Recruitment Context. *Business & Information Systems Engineering*, 63(1):39–54, February 2021.

[2] Kalia Orphanou, Jahna Otterbacher, Styliani Kleanthous, Khuyagbaatar Batsuren, Fausto Giunchiglia, Veronika Bogina, Avital Shulner Tal, Alan Hartman, and Tsvi Kuflik. Mitigating Bias in Algorithmic Systems—A Fish-eye View. *ACM Computing Surveys*, 55(5):87:1–87:37, December 2022.

[3] Marianne Bertrand and Sendhil Mullainathan. Are emily and greg more employable than lakisha and jamal? a field experiment on labor market discrimination. *American Economic Review*, 94(4):991–1013, September 2004.

[4] Kristian Lum and William Isaac. To predict and serve? *Significance*, 13(5):14–19, 2016.

[5] Jeffrey Dastin. Amazon scraps secret AI recruiting tool that showed bias against women, October 2018.

[6] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

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

[8] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *CoRR*, abs/1908.09635, 2019.

[9] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma. Fairness-aware classifier with prejudice remover regularizer. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 35–50, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[10] Indre Zliobaite and Bart Custers. Using sensitive personal data may be necessary for avoiding discrimination in data-driven decision models. *Artificial Intelligence and Law*, 24, 06 2016.

[11] Jing An, Lexing Ying, and Yuhua Zhu. Why resampling outperforms reweighting for correcting sampling bias with stochastic gradients, August 2021. arXiv:2009.13447 [cs, math, stat].

[12] Ajinkya More. Survey of resampling techniques for improving classification performance in unbalanced datasets, August 2016. arXiv:1608.06048 [cs, stat].

[13] Faisal Kamiran, Toon Calders, F Kamiran, Tue Nl, T Calders, and Tue Nl. Classification with No Discrimination by Preferential Sampling.

[14] Evgeny Burnaev, Pavel Erofeev, and Artem Papanov. Influence of Resampling on Accuracy of Imbalanced Classification. page 987521, December 2015. arXiv:1707.03905 [cs, stat].

[15] Qi Kang, XiaoShuang Chen, SiSi Li, and MengChu Zhou. A Noise-Filtered Under-Sampling Scheme for Imbalanced Classification. *IEEE Transactions on Cybernetics*, 47(12):4263–4274, 2017.

[16] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. Capuchin: Causal Database Repair for Algorithmic Fairness, October 2019. arXiv:1902.08283 [cs].

[17] Joymallya Chakraborty, Suvodeep Majumder, and Tim Menzies. Bias in machine learning software: Why? how? what to do? *CoRR*, abs/2105.12195, 2021.

[18] Teresa Salazar, Miriam Seoane Santos, Helder Araujo, and Pedro Henriques Abreu. FAWOS: Fairness-Aware Oversampling Algorithm Based on Distributions of Sensitive Attributes. *IEEE Access*, 9:81370–81379, 2021.

[19] Damien Dablain, Bartosz Krawczyk, and Nitesh Chawla. Towards A Holistic View of Bias in Machine Learning: Bridging Algorithmic Fairness and Imbalanced Learning. 2022.

[20] Bhushan Chaudhari, Akash Agarwal, and Tanmoy Bhowmik. Simultaneous Improvement of ML Model Fairness and Performance by Identifying Bias in Data, October 2022. arXiv:2210.13182 [cs].

[21] A. Alfonsi, G. Melacini, and M. Pitsiali. Re-sampling techniques to mitigate discrimination in training data. https://github.com/Chavelanda/Fair-Resampling.

[22] Professor Dr. Hans Hofmann Institut für Statistik und Ökonometrie Universität Hamburg. German credit card. https://www.kaggle.com/datasets/willianleite/german-credit-card?select=Credit.csv.

[23] Credit card approval (clean data). https://www.kaggle.com/datasets/samuelcortinhas/credit-card-approval-clean-data?resource=download.