Feature Selection Under Fairness and Performance Constraints

Ginel Dorleon[1][0000−0003−2343−4445], Imen Megdiche[1][0000−0002−1331−8662], Nathalie Bricon-Souf[1][0000−0003−2150−0998], and Olivier Teste[1][0000−0003−0338−9886]

Abstract. Feature selection is an essential preprocessing procedure in data analysis. The process refers to selecting a subset of relevant features to improve prediction performance and better understand the data. However, we notice that traditional feature selection methods have limited ability to deal with data distribution over protected features due to data imbalance and indeed protected features are selected. Two problems can occur with current feature selection methods when protected features are considered: the presence of protected features among the selected ones which often lead to unfair results and the presence of redundant features which carry potentially the same information with the protected ones. To address these issues, we introduce in this paper a fair feature selection method that takes into account the existence of protected features and their redundant. Our new method finds a set of relevant features with no protected features with the least possible redundancy under prediction quality constraint. This constraint consists of a trade-off between fairness and prediction performance. Our experiments on well-known biased datasets from the literature demonstrated that our proposed method outperformed the traditional feature selection methods under comparison in terms of performance and fairness.

Keywords: Feature Selection · Fairness · Protected Features · Bias · Machine Learning

1 Introduction

Feature selection is a popular dimensionality reduction technique for processing large dataset. Most of the time, dataset with a lot of features is often problematic because some features can be noisy, irrelevant and redundant; and as result it becomes difficult to extract meaningful conclusions [1].To deal with this problem, "dimensionality reduction" techniques such as feature selection are used [1]. Feature Selection (FS) helps to understand data, to reduce computation time, and to improve prediction performance. Based on the ways they are used, traditional feature selection methods can be widely divided into three categories: filter [2], wrapper [4] and embedded [5]. Methods based on the filter strategy use different statistics criteria to select important features from the features set without training a model, and thus they are known to be simpler and more efficient than both wrapper and embedded [3]. Wrapper methods use a learning algorithm to select relevant features that improve the learning performance [4]. Embedded methods integrate an automatic feature selection in a training process, and has proven to be more effective than the filter and the wrapper methods [5]. The main objective of any FS method is to select a subset of relevant features from the input data that helps improving model's prediction [6]. Fairness is another quality of the prediction model which can be of high importance for the usability of the model. Some specific features, known as protected, could induce problems when dealing with fairness and it has been proved [7] that protected features can lead to unfair decisions against minority groups.

According to [9], protected features are features that are of particular importance either for social, ethical or legal reasons when making decisions. Some examples of protected features are: sex, race, age, religion. With existing feature selection methods, two major problems are identified among the features selected which are: (1) protected features whose presence leads to biased results and (2) the presence of redundant features to the protected whose deletion leads to a loss in the prediction performance. In this study, redundancy is considered in the sense of correlation between non-independent features and the fact that the latter can be strongly correlated with others enabling a classifier to reconstruct them. For fairness, there are many definitions [10,11]. While each of these definition has merit, there is no consensus on what qualifies a model as fair, and this issue is beyond the scope of this article. Our goal is not to address the relative virtues of these definitions of fairness, but rather to assess the strength of the evidence presented by a set of features that a model is unfair to a certain subgroup based on a given metric.

Thus, in our work, we focused on these two problems identified among selected features that directly affect performance and fairness. Dealing with such issues, performance and fairness are computationally related and improving one leads to decreasing the other. In order to solve this, we introduce a method that allows to obtain the best trade-off between performance and fairness. Our method finds a set of relevant features without protected feature and with the least possible redundancy which maximizes the performance while ensuring fairness of the model obtained.

Our contributions in this work can be summarized as follows:

1- we introduce a more flexible way to use threshold for redundancy analysis by defining a threshold space instead of using a single value which could be subjective

2- we define an outcome-fairness algorithm for dealing with protected features in decision support algorithm
   - the algorithm takes into consideration redundant features while making decisions on fairness, so that the overall performance remains high
   - our introduced method to achieve fairness is based on two different fairness metrics in order to ensure the robustness of the approach

3- with our method, we show that it is possible to comply with data privacy policy by not using protected features while remaining efficient and fair

4- our introduced method to achieve fairness is easily adaptable to any decision making problems (regression, clustering...) involving protected features

The rest of this paper is organized as follows: in section 2, we summarize the different existing methods to tackle the issues identified with their limitations. Section 3 presents our new approach on protected features, redundancy and fairness. The experimental results are described and analyzed in section 4.

2 Related Work

Many existing work proposed various feature selection methods to deal with the problem of redundancy and protected features. Here we look at those who have proposed methods for redundancy analysis and those who have proposed methods for handling protected features in FS.

For the first category, the authors in [12] proposed a feature selection method known as "Minimum Redundancy & Maximum Relevance (mRMR). This method, based on Mutual Information (MI) as a correlation measure, makes it possible to select features that have a strong correlation with the output and a weak correlation between them in order to maximize relevancy and minimize redundancy. The authors in [13] proposed another feature selection method called Fast Correlation-Based Filter(FCBF), which uses symmetrical uncertainty as correlation measure and approximate Markov blanket to remove redundancy among features. In another method known as Redundancy Analysis Based Feature Selection (RABFS) [14], the authors use the maximum information coefficient(mic) to establish a threshold, analyze the redundancy between features and create a subset of relevant features.
However, when analyzing the methods cited above, we found that they inappropriately remove redundancy because they require users to set a single-defined threshold. We observed several problems with the strategy of using a single-defined threshold:
1) feature redundancy depends on the threshold set, that being said, different thresholds led to different sets of redundant features; thus, different models.
2) as more redundant features are removed according to the single-specified threshold, we observed a significant loss of performance.

For the second category related to protected features, we noticed some existing work that introduced different strategies to handle the problem posed by protected features. In [15] the authors introduced a naive approach, named "Fairness Through Unawareness", consisting of removing completely all protected features of the dataset to ensure fairness. In [9], the authors used a fair-group strategy based on a bias metric (disparate impact), to improve the fairness of prediction results within each sub-group. In another method [16] called "fair class balancing", the authors tackled the problem at a data processing level by proposing a method that allows to enhance model fairness without using any information about protected attributes.

Again, we noticed various limitations to the approaches cited above trying to improve fairness while considering protected features. Firstly, the approach of [15] of completely removing protected features may not solve the problem because there may be redundant features or even proxies to the protected. Because, as underlined by [7], some features known as proxies such as zip code, for example, can reveal the economic level or even predominant race of a residential area. Thus, this can still lead to racial discrimination in a decision making problem such as loan application despite the fact that zip code appears to be a non-protected feature [17]. Secondly, the approach of [9] using fair-group does not take into account the existence of redundant features or proxies which, potentially carry the same information as the protected and can affect the prediction within groups. As underlined by [7], it is possible to reveal information about a protected feature using its redundant. We notice the same observation for the work in [16], where redundant features to the protected are also ignored; this is dangerous in terms of fair outcomes when dealing with decisions problems involving minority groups.

Given the limitations of these above methods, there is a need for more in-depth research to overcome these limitations. Thus, we propose a new feature selection method which allows the building of efficient and fair models without protected feature and with the least possible redundancy. Our new method is a trade-off between performance and fairness. To compute fairness, we use two different bias metrics that have been proposed in the literature[21]: Demographic Parity and Equality of Odds. As each one of this metric uses a different criteria, they allow us to evaluate different fairness aspects of our approach. We give more details on these metrics in section 3.5. We would like to recall here, as part of our approach, by "fair model" we mean a model whose results are independent of one or more given features, in particular those considered to be protected [18].

3 The Proposed Method

In this section, we present our approach, the different steps are illustrated in Figure 1. Our method takes as input a dataset divided into protected and unprotected features. Then, it performs a redundancy analysis based on a defined threshold space (S). Following the redundancy analysis, two subsets of features are obtained: a list of non redundant features (N) and a list of redundant features (R). These two lists are used subsequently to train various models using all possible partitions between (N) and (R).The partitions are created by taking iterative combinations without duplication between the two subsets (N) and (R). Each partition is used to train a model, then for each model obtained, we calculate its f-score, its fairness and a trade-off score(Δ). We will keep as final model the one which has the highest trade-off (delta) score, i-e, the most efficient and fair one.

With this new method, we propose an efficient solution to the problem related to protected and redundant features on performance and fairness. This method makes it possible to take into account i) redundancy, ii) protected features and iii) fairness. Below, we give more details and explain every step of the proposed approach.

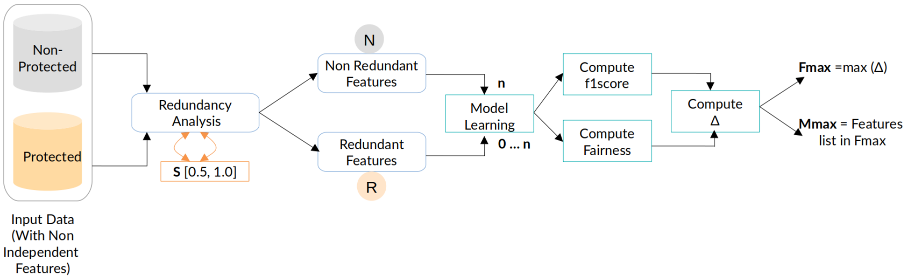


Fig. 1. The proposed approach and its different stages

3.1 Input Data

Once the input data is processed, we divided the input features into protected and non-protected features. In the majority of cases, protected features are known or designated by a system expert. In our case, to select protected features, we referred to the general data protection regulation (GDPR) of the European Parliament and Council on processing of personal data and the protection of privacy. According to article 4(13), (14) and (15) of the GDPR[8], protected features include: gender, race, ethnicity, age and more. In datasets used, gender and race were used as protected.

3.2 Redundancy Analysis

As we saw above in section 2, using a single-defined threshold could be subjective for redundancy analysis because for each chosen threshold, we would have a different features list and thus, a different model. To avoid this subjectivity in our redundancy analysis, we introduced a more flexible way to use threshold for redundancy analysis. To do so, we defined a redundancy space $S = [0.5, 1.0]$ in which we vary different redundancy thresholds (hyper-parameters) with a step $t = 0.05$. This strategy is efficient and allows to vary the thresholds precisely in order to have several graduation of the selected redundancy level. Thus, using the list of non-protected and protected features from our input dataset, we sought to determine the lists of non-redundant (N) and redundant (R) features based on the thresholds space S by using symmetrical uncertainty as measure of correlation [19]. The formula for calculating symmetrical uncertainty (SU) is defined by:

$$SU(X,Y) = 2\Big[\frac{IG(X,Y)}{H(X)+H(Y)}\Big] \tag{1}$$

Symmetrical uncertainty is a non-linear correlation measure based on the information theoretical concept of entropy, a measure of uncertainty of a random variable. The entropy of a random variable X is defined by:

$$H(X) = -\sum_{i} P(x_i) \log_2 P(x_i) \tag{2}$$

and the entropy of X observing another variable Y is:

$$H(X|Y) = \sum_{j} P(y_j) \sum_{i} P(x_i|y_j) \log_2 \big(P(x_i|y_j)\big) \tag{3}$$

Where $P(xi)$ represents the prior probabilities for all values of X and $P(x_i|y_j)$, the conditional probabilities of X being given the values of Y. The statistical difference between $H(X)$ and $H(X|Y)$ is called information gain or mutual information [19], and represents the degree of correlation between X and Y. Thus, using formulas (2) and (3), information gain or mutual information can be defined by:

$$IG(X, Y) = H(X) - H(X|Y) \tag{4}$$

We choose the SU measure explained by equations (2), (3) and (4) for several reasons. Firstly, it produces a normalized result between 0 and 1 and observes not only linear correlations, but also non-linear relationships between features. Secondly, it compensates the bias of information gain towards features with more values and restricts its values between [0,1]. A value of 1 indicates a strong correlation, a value of 0 indicates that X and Y are independent.

3.3 Model Learning

Once the list of non-redundant features (N) and the list of redundant features (R) are obtained from 3.2, we then seek to train various models using all possible partitions between N and R. We start by training a model with N only then we add iteratively every partition (a combination of features) of R until all the iterative combinations between N and R have been used. Like this, we have a list of models, each trained with a different combination of features (partition).

3.4 Computing F1-Score

F1-score is used as measure to assess performance of the learning models obtained in section 3.3. In a classification problem, the f1score (or fscore), is used to find the balance between precision and recall. Precision is the fraction of true positive (TP) examples among the examples that the model classified as positive (TP, FP). Recall, also known as sensitivity, is the fraction of the number of all correct examples classified as positive(TP) out of all positive that could have been classified(TP, FN). Based on the definition of Precision and Recall, the f1-score can be written as:

$$Fscore = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \tag{5}$$

3.5 Computing Fairness

To compute fairness, we use two different bias metrics that have been proposed in the literature [21]. In order to introduce the bias metrics that we used, we

introduce the following concepts. Let X be an input dataset with dimension nxp (n observations and p features):
Let f be a learning model and its performance score $f[X]$ which will be used to predict a binary output $\hat{y} \in \{0,1\}$. Each data point X_i is associated to a protected feature P, here we consider that P is binary: $P \in$ { 0,1}. We consider $P = 0$ to be an unprivileged group and $P = 1$ a privileged group. Likewise, we consider $\hat{y} = 1$ to be the preferred outcome, assuming it represents the more desirable of the two possible outcomes. For instance, $P =$'gender' could be the protected attribute with 'female' = 0, the unprivileged group, and 'male' = 1 the privileged.
Suppose for some data points we know the ground truth; i.e., the true value $y \in \{0,1\}$. Note that these outcomes may be statistically different between different groups, either because the differences are real, or because the model is somewhat biased. Depending on the situation, we may want our estimate $\hat{y}$ to take these differences into account or to compensate them. So, we used the two bias metrics introduced in [21] and defined below:

1. **Demographic Parity**. This metric suggests that a predictor is unbiased if the prediction ŷ is independent of the protected feature P such that Pr(ŷ| P) = Pr(ŷ) (Pr: Prediction rate). This means that the same proportion of each subgroup is classified as positive. To assess fairness from this metric, we use the difference between prediction rates of the subgroups. Let us call this difference Demographic Parity Difference (DPD), we defined it as:

$$DPD = Pr(\hat{y} = 1|P = 1) - Pr(\hat{y} = 1|P = 0) \tag{6}$$

2. **Equality of Odds**. This metric states that the prediction ŷ is conditionally independent of the protected feature P, given the true value y: Pr (ŷ|y, P) = Pr (ŷ|y). This means that the true positive rate and the false positive rate will be the same between unprivileged and privileged groups. To assess fairness from this metric, we use the difference between prediction rates (positive and negative). Let us call it Equality of Odds Difference (EOD), we defined it as:

$$EOD = Pr(\hat{y} = 1|P = 1, y = y_i) - Pr(\hat{y} = 1|P = 0, y = y_i), y_i \in \{0,1\} \tag{7}$$

Using the measurements obtained with the two metrics defined above in formulas (6) & (7), we are ready to compute a fairness score. For these two metrics, the result obtained is between -1 and 1, however the ideal value we would like to obtain is 0. Since the domain of performance values (fscore) is between 0 and 1, we use the absolute value of the measurement obtained in order to normalize the fairness domain between 0 and 1. Then, we invert the value obtained so that its greatest value is 1. For example, let $val \in$ [-1, 1] be the fairness value obtained for a metric, to invert it we proceed like this: new_val = 1-$|val|$.

For the partitions used (section 3.3), when the list N is used alone, i-e when there is no redundant feature, we consider that the fairness score is 1 since there are no protected features nor redundant to the protected ones. When we add partitions from R to N, the added redundant feature (from R) is used as P to assess fairness. If there are multiple redundant features in R to the protected, we calculate an intermediate fairness score for each of the redundant and then average it to obtain a final fairness score. With the other FS methods used for comparison in our experiment, if there is any protected feature (i.e. P), in their list of selected features, it is used to assess fairness using the formulas in (6) & (7). Otherwise, we consider their fairness score to be 1. Since we have two values for fairness from the two bias metrics used, the final fairness value used is an average of the two fairness scores obtained: $fairness = (DPD, EOD)/2$.

3.6 Computing the Trade-off (Δ)

We have defined our trade-off formula as follow:

$$\Delta = (1 + \beta^2) * \frac{f1score * fairness}{\beta^2 * f1score + fairness} \tag{8}$$

The formula is inspired from the traditional F-measure [20] and helps to compute the harmonic mean between our f1score and fairness. The reason we have chosen to use the harmonic mean instead of other means is that it allows us to weigh the fairness higher than the fscore. With the classical arithmetic mean, the higher score would have been more important(sometimes it could be the fscore and sometimes the fairness). But by choosing to do so, we decide to assign a greater importance to fairness (whether this score is smaller than the fscore or not). In the formula defined for delta, two commonly used values for β are 2 which weighs fairness higher than f1score, and 0.5, which weighs fairness lower than f1score. In our experiments, the beta (β) in the formula is then set to 2 ($\beta = 2$). For each trained model in step 3.3, we obtain a fscore and a fairness value. Then we compute a delta for each model using their $f1score$ and $fairness$. The final delta will be the max, which we note $Fmax$, of all the delta. The list of features according to $Fmax$ is noted as $Mmax$.

3.7 Algorithm of the Proposed Method

Algorithm 1 shows the process of the proposed method. Let N be the set of non-redundant features, we denote by f_i a feature of N. Let P the set of known protected features or designated by a system expert before any analysis, we denote by p_i any feature of P. The algorithm takes as input two lists: the list of non-protected features N and the list of protected features annotated P from the input dataset. We use the following defined parameters: t the step ($t = 0.05$) to iterate over S and the hyper-parameter space (S = [0.5, 1.0]). We start by initializing the values of $Fmax$, $Mmax$, d, t and the empty list R (line 2-6). For each hyper-parameter (threshold) d in S, we seek to find the redundancy between the list of protected P and the list of non-protected N, if any redundant feature is found according to d, it is added to R then removed from N (line 7-16). Using N, we iteratively increment over all possible partitions cr of R to train all possible models using all the partitions between the lists N and R, evaluate them and calculate their delta according to the specified formula(17-25). Then we decrement, start over using a new value of d until all possible value in the hyper-parameter S have been used (line 27). For the output of the algorithm, we have **Fmax** which is the max of all the calculated delta, and **Mmax** which is the list of features constituting the model which led to the delta max (Fmax).

Algorithm 1: Pseudo-code of the proposed method

```
   Input: N, P // Non protected and protected Features
   Output: Fmax, Mmax //max performance & feature list
1  Begin
2  t ← 0.05 //iteration step over S
3  d ← 1.0 //highest threshold value in S
4  R ← { } //redundant list
5  Fmax = 0 //max($\Delta$) to maximize
6  Mmax ← { } //feature list of Fmax
7  while d ∈ S do
8      //finding redundant features
9      for $f_i$ ∈ N do
10         for $p_i$ ∈ P do
11             if |compute corr($p_i$, $f_i$)| ≥ d then
12                 R ← {$f_i$} ∪R
13                 N ← N \{$f_i$}
14             end if
15         end do
16     end do
17     //search for the best model with the best trade-off $\hat{f}$
18     for cr ∈ partition(R) ∪{}do
19         compute $\hat{f}$ using N ∪ cr
20         compute $\Delta$ using eq. (1)
21         if $\Delta$ ≥ Fmax then
22             Fmax ← $\Delta$
23             Mmax ← $\hat{f}$
24         end if
25     end do
26     d ← d - t
27   end do
28 End
```

4 Experimental Setup

In this section, we present the experimental approach that we carried out and the comparative results obtained.

Goal: the goal of these experiments is to compare the results obtained with our method with other Feature Selection methods. For that, two other existing feature selection methods were used for comparison: mRMR [12] and FCBF [13]. In particular, this comparison was made based on 3 criterion (performance, fairness and the delta score) using a classification task. We have also compared the numbers of selected features by each method(Table 4.2).

Baseline: the first existing method used for comparison with our proposed method is mRMR [12]. It aims to select a subset of highly relevant features while reducing redundancy between themselves. This method is a two stage process; the first stage includes an incremental feature selection which later is combined to another more sophisticated wrapper feature selectors. We used the backward and forward selectors as in the original paper.

FCBF [13] is the other existing method used for comparison to our proposed one. It uses symmetrical uncertainty as correlation measure and approximate Markov blanket to remove redundancy [20]. This method requires the user to set up a threshold and this is somehow subjective. However, the authors stated in their paper that setting the threshold to a reasonably large value does not sacrifice the goodness of the selected subsets, thus in our experiment, the threshold was set to 0.6.

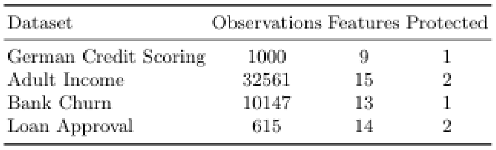
When used, each above method outputs a final subset of relevant features, normally this represents a "partition" of our proposed method. Then we use this final relevant subset from each method to train a model (3.3), compute its f1-score (3.4), its fairness (3.5) and the trade-off score (3.6).

Classifiers: To evaluate and compare the proposed method to existing methods, we proceeded to a learning task by considering a binary classification problem over the 4 datasets that we describe below (section 4.1). For this binary classification, RandomForest [22] and AdaBoost [23] were used as classifiers. This choice is explained by the main advantages of these classifiers which ensures high precision through cross validation, providing an easy interpretation of the obtained result. And also the fact that these two classifiers use two different ensemble strategies (bagging & boosting), this allowed us to seize different aspects of each method on the learning task. Each model is trained and evaluated using the classic cross-validation procedure. The f1-score as described in section 3.4 is used as measure to assess the performance of each trained model.

4.1 Datasets

To evaluate our method, we carried out experiments on four well-known datasets in the literature [25]. They each contain known protected features, which allowed us to evaluate our method on appropriate cases. These datasets were chosen on the basis of the differences they present, their types and the number of observations varying from 615 to 32561 and the number of features. The German credit dataset classifies people described by a set of features as good or bad credit risks. The Adult income dataset task it to predict whether income exceeds "50K/yr" based on census data. The third dataset used is Bank Churn, and the goal is to predict customer churn in a bank. The last dataset used is Loan Approval with the goal of predicting an applicant eligibility to get a loan. Based on the general data protection regulation (GDPR) policy, we have identified which feature is protected in each dataset after processing. In Table 1, we give more details on the datasets used.

Table 1. Experimental datasets used



Dataset	Observations	Features	Protected
German Credit Scoring	1000	9	1
Adult Income	32561	15	2
Bank Churn	10147	13	1
Loan Approval	615	14	2

4.2 Results Analysis

The analysis of the results is based on two criterion: the number of selected features and the trade-off score(fairness and performance).

Selected Features: we present in Table 2 below the first comparison results obtained with the different methods over selected features. For all the methods: **(N)** represents the number of selected features and **(R** $\in \hat{f}$**)** the number of redundant features that is part of the list of features used for the final model. **(P)** is the number of considered protected features for our method and for the two other methods, it is the number of protected features that we observed in their selected list.

We notice that the number of selected features of our method is higher than the other approaches. This is due to our fairness goal without removing any features beforehand while the others remove some features based on their relevancy and redundancy approaches. We also recall that our method, in order to find redundant features, uses protected features. Here comes one of the advantages of using our threshold space for redundancy analysis. It helps us to see different levels of redundancy for a specific feature so we can identify which feature has no influence on the model's fairness and which one boosts performance. Observing the results, we have used two protected features (sex, race) for Adult and Loan Approval dataset while for German Credit and Bank Churner dataset we have used one protected feature(sex). For our method, the protected features are used in the redundancy analysis step only, they help finding the redundant features using the threshold space (S = [0.5, 1.0]).

When using the two other existing methods (mRMR and FCBF), we have observed that their final lists contain protected features **(P)** and also features that have been highlighted as redundant **(R)** by our method. This is explained by the fact that theses methods do not take care nor propose any processing to handle protected features which is not the case for our method. Here comes one of the contribution of our method on not using personal protected or sensitive data with respect to data privacy policy.

Trade-off score(F1-score, Fairness): On Tables 3 & 4 (appendix), we report the results based on the trade-off score(f1-score, fairness), obtained by our method compared to the two other FS methods. The experiments were carried

Table 2. Comparison over number of selected features by each method

	Proposed				**mRMR**				**FCBF**			
Datset	**(N)**	**(P)**	**(R)**	**(R $\in \hat{f}$)**	**(N)**	**(P)**	**(R)**	**(R $\in \hat{f}$)**	**(N)**	**(P)**	**(R)**	**(R $\in \hat{f}$)**
German	7	1	1	1	5	1	0	0	6	1	0	0
Adult	12	2	3	2	7	2	1	1	9	1	1	1
Bank	11	1	2	1	8	0	0	0	7	0	2	2
Loan	11	2	3	1	6	1	1	1	8	1	0	0

out using Random Forest and AdaBoost, on four well known datasets for fairness study. F1-score was used to assess performance of each classifier. We compute fairness based on two different bias metrics (DPD, EOD) as exposed on section 3.5. The score for Delta is calculated using equation (8) and the process explained in section 3.6.

Observing the results with Random Forest, we can see clearly that our method outperforms the two other existing methods on three datasets, except Bank Churner where they have a higher score for Delta. However, this understandable because, since there is no protected feature in their final list, we had to set their fairness score to 1. Observing Table 4 with AdaBoost, our method has a better delta score than the other methods on three datasets except Bank Churn where mRMR and FCBF perform better than the proposed. Again, this is due to the fact that we had to set their fairness score to 1. On Table 4, we notice that FCBF has a better f1-score than our method for German dataset, but still our method has a better fairness thus a better Delta score. We observe the same thing with mRMR on the Adult Income dataset where it has a higher f1-score than our method, but again, our method still has a higher Delta score since our fairness score is higher. Overall, the results of the experiments show that our method performs well and our redundancy analysis guided by protected features gives a better performance value in term of fairness and the trade-off score generally.

We also compared the execution time of our algorithm to other FS methods (Fig. 2). The FCBF method is faster than our method over all the datasets. It is understandable since this method performs a single redundancy analysis using only one defined threshold while our method uses a threshold space to perform redundancy analysis. However, our method is faster than the mRMR method which in fact is a two stages process with two wrapper selectors.

In general, on these 4 datasets, we get satisfactory results and we have maintained a good level of performance (f1-score), a higher fairness guarantying a higher score for the trade-off between f1-score and fairness.

5 Conclusion

In this article, we present a novel feature selection method to improve performance and fairness in the case of protected features while considering their redundant. To achieve our goal, we introduce a trade-off strategy between performance and fairness. This new method, unlike existing methods allows in the presence of protected and redundant features to obtain a model that is both optimal and fair with respect to data privacy policy. The performance of our method was experimentally evaluated on four well known biased datasets. Compared to two other existing feature selection methods, we obtain satisfactory results. The comparative results obtained show our method's effectiveness in boosting fairness while maintaining a high level of performance.

Our future work should focus data distribution over protected and redundant features and sort out the imbalance that can lead to bias.

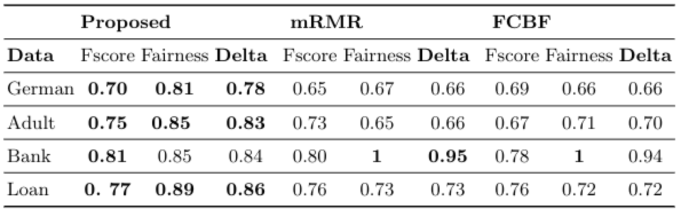
2

Appendix

I - EXPERIMENTAL RESULTS: CLASSIFIERS

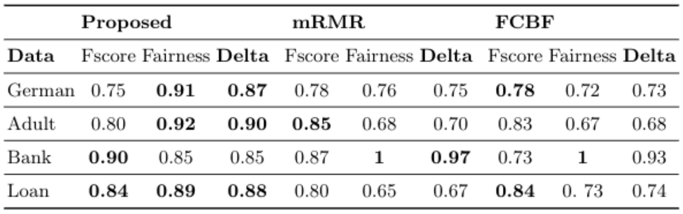
We report in the tables below, the results obtained with the two classifiers RandomForest and AdaBoost with the described datasets in table 1.

Table 3. Random Forest: Comparison based on Fscore, Fairness & Delta



	Proposed			**mRMR**			**FCBF**		
Data	Fscore	Fairness	**Delta**	Fscore	Fairness	**Delta**	Fscore	Fairness	**Delta**
German	**0.70**	**0.81**	**0.78**	0.65	0.67	0.66	0.69	0.66	0.66
Adult	**0.75**	**0.85**	**0.83**	0.73	0.65	0.66	0.67	0.71	0.70
Bank	**0.81**	0.85	0.84	0.80	**1**	**0.95**	0.78	**1**	0.94
Loan	**0. 77**	**0.89**	**0.86**	0.76	0.73	0.73	0.76	0.72	0.72

Table 4. AdaBoost: Comparison based on Fscore, Fairness & Delta



	Proposed			**mRMR**			**FCBF**		
Data	Fscore	Fairness	**Delta**	Fscore	Fairness	**Delta**	Fscore	Fairness	**Delta**
German	0.75	**0.91**	**0.87**	0.78	0.76	0.75	**0.78**	0.72	0.73
Adult	0.80	**0.92**	**0.90**	**0.85**	0.68	0.70	0.83	0.67	0.68
Bank	**0.90**	0.85	0.85	0.87	**1**	**0.97**	0.73	**1**	0.93
Loan	**0.84**	**0.89**	**0.88**	0.80	0.65	0.67	**0.84**	0. 73	0.74

II - EXPERIMENTAL RESULTS: EXECUTION TIME

We have compared the execution time of the proposed algorithm to the two other FS methods. The method named FCBF method is faster than our method over all the datasets. This is understandable since FCBF performs a single redundancy analysis using only one defined threshold while our method uses a threshold space to perform redundancy analysis. However, our method is faster than the mRMR method which in fact is a two stages process with two wrapper selectors.

III - SOURCE CODE AVAILABILITY

The full source code and data of our experiments is available under request on our github repository.

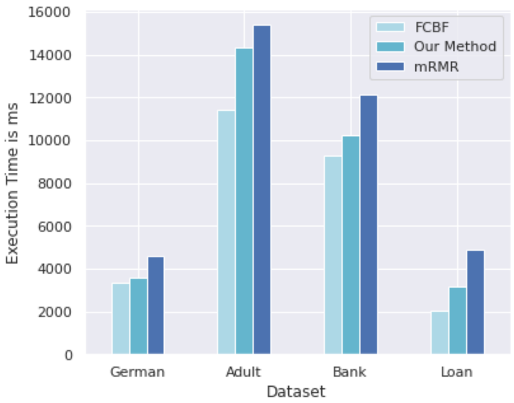


Fig. 2. Comparison of execution time of the three methods.

References

1. G. Thippa Reddy, M. Praveen Kumar Reddy, Kuruva Lakshmanna, Rajesh Kaluri, Dharmendra Singh Rajput, Gautam Srivastava, and Thar Baker. 2020. Analysis of Dimensionality Reduction Techniques on Big Data. IEEE Access 8 (2020), 54776–54788.
2. Alan Jović, Karla Brkić, and Nikola Bogunović. 2015. A review of feature selection methods with applications. In 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO).
3. Xiaofei He, Deng Cai, and Partha Niyogi. Laplacian score for feature selection. In NIPS, pages 507–514, 2005.
4. Alper Unler, Alper Murat, and Ratna Babu Chinnam : A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. Information Sciences, 181(20):4625–4641,2011.
5. Hanyang Peng and Yong Fan. A general framework for sparsity regularized feature selection via iteratively reweighted least square minimization. In AAAI, pages 2471–2477, 2017.
6. I. Guyon and A. Elisseeff. 2003. An Introduction to Variable and Feature Selection. J. Mach. Learn. Res. 3 (2003), 1157–1182.
7. Samuel Yeom, Anupam Datta, and Matt Fredrikson. 2018. Hunting for Discriminatory Proxies in Linear Regression Models. In Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18). Curran Associates Inc., Red Hook, NY, USA, 4573–4583.
8. Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications) (OJ L 201, 31.7.2002, p. 37).
9. Boli Fang, Miao Jiang, Pei-yi Cheng, Jerry Shen, and Yi Fang. 2020. Achieving Outcome Fairness in Machine Learning Models for Social Decision Problems. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, Christian Bessiere (Ed.). ijcai.org, 444–450.

10. Cyrus DiCiccio, Sriram Vasudevan, Kinjal Basu, Krishnaram Kenthapadi, and Deepak Agarwal. 2020. Evaluating Fairness Using Permutation Tests. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining (KDD '20). Association for Computing Machinery, New York, NY, USA, 1467–1477.
11. Kazuto Fukuchi, Satoshi Hara, and Takanori Maehara. 2020. Faking Fairness via Stealthily Biased Sampling. Proceedings of the AAAI Conference on Artificial Intelligence 34, 01 (Apr. 2020), 412–419.
12. Hanchuan Peng, Fuhui Long, and C. Ding. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min- redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 8 (2005)
13. Lei Yu and Huan Liu. 2004. Efficient Feature Selection via Analysis of Relevance and Redundancy. J. Mach. Learn. Res. 5 (2004), 1205–1224
14. Mei Wang, Xinrong Tao, and Fei Han. 2020. A New Method for Redundancy Analysis in Feature Selection. In 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence (ACAI 2020). Association for Computing Machinery, New York, NY, USA, Article 21, 5 pages.
15. Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. 2012. Fairness through Awareness. In Proceedings of the 3rd Innovations in Theoretical Computer Science Conference (ITCS '12). Association for Computing Machinery, New York, NY, USA, 214–226.
16. Shen Yan, Hsien-te Kao, and Emilio Ferrara. 2020. Fair Class Balancing: Enhancing Model Fairness without Observing Sensitive Attributes. In Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management (CIKM '20). Association for Computing Machinery, New York, NY, USA.
17. Lu Zhang, Yongkai Wu, and Xintao Wu. 2016. A causal framework for discovering and removing direct and indirect discrimination. CoRR abs/1611.07509 (2016). arXiv:1611.07509
18. Disi Ji, Padhraic Smyth, and Mark Steyvers. 2020. Can I Trust My Fairness Metric? Assessing Fairness with Unlabeled Data and Bayesian Inference. In Advances in Neural Information Processing Systems, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 18600–18612.
19. Laura Elena Raileanu and Kilian Stoffel. 2004. Theoretical comparison between the gini index and information gain criteria. Annals of Mathematics and Artificial Intelligence 41, 1 (2004), 77–93.
20. Powers, David MW. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." arXiv preprint arXiv:2010.16061 (2020).
21. Rachel E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Kr. Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan N. R., John T. Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush Varshney, and Yunfeng Zhang. 2018. AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias. ArXiv abs/1810.01943 (2018).
22. Mahesh Pal. 2005. Random forest classifier for remote sensing classification. International journal of remote sensing 26, 1 (2005), 217–222.
23. Schapire, Robert E. "Explaining adaboost." Empirical inference. Springer, Berlin, Heidelberg, 2013. 37-52.
24. Montgomery, Douglas C., Elizabeth A. Peck, and G. Geoffrey Vining. Introduction to linear regression analysis. John Wiley Sons, 2021.
25. Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository.