Experiments with different methods for class imbalance and bias in customers' credit risk assessment with little or no credit history

BJORN VAN BRAAK, University of Twente, The Netherlands

Customers defaulting on their loans payment present a big risk for credit providers. Therefore, credit issuers are not inclined to issue financial products or services to retail customers with little or no previous credit history due to the uncertainty about the customers creditworthiness. This results in limited access to financial resources and unfavorable terms for accessible resources. This research focus on improving the predictability of creditworthiness of these underserved groups with the help of ensemble models. Specifically, by applying various methods to handle the imbalance common in dataset used for credit risk assessment. Furthermore, this research attempts to improve fairness of ML model by limiting the group bias through the applications of disparate impact remover, a well known bias mitigation method. In applying three imbalance methods, the class weight was the most effective at handling class imbalance while still maintaining a reasonable performance. Additionally, disparate impact remover can have adverse effect on unprotected bias sensitive features.

Additional Key Words and Phrases: Credit Score, Machine Learning, Ensemble Learning, Credit Risk Assessment, Financial Inclusion

1 INTRODUCTION

The field of credit risk assessment has developed over time from simple statistical models to more complex machine learning (ML) models due the increasing amount of data available [Galindo and Tamayo 2000]. The research on credit risk assessment are related to different types of risks a lender is exposed to, for example credit risk assessment, non-performing assets prediction or fraud detection [Bhatore et al. 2020]. This research will solely focus on credit risk assessment. The risk assessment can be performed by assigning a credit score, assessing the probability of loan default or by classifying groups of lenders into bad and good lenders[Kritzinger and van Vuuren 2018]. There are already models on the market for credit risk assessment both using ML and traditional statistical models. The main source of information used in these models is related the credit, for example: credit history, outstanding credit and type of credit owed. This paper tries to assess the risk for short-term loans when limited credit history is available about a borrower in a retail setting. The limited availability can have several reason, but is often the case in developing countries or for poorer customers [The World Bank 2022]. Although ML models can improve the credit risk assessment they often lack interpretability [Paleologo et al. 2010]. Therefore, these models are susceptible to contain unnoticed bias since the answer cannot be verified.

In this research the use of tree-based ensemble models is explored which suffer from the lack of interpretability. Ensemble methods, however, are proven to have high predictive powers in credit risk assessment [Baesens et al. 2015; Brown and Mues 2012; Peng et al. 2011]. The reason for choosing tree-based ensemble models over neural networks, which have also shown good results, is that for the specific problem at hand the tree-based models are likely more suited. A proof of this suitability is presented by Grinsztajn et al. [2022] who shows that tree-based models work better on tabular data compared to neural networks. The reasoning behind this is tree-based algorithm are not effected by uninformative features and can better identify irregular patterns. The main predictor of creditworthiness is credit history, when little credit history about a person is available the dataset needs to include a lot of weakly correlated feature (sparse high-dimensional dataset) in order to make predictions. Therefore, in this case the datasets contains a lot of uninformative features and require the algorithm to identify irregular patterns since the correlation of individual predictors is weak.

There are multiple types of tree-based ensemble models, such as bagging [Ho 1995, 1998; Paleologo et al. 2010] and boosting [Chen and Guestrin 2016; Ke et al. 2017; Prokhorenkova et al. 2017]. In this paper both types of tree-based ensemble models are explored in predictive frameworks that aim to design an assessment tool that is robust for imbalance datasets. Thus, the main research question is: *How can the credit risk related to retail customers with little to no previous credit history be accurately assessed with a ML model while mitigating group bias?*

The following subquestions have been derived to help answer the main research question:

(1) How can the imbalance in the dataset be handled for a retail customer's ability to pay back loans when little credit history is available?
(2) How can group bias be mitigated for ML models predicting credit risk using datasets including additional information next to credit history?

The 1st sub-research question revolves around handling imbalance in ML. The methods used to answer this question will focus on classify the minority class correctly, as this is often the class of interest. This research compares multiple method to correct for the imbalance in the dataset, namely oversampling the minority class, undersampling the majority class and applying class weight in loss function of the ML model. Additionally, the 2nd sub-research question is about finding ways to mitigate bias when the model lacks interpretability. A naive approach would be to drop sensitive feature. This approach does not always get rid of the underlying bias, since the sensitive features can be correlated with other features. Therefore, this research will also experiment with applying disparate impact remover, a preprocessing technique, to the dataset.

The paper will be structured by first exploring related work in section 2. Secondly, the classification algorithms and disparate impact remover used in this research will be explained in the method

TScIT 39, July 7, 2022, Enschede, The Netherlands

section 3. Third, the experimental set-up will be described in section 4. Fourth, the results of the experiment in section 5. Last, the sections 6 and 7 will be the discussion and conclusion.

2 RELATED WORK

This section is divided into three parts. The first part reviews the literature on approaches to common problems in credit risk datasets, namely class imbalance, outliers, missing data and data sparsity. The second part reviews common ML models used for credit risk assessment. Lastly, the third part reviews possible ways to mitigate group bias in the ML development.

2.1 Common problems for credit risk datasets

Common problems for credit risk datasets are class imbalance, outliers, missing data and sparsity of the data. Consequently, solutions to these problems have been proposed in the literature [Blagus and Lusa 2013; Branco et al. 2015; Haixiang et al. 2017; Ishaq et al. 2021; Melo Junior et al. 2020]. The proposed solutions range from applying different data transformation to adapting the ML algorithms. Therefore, it is important to note that a solution can be applied in different phases of the ML development and deployment.

2.1.1 Imbalance. Class imbalance is a problem encountered in a lot of domains. Standard estimators such as logistic regression (logR), support vector machine (SVM) and Decision Tree (DT) tend to provide sub-optimal classification results when applied to imbalanced datasets [Haixiang et al. 2017]. In other words, when applied to imbalanced datasets these models will provide a good coverage of the majority class while the minority class, which is the class of interest, is misclassified. This problem is prevalent in credit score datasets [Melo Junior et al. 2020]. Branco et al. [2015] suggest several strategies for handling imbalanced datasets, namely data pre-processing, special-purpose learning methods, prediction post-processing and hybrid methods. In this study, the approach of oversampling techniques Synthetic Minority Over-Sampling Technique (SMOTE) [Chawla et al. 2002], which falls into the data preprocessing strategy for handling imbalanced datasets, is chosen for it has been shown to be effective on credit scoring datasets[Crone and Finlay 2012; Marqués et al. 2013]. The original paper of SMOTE also suggest it be used in combination with an undersampling technique. Furthermore, it was tested out on 100-500% of the minority class size. As an alternative to handling imbalance in the data preprocessing stage is during the model training. In this paper applying class weight which influences the value of the loss function during the training of ML model is chosen.

López et al. [2013] also remarks that the reasons for the significant loss of performance caused by the imbalance can be different based on the data characteristics. Additionally, it is also shown that different approaches of combatting imbalance are better depending on the model used. For example, for classifiers like C4.5 and kNN the use of an ensemble of learners been proven to be effective. Lopez et al. suggests that boosting helps in the case there are a lot of small disjuncts. In contrast, for SVM SMOTE has been effective. Furthermore, SMOTE does not always improve the performance of a an algorithm especially on high-dimensional data, but it has been shown to do particularly well with DT based algorithms [Ishaq et al. 2021]. Variable selection should be performed, and should be performed before SMOTE [Blagus and Lusa 2013]. The reason for this is that the samples are no longer independent after applying SMOTE, which is an assumption made is a lot of variable selection methods.

2.1.2 Other issues.

- **Outliers.** Tree-based algorithms are considered to be resistant to outliers [Kuhn and Johnson 2013]. However, as Beyan and Fisher [2015] points out the imbalance in the dataset can cause the model to mistakenly identify noise as a minority example.
- **Missing values.** The handling of missing values can be done in three ways: by removing the data, by imputing missing values or by the algorithm of the model (especially useful in tree-based models)[Kuhn and Johnson 2013]. The first way can either be done for a specific predictor variable or record of data. Secondly, a strategy can be selected for assigning values to missing fields; the strategy selection is based on domain knowledge of the problem. As a last way to handle missing values the specific model can have ways to deal with missing values during training (for example in the C4.5 algorithm [Saar-Tsechansky and Provost 2007]). In any case, the missing values should be handled on a per case basis.
- **Sparse data.** Sparse data is data with high dimensionality where few fields contain meaningful data for the model to train on. This is typically the case when using categorical features, because of the way they are encoded. The sparsity of data increases the space and time complexity of the model training. Furthermore, models trained on high dimensional sparse data are prone to overfitting to the train data. Therefore, general-purpose ML models cannot handle sparsity well [Chen et al. 2019]. Laborda and Ryoo [2021] suggest multiple feature selection techniques to reduce the overfitting caused by high dimensionality of the dataset.

2.2 ML algorithms

In general, the most common set of ML methods used for credit risk assessment are Logistic Regression, SVM, Artificial Neural Network (ANN) and DT [Abellán and Castellano 2017; Brown and Mues 2012; Li and Chen 2020; Tomczak and Zie˛ba 2015; Wang et al. 2011]. The three algorithms considered in this research have DT as baseline for the model and have produced high accuracy models for credit risk assessment [Chen et al. 2019; Li et al. 2020]

2.2.1 Decision and Gradient Boosted Trees. Decision trees algorithms are generally not used in credit scoring applications as the sole classifier because they are very sensitive to noise and prone to overfitting. However, the decision tree has as an advantage that it is easily interpretable by humans, no underlying assumptions are made about the underlying distributions and have similar performance to other widely-used techniques used in credit scoring [Marqués et al. 2012]. Consequently, ensemble learning with DT as a baseline classifier can increase performance significantly. In the literature, ensemble learning has been applied to credit risk

assessment in different ways [Florez-Lopez and Ramon-Jeronimo 2015; Paleologo et al. 2010].

In this research, a focus will be on ensemble learning using the meta-algorithm of boosting. The use of the AdaBoost (ADA) algorithm [Freund and Schapire 1997] has been studied extensively in credit risk assessment [Abellán and Castellano 2017; Tomczak and Zie¸ba 2015; Tyagi 2022; Wang et al. 2011]. Additionally, two commonly used gradient tree boosting algorithms XGBoost [Chen and Guestrin 2016] and LightGBM [Ke et al. 2017] developed in recent years have also been studied in the context of credit scoring [Li and Chen 2020; Soares De Melo Junior et al. 2019]. Furthermore, an alternative that deals better with the sparsity called DeepGBM has also been researched in the field of credit risk assessment [Chen et al. 2019]. Other interesting studies that propose credit risk assessment are related to the use of hybrid models [Machado and Karray 2022] and two-level ensemble methods [Marqués et al. 2012].

2.3 Bias mitigation in ML

As mentioned in the introduction dropping bias sensitive features from the dataset does not ensure that the underlying group bias is removed from the dataset. The model can be trained on data that already contains human prejudice. Furthermore, the existence of indirect discrimination found in non-sensitive features can impose the same disproportionate burdens on minority or disadvantaged groups [Pedreshi et al. 2008]. Fairness in AI has been a topic of exceeding interest in recent years. There have been several bias mitigation strategies proposed in the literature. For example, Hardt et al. [2016] suggest a post-hoc correction by first training on biased predictors and afterwards adjusting the learned predictor to remove discrimination. In contrast, Woodworth et al. [2017] suggests an approach to learning non-discriminatory predictors from the start. In addition, the authors point out that a predictor can be implicitly discriminatory and training a model on such discriminatory predictors is sub-optimal. Training a model to be more fair will lead in many scenarios significantly decrease the result depending on the bias mitigation method used [Chen et al. 2022]. Furthermore, The effectiveness of a bias mitigation method is heavily dependent on the scenario [Menon and Williamson 2017]. However, a trade-off between accuracy and fairness of a model was observed regardless of the bias mitigation method used. Many bias mitigation methods have been implemented in the toolkit of IBM AI Fairness 360 (AIF360) [Mehrabi et al. 2021], which is also used in this research.

3 METHODOLOGY

Three different tree-based ensemble algorithm were used, namely Random Forest, LightGBM and CatBoost. They were selected based on their successful applications to assess credit risk in the past [Li and Chen 2020; Ponsam et al. 2021]. The use of an ensemble model often gives better performance then the best single classifier, but does not guarantee better performance. As a last remark, the use of different type of ensemble learning boosting, bagging and stacking have been explored previously in the literature[Li and Chen 2020; Wang et al. 2011]. However, to the best of our knowledge, there hasn't been any studies exploring the use of ensemble models to assess credit risk for retail customers with no (or little) credit history, while also handling bias mitigation.

3.1 Random Forest

The Random Forest consist of multiple base learners learned in parallel based on a sample subset of the entire dataset draw with replacement (bootstrap sampling) [Zhou 2012]. A Random Forest algorithm uses all decision trees and comes to decision by the voting, in the case of classifiers, of all DT (aggregation). The difference with other bootstrap aggregation (bagging) algorithm is that Random Forest only uses a subset of the features for each boostrap sample. This can also be seen in Algorithm 1, based on Breiman [2001].

Algorithm 1 The random decision tree algorithm used in RF as described in Zhou [2012]

input: Data set $D = (x_1, y_1), (x_2, y_2), ..., (x_m, y_m)$;
Feature subset size K
procedure buildRandomDecisionTree(D, K)
$N \leftarrow$ create a tree node
if all instance in the same class **then**
return N
end if
$F \leftarrow$ the set of features that can be split further
if F is empty **then**
return N
end if
$\tilde{F} \leftarrow$ select K features from F randomly
$N.f \leftarrow$ the feature which has the best split point in $\tilde{F}$
$N.p \leftarrow$ the best split point on N.f
$D_l \leftarrow$ subset of D with values on $N.f$ smaller than $N.p$
$D_r \leftarrow$ subset of D with values on $N.f$ no smaller than $N.p$
$N_l \leftarrow$ buildRandomDecisionTree(D_l, K)
$N_r \leftarrow$ buildRandomDecisionTree(D_r, K)
end procedure

3.2 LightGBM

LightGBM is a gradient boosting algorithm. A gradient boosting algorithm tries to constructs a single strong learner by sequential learning weak learners [Ke et al. 2017]. LightGBM grows the trees leaf-wise, instead of the more common depth-wise. This allows the individual trees to grow more complex. It is much more efficient then XGBoost by requiring less memory and computational resources [Ke et al. 2017].

3.3 CatBoost

CatBoost is also a gradient boosting algorithm. In contrast to LightGBM, it uses a symmetric tree with each split in the same level of the tree having the same split condition [Yandex LLC nd]. It also applies a new algorithm called ordered boosting to deal with the problem of prediction shift common in small dataset [Yandex LLC nd]. Furthermore, it has built in capabilities for handling categorical features.

3.4 Disparate impact remover

Disparate impact remover is a technique to mitigate group bias [Feldman et al. 2015]. It does this by modifying the original dataset to ensure group-based fairness for protected feature(s) while attempting to preserve the ability to classify the label classes. Suppose there exist a dataset D which consist of the protected feature X, the remaining features Y and the class outcome C. The main idea of the disparate impact remover is to try to modify Y, while preserving relative per-feature ordering, in such a way that given Y and C the feature X cannot be predicted [Feldman et al. 2015]. The disparate impact remover can be extended to multiple protected features by operating over the joint distribution. In the experiment of Feldman et al. [2015] the performance loss did not compound on the number of protected attributes included, rather the performance loss dependent on the protected feature with the highest performance loss. A repair level can be specified to define how much the group-based bias need to be corrected for.

4 EXPERIMENTAL SET-UP

Three experiment were conducted. Two related to applying methods to handle the imbalance in the dataset. The other one is about trying to mitigate bias with the disparate impact.

4.1 Dataset

The dataset can be found on Kaggle [Anna Montoya 2018]. The research uses a dataset containing information about the applicant, details about the loan application and credit history. The credit history contained in the dataset are the amount of previous enquiries made to the Credit Bureau about the applicant and the credit score from 3 other sources. In around 12% of the case the credit history is missing or incomplete. The application dataset suffers from high imbalance with 24, 825 positive targets and 282, 686 negative targets ($\sim$ 8% of total records).

4.2 Preprocessing

Both the k-nearest neighbors (kNN), used by SMOTE, and the tree-based algorithms used are non-parametric models, meaning no assumption about the underlying data distribution is made. The baseline assumption was made that the data was independent and identically distributed (i.i.d.) in order to perform statistical tests. In step 1, a correlation analysis was performed since collinearity between predictors can reduce performance and stability of the model [Kuhn and Johnson 2013]. In particular, the kNN algorithm used by SMOTE-NC is vulnerable to collinear predictors in the train set. The numerical predictors were tested with Pearson's correlation coefficient (r) and the categorical predictors with Cramér's V (ϕ_c); the correlation threshold was set at $\tau = |0.9|$. In total, 31 numerical predictors and 3 categorical predictors were dropped. The specific algorithm followed for removing predictors is based on Kuhn and Johnson [2013] and stated in Algorithm 2.

The experiment uses a 5x2 cross-validation in order to estimate the models performance and uses the same splits across all experiments. The preprocessing steps used on each iteration of the 5x2 cross-validation were:

Algorithm 2 Algorithm to drop features based on Kuhn and Johnson [2013]

$X_{corr} = X.corr()$ ▷ correlation matrix
$X_{new} = X$
$num_{cols} = len(X_{new}.columns)$
while $numOutsideThreshold(X_{corr}, \tau) > num_{cols}$ **do**
  $A = findHighestCorrelatedPredictor(X_{corr})$
  $B = findSecondHighestCorrelatedPredictor(X_{corr})$
  $\overline{A} = findAverageAbsCorrelation(X_{corr}, A)$
  $\overline{B} = findAverageAbsCorrelation(X_{corr}, B)$
  $predictor_{drop} = undefined$
  if $\overline{A} > \overline{B}$ **then**
    $predictor_{drop} = A$
  else
    $predictor_{drop} = B$
  end if
  $X_{new} = X_{new}.drop(predictor_{drop})$
  $X_{corr} = X_{new}.corr()$
  $num_{cols} = len(X_{new}.columns)$
end while

(1) the missing values are imputed by using the 'median' strategy for numerical and the 'most frequent' for categorical data
(2) the ordinal values are encoded
(3) the numeric and ordinal values are scaled by applying min-max scaling
(4) SMOTE-NC was used to generated new synthetic samples
(5) (experiment 2 only) The data was randomly undersampled
(6) The nominal values were OneHotEncoded and for the interval and ratio features the interaction between features were added

All numeric values were tested for normality with confidence level 95% and none passed the test, therefore the type of scaling applied was min-max scaling [Gopal et al. 2015]. The ordinal values were chosen to be treated as numeric values. A modified version of SMOTE was used that also handles nominal values called SMOTE-NC, which was also suggested for these situations by the authors of the original method [Chawla et al. 2002] and the exact implementation used can be found in [Lemaitre et al. 2017]. As discussed in section 2.3, several approaches can be taken to prevent group bias . In this study, the approach of disparate impact remover will be the focus. The approach will be evaluated with the performance and fairness metrics, as there is a fairness-performance trade-off [Chen et al. 2022]. The step of removing disparate impact was placed before the cross-validation mentioned in this section. This choice was made due to limitations in the implementations used.

4.3 Imbalance Parameters

This study aims to address the issue of class imbalance within the target variable of the dataset under investigation in experiment 1 and 2. In order to tackle the class imbalance the following three approaches were taken: oversampling with SMOTE, random undersampling and applying class weight. The specific independent variables varied in the experiment are explained in this section. The

class weight is based on Equation 1

$$w_{class} = \frac{n_{total}}{2 \times n_{class}} \tag{1}$$

The percentage of oversampling done is based on a percentage relative to the original imbalance in the dataset. For both oversampling as undersampling the desired end resampling ratio α_{rs} is given by Equation 2.

$$\alpha_{rs} = \frac{n_{rm}}{n_{rM}}, \text{ where} \tag{2}$$
$$n_{rm} = \text{number of minority class after resampling}$$
$$n_{rM} = \text{number of majroity class after resampling}$$

4.4 Hyperparameter tuning

The experiment does not tune any of the hyperparameters of the ML algorithms used. However, the number of estimators fitted set to 100 to limit computational expenses. Additionally, the default setting of CatBoost were computationally much more expensive and therefore minimum data in leaf was set to 20 instead of 1 (the same as LightGBM default) to limit the computational expense. The rest of the hyperparameter were left to their default values [Microsoft Corporation nd; scikit-learn developers nd; Yandex LLC nd]

4.5 Evaluation metrics

4.5.1 Performance metrics. The first metric is accuracy, given in Equation 3. As concluded by multiple studies using only accuracy (Equation 3) alone is insufficient for evaluating model performance on imbalanced dataset [García et al. 2014; López et al. 2013]. However, the reason for still including accuracy is that applying methods to correct for the imbalance in the dataset tends to reduce the overall performance of the model.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Therefore, in addition to accuracy two other performance metrics were selected for model evaluation that are more suited for imbalanced datasets, namely the weighted recall (Equation 5) and F-score (Equation 8).

The recall measures the ability of the model to correctly identify instances of the class. In general, the recall per label class i is defined by Equation 4

$$\text{recall}_i = \frac{p_i}{n_i}, \text{ where} \tag{4}$$
$$p_i = \text{correctly predicted data items in class i}$$
$$n_i = \text{actual number of data items in class i}$$

The weighted recall for the amount of label classes C is presented in Equation 5. The importance of correctly identify a class can be different per class. For example in the specific case of credit risk assessment, it is more important to correctly classify a person as a bad lender then to correctly classify someone as a good lender since the misclassification costs are higher for the former class. As proposed by Gupta et al. [2020] applying a specific weight to the class recall can represent this difference in importance.

$$\text{weighted recall} = \sum_{i=1}^{C} w_i \times \text{recall}_i, \text{ given} \tag{5}$$
$$0 \leq w_i \leq 1 \text{ and } \sum_{i=1}^{C} w_i = 1$$

In the case of a binary classification, such as is the case in this study, the recall for the positive class is given by true positive rate and the recall of the negative class is the true negative rate. Therefore, the given Equation 5 can be rewritten as Equation 6

$$\text{weighted recall} = w_1 \times TPr + w_2 \times TNr, \text{ where} \tag{6}$$
$$TPr = \text{recall minority} = \text{sensitivity} = \frac{TP}{TP + FN}$$
$$TNr = \text{recall majority} = \text{specificity} = \frac{TN}{TN + FP}$$

In this research, the weights were chosen to be inversely proportional to the number of samples present in the dataset. The formula for the weights is defined in equation 7

$$w_i = \frac{1}{f_i \times \sum_{j=1}^{C} f_j^{-1}}, \text{ where} \tag{7}$$
$$f_i = \text{frequency of class } i$$

The weighted recall metric (Equation 6) with the weights defined in Equation 7 will punish the misclassification of sample based on the severity of class imbalance in the dataset.

The F-score is used to measure the recall and precision of a model. Equation 8 is a generalization of the f-score formula [Nazarathy and Klok 2021]. For evaluation, β needs to be set based on the relative weight given to the sensitivity (the ability to recognize all positive cases) and the precision (the ability to correctly identify a case as positive). A higher β value gives higher importance to the ability to recall all positive cases. The β for this experiment is set to 1.

$$F_\beta = (1 + \beta^2) \times \frac{precision \times sensitivity}{\beta^2 \times precision + sensitivity}, \text{ where} \tag{8}$$
$$sensitivity = \frac{TP}{TP + FN}$$
$$precision = \frac{TP}{TP + FP}$$

Both evaluation metrics stated in Equation 5 and Equation 8 can be adjusted based on the misclassification cost for the specific problem.

4.5.2 Bias metrics. The two fairness metrics chosen are the Statistical Parity Difference (SPD) and the Equality of Opportunity Difference (EOD). The reason for not choosing the disparate impact metric (which is also commonly used in the industry) is that the metrics EOD and SPD have the same scale ($[-1, 1]$). In addition, the absolute value of both metrics are taken. The reason for this is two-fold: it makes for easier comparison as a discrepancy in both direction is equally 'worse' and the metrics become an optimization

problem to minimize.

$$a_{unpriv} = \{a_0, b_0, c_0, ...\}$$
$$a_{priv} = \{a_1, b_1, c_1, ...\}$$
$$a_{unpriv} \cap a_{priv} = \emptyset$$

In both Equation 9 and Equation 10, P defines the probability.

$$SPD = P(\hat{Y} = 1|A = a_{unpriv}) - P(\hat{Y}|A = a_{priv}) \tag{9}$$

$$EOD = P(\hat{Y} = 1|y = 1, A = a_{unpriv}) - P(\hat{Y} = 1|y = 1, A = a_{priv}) \tag{10}$$

As established in previous literature typically there exist a trade-off between fairness and performance of a model [Chen et al. 2022; Menon and Williamson 2017]. Therefore, the bias mitigation method is also evaluated against the three performance metrics previously defined.

5 RESULTS

The results will be discussed in four sections. In the first two sections the results of experiment 1 and 2 will be discussed. Thereafter, in section 5.3 the difference in performance and behaviour of the imbalance methods will be discussed. Lastly, the results of experiment 3 is discussed in section 5.4.

5.1 Experiment 1: SMOTE-NC and Class Weight

The results of varying oversampling ratio (from 0 to 400%) and class weight (from $\{0 : 1, 1 : 1\}$ to $\{0 : 0.5, 1 : 6.2\}$ based on Equation 1 for each class) can be found in Figure 1. The performance behaviour in the oversample ratio and class weight gave similar reaction for all ML algorithms, only the magnitude differs. LightGBM, for example, was less sensitive to change in the oversample ratio. CatBoost was the most sensitive to changes in both the oversample and class weight. For example, CatBoost achieved a significantly higher ratio of 0.87 for the weighted recall, but this did come at significant drop in accuracy of 0.30 when strongest class weight ($\{0 : 0.5, 1 : 6.2\}$) was applied in combination with highest oversampling (0.439).

5.2 Experiment 2: SMOTE-NC and Random Undersampling

The results of varying oversampling ratio (from 0 to 400%) and undersampling ratio (from 0.439, which is the ratio achieved with 400% oversampling, to 1 α_{rs}) can be found in figure 2. The behaviour of the ML algorithm across this experiment were very similar. Interestingly, simple random undersampling performs better for the minority class then combined with SMOTE. Similar results have also been found in other research. Blagus and Lusa [2010] offers as a possible explanation that simple undersampling is more effective in removing the gap between the class-specific accuracies for high-dimensional data. The problem of misclassification comes in region where there is overlap between the minority and majority. Algorithm tend to then prefer the majority class. In further research Blagus and Lusa [2013] offered an additional reason why SMOTE does not perform well on high-dimensional data is that high-dimensional data often includes a lot of noise. This noise is then amplified by the SMOTE algorithm.

5.3 Comparing class weight to random undersampling

Regardless of the imbalance method applied, trying to account for the imbalance of the dataset comes with an overall performance loss. This might indicate that the ML algorithms do not become better at predicting the minority class but instead just predicting the minority class more often. The third metric F1-score support this argument. The gain in sensitivity (recall of the minority) to the class as seen by the increase in weighted recall for both experiment does not lead to a similar increase in F1-score. In other words, the gain in sensitivity is offset by a lower precision for the minority class. Furthermore, the value of the F1-score is low, which indicates that the features to predict the class is not good at distinguishing the minority class from the majority class.

Furthermore, when combining class weight and SMOTE, as in experiment 2, the effect of forcing the model to predict the minority class more often is amplified. This is because the class weight is based on the original imbalance (so the imbalance prior to applying SMOTE) and the class weight will also be applied on the new minority samples generated by SMOTE. The use of both more samples as well heavily penalize incorrect mistakes of the minority class value makes the model focus mostly on the minority class. In contrast, the amount of undersampling done to reach the specified ratio is dependent on the oversample ratio. Thus, it does not suffer from the same problem when combining oversampling and undersampling together.

In conclusion, in both experiment SMOTE had limited impact on the performance. This is likely due to the high-dimensionality of the data. The combination of class weight and SMOTE should be applied with care, since the effects of noise in the minority class is amplified by both SMOTE and class weight.

5.4 Experiment 3: Disparate Impact Remover

As a last experiment the best estimators from Experiment 1 and 2 for each model are corrected for bias. The best estimator per experiment per ML algorithm is defined as the result with the highest weighted recall (Equation 6) where the accuracy (Equation 3) is still above 0.8. In the dataset three bias sensitive feature have been identified, namely gender, education level and region rating. The experiment will vary the repair level between 0 and 1 to correct for the sensitive feature gender. Although it is theoretically possible to correct for more sensitive features with the disparate impact remover, the implementation of AIF360 does not support this. Therefore, the experiment also looks at what happens to unprotected bias sensitive features. The fairness metrics are calculate for each (un)protected bias sensitive feature based on Equation 9 and 10. The results were similar in both cases and are depicted in Figure 3 for Experiment 1 and in Figure 4 for Experiment 2. The results for each model is depicted in one column. In each column there are 3 graphs, the first depicts the fairness metrics on the protected feature, the second the fairness metrics on the unprotected feature and the last the performance metrics. Note that in all figures y-axis is the difference with the baseline (repair level 0). Based on previous literature expected is that the performance metrics will go down with respect to the baseline, while the disparate impact will be removed for the protected feature and therefore the difference with the baseline will

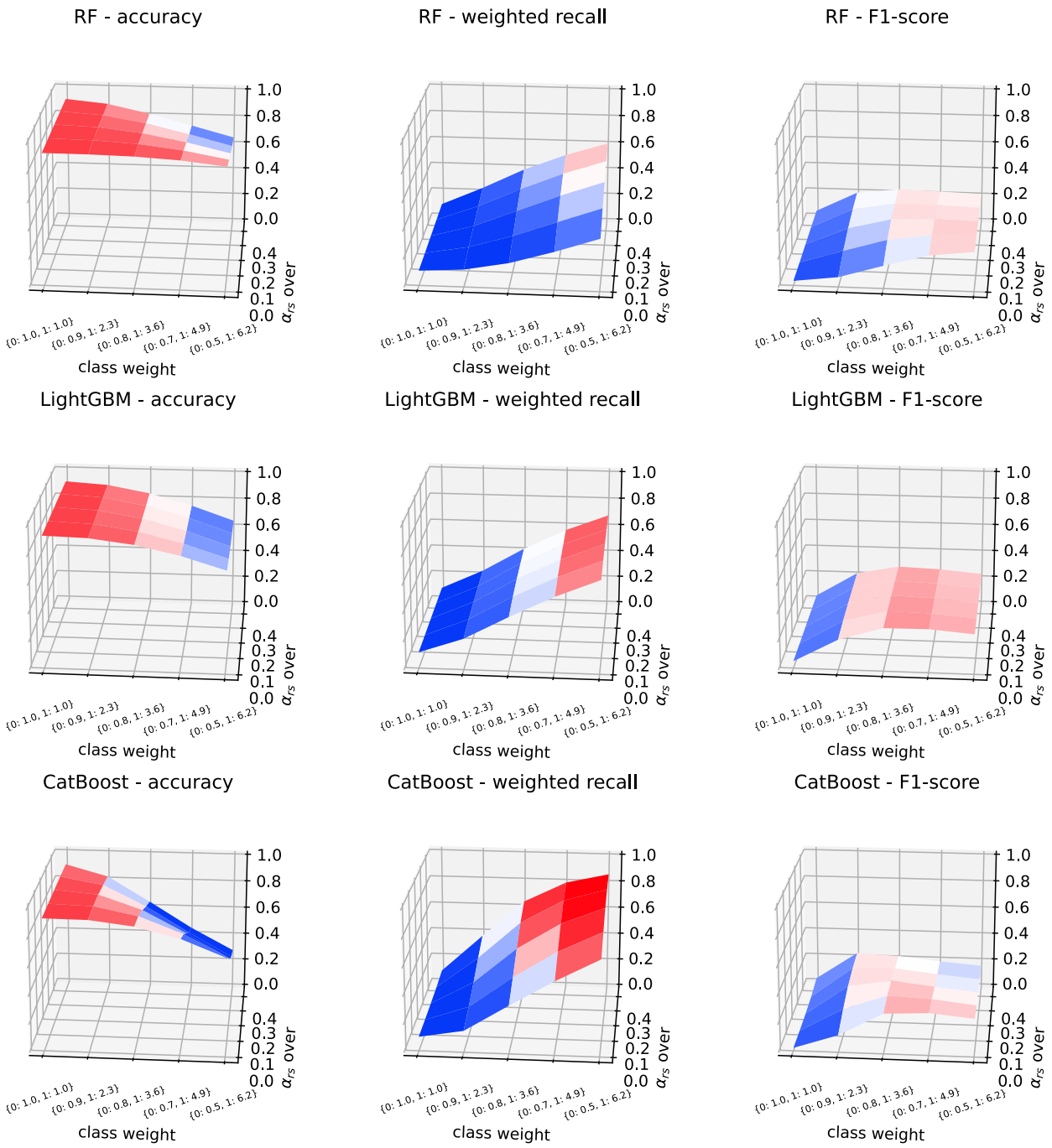

Fig. 1. Results with SMOTE-NC and class weight

go up for the protected feature. In general, the behaviour of the bias mitigation is independent of the exact method used for imbalance and dependent on the ML algorithm. This is expected because the importance of feature can differ significantly for each ML algorithm. Interestingly, the *SPD* and *EOD* of the protected attribute is not significantly improved, while corresponding values for the unprotected attribute (between −0.05 and 0) and the performance metric (between −0.02 and 0) are worse. The cause of ineffectiveness for removing disparate impact for the protected attribute is most likely the placement of the disparate impact remover in the machine learning

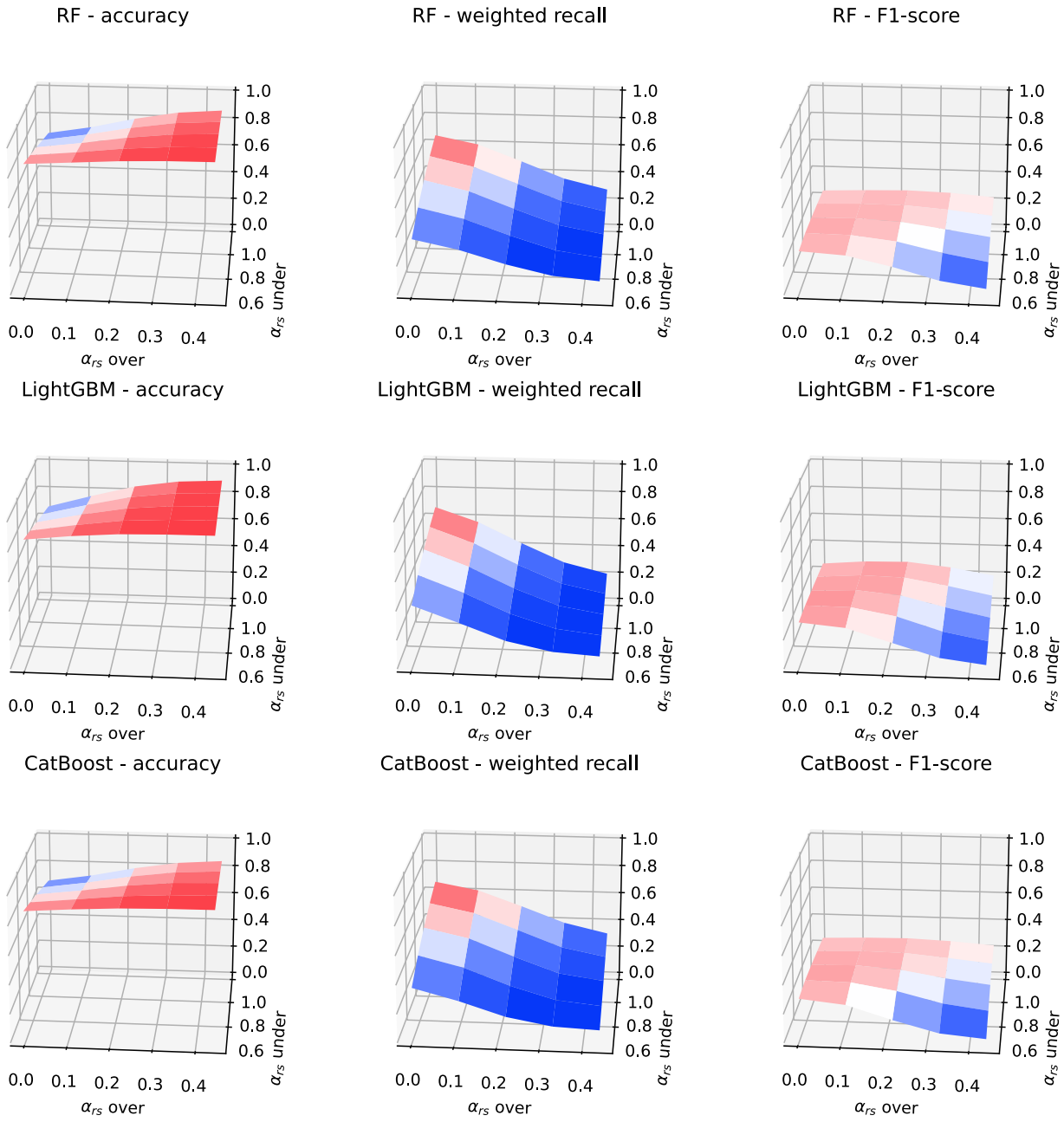

Fig. 2. Results with SMOTE-NC and random undersampling

pipeline before SMOTE and random undersampling. Both imbalance methods can change the probability density functions of the protected feature classes different, which cancel the effect to make the distribution of the protected feature classes similar. One notable exception is the weighted recall in Figure 3 which significantly improved while also dropping significantly in accuracy. This is similar to the case for experiment 1 and experiment 2 where in the case of a significant loss of accuracy due to method used to handle imbalance the model can start randomly predicting the minority class more often (based on the f-score not increasing). In conclusion, having a disparate impact remover before the preprocessing steps altering the distribution of the features in the model can be detrimental for the effectiveness of removing disparate impact. Additionally, the disparate impact remover will generally cause a small performance loss while also hurting the fairness for unprotected bias features.

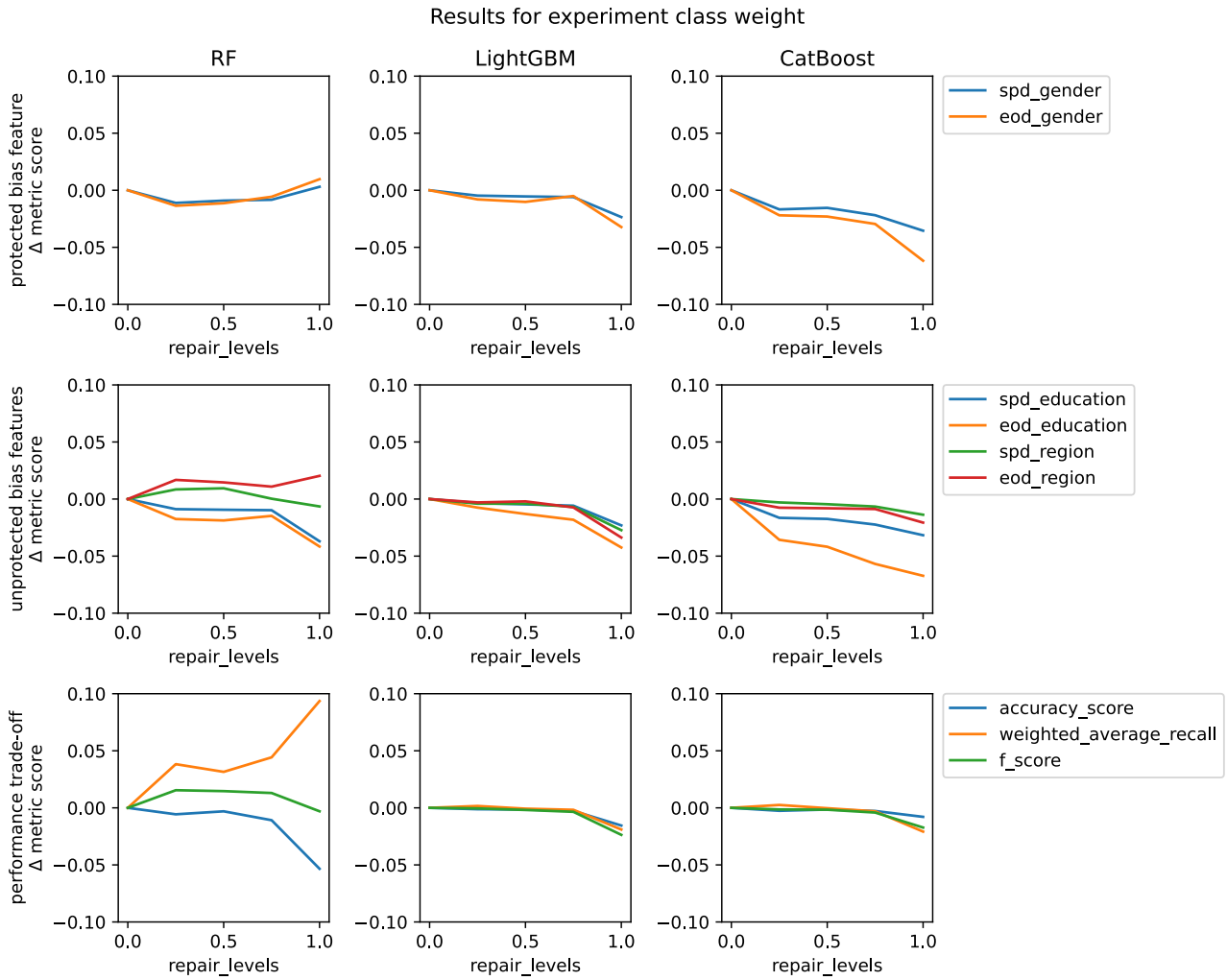

Fig. 3. Experiment 3 for the best estimator for each ML algorithm of Experiment 1

6 DISCUSSION

The hyperparameters were left at their default and only changed slightly for performance reason. The assumption is made that the algorithm have similar behaviour when tuned for optimal performance. The experiment set-up limited the way in which the algorithm could be used. For example, in theory all underlying tree algorithm used by ensemble models can handle categorical features internally. However, due to the specific implementation used for the RF model not supporting this the categorical features were One-HotEncoded. This will both increase the number of split and can have an impact on performance for all models. Another example is that LightGBM grows the trees leaf-wise and CatBoost grows the trees depth-wise. As also stated in LightGBM documentation [Microsoft Corporation 2023], for a fixed number of leaves in a tree the leaf-wise tree typically grows much deeper. Therefore, the between-model comparison is of limited significant when using the models in real world context. As a last example, the value for number of estimators is chosen to be 100 for each algorithm. However, the way a boosting algorithm and bagging algorithm use the estimator is different. Normally the number of estimators is bigger for a boosting algorithm. The dataset only contains records of accepted application. Therefore, there exist sample bias in the dataset which cannot be mitigated during the experimental set-up. Consequently, the ML model trained may include bias from the previous procedure for reviewing applications. Simple imputing strategies were used to handle missing values. This might limit the effectiveness of the kNN algorithm used by SMOTE, since this is dependent on proximity. This can also influence fairness. For example, if the application process was previously reviewed by a human reviewer personal bias and stereotype can lead to application being rejected and therefore not showing. It is unbeknown to the research how the samples included in the dataset were acquire. Another important point to highlight is that when using class weight for imbalanced dataset

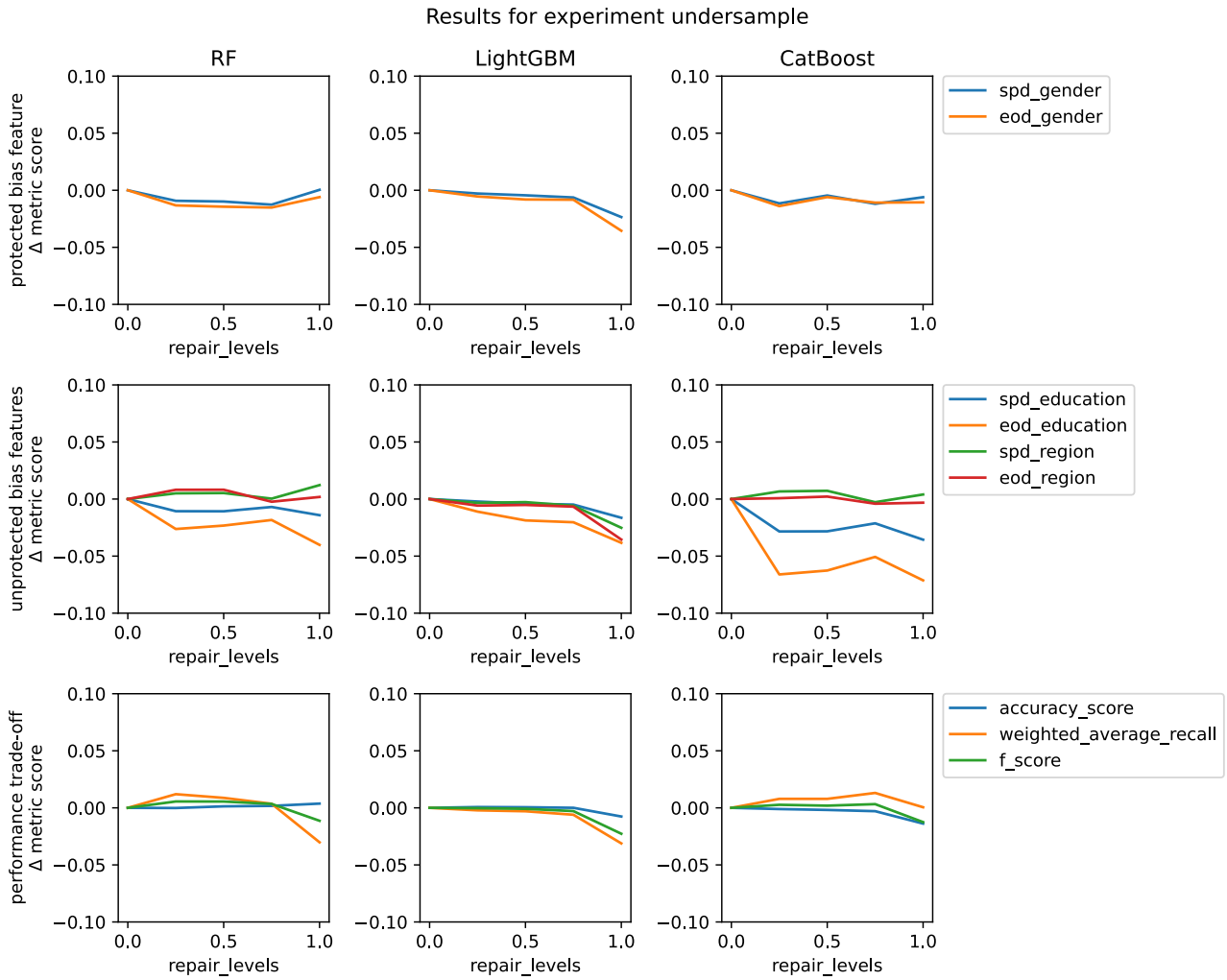

Fig. 4. Experiment 3 for the best estimators for each ML algorithm of Experiment 2

and the algorithm uses sample weight (for example in the case of ADA) the class weight will be multiplied by the sample weight in the loss function. Therefore, models both using class weight and sample weight might become unstable. The disparate impact remover is placed outside the preprocessing-pipeline described in section 4.2 and will therefore run only once per repair level per model, not on each fold. This gives a data leak between the train and test set.

7 CONCLUSIONS

To answer the 1st sub-research question, as also concluded by other researchers [Blagus and Lusa 2013; Ishaq et al. 2021] SMOTE has limited positive effect on the overall performance, in some cases even negative, if the high-dimensional dataset consist of weakly correlated predictors and tends to amplify noise. In the case of experiment 1, the high class weight for the minority class combined with SMOTE caused the loss function to be very sensitive to the minority class, while decreasing performance significantly. The magnitude of change is also heavily dependent on the model used for experiment 1. This is logically since the loss function is modified and the impact is therefore dependent on the ML algorithm. Therefore, it is recommended to not use SMOTE for high-dimensional dataset with weakly correlating predictors. Furthermore, applying a balanced class weight will benefit is generally better than random under sampling the data since it obtains similar results for weighted recall while maintaining a better accuracy. However, random undersampling provided for way more consistent results across all models.

In regards to the 2nd sub-research question, the disparate impact remover should be placed in the preprocessing steps with care, preferably after any oversampling or undersampling. The disparate impact remover caused a small performance loss, but more importantly increased the group bias for the unprotected bias features. Consequently, not recognizing a feature sensitive to group bias,

which is easily possible in cases of indirect discrimination, can lead to increased group bias when applying the disparate impact remover method.

7.1 Further work

The way class weight was implemented in combination with SMOTE caused the loss function to overly focus on the minority class. It might be interest to keep the class weight for the minority class the same (or close to 1), while lowering the class weight for the majority class. This has a similar effect to undersampling without throwing away potentially important data prior to the ML algorithm step. As an alternative approach smarter ways of undersampling can also be explored. For example, in Batista et al. [2004] the tested TOMEK + SMOTE as on of the best for tackling the imbalance. The difference between models was only visible in experiment 1. Therefore, it is also a possibility to study the behaviour of applying class weight for different types of ML methods.

ACKNOWLEDGMENTS

The author would like to thank Marcos Machado for supervising the research project.

REFERENCES

Joaquín Abellán and Javier G. Castellano. 2017. A comparative study on base classifiers in ensemble methods for credit scoring. *Expert Systems with Applications* 73 (5 2017), 1–10. https://doi.org/10.1016/J.ESWA.2016.12.020

KirillOdintsov Martin Kotek Anna Montoya, inversion. 2018. Home Credit Default Risk. https://kaggle.com/competitions/home-credit-default-risk

B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen. 2015. Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society* 247, 1 (2015), 124–136. https://doi.org/10.1016/j.ejor.2015.05.030

Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter* 6, 1 (6 2004), 20–29. https://doi.org/10.1145/1007730.1007735

Cigdem Beyan and Robert Fisher. 2015. Classifying imbalanced data sets using similarity based hierarchical decomposition. *Pattern Recognition* 48, 5 (5 2015), 1653–1672. https://doi.org/10.1016/J.PATCOG.2014.10.032

Siddharth Bhatore, Lalit Mohan, and · Y Raghu Reddy. 2020. Machine learning techniques for credit risk evaluation: a systematic literature review. *Journal of Banking and Financial Technology 2020 4:1* 4, 1 (6 2020), 111–138. https://doi.org/10.1007/S42786-020-00020-3

Rok Blagus and Lara Lusa. 2010. Class prediction for high-dimensional class-imbalanced data. *BMC bioinformatics* 11, 1 (10 2010), 523. https://doi.org/10.1186/1471-2105-11-523/FIGURES/6

Rok Blagus and Lara Lusa. 2013. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics* 14, 1 (3 2013), 1–16. https://doi.org/10.1186/1471-2105-14-106/FIGURES/7

Paula Branco, Luís Torgo, and Rita P Ribeiro. 2015. A Survey of Predictive Modelling under Imbalanced Distributions. (5 2015). https://arxiv.org/abs/1505.01658v2

Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (10 2001), 5–32. https://doi.org/10.1023/A:1010933404324/METRICS

Iain Brown and Christophe Mues. 2012. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Systems with Applications* 39, 3 (2 2012), 3446–3453. https://doi.org/10.1016/J.ESWA.2011.09.033

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal Of Artificial Intelligence Research* 16 (6 2002), 321–357. https://doi.org/10.1613/jair.953

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 13-17-August-2016 (8 2016), 785–794. https://doi.org/10.1145/2939672.2939785

Xue Chen, Zhenlong Liu, Ming Zhong, Xin Liu, and Peng Song. 2019. A Deep Learning Approach Using DeepGBM for Credit Assessment. In *Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence (RICAI '19)*. Association for Computing Machinery, New York, NY, USA, 774–779. https://doi.org/10.1145/3366194.3366333

Zhenpeng Chen, Jie M. Zhang, Federica Sarro, and Mark Harman. 2022. A Comprehensive Empirical Study of Bias Mitigation Methods for Machine Learning Classifiers. (7 2022). https://doi.org/10.1145/nnnnnnn.nnnnnnn

Sven F. Crone and Steven Finlay. 2012. Instance sampling in credit scoring: An empirical study of sample size and balancing. *International Journal of Forecasting* 28, 1 (1 2012), 224–238. https://doi.org/10.1016/j.ijforecast.2011.07.006

Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. 2015. Certifying and removing disparate impact. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 2015-August (12 2015), 259–268. https://doi.org/10.1145/2783258.2783311

Raquel Florez-Lopez and Juan Manuel Ramon-Jeronimo. 2015. Enhancing accuracy and interpretability of ensemble strategies in credit risk assessment. A correlated-adjusted decision forest proposal. *Expert Systems with Applications* 42, 13 (8 2015), 5737–5753. https://doi.org/10.1016/j.eswa.2015.02.042

Yoav Freund and Robert E. Schapire. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. System Sci.* 55, 1 (8 1997), 119–139. https://doi.org/10.1006/JCSS.1997.1504

J. Galindo and P. Tamayo. 2000. Credit risk assessment using statistical and machine learning: Basic methodology and risk modeling applications. *Computational Economics* 15, 1-2 (2000), 107–143. https://doi.org/10.1023/A:1008699112516/METRICS

Vicente García, Ramón A. Mollineda, and J. Salvador Sánchez. 2014. A bias correction function for classification performance assessment in two-class imbalanced problems. *Knowledge-Based Systems* 59 (3 2014), 66–74. https://doi.org/10.1016/J.KNOSYS.2014.01.021

S Gopal, Krishna Patro, and Kishore Kumar Sahu. 2015. Normalization: A Preprocessing Stage. *IARJSET* (3 2015), 20–22. https://doi.org/10.17148/iarjset.2015.2305

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. 2022. Why do tree-based models still outperform deep learning on tabular data? (7 2022). https://arxiv-org.ezproxy2.utwente.nl/abs/2207.08815v1

Akhilesh Gupta, Nesime Tatbul, Ryan Marcus, Shengtian Zhou, Insup Lee, and Justin Gottschlich. 2020. CLASS-WEIGHTED EVALUATION METRICS FOR IMBALANCED DATA CLASSIFICATION. (2020).

Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. 2017. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications* 73 (5 2017), 220–239. https://doi.org/10.1016/J.ESWA.2016.12.035

Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of Opportunity in Supervised Learning. *Advances in Neural Information Processing Systems* (10 2016), 3323–3331. https://arxiv.org/abs/1610.02413v1

Tin Kam Ho. 1995. Random decision forests. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR* 1 (1995), 278–282. https://doi.org/10.1109/ICDAR.1995.598994

Tin Kam Ho. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 8 (1998), 832–844. https://doi.org/10.1109/34.709601

Abid Ishaq, Saima Sadiq, Muhammad Umer, Saleem Ullah, Seyedali Mirjalili, Vaibhav Rupapara, and Michele Nappi. 2021. Improving the Prediction of Heart Failure Patients' Survival Using SMOTE and Effective Data Mining Techniques. *IEEE Access* 9 (2021), 39707–39716. https://doi.org/10.1109/ACCESS.2021.3064084

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. *Advances in Neural Information Processing Systems* 30 (2017). https://github.com/Microsoft/LightGBM.

Nico Kritzinger and Gary W. van Vuuren. 2018. An optimised credit scorecard to enhance cut-off score determination. *South African Journal of Economic and Management Sciences* 21, 1 (6 2018), 15. https://doi.org/10.4102/SAJEMS.V21I1.1571

Max Kuhn and Kjell Johnson. 2013. Data Pre-processing. In *Applied Predictive Modeling*. Springer New York, New York, NY, 27–59. https://doi.org/10.1007/978-1-4614-6849-3{_}3

Juan Laborda and Seyong Ryoo. 2021. Feature Selection in a Credit Scoring Model. *Mathematics 2021, Vol. 9, Page 746* 9, 7 (3 2021), 746. https://doi.org/10.3390/MATH9070746

Guillaume Lemaitre, Fernando Nogueira, and Christos K Aridas char. 2017. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning. *Journal of Machine Learning Research* 18, 17 (2017), 1–5. http://jmlr.org/papers/v18/16-365.html

Jianping Li, Mingxi Liu, Cheng Few Lee, and Dengsheng Wu. 2020. Support vector machines based methodology for credit risk analysis. *Handbook of Financial Econometrics, Mathematics, Statistics, and Machine Learning (In 4 Volumes)* (1 2020), 791–822. https://doi.org/10.1142/9789811202391{_}0020

Yiheng Li and Weidong Chen. 2020. A Comparative Performance Assessment of Ensemble Learning for Credit Scoring. *Mathematics* 8, 10 (10 2020), 1–19. https://doi.org/10.3390/MATH8101756

Victoria López, Alberto Fernández, Salvador García, Vasile Palade, and Francisco Herrera. 2013. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf. Sci.* 250 (11 2013), 113–141. https://doi.org/10.1016/J.INS.2013.07.007

Marcos Roberto Machado and Salma Karray. 2022. Assessing credit risk of commercial customers using hybrid machine learning algorithms. *Expert Systems with Applications* 200 (8 2022), 116889. https://doi.org/10.1016/J.ESWA.2022.116889

A. I. Marqués, V. García, and J. S. Sánchez. 2012. Two-level classifier ensembles for credit risk assessment. *Expert Systems with Applications* 39, 12 (9 2012), 10916–10922. https://doi.org/10.1016/J.ESWA.2012.03.033

A. I. Marqués, V. García, and J. S. Sánchez. 2013. On the suitability of resampling techniques for the class imbalance problem in credit scoring. *Journal of the Operational Research Society* 64, 7 (2013), 1060–1070. https://doi.org/10.1057/JORS.2012.120

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys (CSUR)* 54, 6 (7 2021). https://doi.org/10.1145/3457607

Leopoldo Melo Junior, Franco Maria Nardini, Chiara Renso, Roberto Trani, and Jose Antonio Macedo. 2020. A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems. *Expert Syst. Appl.* 152 (8 2020). https://doi.org/10.1016/J.ESWA.2020.113351

Aditya Krishna Menon and Robert C. Williamson. 2017. The cost of fairness in classification. (5 2017). https://arxiv.org/abs/1705.09055v1

Microsoft Corporation. 2023. Parameters Tuning — LightGBM 3.3.5 documentation. https://lightgbm.readthedocs.io/en/v3.3.5/Parameters-Tuning.html

Microsoft Corporation. nd. lightgbm.LGBMClassifier — LightGBM 3.3.5 documentation. https://lightgbm.readthedocs.io/en/v3.3.5/pythonapi/lightgbm.LGBMClassifier.html

Yoni Nazarathy and Hayden Klok. 2021. Statistics with Julia. (2021). https://doi.org/10.1007/978-3-030-70901-3

Giuseppe Paleologo, André Elisseeff, and Gianluca Antonini. 2010. Subagging for credit scoring models. *European Journal of Operational Research* 201, 2 (3 2010), 490–499. https://doi.org/10.1016/J.EJOR.2009.03.008

Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. 2008. Discrimination-aware data mining. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2008), 560–568. https://doi.org/10.1145/1401890.1401959

Yi Peng, Guoxun Wang, Gang Kou, and Yong Shi. 2011. An empirical study of classification algorithm evaluation for financial risk prediction. *Appl. Soft Comput.* 11, 2 (3 2011), 2906–2915. https://doi.org/10.1016/J.ASOC.2010.11.028

J. Godwin Ponsam, S. V.Juno Bella Gracia, G. Geetha, S. Karpaselvi, and K. Nimala. 2021. Credit Risk Analysis using LightGBM and a comparative study of popular algorithms. *Proceedings of the 2021 4th International Conference on Computing and Communications Technologies, ICCCT 2021* (2021), 634–641. https://doi.org/10.1109/ICCCT53315.2021.9711896

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. 2017. CatBoost: unbiased boosting with categorical features. *Advances in Neural Information Processing Systems* 2018-December (6 2017), 6638–6648. https://arxiv.org/abs/1706.09516v5

Maytal Saar-Tsechansky and Foster Provost. 2007. Handling Missing Values when Applying Classification Models. *Journal of Machine Learning Research* 8, 57 (2007), 1623–1657. http://jmlr.org/papers/v8/saar-tsechansky07a.html

scikit-learn developers. n.d.. sklearn.ensemble.RandomForestClassifier — scikit-learn 1.2.2 documentation. https://scikit-learn.org/1.2/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn-ensemble-randomforestclassifier

Leopoldo Soares De Melo Junior, Franco Maria Nardini, Chiara Renso, and Jose Antonio Fernandes De MacEdo. 2019. An Empirical Comparison of Classification Algorithms for Imbalanced Credit Scoring Datasets. *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)* (12 2019), 747–754. https://doi.org/10.1109/ICMLA.2019.00133

The World Bank. 2022. Financial Inclusion Overview. https://www.worldbank.org/en/topic/financialinclusion/overview#1

Jakub M. Tomczak and Maciej Zie ̨ba. 2015. Classification Restricted Boltzmann Machine for comprehensible credit scoring model. *Expert Systems with Applications* 42, 4 (2015), 1789–1796. https://doi.org/10.1016/J.ESWA.2014.10.016

Swati Tyagi. 2022. Analyzing Machine Learning Models for Credit Scoring with Explainable AI and Optimizing Investment Decisions. *ArXiv* (2022). https://doi.org/10.48550/ARXIV.2209.09362

Gang Wang, Jinxing Hao, Jian Ma, and Hongbing Jiang. 2011. A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications* 38, 1 (1 2011), 223–230. https://doi.org/10.1016/J.ESWA.2010.06.048

Blake Woodworth, Suriya Gunasekar, Mesrob I. Ohannessian, and Nathan Srebro. 2017. Learning Non-Discriminatory Predictors. (2 2017). https://arxiv.org/abs/1702.06081v3

Yandex LLC. nd. Overview - CatBoostClassifier | CatBoost. https://catboost.ai/en/docs/concepts/python-reference_catboostclassifier

Zhihua Zhou. 2012. *Ensemble Methods: Foundations and Algorithms* (1 ed.). CRC Press, Boca Raton. 48–59 pages.