Journal of Data Science 0 (0), 1–16
??? 2024
DOI: 10.6339/24-JDS1154
Statistical Data Science

Is Augmentation Effective in Improving Prediction in Imbalanced Datasets?

Gabriel O. Assunção[1,*], Rafael Izbicki[2], and Marcos O. Prates[1]

[1]*Department of Statistics, Universidade Federal de Minas Gerais, Belo Horizonte, Brazil*
[2]*Department of Statistics, Universidade Federal de São Carlos, São Carlos, Brazil*

Abstract

Imbalanced datasets present a significant challenge for machine learning models, often leading to biased predictions. To address this issue, data augmentation techniques are widely used to generate new samples for the minority class. However, in this paper, we challenge the common assumption that data augmentation is necessary to improve predictions on imbalanced datasets. Instead, we argue that adjusting the classifier cutoffs without data augmentation can produce similar results to oversampling techniques. Our study provides theoretical and empirical evidence to support this claim. Our findings contribute to a better understanding of the strengths and limitations of different approaches to dealing with imbalanced data, and help researchers and practitioners make informed decisions about which methods to use for a given task.

Keywords *balanced accuracy; data augmentation; oversampling*

1 Introduction

Imbalanced datasets are a widespread issue encountered in real-life text datasets, where certain classes contain more observations than others. For instance, positive reviews of a product tend to outweigh negative ones. However, machine learning models trained on imbalanced data can lead to biased predictions, and addressing this concern is crucial. This fact impacts different fields, such as natural language processing (NLP), where imbalanced classes arise in applications such as spam filtering (Al Najada and Zhu, 2014) and sentiment analysis (Wang et al., 2013). Similarly, in image classification, detecting objects in images with minimal examples poses a significant challenge (Gao et al., 2014).

Dealing with imbalanced data can be achieved through various approaches, which can be classified into three categories: preprocessing, cost-sensitive, and algorithmic methods (Kaur et al., 2019). A commonly used approach is preprocessing, which involves generating new samples to balance the dataset. This method is also known as sampling, and it aims to generate new data synthetically for the minority class.

Two popular sampling methods are Synthetic Minority Oversampling Technique (SMOTE) (Chawla et al., 2002) and Random Oversampling, which have been extensively used and studied (Li et al., 2010; Padurariu and Breaban, 2019). These techniques increase the representation of minority classes, which can reduce bias towards the majority class and improve the model's performance (Feng et al., 2021). Indeed, such techniques are recommended by a vast amount of papers (Abdoh et al., 2018; Tan et al., 2019; Rupapara et al., 2021; Akkaradamrongrat et al., 2019; Mohasseb et al., 2018; Tesfahun and Bhaskari, 2013).

*Corresponding author. Email: gabrieloliveira1995@gmail.com.

Received April 25, 2024; Accepted September 6, 2024

Oversampling techniques are widely used to enhance prediction accuracy, especially when the classification rule relies on assigning a new observation to the label with the highest estimated probability (Abdoh et al., 2018; Tan et al., 2019; Rupapara et al., 2021; Akkaradamrongrat et al., 2019; Mohasseb et al., 2018; Tesfahun and Bhaskari, 2013). However, in this paper, we argue that contrary to common belief, data augmentation is usually unnecessary to improve predictions on imbalanced datasets. Our argument is that oversampling techniques are widely used because most machine learning software relies on assigning a new instance to the label with the highest estimated probability by default. This is not optimal. Indeed, we show that by changing how the probabilities estimated using the unaugmented data are used to create a classifier, we can maximize the classification performance for most datasets. As a result, the purported benefits of data augmentation may be misleading.

To support our argument, we provide both theoretical and empirical evidence that challenges the common belief that data augmentation is always beneficial for imbalanced datasets. We believe that our findings will contribute to a better understanding of the strengths and limitations of different approaches to dealing with imbalanced data, and will help researchers and practitioners to make more informed decisions about which methods to use for a given task.

Let Y denote the label of interest, and $\mathbf{x}$ is the vector of features. For simplicity, let us assume that Y takes binary values, 0 or 1. In this study, we prove that if we can accurately estimate $\mathbb{P}(Y = 1|\mathbf{x})$, changing the cutoffs of the classifier without data augmentation produces the same results as using Random Oversampling (Section 2). That is, classifying an instance as belonging to the positive class based on $\mathbb{P}(Y = 1|\mathbf{x}) > c$, where c is properly chosen (see Section 2 for details), is equivalent to performing data augmentation and classifying with a cutoff of 0.5, the default majority rule used by most software.

To illustrate this point, Figure 1 presents the results obtained from the Android App reviews dataset (Grano et al., 2017). The left panel of the figure shows the improvement in balanced accuracy achieved by a model that uses Random Oversampling compared to a base model that does not use augmentation when a threshold of 0.5 is used for classification. However, as shown in the right panel, by properly choosing the value of c, we can obtain improved results without using data augmentation.

Therefore, the only way random oversampling can produce better classifications is if it can create a more accurate estimate of $\mathbb{P}(y|\mathbf{x})$. However, there is no evidence to support the fact that $\mathbb{P}(y|\mathbf{x})$ will be better estimated using oversampling: oversampling only reuses the same training points, so there is no new information on the augmented dataset. Indeed, Section 3 presents empirical data that suggests random oversampling does not improve the accuracy of the probability estimate. Moreover, that section also goes beyond Random Oversampling and provides empirical evidence that even other state-of-the-art augmentation methods achieve similar results to a model without oversampling as long as the cutoff is properly chosen. Finally, Section 4 ends with a conclusion and discussion of our discoveries.

2 Random Oversampling

Let $\mathbf{X} \in \mathcal{X}$ denote the features (covariates), and $Y \in \{0, \ldots, K-1\} := \mathcal{Y}$ be the label of interest. We start by proving that the probabilities $\mathbb{P}(Y = k|\mathbf{x})$ (henceforth the *conditional probabilities*) induced by the Random Oversampling technique have a one-to-one relationship with the original probabilities associated with the non-augmented dataset. We will show that, in the binary case, this implies that any classification rule based on the augmented conditional probabilities (such as

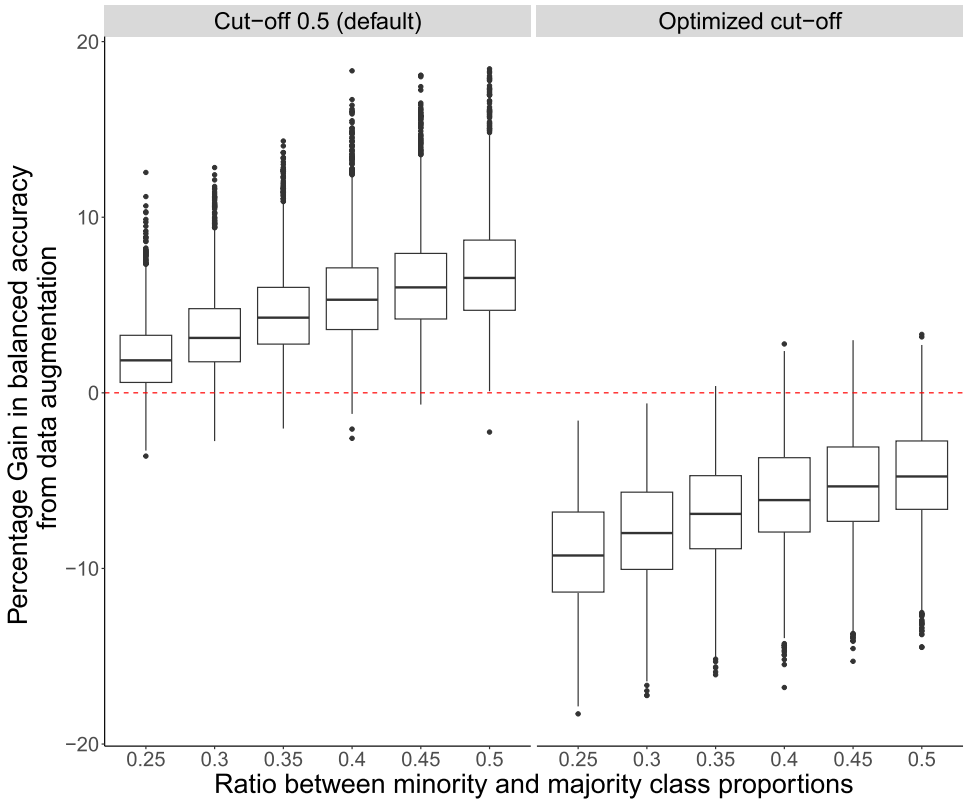

Figure 1: Gain in balanced accuracy when performing data augmentation for the Android App reviews Dataset (see Section 3 for details). If the default cutoff $c = 0.5$ for $\widehat{P}(Y = 1|\mathbf{x})$ is chosen to perform classification, it appears that data augmentation is effective (left panel). However, the right panel reveals that when c is appropriately chosen, there is no improvement in performance through data augmentation.

classifying an instance as positive if its conditional probability is larger than 0.5) is equivalent to a rule based on the non-augmented probabilities; one only needs to select an appropriate threshold point. We will assume we know $\mathbb{P}$. That is, it does not need to be estimated. All proofs are presented in the Supplementary Material S.1.

Let $\mathbb{P}$ be the probability distribution associated with the original (non-augmented) data with n observations:

$$\mathcal{T} = \big\{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\big\}.$$

We denote the dataset with m augmented samples by

$$\mathcal{T}' = \big\{(\mathbf{X}'_1, Y'_1), \ldots, (\mathbf{X}'_m, Y'_m)\big\}.$$

We assume each sample from $\mathcal{T}'$ is sampled using the following rule:

1. Draw $k \in \mathcal{Y}$, such that each element of $\mathcal{Y}$ has probability w_k of being drawn, where w_k's are predefined non-negative numbers such that $\sum_{k=0}^{K-1} w_k = 1$.
2. Randomly choose an observation from $\{(\mathbf{X}_i, Y_i) \in \mathcal{T} : Y_i = k\}$.

The augmented dataset is then given by merging $\mathcal{T}$ with $\mathcal{T}'$:

$$\mathcal{T}_{\text{aug}} = \mathcal{T} \cup \mathcal{T}'.$$

This scheme is a generalization of vanilla random oversampling:

Example 1 (Random Oversampling for binary labels). In a binary classification problem $\mathcal{Y} = \{0, 1\}$, suppose that label 1 is the minority class. To oversample the data, we set $w_1 = 1$ and $w_0 = 0$, and to make the augmented data to be 50-50, we choose $m = n(\mathbb{P}(Y = 0) - \mathbb{P}(Y = 1))$.

This data-generating scheme, together with $\mathbb{P}$, induce a distribution over each $(\mathbf{X}, Y) \in \mathcal{T}_{\text{aug}}$, denoted by $\mathbb{P}_{\text{a}}$. $\mathbb{P}_{\text{a}}$ is a version of $\mathbb{P}$ with prior probability shift (Vaz et al., 2019; Quiñonero-Candela et al., 2022). The following theorem shows how $\mathbb{P}_{\text{a}}$ relates to $\mathbb{P}$.

Theorem 1. *For each $j \in \mathcal{Y}$,*

$$\mathbb{P}_a(Y = j|\mathbf{x}) = \frac{\frac{\mathbb{P}_a(Y=j)}{\mathbb{P}(Y=j)}\mathbb{P}(Y = j|\mathbf{x})}{\sum_{k=0}^{K-1} \frac{\mathbb{P}_a(Y=k)}{\mathbb{P}(Y=k)}\mathbb{P}(Y = k|\mathbf{x})}, \tag{1}$$

where

$$\mathbb{P}_a(Y = j) = \frac{n}{n+m}\mathbb{P}(Y = j) + \frac{m}{n+m}w_j.$$

Conversely, it holds that

$$\mathbb{P}(Y = j|\mathbf{x}) = \frac{\frac{\mathbb{P}(Y=j)}{\mathbb{P}_a(Y=j)}\mathbb{P}_a(Y = j|\mathbf{x})}{\sum_{k=1}^{K} \frac{\mathbb{P}(Y=k)}{\mathbb{P}_a(Y=k)}\mathbb{P}_a(Y = k|\mathbf{x})}.$$

In practice, classification rules for probabilistic classifiers are typically obtained by minimizing the expected value of a loss function $L : \mathcal{Y} \times \mathcal{Y} \longrightarrow \mathbb{R}$, where $L(k, j)$ represents the loss of classifying an instance of the class k as being of the class j and $L(k, k) = 0$ for every $k \in \mathcal{Y}$. The optimal classifier under $\mathbb{P}_{\text{a}}$ is given by Ripley (2007)

$$g^*(\mathbf{x}) := \arg\min_{j \in \mathcal{Y}} \sum_{k \in \mathcal{Y}} L(k, j)\mathbb{P}_{\text{a}}(Y = k|\mathbf{x}). \tag{2}$$

In the binary case, the optimal decision rule corresponds to checking whether the probability is greater than the threshold $\frac{L(0,1)}{L(0,1)+L(1,0)}$ as presented in Ripley (2007, pg. 19, Proposition 2.1). In the binary case (that is, $K = 2$), criteria (2) leads to the following decision rule:

$$g^*(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbb{P}_{\text{a}}(Y = 1|\mathbf{x}) \geqslant \frac{L(0,1)}{L(0,1)+L(1,0)} \\ 0, & \text{otherwise.} \end{cases}$$

In practice, the decision rule used is often to choose the label that maximizes the $\mathbb{P}_a(Y = k|\mathbf{x})$. This classifier is a special case when the errors have the same cost, see Proposition 2.1 in Ripley (2007, pg 19).

Under the 0-1 loss (that is, all misclassification errors have the same cost $L(k, j) = \mathbb{I}(k \neq j)$), Equation (2) leads to the following optimal classifier:

$$g^*(\mathbf{x}) := \arg\min_{j \in \mathcal{Y}} \sum_{k \in \mathcal{Y} | k \neq j} \mathbb{P}_{\text{a}}(Y = k|\mathbf{x}),$$

which, in the binary case, is equivalent to

$$g^*(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbb{P}_{\text{a}}(Y = 1|\mathbf{x}) \geqslant \frac{1}{2} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The following theorem shows that we can obtain the same classifier from Equation (2), g^*, using the probability associated with the original data but with a different loss.

Theorem 2. *Let g^* be the optimal classifier according to the loss L and the distribution induced by the augmentation procedure $\mathbb{P}_a$ (Equation (2)). Then*

$$g^*(\mathbf{x}) = \arg\min_{j \in \mathcal{Y}} \sum_{k \in \mathcal{Y}} L'(k, j)\mathbb{P}(Y = k|\mathbf{x}),$$

where $L'(k, j) = L(k, j)\mathbb{P}_a(Y = k)/P(Y = k)$.

In particular, from Proposition 2.1 (Ripley, 2007, pg. 19) and Theorem 2 it follows that, in the binary case, the optimal classifier for the augmented dataset according to the 0-1 loss can be computed by comparing the original conditional probabilities to the threshold $\mathbb{P}(Y = 1)$:

Corollary 1. *The classifier of Equation (3) is equivalent to*

$$g^*(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbb{P}(Y = 1|\mathbf{x}) \geqslant \mathbb{P}(Y = 1) \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

This classifier is the one that maximizes the balanced accuracy.

Figure 2 illustrates the relationship between the original probability $\mathbb{P}(Y = 1|\mathbf{x})$ and the augmented probability $\mathbb{P}_a(Y = 1|\mathbf{x})$ for various values of $\mathbb{P}(Y = 1)$. In particular, it illustrates Corollary 1 by showing that $\mathbb{P}_a(Y = 1|\mathbf{x}) = 0.5$ always corresponds to $\mathbb{P}(Y = 1|\mathbf{x}) = \mathbb{P}(Y = 1)$, and therefore checking whether $\mathbb{P}_a(Y = 1|\mathbf{x}) > 0.5$ is equivalent to checking whether $\mathbb{P}(Y = 1|\mathbf{x}) > \mathbb{P}(Y = 1)$.

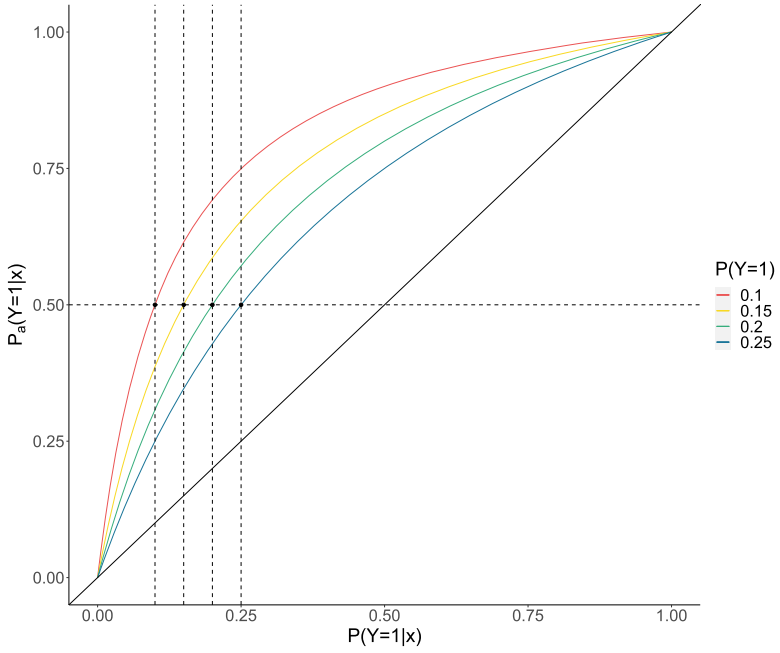

Figure 2: Relationship between the original probability $\mathbb{P}(Y = 1|\mathbf{x})$ and the augmented probability $\mathbb{P}_a(Y = 1|\mathbf{x})$ (Theorem 1) for different values of $\mathbb{P}(Y = 1)$. The dashed lines illustrate Corollary 1: setting the threshold on the augmented data to 0.5 corresponds to using the threshold of $\mathbb{P}(Y = 1)$ on the original probability.

Corollary 1 proves that, if we know either $\mathbb{P}(Y = 1|\mathbf{x})$ or $\mathbb{P}_a(Y = 1|\mathbf{x})$, then comparing the former to an optimized threshold will yield the same loss as using a threshold of 0.5 and applying Random Oversampling. However, in practice, we must estimate these probabilities using models, which could potentially lead to different results. Nevertheless, empirical results in the following section indicate that selecting the appropriate threshold for $\widehat{\mathbb{P}}(Y = 1|\mathbf{x})$ is sufficient to achieve the same outcome as using $\widehat{\mathbb{P}}_a(Y = 1|\mathbf{x})$, even if the probabilities are estimated. This is because Random Oversampling does not provide any additional information about the dataset; it merely reuses the same sample points from the original training set. Surprisingly, the empirical results in the next section suggest that this is also the case even when using more advanced and complex data augmentation methods.

3 Experiments

We perform an empirical comparison of several oversampling techniques on real datasets with the aim of demonstrating that, by appropriately selecting the threshold, there is generally no advantage in employing oversampling. The perceived benefits are often illusory and arise from the adoption of default threshold values.

In our experiments, we use eight datasets, all available online:

- Sentiment Analysis in Twitter (Barbieri et al., 2020).
- Women's E-commerce Clothing Reviews (Agarap, 2018).
- Android Apps and User Feedback (Grano et al., 2017).
- Hate Speech Offensive (Davidson et al., 2017).
- Default of Credit Card Client (Yeh and Lien, 2009).
- CDC Diabetes Health Indicators (Available on UCI).
- Customer Churn (Available on Kaggle).
- Conversion digital marketing Campaign (Available on Kaggle).

Table 1 shows a description of the dataset minority class ratio, size of the dataset and the main goal of the used datasets.

Table 1: Description of the datasets used to compare the augmentation methods.

Dataset Name	Goal	Size	Ratio
Sentiment Analysis in Twitter	Classifies a tweet sentiment in negative, positive or neutral	24K	28% of the tweets are negative (we used only positive and negative tweets)
Women's E-Commerce Clothing Reviews	Based on a review classify if the user recommends the product	23K	18% of the clients do not recommend a product.
Android Apps and User Feedback	Classify an app review in positive or negative	288K	18% of the reviews are negative (less than 3 stars)
Hate speech offensive	Label a tweet as hate speech, offensive or neither	20K	7% of the tweets are labeled as hate speech. We used only hate speech and offensive tweets
Default Credit Card Client	Predict a default payment of a credit card based on demographic factors, credit data, history of payment	30K	21% of the samples is a default payment
CDC Diabetes Health Indicators	Classify a patient with diabetes or prediabetes	253K	14% of the samples are patients with diabetes
Customer Churn	Predict a client churn based on features like age, gender, tenure, usage frequency, support calls, payment delay	10K	21% of the clients has churned
Conversion digital marketing Campaign	Predict Marketing Campaign conversion	8K	11% of the clients has not converted

To assess the efficacy of different oversampling methodologies, we conducted a random sampling of the original datasets fifty times, with two sample sizes: 500 and 2000. Each sample was utilized to generate a distinct model. To validate the models, we established a separate validation dataset by randomly sampling from the original dataset, comprising two distinct sizes: 125 for models trained with 500 sample points and 500 for models trained with 2000 sample points. All models were compared using an independent sample with 1000 sample points.

We opted for limited sample sizes because it is the typical setting commonly used for augmentation techniques (Chen et al., 2023a; Stylianou et al., 2023; Wu and Huang, 2022; Kokol et al., 2022; Tepper et al., 2020; Kumar et al., 2020; Hu et al., 2019; Shleifer, 2019; Shu et al., 2018; Zhou, 2018). Therefore, we can accurately measure the uncertainty surrounding the prediction errors for each method. See, however, Section 3.3 for an analysis of the full datasets.

The following oversampling methods were used in the experiments:

- **Random Sampling**: sentences of the training dataset are randomly sampled.
- **SMOTE** (Chawla et al., 2002): creates synthetic data considering the neighbors of observations from the minority class, the synthetic data is created between the observation and its neighbor.
- **Borderline SMOTE** (Han et al., 2005): is a variation of SMOTE and uses borderline samples to generate synthetic data, that is, the observations used to create the new samples are those that are misclassified by a KNN classifier (Fix and Hodges, 1952).
- **ADASYN** (He et al., 2008): is similar to SMOTE, but the focus is to generate new samples that are hard to classify based on the KKN classifier. The main difference between ADASYN and Bordeline SMOTE is that Bordeline SMOTE exclusively sample instances that are near to the majority class.

All oversampling methods are compared using a random forest classifier (Breiman, 2001) with a Bag-of-Words input for the text datasets. We opted for the previous techniques because it does not depend on fining tuning (Sumathi et al., 2020) and allow us to focus the study on the augmentation method's capacity to improve or not the classification.

To evaluate the effectiveness of a classifier $g(\mathbf{x}) \in 0, 1$, we use the balanced accuracy metric, which calculates the average of the sensitivity and specificity of the classifier:

$$\text{ba}(g) = \frac{\mathbb{P}(g(\mathbf{X}) = 1|Y = 1) + \mathbb{P}(g(\mathbf{X}) = 0|Y = 0)}{2}.$$

In the supplementary material we demonstrate the results for F1-score, sensitivity, specificity and accuracy.

In practice, we estimate $\text{ba}(g)$ using the test data. To enhance interpretability, we compare the performance of the classifier obtained via data augmentation, g_{aug}, with that of a non-augmented classifier g_{base} by reporting the percentage gain in balanced accuracy achieved by the augmented classifier over the non-augmented one:

$$\text{PercentageGain} = \frac{\text{ba}(g_{\text{aug}}) - \text{ba}(g_{\text{base}})}{\text{ba}(g_{\text{base}})}. \tag{5}$$

We also test whether the augmentation methods improve the estimates of the probabilities $\mathbb{P}(Y = 1|\mathbf{x})$ by using the percentage gain (Equation (5)) in terms of the Area Under ROC (AUC) (Lusted, 1971) and the Brier Score (Brier et al., 1950):

$$\text{BrierScore} = \mathbb{E}\big[\big(Y - \hat{\mathbb{P}}(Y = 1|\mathbf{X})\big)^2\big].$$

Again we use the test data to estimate such quantities. Moreover, to make results meaningful, to compute such scores we first map the augmented estimates $\widehat{\mathbb{P}}_a(Y = 1|\mathbf{x})$ back to $\widehat{\mathbb{P}}(Y = 1|\mathbf{x})$ using the last equation of Theorem 1.

We augmented all train datasets 40 times using each technique to achieve a 50-50 ratio for the minority class. To assess whether the augmented method truly enhances the model's performance, we conducted a formal hypothesis test to evaluate whether the percentage gain is zero (for further information regarding this hypothesis test, refer to the Supplementary Material S.2). The level of significance was set to $\alpha = 1\%$. We chose not to apply multiple corrections because each test compares different methods on different datasets, making such corrections potentially unnecessary. Additionally, using Bonferroni corrections would result in fewer rejections, favoring our thesis that augmentation is unnecessary. Not using these corrections adheres to more stringent criteria.

3.1 The importance of choosing a suitable threshold

To evaluate the significance of selecting an appropriate threshold, we employed two classification rules in our methodology:

- **($c = 0.5$)** The first rule is the widely-used classifier that assigns labels based on the class with a higher probability. Because we only deal with binary classification, this corresponds to using the value of $c = 0.5$ as a threshold.
- **(Optimized c)** The second rule, inspired by Corollary 1, involves choosing a threshold c that maximizes the balanced accuracy on a validation dataset.

We perform three comparative evaluations of the following classifiers:

1. **($c = 0.5$ for augmented and base models)** We evaluate the standard classifier that uses $c = 0.5$ on both augmented and non-augmented models. This is the approach usually taken by blog posts and papers that advocate for the use of data augmentation, as well as the default options in most machine learning packages, such as Python Scikit-learn models (Pedregosa et al., 2011); XGBoost API (Chen et al., 2023b); R Random Forest package (Liaw and Wiener, 2002)
2. **($c = 0.5$ for the augmented model, and optimized for base model)** We compare the non-augmented method, but now using the threshold that optimizes the balanced accuracy, against the augmented method with a classification rule of 0.5. The goal is to check whether choosing c in the non-augmented model is enough to achieve the same accuracy as we would get by doing data augmentation. This is what we expect to happen on the Random Oversampling method (Corollary 1).
3. **(Optimized c augmented and base models)** We optimize the cutoffs for both the base and the augmented models. This comparison aims to determine whether the other techniques have any advantages over the non-augmented model that could not be achieved by choosing an appropriate threshold.

Figure 3 displays the average percentage gain attained for the augmented techniques in each dataset. The top rows represent the results achieved on models trained with a sample size of 500, while the bottom rows correspond to models trained with a sample size of 2000. Results demonstrating no statistical significance are identified by an asterisk and presented against a white background. Significance is denoted by a color scale, wherein red denotes instances where the non-augmented method produces superior outcomes and blue indicates cases in which the augmented method yields better results.

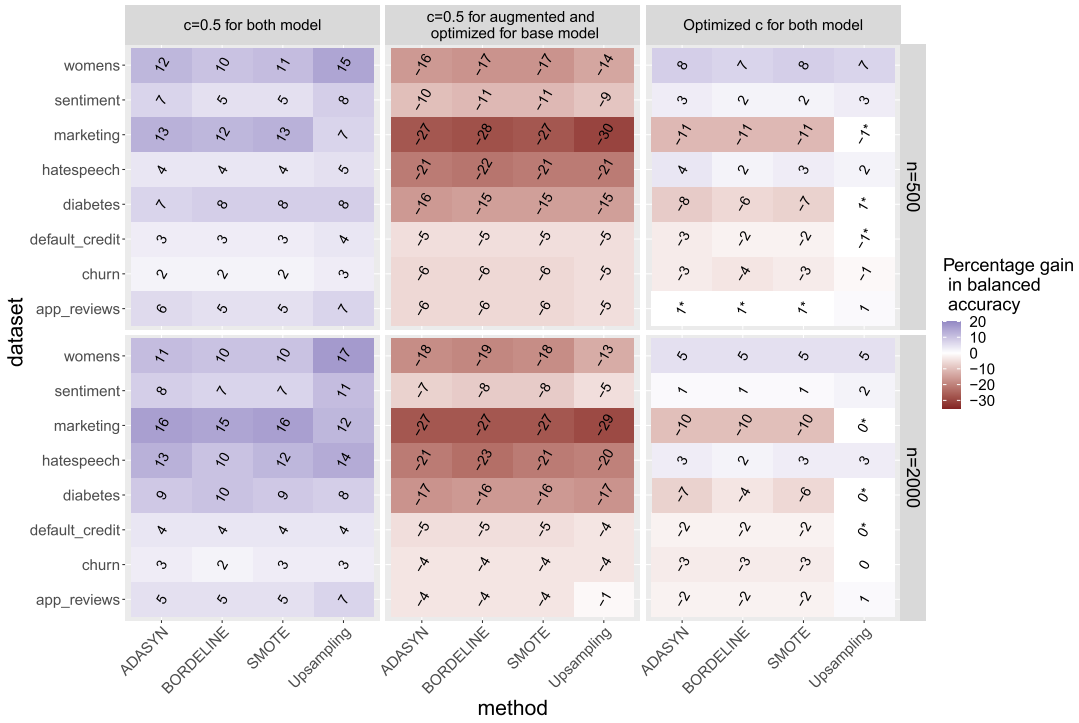

Figure 3: Heatmap of the mean percentage gain in balanced accuracy when comparing the augmented methods with the non-augmented model for classification rules. Positive values indicate superior performance by the augmented method. Non-significant, at a significance level of 1%, gains are marked with asterisks and displayed in white. Our findings indicate that data augmentation provides a noticeable benefit only when using the default threshold of $c = 0.5$ (left column); optimizing the threshold on non-augmented data eliminates the need for augmentation.

In the first column, where both classifiers have $c = 0.5$, we observe the augmented data are significantly better than the base classifier, both for $n = 500$ and $n = 2{,}000$, on all datasets. However, when the cutoff is optimized for the base model, the second column shows that essentially all data augmentation techniques have a worse balanced accuracy. Thus, the apparent benefit of data augmentation in the first column is illusory.

Furthermore, we optimized the threshold on the augmented sample to see if data augmentation could benefit the classifiers in the third column. However, the results indicate that this did not help either. In most of the cases, selecting a good cutoff for the base model was sufficient, and data augmentation did not improve the performance of the classifiers.

In Section S.3 of the Supplementary Material the same analysis for a logistic model is presented and similar results of the Random Forest are obtained.

3.2 Does data-augmentation improve the estimation of $\mathbb{P}(Y = y|\mathbf{x})$?

The preceding findings demonstrate the influence of data augmentation on classification accuracy through a comparison of $\widehat{\mathbb{P}}(Y = 1|\mathbf{x})$ against a threshold. We conclude that data augmentation scarcely enhances performance if the threshold is properly selected on a non-augmented model. It is important to note, however, that a lack of improvement in the metrics does not necessarily

imply a lack of improvement in the estimation. This section aims to explore the potential of data augmentation in improving the estimate of $\mathbb{P}(Y = 1|\mathbf{x})$.

The left column of Figure 4 shows the percentage gain in AUC for the augmented models. As before, results are marked with asterisks to indicate non-significant percentage gains according to our formal hypothesis test, while values with a blue background signify significant improvements and those with a red background indicate a significant lack of improvement. Our results reveal that, in almost all cases, the gain in AUC for augmented models was non-significant. This indicates that there is often little to no improvement in estimating $\mathbb{P}(Y = 1|\mathbf{x})$ when using data augmentation.

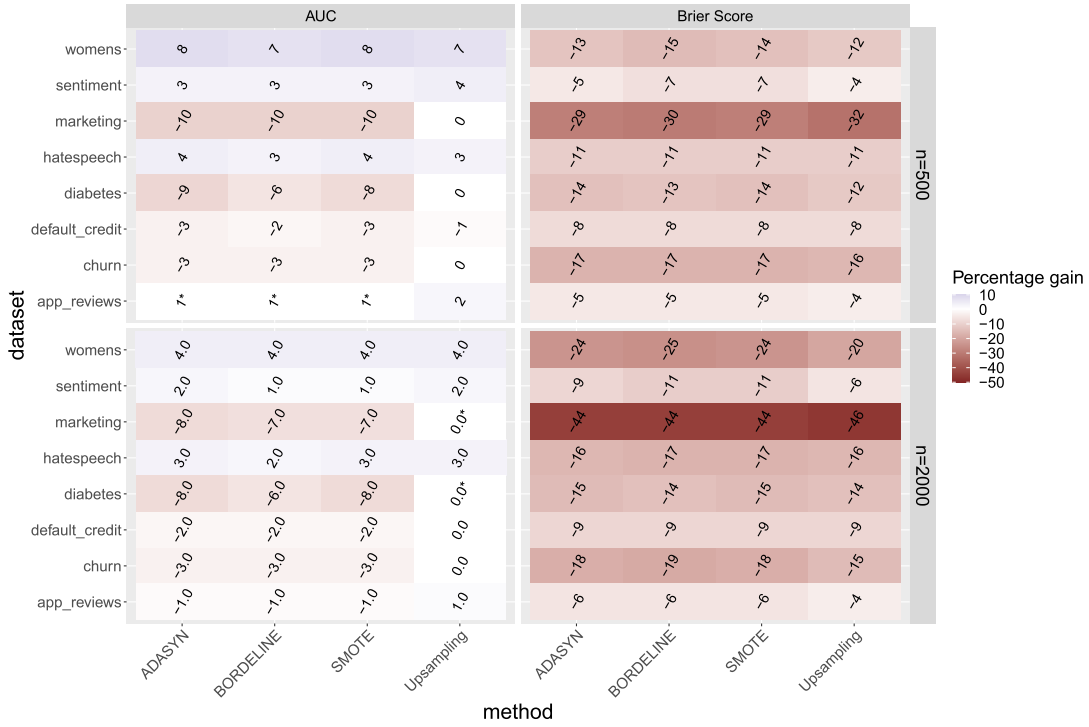

Figure 4: Heatmap of the average percentage improvement in the AUC (left column) and Brier Score (right column) when comparing the augmented methods with the non-augmented ones. Positive values indicate superior precision in estimating $\mathbb{P}(Y = 1|\mathbf{x})$ using the augmented method. Non-significant, at a significance level of 1%, gains are marked with asterisks and displayed in white. The AUC results indicate that data augmentation only rarely improves $\mathbb{P}(Y = 1|\mathbf{x})$ estimates, while the Brier Score suggests that it never improves them and often leads to worse results.

Next, we compare the behavior of ROC curves for the base and the augmented models. In order to take into account the uncertainty in the estimate of the ROC curves, we functional boxplots (Sun and Genton, 2011) on the 40 trials per train repetition that were created. The results for the Churn dataset with a sample size of 500 are illustrated in Figure 5, for the others dataset see Supplementary Material S.4. The plots show that almost all ROC curves for the non-augmented model are within the purple region, indicating that, in most cases, it does not differ from the ROC curves of the augmented models.

We next evaluated the performance of estimating $\mathbb{P}(Y = 1|\mathbf{x})$ using the Brier Score (right column of Figure 4). Our results suggest that data augmentation did not lead to a statistically

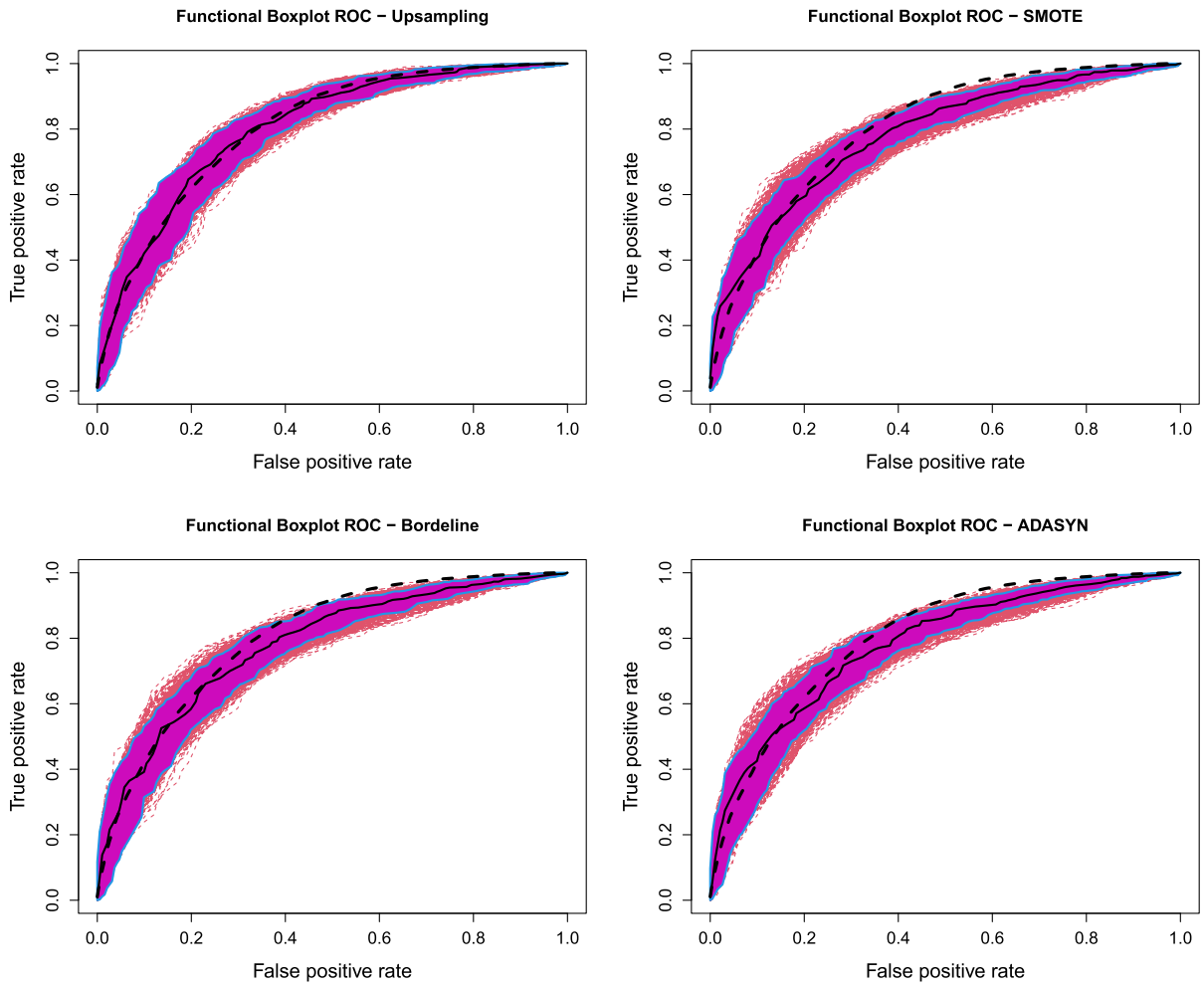

Figure 5: Functional Boxplot of the ROC curves for the Churn Dataset on the train size of 500. The blue curves, the straight black curve, and the red curve define, respectively, the interval, median, and outliers of the ROC curve for the augmented model, the purple area bounded by the blue curves represents the inter-quartile range; the dashed black curve represents the mean ROC curve of the base model. There is no benefit in doing data augmentation.

significant improvement in estimating $\mathbb{P}(Y = 1|\mathbf{x})$. In fact, in all cases, it resulted in worse performance.

In the next section we analyze the impact of the augmentation over the full applications datasets.

3.3 Threshold selection for the full dataset

In this section, we present an analysis of the impact of augmentation when using the entire dataset.

Figure 6 shows a heatmap depicting changes in balanced accuracy resulting from three different threshold selection methods. Our results indicate that setting a threshold of 0.5 notably improves model performance when using augmentation techniques. In contrast, models

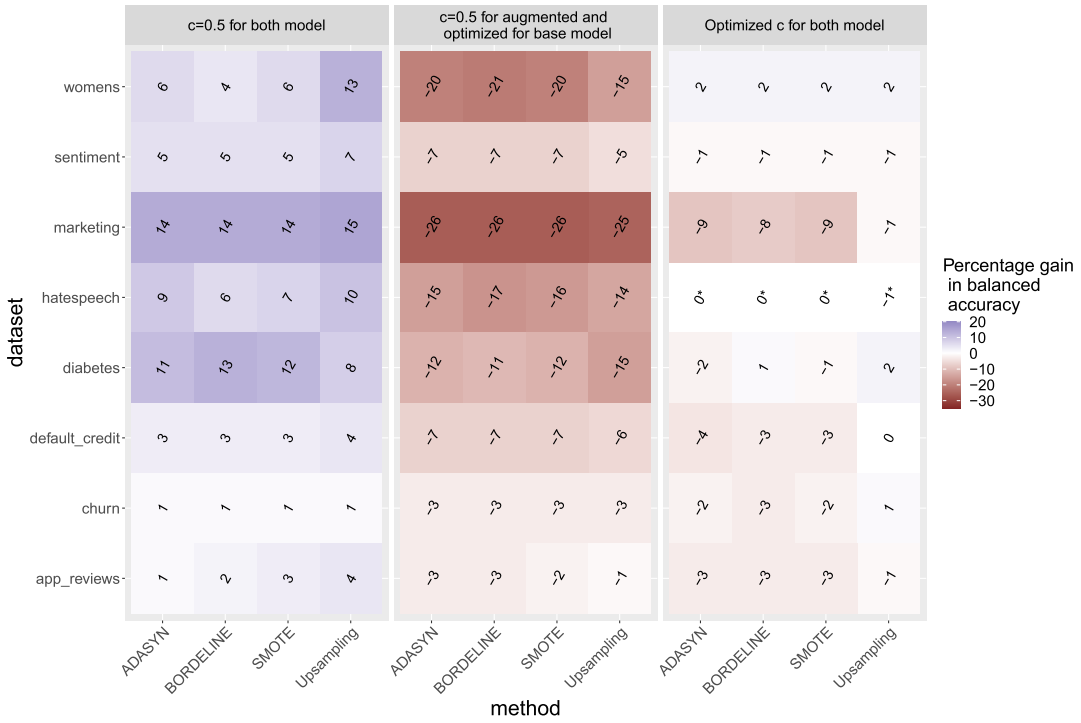

Figure 6: Heatmap of the mean percentage gain in Balanced Accuracy when comparing the augmented methods with the non-augmented model for classification rules, for models created using the full dataset. Positive values indicate superior performance by the augmented method. Non-significant gains are marked with asterisks and displayed in white. The results show that data augmentation provides a benefit when the threshold is the default $c = 0.5$, but optimizing the threshold on non-augmented data eliminates the need of augmentation.

created with the original dataset perform similarly when we employ an optimized threshold. The heatmap is predominantly red, suggesting that optimizing thresholds leads to comparable or better results than augmentation.

It is worth noting that all observed changes, apart of zero, are statistically significant. This observation is attributed to the little variability in the difference of balanced accuracy when employing the full vs augmented datasets.

4 Conclusions

By utilizing both theoretical derivations and empirical evaluations, we conclude that selecting a threshold that optimizes specific metrics of the model directly in the imbalanced dataset is enough to achieve good prediction accuracy; there is no benefit in doing data augmentation. This conclusion is reinforced by the comparable results achieved between the balanced accuracy metric and the F1-Score, as detailed in Supplementary Material S.5. Furthermore, our comprehensive analysis indicates that incorporating augmentation methods into the model does not yield a superior estimation of probabilities.

Despite a limited discussion about the efficacy of oversampling the techniques in the literature, our theoretical and empirical results align with available conclusions about the necessity of

augmentation in other research areas. For example, van den Goorbergh et al. (2022) conducted a simulated investigation to examine the impact of widely used imbalanced correction techniques in predicting ovarian cancer diagnosis, arriving at conclusions akin to ours. Also, Newaz et al. (2022) presents a broad empirical investigation around imbalanced datasets and concludes that there are no major gains from using data augmentation methods.

Although we found that none of the data augmentation methods were able to substantially improve the performance of estimates of $\mathbb{P}(y|\mathbf{x})$ and resulting classifiers, this does not mean that it is impossible to develop methods that can improve performance. In fact, if it were possible to generate new data in an i.i.d. fashion, estimates of $\mathbb{P}(y|\mathbf{x})$ (and therefore the performance of resulting classifiers) would be superior. This raises the question of how to perform data augmentation in a way that approximates i.i.d. and opens up the discussion of new techniques to improve performance.

Supplementary Material

The supplementary materials include a zipped file containing the proofs of the theorems and complementary analysis and a folder containing the code to reproduce our experiment. The code is also available in https://github.com/gabrieloa/augmentation-effective, the instructions to run the code are in the `README.md` file.

Funding

Marcos O. Prates would like to acknowledge (Conselho Nacional de Desenvolvimento Científico e Tecnológico) CNPq grant 309186/2021-8 and FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais) grant APQ-01837-22 and CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) for financial support. Rafael Izbicki is grateful for the financial support of CNPq (422705/2021-7 and 305065/2023-8) and FAPESP (grant 2023/07068-1).

References

Abdoh SF, Rizka MA, Maghraby FA (2018). Cervical cancer diagnosis using random forest classifier with SMOTE and feature reduction techniques. *IEEE Access*, 6: 59475–59485.

Agarap AF (2018). Statistical analysis on e-commerce reviews, with sentiment classification using bidirectional recurrent neural network (RNN). arXiv preprint: https://arxiv.org/abs/1805.03687. Dataset: https://www.kaggle.com/datasets/nicapotato/womens-ecommerce-clothing-reviews.

Akkaradamrongrat S, Kachamas P, Sinthupinyo S (2019). Text generation for imbalanced text classification. In: *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 181–186. IEEE.

Al Najada H, Zhu X (2014). iSRD: Spam review detection with imbalanced data distributions. In: James Joshi, Elisa Bertino, Bhavani Thuraisingham, Ling Liu, editors, *Proceedings of the 2014 IEEE 15th International Conference on Information Reuse and Integration (IEEE IRI 2014)*, pages 553–560. IEEE.

Barbieri F, Camacho-Collados J, Anke LE, Neves L (2020). Tweeteval: Unified benchmark and comparative evaluation for tweet classification. In: Trevor Cohn, Yulan He, Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650.

Breiman L (2001). Random forests. *Machine Learning*, 45(1): 5–32.

Brier GW, et al. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1): 1–3.

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002). SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16: 321–357.

Chen J, Tam D, Raffel C, Bansal M, Yang D (2023a). An empirical survey of data augmentation for limited data learning in NLP. *Transactions of the Association for Computational Linguistics*, 11: 191–211.

Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, et al. (2023b). *xgboost: Extreme gradient boosting*. R package version 1.7.5.1.

Davidson T, Warmsley D, Macy M, Weber I (2017). Automated hate speech detection and the problem of offensive language. In: *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 512–515. Dataset: https://huggingface.co/datasets/hate_speech_offensive.

Feng SY, Gangal V, Wei J, Chandar S, Vosoughi S, Mitamura T, et al. (2021). A survey of data augmentation approaches for NLP. In: Chengqing Zong, Fei Xia, Wenjie Li, Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988.

Fix E, Hodges JL Jr (1952). Discriminatory analysis-nonparametric discrimination: Small sample performance. Technical report, California Univ Berkeley.

Gao Z, Zhang L-f, Chen M-y, Hauptmann A, Zhang H, Cai A-N (2014). Enhanced and hierarchical structure algorithm for data imbalance problem in semantic extraction under massive video dataset. *Multimedia Tools and Applications*, 68: 641–657.

Grano G, Di Sorbo A, Mercaldo F, Visaggio CA, Canfora G, Panichella S, (2017). Android apps and user feedback: A dataset for software evolution and quality improvement. In: Federica Sarro, Emad Shihab, Meiyappan Nagappan, Marie C. Platenius, Daniel Kaimann, editors, *Proceedings of the 2nd ACM SIGSOFT International Workshop on App Market Analytics*, pages 8–11. Dataset: https://huggingface.co/datasets/app_reviews.

Han H, Wang W-Y, Mao B-H (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: De-Shuang Huang, Xiao-Ping Zhang, Guang-Bin Huang, editors, *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005*, Hefei, China, August 23–26, 2005, Proceedings, Part I 1, pages 878–887. Springer.

He H, Bai Y, Garcia EA, Li S (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328. IEEE.

Hu Z, Tan B, Salakhutdinov RR, Mitchell TM, Xing EP (2019). Learning data manipulation for augmentation and weighting. *Advances in Neural Information Processing Systems*, 32: 15764–15775.

Kaur H, Pannu HS, Malhi AK (2019). A systematic review on imbalanced data challenges in machine learning: Applications and solutions. *ACM Computing Surveys (CSUR)*, 52(4): 1–36.

Kokol P, Kokol M, Zagoranski S (2022). Machine learning on small size samples: A synthetic knowledge synthesis. *Science Progress*, 105(1). https://doi.org/10.1177/00368504211029777.

Kumar V, Choudhary A, Cho E (2020). Data augmentation using pre-trained transformer models. In: William M. Campbell, Alex Waibel, Dilek Hakkani-Tur, Timothy J. Hazen, Kevin Kilgour, Eunah Cho, Varun Kumar, Hadrien Glaude, editors, *Proceedings of the 2nd Work-*

shop on Life-long Learning for Spoken Language Systems, pages 18–26.

Li Y, Sun G, Zhu Y (2010). Data imbalance problem in text classification. In: Qingling Li, Fei Yu, Yun Liu, editors, *2010 Third International Symposium on Information Processing*, pages 301–305. IEEE.

Liaw A, Wiener M (2002). Classification and regression by randomforest. *R News*, 2(3): 18–22.

Lusted LB (1971). Decision-making studies in patient management. *New England Journal of Medicine*, 284(8): 416–424.

Mohasseb A, Bader-El-Den M, Cocea M, Liu H (2018). Improving imbalanced question classification using structured SMOTE based approach. In: *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 2, pages 593–597. IEEE.

Newaz A, Hassan S, Haq FS (2022). An empirical analysis of the efficacy of different sampling techniques for imbalanced classification. arXiv preprint: https://arxiv.org/abs/2208.11852.

Padurariu C, Breaban ME (2019). Dealing with data imbalance in text classification. *Procedia Computer Science*, 159: 736–745.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Quiñonero-Candela J, Sugiyama M, Schwaighofer A, Lawrence ND (2022). *Dataset Shift in Machine Learning*. MIT Press.

Ripley BD (2007). *Pattern Recognition and Neural Networks*. Cambridge University Press.

Rupapara V, Rustam F, Shahzad HF, Mehmood A, Ashraf I, Choi GS (2021). Impact of SMOTE on imbalanced text features for toxic comments classification using RVVC model. *IEEE Access*, 9: 78621–78634.

Shleifer S (2019). Low resource text classification with ulmfit and backtranslation. arXiv preprint: https://arxiv.org/abs/1903.09244.

Shu J, Xu Z, Meng D (2018). Small sample learning in big data era. arXiv preprint: https://arxiv.org/abs/1808.04572.

Stylianou N, Chatzakou D, Tsikrika T, Vrochidis S, Kompatsiaris I (2023). Domain-aligned data augmentation for low-resource and imbalanced text classification. In: *European Conference on Information Retrieval*, pages 172–187. Springer.

Sumathi B, et al. (2020). Grid search tuning of hyperparameters in random forest classifier for customer feedback sentiment prediction. *International Journal of Advanced Computer Science and Applications*, 11(9): 173–178.

Sun Y, Genton MG (2011). Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2): 316–334.

Tan X, Su S, Huang Z, Guo X, Zuo Z, Sun X, et al. (2019). Wireless sensor networks intrusion detection based on SMOTE and the random forest algorithm. *Sensors*, 19(1): 203.

Tepper N, Goldbraich E, Zwerdling N, Kour G, Tavor AA, Carmeli B (2020). Balancing via generation for multi-class text classification improvement. In: Trevor Cohn, Yulan He, Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1440–1452.

Tesfahun A, Bhaskari DL (2013). Intrusion detection using random forests classifier with SMOTE and feature reduction. In: Vidyasagar Potdar, Pritam Shah, Rajesh Ingle, Fang Liu, editors, *2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies*, pages 127–132. IEEE.

van den Goorbergh R, van Smeden M, Timmerman D, Van Calster B (2022). The harm of class imbalance corrections for risk prediction models: Illustration and simulation using logistic

regression. *Journal of the American Medical Informatics Association*, 29(9): 1525–1534.

Vaz AF, Izbicki R, Stern RB (2019). Quantification under prior probability shift: The ratio estimator and its extensions. *Journal of Machine Learning Research*, 20(79): 1–33.

Wang S, Li D, Zhao L, Zhang J (2013). Sample cutting method for imbalanced text sentiment classification based on BRC. *Knowledge-Based Systems*, 37: 451–461.

Wu J-L, Huang S (2022). Application of generative adversarial networks and Shapley algorithm based on easy data augmentation for imbalanced text data. *Applied Sciences*, 12(21): 10964.

Yeh I-C, Lien C-h (2009). The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. *Expert Systems with Applications*, 36(2): 2473–2480.

Zhou Z-H (2018). A brief introduction to weakly supervised learning. *National Science Review*, 5(1): 44–53.