SMOTE-CD: SMOTE for compositional data

Teo Nguyen, Kerrie Mengersen, Damien Sous, Benoit Liquet

PLOS ONE

RESEARCH ARTICLE

SMOTE-CD: SMOTE for compositional data

Teo Nguyen[1,2]*, **Kerrie Mengersen**[1,3], **Damien Sous**[4,5], **Benoit Liquet**[1,2]

1 Laboratoire de Mathématiques et de leurs Applications, Université de Pau et des Pays de l'Adour, E2S UPPA, CNRS, Anglet, France, **2** School of Mathematics and Physical Sciences, Macquarie University, Sydney, NSW, Australia, **3** School of Mathematical Sciences, Queensland University of Technology, Brisbane, QLD, Australia, **4** Laboratoire des Sciences Pour l'ingénieur Appliquées à la Mécanique et au Génie Électrique, Université de Pau et des Pays de l'Adour, E2S UPPA, Anglet, France, **5** Mediterranean Institute of Oceanography, Université de Toulon, Aix Marseille Université, CNRS, IRD, La Garde, France

* teo.nguyen@univ-pau.fr

Citation: Nguyen T, Mengersen K, Sous D, Liquet B (2023) SMOTE-CD: SMOTE for compositional data. PLoS ONE 18(6): e0287705. https://doi.org/10.1371/journal.pone.0287705

Editor: Sathishkumar V E, Jeonbuk National University, KOREA, REPUBLIC OF

Received: April 5, 2023

Accepted: June 12, 2023

Published: June 29, 2023

Data Availability Statement: The data and package underlying the results presented in the study are available from https://github.com/teongu/smote_cd.

Funding: Funding was provided by the Energy Environment Solutions (E2S-UPPA: https://e2s-uppa.eu/fr/index.html) consortium and the international chair Kerrie Mengersen from E2S-UPPA. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

Compositional data are a special kind of data, represented as a proportion carrying relative information. Although this type of data is widely spread, no solution exists to deal with the cases where the classes are not well balanced. After describing compositional data imbalance, this paper proposes an adaptation of the original Synthetic Minority Oversampling TEchnique (SMOTE) to deal with compositional data imbalance. The new approach, called SMOTE for Compositional Data (SMOTE-CD), generates synthetic examples by computing a linear combination of selected existing data points, using compositional data operations. The performance of the SMOTE-CD is tested with three different regressors (Gradient Boosting tree, Neural Networks, Dirichlet regressor) applied to two real datasets and to synthetic generated data, and the performance is evaluated using accuracy, cross-entropy, F1-score, R2 score and RMSE. The results show improvements across all metrics, but the impact of oversampling on performance varies depending on the model and the data. In some cases, oversampling may lead to a decrease in performance for the majority class. However, for the real data, the best performance across all models is achieved when oversampling is used. Notably, the F1-score is consistently increased with oversampling. Unlike the original technique, the performance is not improved when combining oversampling of the minority classes and undersampling of the majority class. The Python package *smote-cd* implements the method and is available online.

Introduction

Context

Over the past few years, data imbalance problems have been widely studied in classification tasks [1]. An imbalance distribution over the classes will often cause the models to prioritize their performance on the majority classes, at the expense of the minority ones. Different methods exist to deal with imbalanced datasets [2]: algorithm-level methods, where the algorithm reduces the bias by inducing a weight on the classes; data-level methods, where the data are modified to reach a more balanced state; and hybrid methods, combining both algorithm-level methods and data-level methods. Among data-level methods, Synthetic Minority

Oversampling TEchnique (SMOTE) [3], with all its variations [4], is one of the most popular for classification problems. The SMOTE algorithm generates synthetic data points for a particular class by combining the features of two existing points belonging to the same class through linear interpolation.

Most algorithms designed to tackle class imbalance problems, such as SMOTE, are often limited to the classification tasks; for instance [5–9]. However, even though regression problems are also very common in real-life problems, only a few resampling strategies exist for regression tasks [10, 11].

In this paper, we address the special issue of dealing with an imbalanced dataset in regression problems in the case where the labels are compositional. Compositional data are data carrying relative information [12], presented as proportions or percentages, making them different from other types of data. Compositional data are encountered in various fields, including biology [13–15], chemistry [16, 17], ecology [18, 19], geology [20, 21], and social sciences [22–24], among others. However, the class imbalance problem in compositional data regression remains a major challenge in the development of effective models. Existing adaptations of SMOTE and other oversampling techniques have focused on addressing imbalanced datasets in single-label regression [25–28], multi-label classification [29, 30], or when the features are compositional data [31]. However, to the best of our knowledge, no oversampling technique exists for addressing the issue of class imbalance in multi-label regression problems with compositional labels. Therefore, we propose a new oversampling technique called SMOTE for Compositional Data (SMOTE-CD), specifically designed to address this particular situation.

Here, we will measure class imbalance by summing the values of the labels (probability values) for each class on the whole dataset, and summarizing it as a percentage. In that sense, in a perfectly balanced dataset, the percentage of the sum of each class would be $1/K$, with K being the number of classes.

The proposed method is evaluated using five different performance metrics, including accuracy, cross-entropy, F1-score, R2 score, and RMSE, to three different models (Gradient Boosting tree, Neural Networks, Dirichlet regressor) on both simulated and real datasets. Since no other oversampling algorithm currently exists for compositional data, the evaluation of SMOTE-CD is limited to comparing its performance against the case where no oversampling technique is applied. The results show that the performance of the models is overall greater when applying SMOTE-CD, thus demonstrating the effectiveness of the proposed method. This is an important contribution to the field, as it provides a solution for dealing with compositional data imbalance, which has not been addressed before. The use of five different evaluation metrics, as well as the application of three different models to both simulated and real datasets, further strengthens the reliability and generalizability of the proposed method.

The entire paper is arranged as follows. The paper's first section introduces the proposed method and the motivation example. Section 2 presents the compositional data and the SMOTE-CD algorithm. Section 3 presents the metrics, the simulation study and its results. Sections 4 and 5 present the result on the real datasets. Section 6 presents the discussion and conclusion.

Motivation example: Maupiti island

Description of Maupiti island. The overall purpose of our research project is to develop an automated mapping tool able to provide a classification map from a given satellite image, with a particular focus on a coral reef-lagoon system. The test field site is the Maupiti island, the westernmost Leeward island of the Society archipelago, French Polynesia. The site has a size of approximately 8km by 8km. Maupiti data, that we use here, is just an example, but compositional data can be found, for instance, in health or chemistry fields.

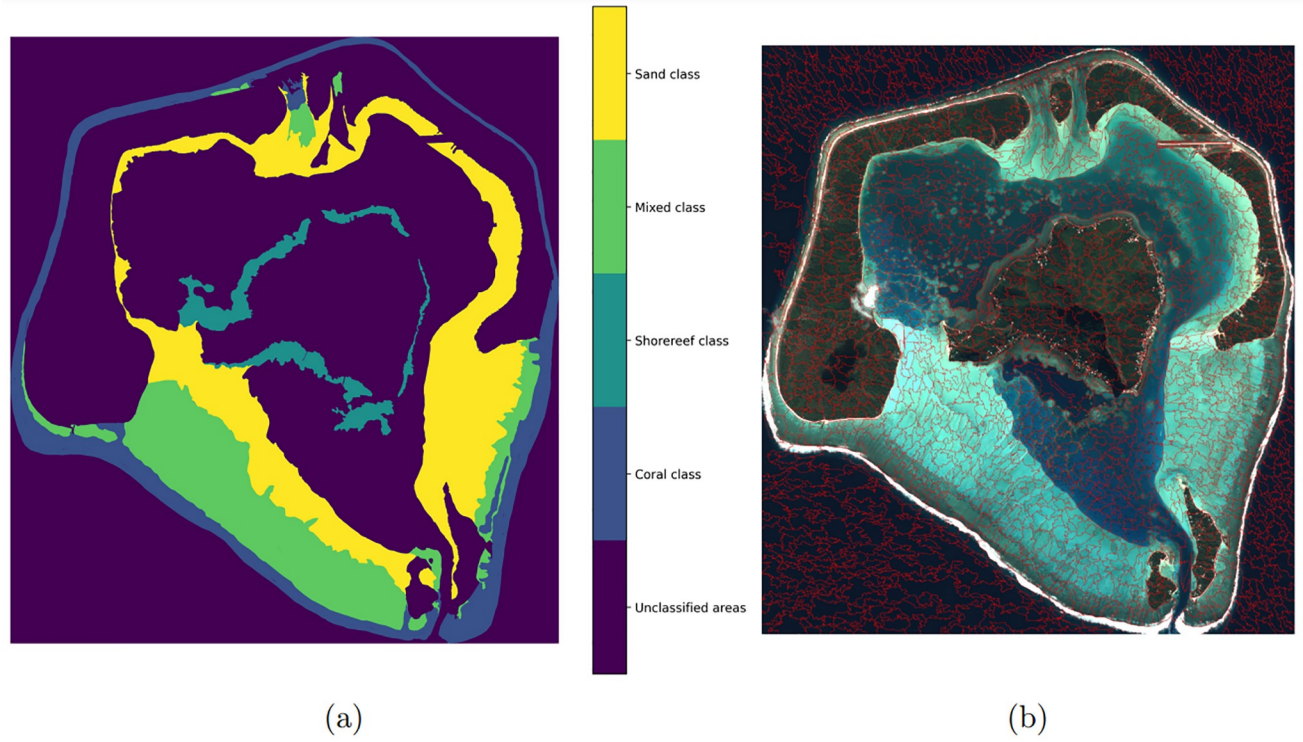


Fig 1. (a) Expert-based mapped image of Maupiti island and (b) Pleiades image of Maupiti island segmented with Felzenszwalb's method.

https://doi.org/10.1371/journal.pone.0287705.g001

An expert-based mapping of Maupiti island was used as a training dataset to develop the model. The satellite image used is a 4-band image captured on June, 14 2021 by the Pleiades satellite. The expert-based mapping of the image relies on the combination of several field observation campaigns [32] and direct examination of the satellite image. The present analysis focuses on the shallow regions of the lagoon, displaying more interpretable imaging. In the selected areas, four seabed type classes were established (Fig 1a):

- **Class 1: Coral**, marked by a overwhelming dominance of coral reef cover.
- **Class 2: Sand**, describing areas covered by detritic sand.
- **Class 3: Shorereef**, gathering shore reef and transitional shore reef.
- **Class 4: Mixed**, representing area covered by a combination of sand and coral.

Automatic mapping. To perform the automatic mapping, the image was first segmented using Felzenszwalb's method [33], which gives Fig 1. For each segment, two different operations were applied:

- The four statistical moments (mean, variance, skewness, kurtosis) were computed on each band; these 16 values will be the features of the dataset.
- The percentage of pixels belonging to each class were computed, according to the expert-based classification; this results in a vector that sums up to 1 that will be the labels of the dataset.

Table 1. Percentage of the number of pixels of each class on Maupiti data, based on expert mapping.

Class	Class 1	Class 2	Class 3	Class 4
Percentage	0.117	0.040	0.482	0.361

https://doi.org/10.1371/journal.pone.0287705.t001

To be able to map the satellite image, the idea was to train a regressor to retrieve, for each segment, the percentage of pixels belonging to each class (i.e., a vector of probabilities). As shown in Table 1, the data are not balanced: one of the class represents 49.5% of the dataset, while another one represents only 3.6%. To overcome this issue, we developed an oversampling technique in order to improve the performance of the regression model on this special kind of data.

Materials and method

Compositional data

Mathematically, we define a D-part compositional dataset as a vector $x = (x_1, x_2, \ldots, x_D) \in \mathbb{R}^D$ such that,

$$\begin{cases} x_i \geq 0, \quad \forall\, i \in \{1, 2, \ldots, D\}, \\ \sum_{i=1}^{D} x_i = 1. \end{cases}$$

A simplex S^D is defined as the ensemble of all the D-part compositional data, i.e.

$$S^D = \left\{ x = (x_1, x_2, ..., x_D) \mid \forall i \in \{1, 2, \ldots, D\}, x_i \geq 0; \sum_{i=1}^{D} x_i = 1 \right\}.$$

The operations performed in S^D must be adapted to follow the properties of the simplex [12]. For instance, before performing the Euclidian operations, it is possible to first apply the centred log-ratio transform $clr(\cdot)$ to the data,

$$\begin{aligned} clr : S^D &\to \mathbb{R}^D \\ (x_1, \ldots, x_D) &\mapsto \left(\log\left(\frac{x_1}{g(x)}\right), \ldots, \log\left(\frac{x_D}{g(x)}\right) \right). \end{aligned}$$

where the function $g(\cdot)$ is the geometric mean $g(x) = (\prod_{i=1}^{D} x_i)^{\frac{1}{D}}$. The $clr(\cdot)$ function is only defined for vectors where none of the value is equal to 0. Several methods exist to overcome this issue [34], but in practice we just replace the 0 by a tiny value such as 10^{-20}. The definition of the $clr(\cdot)$ function involves the existence of the inverse function $clr^{-1}(\cdot)$, that turns to be the softmax function, defined for $z = (z_1, \ldots, z_D) \in \mathbb{R}^D$ as

$$\text{softmax}(z) = \frac{1}{\sum_{i=1}^{D} \exp(z_i)} \cdot (\exp(z_1), \ldots, \exp(z_D)).$$

It is also possible to directly define operators on S^D. Let C be the closure operator,

$$\forall k \in \mathbb{N}, C(x_1, \ldots, x_k) = (x_1, \ldots, x_k)/(x_1 + \cdots + x_k).$$

For two D-part compositions $x, y \in S^D$, the perturbation $x \oplus y$ is defined by

$$x \oplus y = C(x_1 y_1, \ldots, x_D y_D), \tag{1}$$

and, given $\alpha \in \mathbb{R}$, the power transformed composition $\alpha \oplus x$ is

$$\alpha \otimes x = C(x_1^\alpha, \ldots, x_D^\alpha). \tag{2}$$

SMOTE for compositional data

In this section, we denote by n the number of samples in the dataset, p the number of features and K the number of classes. The matrix $X \in \mathbb{R}^{n \times p}$ contains the n observations of the p features and $Y \in \mathbb{R}^{n \times K}$ contains their labels. For any $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, K\}$, we denote by $y_{i,j}$ the value of Y at row i and column j, and $y_{i,\cdot} = (y_{i,1}, \ldots, y_{i,K})$ the probability vector label of row i. Similarly, for $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, p\}$, $x_{i,j}$ is the value of X at row i and column j, and $x_{i,\cdot} = (x_{i,1}, \ldots, x_{i,p})$. In order to simplify the notation, we define

$$\operatorname{argmax}(y_{i,\cdot}) = \underset{j \in \{1,\ldots,K\}}{\operatorname{argmax}}(y_{i,j}),$$

which represents the majority class of a given label $y_{i,\cdot} \in [0, 1]^K$. We also define the sum vector $S \in \mathbb{R}^K$ as the sum of the values for each class,

$$S = \left(\sum_{i=1}^{n} y_{i,1}, \sum_{i=1}^{n} y_{i,2}, \ldots, \sum_{i=1}^{n} y_{i,K} \right). \tag{3}$$

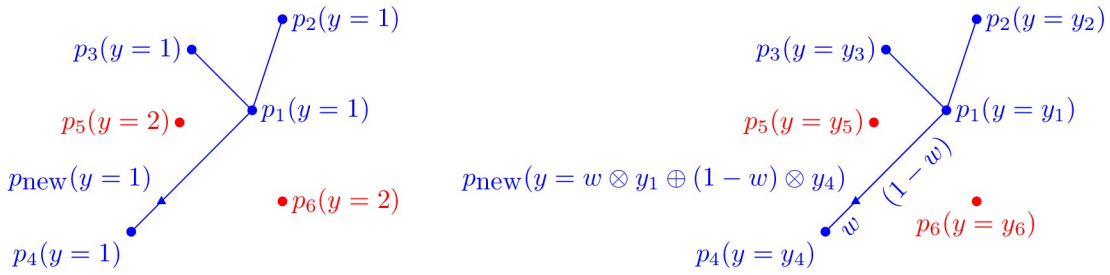
The majority class of the dataset is thus defined as argmax(S), and the minority class as argmin (S).

Before introducing the SMOTE-CD algorithm, let's first summarize the idea behind the original SMOTE algorithm. As shown in Fig 2(a), the SMOTE algorithm creates a new point that belongs to class 1 (represented by blue points). To achieve this, the algorithm first selects a point at random (in this case, p_1) and identifies its nearest neighbors (p_2, p_3, p_4). Note that only neighbors with the same label as p_1 (i.e., class 1) are considered, while points labeled as class 2 (represented by red points) are ignored. The algorithm then chooses one of these neighbors (p_4) and creates a new point along the line that connects p_1 and p_4. The features of the new point are determined through a linear combination of the features of p_1 and p_4, and its label is assigned as 1. Algorithm 1 describes the SMOTE algorithm.

Algorithm 1 Original SMOTE [3]

Require: $X \in \mathbb{R}^{n \times p}$ the features.

Require: $Y \in \{1, \ldots, J\}^n$ the class label outputs.



(a) Original SMOTE. (b) SMOTE for Compositional Data.

Fig 2. Difference between the original SMOTE algorithm and SMOTE-CD. The blue points are the points to oversample.(a) The points to oversample belong to the same class (here, class 1). (b) The points to oversample are the ones that have the same class as their majority class in their compositional vector label.

https://doi.org/10.1371/journal.pone.0287705.g002

Require: $k \in \mathbb{N}$ the number of neighbors to select for the k-Nearest Neighbors.
Ensure: Generated data $X_{new} \in \mathbb{R}^{q \times p}$ and $Y_{new} \in \{1, \ldots, J\}^q$ with q the number of points created.
1: Denote by S_j the number of points labeled as class j.
2: M ← the majority class of dataset.
3: Initialize X_{new} and Y_{new} as empty matrices.
4: **for** every class m that needs to be oversampled **do**
5: **while** $S_m < S_M$ **do**
6: Compute $\mathcal{D} = \{i \mid y_i = m\}$, the set of points labeled as class m.
7: Randomly choose $r_1 \in \mathcal{D}$ and find the indices of its k nearest neighbors.
8: Randomly choose an index r_2 among these neighbors.
9: $x^{new} \leftarrow w \times x_{r_1,\cdot} + (1 - w) \times x_{r_2,\cdot}$ with $w \in [0, 1]$ randomly drawn.
10: $y^{new} \leftarrow m$.
11: $S_m \leftarrow S_m + 1$.
12: Append x^{new} to X_{new}, append y^{new} to Y_{new}.
13: **end while**
14: **end for**
15: **return** X_{new}, Y_{new}

The SMOTE-CD algorithm keeps the main ideas from the original SMOTE: 1) select a point from the class to be oversampled, 2) select one of its k-Nearest Neighbors ($k \in \mathbb{N}$ specified by the user) and 3) create a synthetic point in-between those two points. Because of the label that is compositional, these three steps have to be adapted:

1. Select a point r_1 whose majority class is m, where m is the minority class of the dataset.
2. Compute the k-Nearest Neighbors of r_1 among the points that also have m as their majority class. Then select a point r_2 in one of these k neighbors.
3. Randomly draw $w \in [0, 1]$. The features of the point to be created is a linear combination of the two points selected before, with w being the weight of r_2 and $(1 - w)$ the weight of r_1. Similarly, the labels of the point to be created is a linear combination, but using the operators from Eqs (1) and (2).

Fig 2(b) depicts an example of how SMOTE-CD creates a new point. As we are dealing with compositional data label, every point p_i has a vector label y_i. All the blue points are the points having the class m as the majority class of their label y_i, where m is the minority class of the dataset. The algorithm computes the 3 nearest neighbors of p_1 only considering the blue points, and then a point is created on the line between p_1 and p_4. The label of the new point is a linear combination of the labels y_1 and y_4 using the operations defined on the simplex (Eqs (1) and (2)).

Algorithm 2 describes the SMOTE-CD algorithm, using the same notation.

Algorithm 2 SMOTE for compositional data
Require: $X \in \mathbb{R}^{n \times p}$ the features.
Require: $Y \in \mathbb{R}^{n \times K}$ the labels (compositional data).
Require: $k \in \mathbb{N}$ the number of neighbors to select for the k-Nearest Neighbors.
Ensure: Generated data $X_{new} \in R^{q \times p}$ and $Y_{new} \in R^{q \times K}$ with q the number of points created.
1: Compute the label sum vector $S \in \mathbb{R}^D$ as defined in Eq (3).
2: M ← argmax(S), the majority class of dataset (hence S_M is the sum of the majority class).
3: Initialize X_{new} and Y_{new} as empty matrices.
4: **while** min(S) < S_M **do**

5: $m \leftarrow$ argmin(S), the minority class of dataset.
6: Compute $\mathcal{D} = \{i \mid \text{argmax}(y_{i,\cdot}) = m\}$, the set of points whose majority class is m.
7: Randomly choose an index $r_1 \in \mathcal{D}$.
8: Find the indices of the k nearest neighbors of r_1 in $\mathcal{D}$, using the Euclidian distance on X.
9: Randomly choose an index r_2 among these indexes.
10: Uniformly draw a number $w \in [0, 1]$.
11: $x^{new} \leftarrow w \times x_{r_{1,\cdot}} + (1 - w) \times x_{r_{2,\cdot}}$.
12: $y^{new} \leftarrow w \otimes y_{r_{1,\cdot}} \otimes (1 - w)c \otimes y_{r_{2,\cdot}}$.
13: $S \leftarrow S + y^{new}$.
14: Append x^{new} to X_{new}, append y^{new} to Y_{new}.
15: **end while**
16: **return** X_{new}, Y_{new}

The step that creates the label of the new point (line 12) uses the definitions of Eqs (1) and (2). Nevertheless, it is also possible to create the label by using the Euclidian operations on the logratio transformed labels, and to apply the inverse transformation afterwards: $clr^{-1}(w \times clr(y_{r_{1,\cdot}}) + (1 - w) \times clr(y_{r_{2,\cdot}}))$. Although the label could be created by directly performing Euclidian operations on the compositional label, however this would be mathematically irrelevant because it would not respect the rules of compositional data analysis [35].

The proof of convergence holds in the fact that, at each iteration, the increase of the major class of S is smaller that the increase of its minor one, causing the sum of the minor class to converge to the sum of the major one. In other words, we have to be assured that, at each iteration, $y_m^{\text{new}} > y_M^{\text{new}}$, with m (resp. M) the minority (resp. majority) class of the dataset.

This is straightforward by noticing that the two indices r_1 and r_2 used for generating a new point are chosen in $\mathcal{D} = \{i \mid \text{argmax}(y_{i,\cdot}) = m\}$:

$$
\begin{aligned}
r_1, r_2 \in \mathcal{D} \quad &\Rightarrow \begin{cases} y_{r_1,m} > y_{r_1,M} \\ y_{r_2,m} > y_{r_2,M} \end{cases} \\
&\Rightarrow \begin{cases} w \otimes y_{r_1,m} > w \otimes y_{r_1,M} \\ (1-w) \otimes y_{r_2,m} > (1-w) \otimes y_{r_2,M} \end{cases} \\
&\Rightarrow w \otimes y_{r_1,m} + (1-w) \otimes y_{r_2,m} > w \otimes y_{r_1,M} + (1-w) \otimes y_{r_2,M} \\
&\Rightarrow y_m^{\text{new}} > y_M^{\text{new}}.
\end{aligned}
$$

Simulation study

Data simulation

The simulated data are generated by using a multinomial logistic regression. The main idea is to create a probability distribution from a multinomial logistic regression, and then use a Dirichlet distribution with those probabilities to generate the actual label of the new point.

The notation is the same as in the previous section: the number of features (resp. classes) is p (resp. K), and the number of samples is n. The user has to specify a matrix $B \in [0, 1]^{(p+1)\times K}$ which corresponds to the regression coefficients, where $B_{i,k}$ is associated with the ith feature and the kth class. For instance, for a class k, the regression coefficients will be ($B_{0,k}$, $B_{1,k}$, . . ., $B_{p,k}$). Note that $B_{0,k}$ is the intercept, hence explaining the $(p + 1) \times K$ dimension of B.

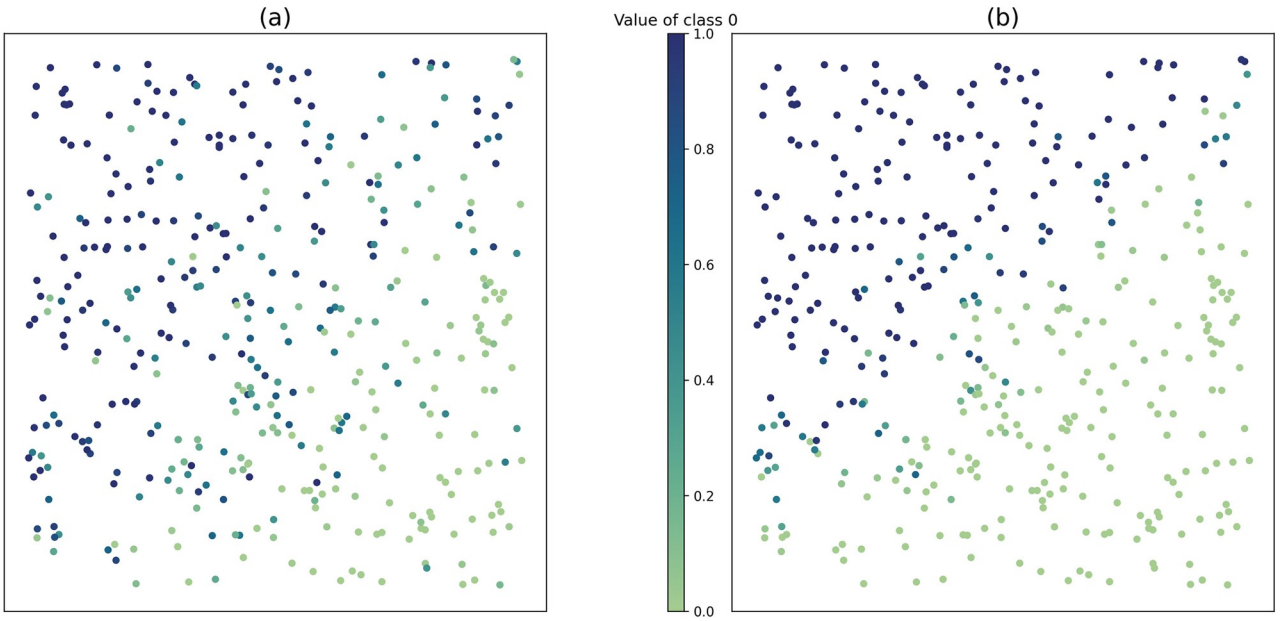


Fig 3. Simulation of 400 points using $B^{(a)}$ (a) and $B^{(b)}$ (b).

https://doi.org/10.1371/journal.pone.0287705.g003

For a given point $x = (x_1, \ldots, x_p) \in \mathbb{R}^p$, we define $x' = (1, x_1, \ldots, x_p) \in \mathbb{R}^{p+1}$ and a vector α as:

$$\begin{aligned}\alpha &= \text{softmax}(B_{0,1} + B_{1,1}x_1 + \cdots + B_{p,1}x_p, \ldots, B_{0,K} + B_{1,K}x_1 + \cdots + B_{p,K}x_p) \\ &= \text{softmax}(x' \cdot B_{\cdot,1}, \ldots, x' \cdot B_{\cdot,K}).\end{aligned}$$

We are then able to randomly draw a label for x with a Dirichlet distribution with parameter α. Algorithm 3 generates a random dataset using this method.

To better understand how the regression coefficients B can change the configuration of the data, we give an example of simulated data with 2 features and 2 labels. Two different values $B^{(a)}$ and $B^{(b)}$ are tested:

$$B^{(a)} = \begin{bmatrix} 0.4 & 0.4 \\ 0.2 & 0.4 \\ 0.5 & 0.3 \end{bmatrix}, \qquad B^{(b)} = \begin{bmatrix} 0.1 & 0.9 \\ 0.0 & 0.5 \\ 0.8 & 0.1 \end{bmatrix}.$$

Each column of a matrix B represents the coefficients for one class. There are 3 lines here because there are 2 features and the first value corresponds to the intercept of the regression. In $B^{(a)}$, the coefficients of each class are purposely close to each other, while they are easily separable in $B^{(b)}$. Fig 3 shows the value of the labels when generating the same 400 points with each matrix, using the function *generate_dataset* of our *smote-cd* Python package, with *random_state* = 2. The points created with $B^{(b)}$ have a clearer border between the points fully belonging in one class or the other. As there are only two classes and their sum is 1, it is only necessary to represent the value of one of them with the gradient of color.

Algorithm 3 Function to generate a synthetic dataset with compositional labels

Require: $K \in \mathbb{N}$ the number of classes.
Require: $p \in \mathbb{N}$ the number of features.
Require: $n \in \mathbb{N}$ the number of samples.

Require: $B \in [0, 1]^{(p+1)\times K}$ the regression coefficients, where $B_{m,k}$ is associated with the mth feature and the kth class.
Ensure: Generated data $X \in R^{n\times p}$ and $Y \in R^{n\times K}$
1: Create a random matrix of points $X \in R^{n\times p}$ such that for all i, j, $x_{i,j}$ is a random number uniformly drawn in a chosen interval (for instance [-10, 10])
2: Initialize Y as an empty matrix of size $(n \times K)$.
3: **for** every row x in X (and its associated row index i) **do**
4: Compute $\alpha = \text{softmax}(x' \cdot B_{\cdot,1}, \ldots, x' \cdot B_{\cdot,K})$ where $x' = (1, x_1, x_2, \ldots, x_p)$
5: Randomly draw a vector from a Dirichlet distribution with parameter α and attribute it to $y_{i,\cdot}$, the ith row of Y.
6: **end for**
7: **return** X, Y

Performance measures

The value of row i column j of Y is still denoted by $y_{i,j}$, and is the probability that the ith sample belongs to class j. Let $\hat{y}_{i,j}$ be the estimate of this probability by a model.

Different metrics can be used to measure the performance of the model. A popular metric is the cross-entropy:

$$CrossEntropy = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K} y_{i,j}\log(\hat{y}_{i,j} + \varepsilon). \tag{4}$$

The ε is added here to overcome the case where $\hat{y}_{i,j} = 0$. We chose $\varepsilon = 10^{-20}$. As the cross-entropy is a loss function, the smaller it is, the better the model performs. The cross-entropy loss may not always be suitable for our model because it treats each sample as equally important, without taking into account the imbalance of the test set. For instance, consider a model predicting three different classes (1, 2 and 3), and imagine that this model performs quite well on class 1 but poorly on classes 2 and 3. If the test set is imbalanced and has a large proportion of class 1 samples, the cross-entropy loss of this model will be low even though it performs poorly overall. The coefficient of determination R^2 allows assessment of the performance of a model on each of the K classes. For a class j, the coefficient of determination is given by

$$R_j^2 = 1 - \frac{\sum_{i=1}^{n}(y_{i,j} - \hat{y}_{i,j})^2}{\sum_{i=1}^{n}(y_{i,j} - \bar{y}_j)^2},$$

where $\bar{y}_j$ is the mean of the values of the jth class. The final R^2 will be equal to the average of the R_j^2 for each class j.

In addition, we also use the Root Mean Squared Error (RMSE) to measure the accuracy of the models. Since we are dealing with multi-class compositional vectors, we define the RMSE between a true and estimated vector as the average of RMSEs calculated across all their classes. Specifically, this is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\frac{1}{K}\sum_{i=1}^{n}\sum_{j=1}^{K}(y_{i,j} - \hat{y}_{i,j})^2}.$$

Even though we are working on a regression problem, classification metrics can be a good tool to understand the efficiency of the models. To do so, it is easy to transform a

compositional label $y_{i,\cdot}$ into a class y'_i by applying the argmax,

$$y'_i = \underset{j}{\text{argmax}}\ y_{i,j}.$$

The usual classification metrics can then be applied to y'. Here, we will use the accuracy (the number of correct points divided by the total number of points) and the F1-score which is computed per class,

$$\text{F1} - \text{score} \ = \frac{TP}{TP + \frac{1}{2}(FN + FP)},$$

where TP are the true positive, FN the false negative and FP the false positive. As with the R^2, the F1-score will be computed for each class and then averaged.

Results

First, to investigate the effect of the oversampling technique, synthetic data were generated with 2 features and 2 classes. To make the dataset imbalanced, 90% of the points that had class 0 as a majority class were deleted. We obtain a dataset in which 93% of the points have class 1 as their majority class (Fig 4(a)), which is then oversampled by selecting a number of nearest neighbors $k = 10$. Fig 4(b) displays the balanced dataset after applying SMOTE-CD, where the original points are displayed as circles and the synthetic created points are displayed as crosses. As in Fig 3, the gradient of color represents the value of one of the two classes.

To evaluate the performance of SMOTE-CD, a 5-fold cross validation was used for three models: Gradient Boosting tree (GB), Neural Network (NN) with one hidden layer, and Dirichlet regression model [36]. The first and second models are chosen because Random Forest and NN are known to be the most efficient to map coral reefs from multispectral satellites [37, 38] and because NN are used in literature for the task of predicting compositional labels [39, 40], and the third is chosen because it is used to generate the simulated data. For each model, the performance is compared between the raw and oversampled data. For the models

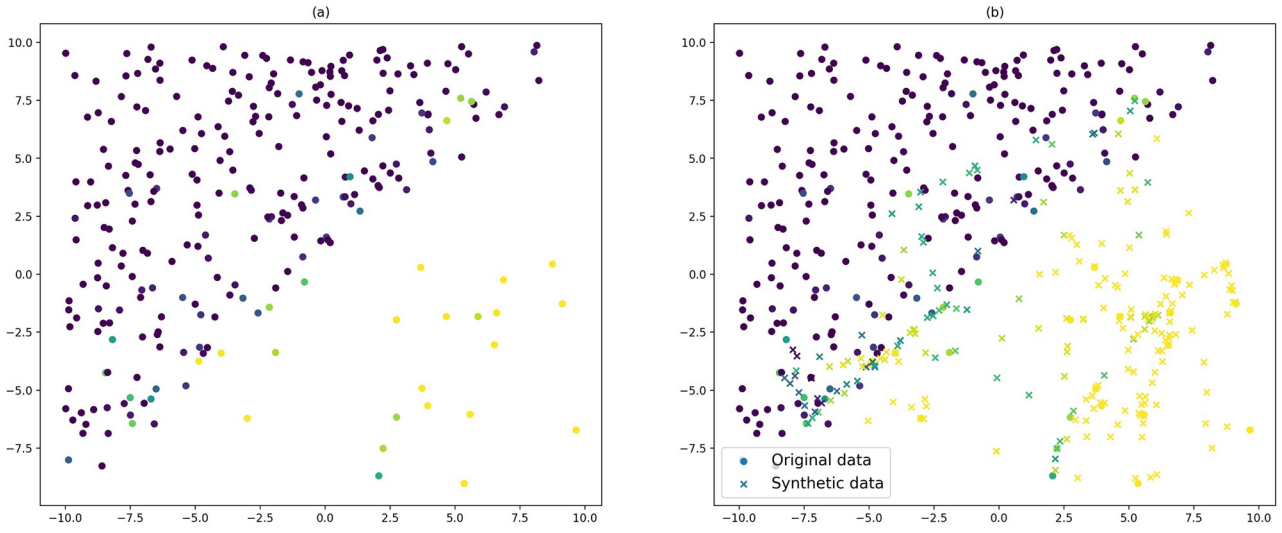


Fig 4. An example of SMOTE-CD. (a) The original imbalanced dataset, (b) the output balanced dataset with the created points displayed as a cross.

https://doi.org/10.1371/journal.pone.0287705.g004

Table 2. Comparison of simulated raw data (4 classes) and oversampled data, repeated 100 times. Displayed results are mean (s.d.).

	Accuracy	Cross-entropy	F1-score	RMSE	R^2
GB (raw)	0.692 (0.018)	5.272 (1.539)	0.532 (0.045)	0.363 (0.011)	0.137 (0.067)
GB (logratio)	0.724 (0.017)	2.508 (0.553)	0.658 (0.027)	0.341 (0.011)	0.198 (0.074)
GB (compositional)	0.683 (0.016)	3.657 (1.055)	0.604 (0.038)	0.359 (0.011)	0.139 (0.085)
NN (raw)	0.772 (0.026)	3.340 (1.370)	0.611 (0.057)	0.315 (0.020)	0.298 (0.103)
NN (logratio)	0.784 (0.023)	1.700 (0.380)	0.729 (0.033)	0.304 (0.018)	0.301 (0.108)
NN (compositional)	0.750 (0.054)	3.483 (1.367)	0.690 (0.063)	0.332 (0.040)	0.198 (0.207)
Dirichlet (raw)	0.789 (0.016)	0.685 (0.017)	0.605 (0.039)	0.287 (0.004)	0.416 (0.022)
Dirichlet (logratio)	0.875 (0.010)	0.754 (0.017)	0.824 (0.019)	0.303 (0.004)	0.380 (0.022)
Dirichlet (compositional)	0.874 (0.011)	0.755 (0.017)	0.824 (0.019)	0.303 (0.004)	0.379 (0.022)

https://doi.org/10.1371/journal.pone.0287705.t002

on which it is possible (GB and NN), hyperparameter tuning was been performed for each data (raw or oversampled). The hyperparameters are detailed in S1 and S2 Tables.

The simulated data were generated with the same shape as the Maupiti data. We selected a matrix B such that the imbalance of the classes was similar to the one of the real data (see Table 1). Then, 550 points were created with 16 features and 4 classes to train the models. Testing was performed with 11000 points (20 times the training set size). This operation was repeated 100 times with the same B. The results and metrics (accuracy, cross-entropy, average F1, RMSE and R^2) are presented in Table 2.

For both the Gradient Boosting and Neural Network models, the oversampling with logratio distance significantly improves all metrics except for R^2 on the Neural Network ($p < 0.0006$). With the compositional distance on the Neural Network, only the F1-score significantly increases ($p \ll 10^{-10}$), while accuracy, RMSE, and R^2 decrease. The GB model shows significant improvement for cross-entropy, F1-score, and RMSE ($p < 0.008$), but a decrease in accuracy. The Dirichlet model with oversampling significantly increases accuracy and F1-score ($p \ll 10^{-10}$) but decreases cross-entropy, RMSE, and R^2.

In order to understand the effects of the imbalance of the dataset on the performance of the oversampling method, three metrics (accuracy, F1 and R^2) were evaluated with different imbalance ratios. First, a matrix B was created to generate a balanced dataset with 16 features and 4 classes. Then, the ratio of class 0 was increased by incrementing the value of $B_{1,1}$. At each step (for a total of ten steps), the following operation was repeated 100 times: 550 points were created to train the models on the raw or oversampled data, and the models were tested on a set of 11000 points. The result appears in Fig 5.

It is apparent that the efficiency of SMOTE-CD depends on the data and the model used. The oversampling technique only improves the R^2 score when the dataset is slightly imbalanced (largest class representing less than 40%), but performs poorly when it is highly imbalanced. On the other hand, the more the dataset is imbalanced, the more the oversampling technique will improve the F1-score. The improvement in accuracy peaks at a certain value of imbalance (when the largest class represents 50% of the dataset), but drops above that threshold.

In order to explain the low R^2 score for the oversampled data, the R^2 per class was calculated for each of the ten steps mentioned above and then averaged. Fig 6 displays the result. The average imbalance ratio is 52% for class 0 (and thus approximately 16% for the three other classes).

For the largest class, the R^2 score is decreased from 0.5 to 0.3 by the oversampling technique, which explains why the raw score is higher than the oversampled score in Fig 5.

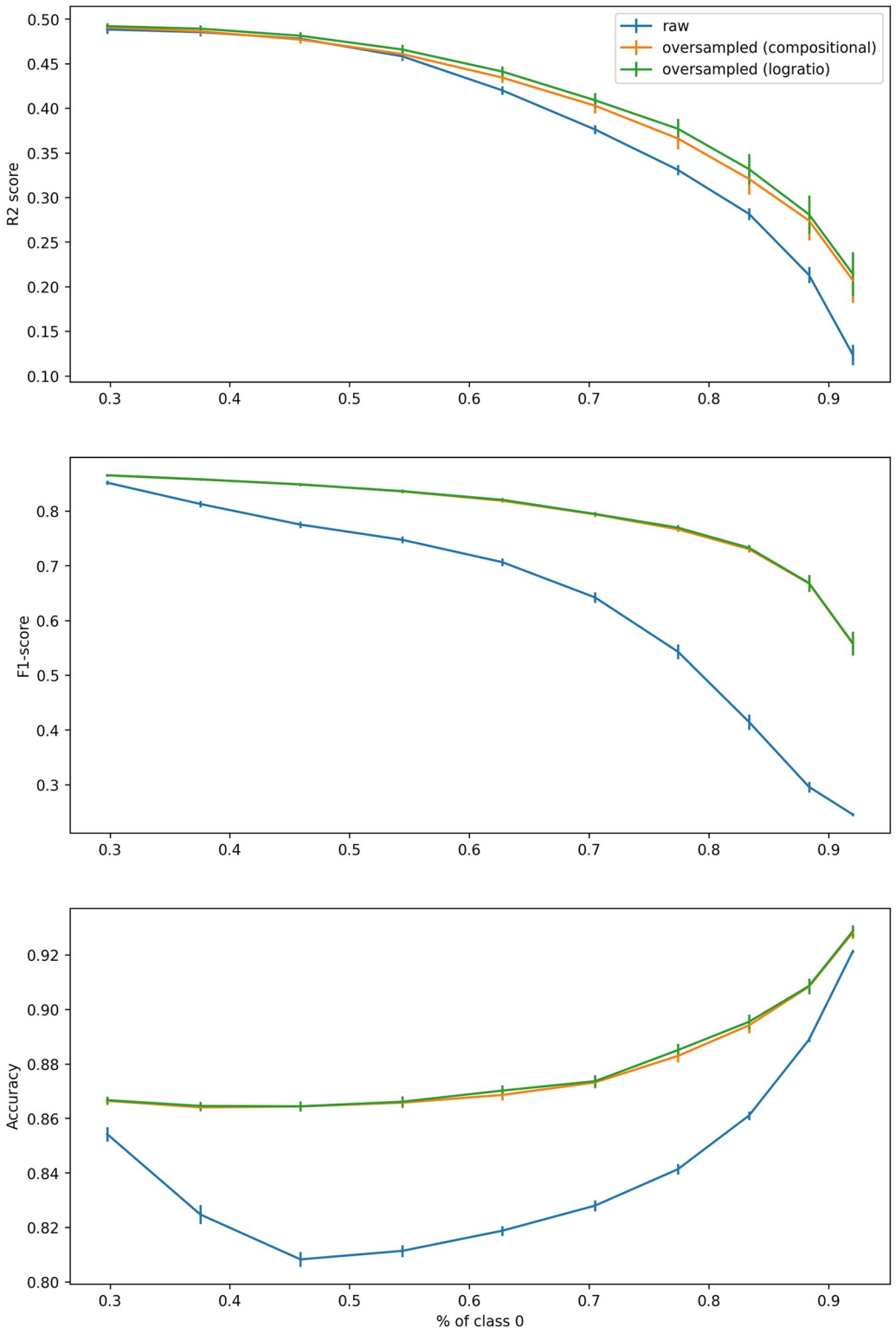


Fig 5. Performance of Dirichlet model on raw and oversampled data, depending on the imbalance of the dataset (indicated by % of observations in class 0), based on 16 features and 4 classes.

https://doi.org/10.1371/journal.pone.0287705.g005

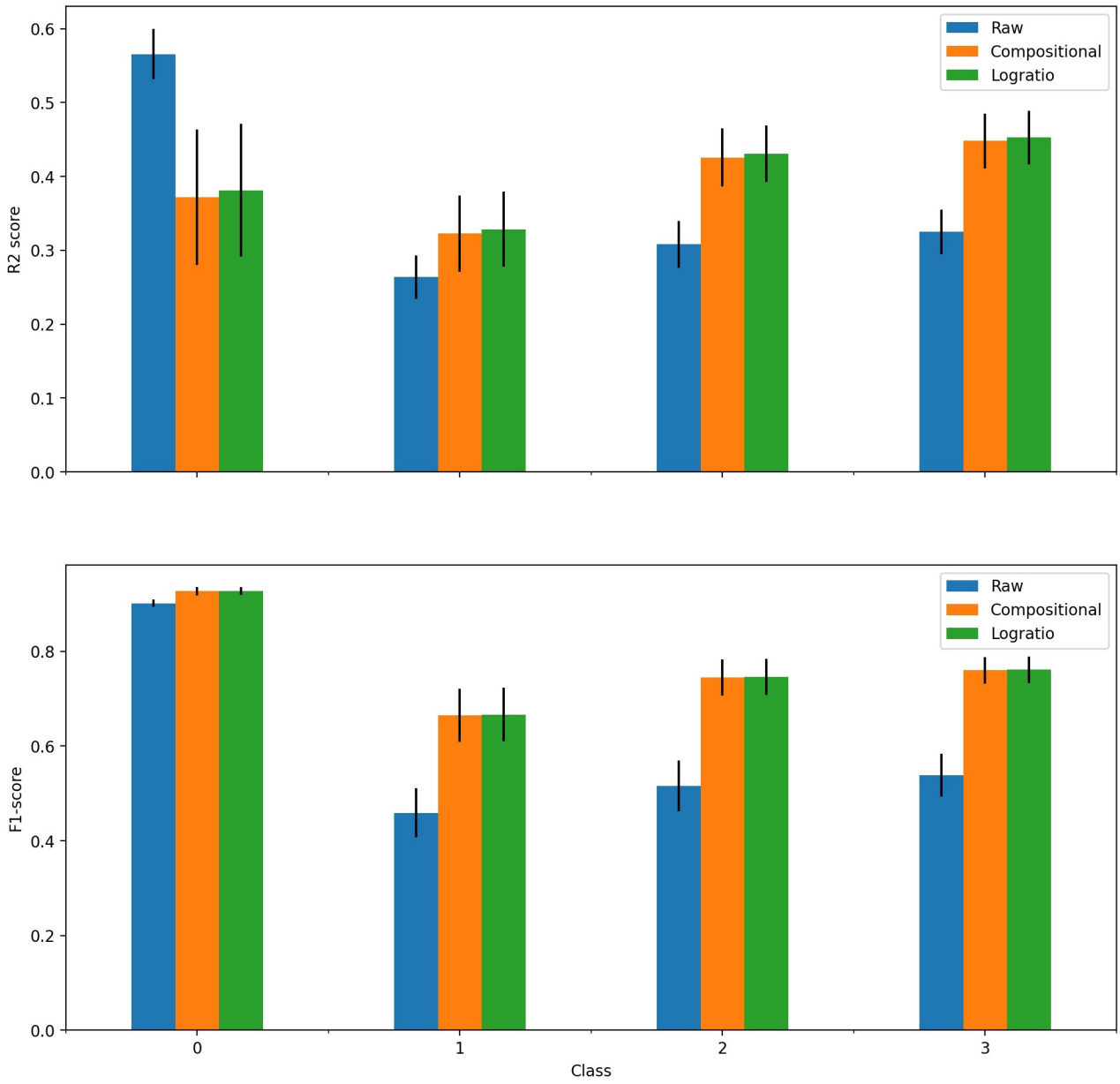


Fig 6. Average R^2 and F1-score per class of Dirichlet model on raw and oversampled simulated data. Bars represent the mean score, vertical lines represent the standard deviation.

https://doi.org/10.1371/journal.pone.0287705.g006

However, for the three minority classes, the R^2 is increased by approximately 0.05, which is the initial goal of the method.

Similarly, Fig 6 also depicts the F1-score per class, averaged over the seven steps. The difference is that the F1-score of the majority class is not decreased by the oversampling technique, while the score of the minority classes is increased by approximately 0.08.

Application to Maupiti data

The performance of the three models on the raw dataset was compared with the oversampled dataset (with either the logratio distance used to create the new labels, or the compositional

Table 3. Results comparing raw Maupiti data (4 classes) and oversampled with a 5-fold cross validation. Displayed results are mean (s.d.).

	Accuracy	Cross-entropy	F1-score	RMSE	R^2
GB (raw)	0.857 (0.003)	2.538 (0.196)	0.809 (0.031)	0.229 (0.003)	0.583 (0.018)
GB (logratio)	0.859 (0.003)	2.504 (0.182)	0.822 (0.028)	0.226 (0.003)	0.596 (0.019)
GB (compositional)	0.859 (0.004)	2.486 (0.149)	0.822 (0.028)	0.226 (0.003)	0.596 (0.018)
NN (raw)	0.877 (0.003)	4.048 (0.416)	0.831 (0.008)	0.214 (0.003)	0.624 (0.018)
NN (logratio)	0.877 (0.003)	3.982 (0.456)	0.835 (0.009)	0.214 (0.003)	0.623 (0.017)
NN (compositional)	0.878 (0.003)	3.956 (0.406)	0.834 (0.010)	0.213 (0.003)	0.622 (0.020)
Dirichlet (raw)	0.801 (0.056)	1.676 (0.874)	0.684 (0.127)	0.262 (0.033)	0.420 (0.163)
Dirichlet (logratio)	0.810 (0.049)	1.663 (0.851)	0.762 (0.064)	0.262 (0.036)	0.423 (0.174)
Dirichlet (compositional)	0.810 (0.049)	1.654 (0.839)	0.762 (0.064)	0.262 (0.036)	0.423 (0.174)

https://doi.org/10.1371/journal.pone.0287705.t003

distance). The results are shown in Table 3. With the Maupiti dataset, the NN is defined with 2 hidden layers of size 80 and 40, and the relu activation function.

With the GB model, all the metrics are significantly improved ($p < 0.03$) when using the oversampling technique, excepted for the cross-entropy for which the differences are not statistically significant ($p = 0.14$ and $p = 0.38$ respectively for the compositional and the logratio distance). The SMOTE-CD shows less results with the NN and Dirichlet model, where only the difference on the F1 is statistically significant (respectively $p < 0.044$) and $p < 10^{-10}$). This improvement is quite important for the Dirichlet model though, as it represents a difference of almost 0.08.

We analyze the per-class R^2 of the Gradient Boosting tree as it is the best model. Fig 7 compares the R^2 between the raw and oversampled data. The oversampling technique decreases

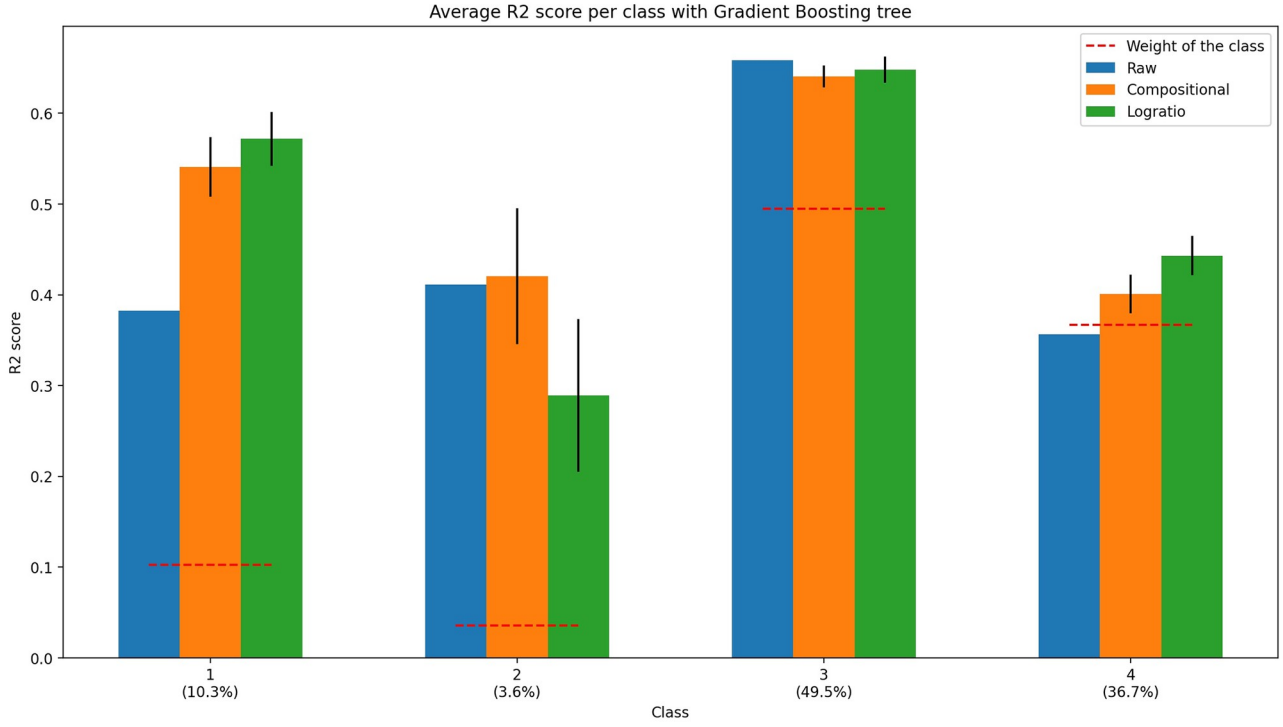


Fig 7. Average R^2 score per class of Gradient Boosting tree on raw and oversampled Maupiti data. The red dotted lines represent the weight of each class, and the value below the class is its weight. Bars represent the mean score, vertical lines represent the standard deviation.

https://doi.org/10.1371/journal.pone.0287705.g007

the performance of the model for the smallest class (Class 2) for the logratio distance, does not change for the largest class (Class 3) and increases the performance on the others (Classes 1 and 4).

We conclude that SMOTE-CD does not improve the performance for a class that is too small: in order to perform ideally, it requires enough points to oversample.

Application to Tecator dataset

To fully evaluate the effectiveness of the SMOTE-CD technique, we applied it to the Tecator meat sample dataset [41], which consists of 240 meat samples. Each sample has absorbance values measured at 100 different wavelengths, as well as corresponding information on the composition of moisture (water), fat, and protein contents. The objective of this analysis is to predict a 3-class compositional data vector from a feature vector of size 100. Because the Dirichlet regression model can be very slow when dealing with a high number of features, we opted to improve its speed by using only the 22 principal components provided in the dataset instead of the 100 features.

To account for the small size of the dataset, a 10-fold cross validation is applied for each model, iterated over 100 times to vary the folds. The results are displayed in Table 4. The neural network is configured with three hidden layers, each having 70 neurons and using the hyperbolic tangent (tanh) activation function, which were selected through hyperparameter tuning.

With the NN, the raw data gives slightly better performances than the oversampled data. However, given the really poor performances of the NN (a negative R^2 and a really high RMSE), we also note that this model was probably not suited for this dataset.

The analysis of the GB and Dirichlet models reveals interesting differences. In both cases, using either the raw or oversampled datasets leads to statistically significant differences ($p < 10^{-4}$). Specifically, for the GB model, using the oversampled data results in better performance, while for the Dirichlet model, oversampling decreases the performance. Notably, among all the models tested, the GB model trained on oversampled data with compositional distance yields the best results. Compared to the Dirichlet model trained on raw data, this approach achieves significantly better accuracy ($p < 0.006$), RMSE ($p \ll 10^{-10}$), and R^2 ($p < 10^{-4}$), with only a slight difference of 1% in cross-entropy and F1-score.

In light of these results, it is apparent that SMOTE-CD can improve the performance for a model that does not perform too poorly (e.g. a R^2 above 0.3). Indeed, if a model has low performance, it is more likely that this is due to poor fit to the data than from the imbalance of the dataset.

Table 4. Results comparing raw Tecator data (3 classes) and oversampled with a 10-fold cross validation, iterated 100 times. Displayed results are mean (s.d.).

Model	Accuracy	Cross-entropy	F1-score	RMSE	R^2
GB (raw)	0.932 (0.006)	0.860 (0.001)	0.701 (0.042)	0.046 (0.001)	0.717 (0.023)
GB (logratio)	0.957 (0.008)	0.860 (0.001)	0.830 (0.046)	0.044 (0.002)	0.730 (0.027)
GB (compositional)	0.957 (0.008)	0.860 (0.002)	0.834 (0.046)	0.044 (0.002)	0.730 (0.026)
NN (raw)	0.908 (0.000)	0.928 (0.009)	0.512 (0.036)	0.113 (0.005)	-1.230 (0.484)
NN (logratio)	0.904 (0.014)	0.938 (0.010)	0.513 (0.044)	0.122 (0.007)	-1.156 (0.449)
NN (compositional)	0.904 (0.016)	0.937 (0.010)	0.512 (0.044)	0.122 (0.007)	-1.158 (0.466)
Dirichlet (raw)	0.954 (0.007)	0.852 (0.003)	0.846 (0.037)	0.048 (0.003)	0.708 (0.045)
Dirichlet (logratio)	0.940 (0.011)	0.878 (0.006)	0.800 (0.044)	0.072 (0.006)	0.310 (0.224)
Dirichlet (compositional)	0.940 (0.011)	0.877 (0.005)	0.802 (0.037)	0.072 (0.005)	0.323 (0.413)

https://doi.org/10.1371/journal.pone.0287705.t004

Discussion

The results on the synthetic datasets show that the SMOTE-CD technique can significantly improve the F1-score and accuracy, but it has a mixed effect on other metrics depending on the model and dataset imbalance level. SMOTE-CD improves the overall performance of the model, especially with respect to the accuracy and the F1-score in the cases where the dataset is not too heavily imbalanced. The R^2 score of the majority class remains similar, but the R^2 of a very small class (3% of the dataset) will be decreased. The R^2 of all the other classes is improved, which is the desired goal of the method.

The results on the real datasets show that the SMOTE-CD technique can significantly improve the performance of the Gradient Boosting model for all metrics, while it has a less pronounced effect on the other models. The per-class analysis of the R^2 score reveals that the SMOTE-CD technique can improve the performance for some classes but not for others, depending on the model and distance metric used.

Further tests are required with other datasets having compositional labels, but these are often hard to find because they are not publicly available. Our oversampling technique could be used with datasets in biology and metabolomics, in poll studies or in soil analysis, but its effectiveness depends on several factors that should be carefully considered.

The original SMOTE paper [3] proposes to undersample the dataset before applying the oversampling technique, which we similarly tested here. The synthetic dataset was first undersampled by randomly withdrawing some points from the majority class, until the total sum of the largest class was equal to the sum of the second largest one. SMOTE-CD was then applied. The results are summarised in S3 Table and compared with those in Table 2 when not using undersampling (S4 Table). No significant difference can be seen when using undersampling before the oversampling, be it positive or negative. The results are similar when undersampling not only the points having the largest class as their majority class, but the points having one of the n largest classes as their majority class (with $n \in [1, \ldots, 3]$). At this point, we are not able to exclude the utility of the undersampling and suggest it could once more depend on the dataset or on the way the removed points are chosen. For instance, when performing random undersampling, consideration could be given to an Edited Nearest Neighbor approach [42]; see [43].

Work has still to be done regarding the initial selection of the points, because it can influence the performance of the original SMOTE algorithm. For instance, we could imagine attributing a "safe" level to each point by exploring its k nearest neighbors and using it in the creation of a new point [44]. It would also be possible to only oversample the points on the border [45], where the border would here be defined by the points having a given amount of neighbors that have the largest class as their majority class.

Conclusion

The SMOTE algorithm has been adapted to deal with the special case in which the dataset labels are compositional, which had not been done before. The present study investigates its effectiveness on imbalanced datasets for three different models: Gradient Boosting tree, Neural Networks, and Dirichlet Regression. The evaluation was performed on both synthetic and real datasets, and several metrics, including accuracy, F1-score, RMSE, cross-entropy, and R^2, were used to assess the performance of the models.

The study suggests that the effectiveness of the SMOTE-CD technique depends on several factors, including the model, distance metric, dataset imbalance level, and class distribution. The SMOTE-CD technique can improve the performance of a model that does not perform too poorly, but it may not be effective for a model with very low performance.

An implementation is proposed in the Python package *smote-cd* available on PyPi: https://pypi.org/project/smote-cd. The Jupyter notebooks used to simulate the data and perform the analyses can be found on the GitHub page of the package: https://github.com/teongu/smote_cd.

Supporting information

S1 Table. Hyperparameters of the Gradient Boosting tree. The hyperparameters listed here are those applied to the Gradient Boosting tree of the Python package *scikit-learn*, tuned with the *hyperopt* package. The value of the *random_state* is 2.
(PDF)

S2 Table. Hyperparameters of the Neural Networks. The hyperparameters listed here are those applied to the MLPRegressor of the Python package *scikit-learn*, tuned with the *hyperopt* package. The value of the *random_state* is 2.
(PDF)

S3 Table. Results comparing simulated raw data (4 classes) and oversampled repeated 100 times, when applying undersampling beforehand.
(PDF)

S4 Table. Difference when applying undersampling+oversampling, and oversampling only. Results are in bold when the undersampling provides better results.
(PDF)

Author Contributions

Data curation: Damien Sous.

Methodology: Teo Nguyen, Benoit Liquet.

Resources: Damien Sous.

Supervision: Kerrie Mengersen, Benoit Liquet.

Validation: Benoit Liquet.

Writing – original draft: Teo Nguyen.

Writing – review & editing: Kerrie Mengersen, Damien Sous, Benoit Liquet.

References

1. Haixiang G, Yijing L, Shang J, Mingyun G, Yuanyue H, Bing G. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications. 2017; 73:220–239. https://doi.org/10.1016/j.eswa.2016.12.035
2. Johnson JM, Khoshgoftaar TM. Survey on deep learning with class imbalance. Journal of Big Data. 2019; 6(1):1–54. https://doi.org/10.1186/s40537-019-0192-5
3. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research. 2002; 16:321–357. https://doi.org/10.1613/jair.953
4. Fernández A, Garcia S, Herrera F, Chawla NV. SMOTE for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. Journal of Artificial Intelligence Research. 2018; 61:863–905. https://doi.org/10.1613/jair.1.11192
5. Bountzouklis C, Fox DM, Di Bernardino E. Predicting wildfire ignition causes in Southern France using eXplainable Artificial Intelligence (XAI) methods. Environmental Research Letters. 2023; 18(4):044038. https://doi.org/10.1088/1748-9326/acc8ee
6. Chemchem A, Alin F, Krajecki M. Combining SMOTE sampling and machine learning for forecasting wheat yields in France. In: 2019 IEEE second international conference on artificial intelligence and knowledge engineering (AIKE). IEEE; 2019. p. 9–14.

7. Ijaz MF, Alfian G, Syafrudin M, Rhee J. Hybrid prediction model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, synthetic minority over sampling technique (SMOTE), and random forest. Applied Sciences. 2018; 8(8):1325. https://doi.org/10.3390/app8081325
8. Kogut T, Tomczak A, Słowik A, Oberski T. Seabed modelling by means of airborne laser bathymetry data and imbalanced learning for offshore mapping. Sensors. 2022; 22(9):3121. https://doi.org/10.3390/s22093121 PMID: 35590809
9. Phanomsophon T, Jaisue N, Worphet A, Tawinteung N, Shrestha B, Posom J, et al. Rapid measurement of classification levels of primary macronutrients in durian (Durio zibethinus Murray CV. Mon Thong) leaves using FT-NIR spectrometer and comparing the effect of imbalanced and balanced data for modelling. Measurement. 2022; 203:111975. https://doi.org/10.1016/j.measurement.2022.111975
10. Torgo L, Branco P, Ribeiro RP, Pfahringer B. Resampling strategies for regression. Expert Systems. 2015; 32(3):465–476. https://doi.org/10.1111/exsy.12081
11. Perez-Ortiz M, Gutierrez PA, Hervas-Martinez C, Yao X. Graph-based approaches for over-sampling in the context of ordinal regression. IEEE Transactions on Knowledge and Data Engineering. 2014; 27 (5):1233–1245. https://doi.org/10.1109/TKDE.2014.2365780
12. Aitchison J. The statistical analysis of compositional data. Journal of the Royal Statistical Society: Series B (Methodological). 1982; 44(2):139–160.
13. Shi P, Zhang A, Li H. Regression analysis for microbiome compositional data. The Annals of Applied Statistics. 2016; 10(2):1019–1040. https://doi.org/10.1214/16-AOAS928
14. Tsilimigras MC, Fodor AA. Compositional data analysis of the microbiome: fundamentals, tools, and challenges. Annals of epidemiology. 2016; 26(5):330–335. https://doi.org/10.1016/j.annepidem.2016.03.002 PMID: 27255738
15. Xia F, Chen J, Fung WK, Li H. A logistic normal multinomial regression model for microbiome compositional data analysis. Biometrics. 2013; 69(4):1053–1063. https://doi.org/10.1111/biom.12079 PMID: 24128059
16. Acquah GE, Via BK, Fasina OO, Adhikari S, Billor N, Eckhardt LG. Chemometric modeling of thermogravimetric data for the compositional analysis of forest biomass. PLOS ONE. 2017; 12(3):1–15. https://doi.org/10.1371/journal.pone.0172999 PMID: 28253322
17. Francis I, Newton J. Determining wine aroma from compositional data. Australian Journal of Grape and Wine Research. 2005; 11(2):114–126. https://doi.org/10.1111/j.1755-0238.2005.tb00283.x
18. Jackson DA. Compositional data in community ecology: the paradigm or peril of proportions? Ecology. 1997; 78(3):929–940. https://doi.org/10.1890/0012-9658(1997)078%5B0929:CDICET%5D2.0.CO;2
19. Vercelloni J, Liquet B, Kennedy EV, González-Rivero M, Caley MJ, Peterson EE, et al. Forecasting intensifying disturbance effects on coral reefs. Global change biology. 2020; 26(5):2785–2797. https://doi.org/10.1111/gcb.15059 PMID: 32115808
20. Buccianti A, Pawlowsky-Glahn V. New perspectives on water chemistry and compositional data analysis. Mathematical Geology. 2005; 37:703–727. https://doi.org/10.1007/s11004-005-7376-6
21. Coakley JP, Rust B. Sedimentation in an Arctic lake. Journal of Sedimentary Research. 1968; 38 (4):1290–1300.
22. de Faria FR, Barbosa D, Howe CA, Canabrava KLR, Sasaki JE, dos Santos Amorim PR. Time-use movement behaviors are associated with scores of depression/anxiety among adolescents: A compositional data analysis. PLOS ONE. 2022; 17(12):1–12. https://doi.org/10.1371/journal.pone.0279401 PMID: 36584176
23. Wei Y, Wang Z, Wang H, Yao T, Li Y. Promoting inclusive water governance and forecasting the structure of water consumption based on compositional data: A case study of Beijing. Science of the Total Environment. 2018; 634:407–416. https://doi.org/10.1016/j.scitotenv.2018.03.325 PMID: 29627564
24. Wei Y, Wang Z, Wang H, Li Y, Jiang Z. Predicting population age structures of China, India, and Vietnam by 2030 based on compositional data. PLOS ONE. 2019; 14(4):1–42. https://doi.org/10.1371/journal.pone.0212772 PMID: 30973941
25. Camacho Luís and Douzas Georgios and Bacao Fernando. Geometric SMOTE for regression. Expert Systems with Applications. 2022; 193:116387. https://doi.org/10.1016/j.eswa.2021.116387
26. Huang Y, Liu DR, Lee SJ, Hsu CH, Liu YG. A boosting resampling method for regression based on a conditional variational autoencoder. Information Sciences. 2022; 590:90–105. https://doi.org/10.1016/j.ins.2021.12.100
27. Moniz N, Ribeiro R, Cerqueira V, Chawla N. Smoteboost for regression: Improving the prediction of extreme values. In: 2018 IEEE 5th international conference on data science and advanced analytics (DSAA). IEEE; 2018. p. 150–159.

28. Torgo L, Ribeiro RP, Pfahringer B, Branco P. Smote for regression. In: Progress in Artificial Intelligence: 16th Portuguese Conference on Artificial Intelligence, EPIA 2013, Angra do Heroísmo, Azores, Portugal, September 9-12, 2013. Proceedings 16. Springer; 2013. 378–389.

29. Charte F, Rivera AJ, del Jesus MJ, Herrera F. MLSMOTE: Approaching imbalanced multilabel learning through synthetic instance generation. Knowledge-Based Systems. 2015; 89:385–397. https://doi.org/10.1016/j.knosys.2015.07.019

30. Deng M, Guo Y, Wang C, Wu F. An oversampling method for multi-class imbalanced data based on composite weights. PLOS ONE. 2021; 16(11):1–15. https://doi.org/10.1371/journal.pone.0259227 PMID: 34767567

31. Gordon-Rodriguez E, Quinn T, Cunningham JP. Data Augmentation for Compositional Data: Advancing Predictive Models of the Microbiome. Advances in Neural Information Processing Systems. 2022; 35:20551–20565.

32. Sous D, Bouchette F, Doerflinger E, Meulé S, Certain R, Toulemonde G, et al. On the small-scale fractal geometrical structure of a living coral reef barrier. Earth Surface Processes and Landforms. 2020; 45 (12):3042–3054. https://doi.org/10.1002/esp.4950

33. Felzenszwalb PF, Huttenlocher DP. Efficient graph-based image segmentation. International Journal of Computer Vision. 2004; 59(2):167–181. https://doi.org/10.1023/B:VISI.0000022288.19776.77

34. Scealy J, Welsh A. Regression for compositional data by using distributions defined on the hypersphere. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2011; 73(3):351–375. https://doi.org/10.1111/j.1467-9868.2010.00766.x

35. Aitchison J, Barceló-Vidal C, Martín-Fernández JA, Pawlowsky-Glahn V. Logratio analysis and compositional distance. Mathematical Geology. 2000; 32(3):271–275. https://doi.org/10.1023/A:1007529726302

36. Maier, M. DirichletReg: Dirichlet regression for compositional data in R. 2014

37. Nguyen T, Liquet B, Mengersen K, Sous D. Mapping of Coral Reefs with Multispectral Satellites: A Review of Recent Papers. Remote Sensing. 2021; 13(21):4470. https://doi.org/10.3390/rs13214470

38. Li J, Knapp DE, Fabina NS, Kennedy EV, Larsen K, Lyons MB, et al. A global coral reef probability map generated using convolutional neural networks. Coral Reefs. 2020; 39:1805–1815. https://doi.org/10.1007/s00338-020-02005-6

39. Ma S, Zhou C, Chi C, Liu Y, Yang G Estimating physical composition of municipal solid waste in China by applying artificial neural network method. Environmental science & technology. 2020; 54(15):9609–9617. https://doi.org/10.1021/acs.est.0c01802

40. Hoy ZX, Woon KS, Chin WC, Hashim H, Van Fan Y Forecasting heterogeneous municipal solid waste generation via Bayesian-optimised neural network with ensemble learning for improved generalisation. Computers & Chemical Engineering. 2022; 166:107946. https://doi.org/10.1016/j.compchemeng.2022.107946

41. Tecator meat sample dataset. http://lib.stat.cmu.edu/datasets/tecator

42. Wilson DL. Asymptotic properties of nearest neighbor rules using edited data. IEEE Transactions on Systems, Man, and Cybernetics. 1972; SMC-2(3):408–421. https://doi.org/10.1109/TSMC.1972.4309137

43. Batista GE, Prati RC, Monard MC. A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD explorations newsletter. 2004; 6(1):20–29. https://doi.org/10.1145/1007730.1007735

44. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: Pacific-Asia conference on knowledge discovery and data mining. Springer; 2009. p. 475–482.

45. Han H, Wang WY, Mao BH. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: International conference on intelligent computing. Springer; 2005. p. 878–887.