Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

RESEARCH REPORT

The harm of SMOTE and other imbalance adjustments: a case study

Ruben van den Goorbergh

Correspondence
Ruben van den Goorbergh, Julius Center for Health Sciences and Primary Care,University Medical Center Utrecht,Utrecht University, P.O. Box 85500, 3508GA Utrecht, The Netherlands. Email: r.w.vandengoorbergh@uu.nl

Summary

In machine learning applications, class imbalance is often adjusted for using data pre-processing methods. I present a case study showing the effect of pre-processing methods on the model performance in low dimensional data. All pre-processing methods yield prediction models that overestimate the risk predictions while not improving the discriminative ability. I advise to apply great caution when using imbalance adjustments if the calibration of the prediction model is of interest.

KEYWORDS:
Clinical risk prediction models, logistic regression, random forest, gradient boosting machines, XGboost, class imbalance, SMOTE, oversampling, undersampling

1 | INTRODUCTION

In the field of prediction modelling and machine learning, class imbalance is a well-known phenomenon.[?] Class imbalance refers to the situation in which the event rate in a data set is not 50%.[?] In medical data, the event of interest is often rare or sometimes very rare, which can result in severe class imbalance. The traditional approach for developing dichotomous clinical risk prediction models involves the use of logistic regression models[?], which do not necessarily suffer from unevenly distributed classes.[?] However, imbalanced data may lead to problems for many other machine learning algorithms because of their accuracy-oriented design, resulting in overlooking the minority class.[?] Therefore, solutions to imbalanced data could be of great interest for clinical prediction modelers, as interest in exploiting complex machine learning models is growing.[?]

One category of solutions for dealing with imbalanced data is to pre-process the data using resampling techniques, where part of the data in the majority class is discarded (undersampling) or the data of the minority class is duplicated (oversampling). This results in an artificial, more balanced data set.[?] [?] However, in a medical setting, it is of paramount importance that the prediction model is not only able to accurately distinguish events and non-events, but also able to accurately estimate the risk of event (i.e. the calibration of the model is important). This means that there should be an overall agreement between the observed outcomes and risk estimates.[?] Combining the notion of calibration with the fact that, in medicine, the diseases and other health outcomes we want to predict often lead to imbalanced data sets, raises the question how adjustments made for class imbalance affect the performance of the models used for making these predictions.

In this research report, I will investigate the performance of two regression-based models (regular- and ridge logistic regression) and two tree-based models (random forest and gradient boosting) for dichotomous risk prediction on imbalanced data. The study focuses on both the discriminative performance and calibration of these models when the imbalance is either adjusted for by one of the following data pre-processing methods: random undersampling (RUS), random oversampling (ROS) or Synthetic Minority Oversampling Technique (SMOTE).[?]

This report is structured as follows. In the next section, I will describe the used models, class imbalance approaches, performance metrics and data. In section three I present the results of the analysis. In the last section the results are discussed and a simulation study is proposed.

2 | METHODS

2.1 | Models

2.1.1 | Regression-based models

In logistic regression the probability ($\pi_i(\mathbf{x}_i)$) of a positive outcome for person i ($y_i = 1$) is modelled using as set of R predictors $\mathbf{x}_i$ by estimating regression coefficients β_j and intercept α. Let $\boldsymbol{\beta}$ be a column vector containing intercept α and coefficients β_j, the regression coefficients are estimated maximizing log-likelihood function of the following form

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \{y_i log(\pi_i(\boldsymbol{\beta})) + (1 - y_i) log(1 - \pi_i(\boldsymbol{\beta}))\}$$

In ridge logistic regression? ?, the following penalized version of the log likelihood function is used for estimating $\boldsymbol{\beta}$, tending to shrink the coefficients towards zero

$$l(\boldsymbol{\beta}) - \lambda \sum_{j=1}^{2} \beta_j^2$$

The hyperparameter λ needs to be tuned. In this article it is estimated by minimizing the deviance using 10-fold cross-validation with a grid of 251 possible values for λ ranging from 0 (no shrinkage) to 64 (large shrinkage).

2.1.2 | Tree-based models

Random forest? and Gradient Boosting Machines both use an ensemble of classification and regression trees (CART) to predict a positive outcome. The Random forest algorithm does this by training each tree on a bootstrapped sample of the development set, using a subset of the predictors to reduce correlation between trees. This process is illustrated with the pseudo-code in algorithm 1. To get to a prediction, the results of all trees are combined by means of a majority vote. The estimated probability is simply the fraction of trees that voted positive given a particular combination of predictor values. For fitting the model the randomForest? package was used with default hyperparameters.

Unlike random forest, Gradient Boosting Machines directly predict probabilities using feature vector $\mathbf{x}$. In this process, an initial tree predicts the probabilities. Each next tree then tries to minimize the differences between the predicted probabilities and the outcome of each observation. In the XGboost? implementation of gradient boosting, each new tree is fitted by minimizing the loss function

$$\mathcal{L}^t = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

Where l is the differentiable loss function that measures the difference between estimated probability $\hat{y}_i$ and the observed score y_i for each person i. We may notice that the predicted score consists of the score predicted in the last iteration ($\hat{y}_i^{(t-1)}$) plus the leaf scores of the tree in the current iteration ($f_t(\mathbf{x}_i)$). $\Omega(f_t)$ is a regularization term that prevents the model from overfitting and can be defined as

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2$$

Where T is the number of terminal leaves, ω_j is the leaf score of leaf j and γ and λ are tuneable hyper parameters. For this research both hyperparameters γ and λ were set to 0, omitting the regularization term.

Other hyperparameters in the XGboost package that were manually tuned were maximum tree depth, learning rate, number of threads and number of rounds. Tuning these hyperparameters did not seem to substantially affect the results. As every new tree tries to explain the difference between the previous prediction and the outcome, the sum of the predictions of all trees results in the final estimated probability given a combination of predictor values.

Algorithm 1 Pseudo-code for Random Forest[?]

```
for i = 1 to c do
    Randomly sample training data $D$ with replacement to produce $D_i$
    Create root node $N_i$, containing $D_i$
    Call BuildTree($N_i$)
end for
      BuildTree(N)
if $N$ contains instances of only one class then
    Return
else
    Randomly select x% of possible splitting features in N
    Select feature F
    Create f child nodes of $N$, $N_1, ..., N_f$, where $F$ has $f$ possible values $(F_1, ..., F_f)$
    for $i = 1$ to $f$ do
        Set the contents of $N_i$ to $D_i$, where $D_i$ is all instances in $N$ that match $F_i$
        Call BuildTree($N_i$)
    end for
end if
```

2.2 | Solutions imbalanced data

To create a balanced data set, that is, a data set where there are the same number of events as non-events, several approaches have been proposed in the machine learning literature. Three of these approaches are examined in this paper.

In ROS, the size of the minority class is increased by resampling cases from the minority class with replacement until it has the same size as the majority class. This means that the new data set contains duplicate cases for the minority class. In RUS, the size of the majority class is reduced to the size of the minority class by randomly discarding cases from the majority class.

SMOTE is a form of oversampling where new synthetic cases are created. These synthetic cases are created by operating in the "feature space" instead of the "data space".[?,?] That is, each synthetic case is created by sampling a random case from the minority class after which its k nearest neighbours are determined. In the `smotefamily` R package I used to implement SMOTE, the nearest neighbours are determined based on the Euclidean distance.[?,?] Then the differences between the feature vectors of the sampled case and those of its k nearest neighbours is taken. These differences are then multiplied by a random number between 0 and 1 and are added to the feature vector of the initial case. This results in k synthetic cases that are interpolated from the original minority class cases. By adding these cases to the data set, balance is achieved.

2.3 | Data

For this study, I used a subset of the data from the International Ovarian Tumor Analysis (IOTA) consortium containing only woman who are not yet in their menopause.[?] The data are derived from 5914 patients between 1999 and 2012. The subset contains data of 3488 patients. A subset of the data was used to achieve a data set with a higher imbalance ratio than the full data set. As outcome variable I used the final diagnoses whether the tumor was benign (n = 2785, 79.8%) or malignant (n = 703, 20.2%). The features considered as predictors are: age and diameters of the ovary (ovaryd1 and ovaryd3).

2.4 | Analytic strategy

To assess the performance of the different approaches to deal with imbalanced data in combination with the different models, the data was first randomly split up in to a validation and a development set using a 1:4 ratio. Afterwards the development data set was used to create multiple training data sets using random oversampling, random undersampling and SMOTE, resulting in four different data sets; $D_{unadjusted}$, D_{over}, D_{under}, D_{SMOTE}.

For every development set D_i, the prediction models as described in section 2.1 were trained with the aim to distinguish between benign and malignant tumors based on the predictors, resulting in 16 (4 x 4) different models. Subsequently all models were assessed on their performance predicting the outcome of the cases contained in the validation set. More information on how the performance evaluation was done can be found in section **??**.

2.5 | Performance evaluation

To quantify the out of sample performance of the model in terms of discrimination, I considered the c-statistic (area under the ROC curve).[?] To asses calibration, I made use of both a visual representation of the calibrative performance of the models in the form of calibration plots and quantitative performance measures in the form of the Estimated Calibration Index (ECI)[?], calibration slopes and calibration intercepts.[?] [?]

The c-statistic can be interpreted as the probability that a random pair of a positive and a negative case are correctly ranked (i.e. that the positive case has got assigned a higher probability than the negative case). The confidence interval for the c-statistic is estimated using the method as described by Pepe[?] using the `auRoc`[?] R package.

To asses the performance regarding correct classification, I used accuracy, sensitivity and specificity scores. Accuracy can be interpreted as the fraction of correctly classified cases compared to the total number of cases. Sensitivity is the ratio of true positives over the number of positive cases in the data set. Specificity is the ratio of true negatives over the total number of negative cases. All classifications were made using the conventional decision threshold of 0.5. A description of the interpretation of all performance measures can be found in Table **??**.

TABLE 1 Performance measures

Performance measure	**Interpretation**
Accuracy $\left(\frac{TP+TN}{TP+FP+TN+FN}\right)$	Fraction correctly classified cases
Sensitivity $\left(\frac{TP}{TP+FN}\right)$	Fraction correctly classified positive cases over all positive cases
Specificity $\left(\frac{TN}{FP+TN}\right)$	Fraction correctly classified negative cases over all negative cases

c-statistic	c-statistic = 1: perfect discrimination
	c-statistic = 0.5: no discriminative performance

ECI	ECI = 0: No difference between observed and predicted probability ECI = 100: maximum difference between observed and predicted probability
Calibration slope	Calibration slope < 1: underfitting Calibration slope > 1: overfitting
Calibration intercept	Calibration intercept = 0: mean calibration is perfect Calibration intercept > 0: probabilities are systematically too high Calibration intercept < 0: probabilities are systematically too low
Calibration curve	The further the curve from the diagonal, the worse the calibrative performance

Abbreviations: TP = True postive, TN = True negative, FP = False positive, FN = False negative, ECI = Estimated Calibration Index

2.6 | Software

All analyses were performed using `R version 3.6.2.`[?] To fit the regression and machine learning models, I used the following R packages: `glmnet`[?], `randomForest`[?], `XGboost`[?]. For the random over- and undersampling procedure I used the `caret`[?] R package and to implement SMOTE I used the `smotefamily`[?] R package.

3 | RESULTS

For all imbalance adjustment approaches and prediction models, it can be seen from the calibration plots that the predicted probabilities exceeded the observed probabilities over the whole scale (Figure 1). This means that the estimated probabilities tended to be to high compared to the observed probability. This observation was confirmed by the calibration in the large (CIL) statistics in Table **??**, which were all >0 for the models trained on the adjusted data sets.

Although the ECI statistic on itself has no straightforward interpretation, it can be used to compare overall performance in terms of calibration. The ECI scores in Table **??** showed the same picture of the calibration being affected by data pre-processing, as for all modelling algorithms the lowest ECI was the one for the model that was trained on the data set where the data remained unadjusted for class imbalance.

The c-statistics did not show a noticeable difference over the different models, indicating that the imbalance approaches did not improve the overall discriminative performance of the models. The accuracy tended to be the highest for the models trained on data sets where the imbalance was not adjusted for (Table **??**), indicating that the number of correct predictions was the highest using these models. However, differences did occur when looking at the sensitivity and specificity over different scenarios. In the cases where the data was adjusted, the sensitivity and specificity tended to have more similar values, where the in the cases where the data remained unadjusted the minority class was overlooked. For instance the sensitivity of the logistic regression model trained on the unadjusted data had a sensitivity score of no more than 0.15, meaning that only 15% of the minority class cases in the validation set got detected by the model.

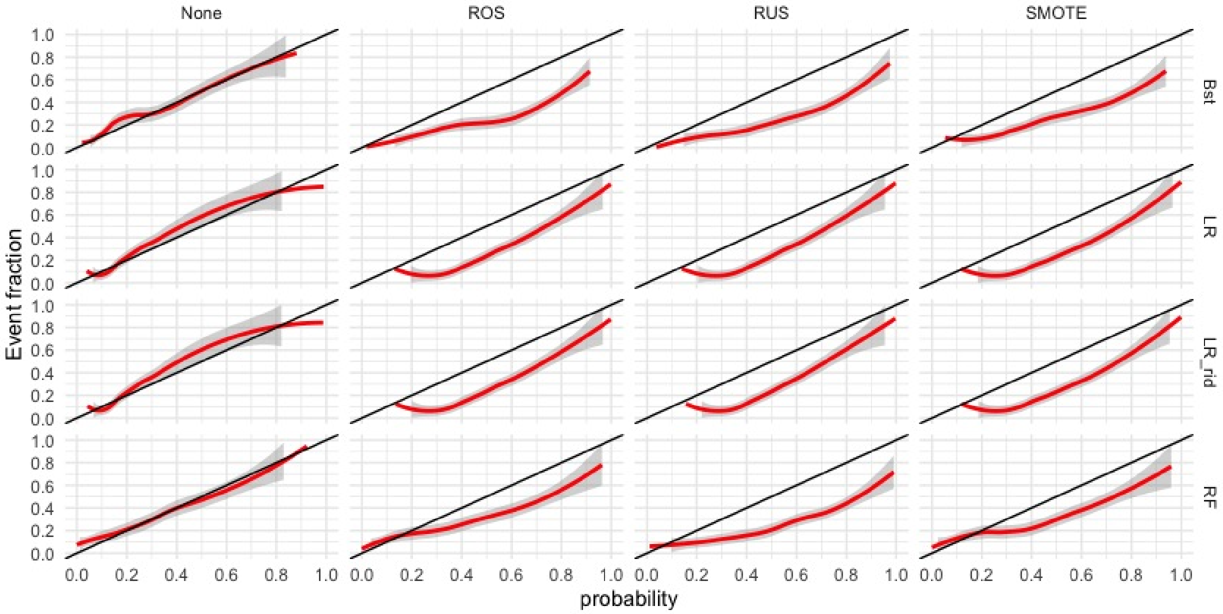

FIGURE 1 Calibration plots. The red line shows the loess curve fitted on the predicted probabilities. The dark grey area shows the 95% confidence interval of the loess curve. The black line shows the hypothetical situation of perfect predictions. Abbreviations: LR = Regular Logistic Regression, LR_rid = Ridge Logistic Regression, RF = Random Forest, Bst = Gradient Boosting Machine, ROS = Random Oversampling, RUS = Random Undersampling, SMOTE = Synthetic Minority Oversampling Technique

TABLE 2 Performance of models in combination with data adjustments for handling class imbalance

	Adjustment	Accuracy	Sensitivity	Specificity	c-statistic (CI)	CIL (CI)	Calibration slope (CI)	ECI
LR	None	0.81	0.15	0.98	0.76 (0.71 — 0.80)	0.09 (-0.11 — 0.28)	1.18 (0.93 — 1.45)	0.16
	ROS	0.72	0.64	0.74	0.76 (0.71 — 0.80)	-1.28 (-1.48 — -1.09)	1.13 (0.88 — 1.39)	5.77
	RUS	0.72	0.64	0.74	0.76 (0.71 — 0.80)	-1.28 (-1.48 — -1.09)	1.18 (0.92 — 1.45)	5.93
	SMOTE	0.72	0.66	0.74	0.76 (0.71 — 0.80)	-1.29 (-1.49 — -1.09)	1.08 (0.85 — 1.33)	5.77
LR_{rid}	None	0.81	0.14	0.98	0.76 (0.71 — 0.80)	0.09 (-0.11 — 0.28)	1.22 (0.96 — 1.50)	0.19
	ROS	0.72	0.64	0.75	0.76 (0.71 — 0.80)	-1.28 (-1.48 — -1.08)	1.14 (0.90 — 1.40)	5.78
	RUS	0.72	0.64	0.74	0.76 (0.71 — 0.80)	-1.28 (-1.48 — -1.09)	1.24 (0.98 — 1.53)	6.01
	SMOTE	0.72	0.66	0.74	0.76 (0.71 — 0.80)	-1.29 (-1.49 — -1.09)	1.09 (0.86 — 1.35)	5.78
RF	None	0.80	0.23	0.95	0.73 (0.68 — 0.77)	0.27 (0.05 — 0.48)	0.6 (0.46 — 0.74)	0.27
	ROS	0.78	0.41	0.88	0.72 (0.67 — 0.76)	-0.48 (-0.69 — -0.27)	0.57 (0.43 — 0.71)	1.50
	RUS	0.70	0.64	0.71	0.73 (0.68 — 0.77)	-1.37 (-1.58 — -1.16)	0.67 (0.51 — 0.83)	6.16
	SMOTE	0.78	0.42	0.88	0.71 (0.66 — 0.75)	-0.49 (-0.7 — -0.28)	0.57 (0.43 — 0.71)	1.59
Bst	None	0.82	0.28	0.97	0.76 (0.72 — 0.80)	0.14 (-0.07 — 0.34)	0.92 (0.74 — 1.12)	0.15
	ROS	0.72	0.58	0.76	0.74 (0.69 — 0.78)	-1.22 (-1.43 — -1.02)	0.77 (0.60 — 0.94)	4.77
	RUS	0.69	0.64	0.71	0.73 (0.69 — 0.78)	-1.36 (-1.57 — -1.15)	0.77 (0.60 — 0.94)	5.79
	SMOTE	0.74	0.59	0.78	0.74 (0.70 — 0.79)	-1.17 (-1.38 — -0.96)	0.78 (0.62 — 0.96)	4.18

Abbreviations: LR = Regular Logistic Regression, LR_{rid} = Ridge Logistic Regression, RF = Random Forest, Bst = Gradient Boosting Machine, ROS = Random Oversampling, RUS = Random Undersampling, SMOTE = Synthetic Minority Oversampling Technique, CIL = Calibration in the large, ECI = Estimated Calibration Index, CI = Confidence Interval

4 | DISCUSSION

In this report I conducted a case study to examine the effect of data pre-processing methods to adjust for class imbalance on the performance of medical prediction models. I found that data pre-processing methods can seriously affect the calibration and therefore has implications for the development of these models.

The results show that, when adjusting the data for class imbalance, models tend to overestimate probabilities estimated for the individual cases and thus are poorly calibrated. Van Calster et al.[?] describe this as the lack of mean calibration, the most basal type of calibration. Moreover, imbalance solutions did not seem to improve the performance of the models in terms of the c-statistic. The pre-processing methods did show to bring the sensitivity and specificity closer to each other, indicating that these methods do help to increase the model performance in terms of correctly classifying the minority class. However, it can be argued that this could also be done by moving the decision threshold value so that the calibration is not affected.[?]

This study has also some limitations. First of all, the absence of weighting as a solution for imbalanced data in this paper can be seen as a limitation, since it is a method that is sometimes used to deal with class imbalance.[?] Yet, as Maloof points out[?], weighting and oversampling are conceptually closely related and have very similar results. Secondly, as a result of being a case study based on only one data set, it is not possible to generalize findings to a broad set of scenarios that may be encountered in the development of medical prediction models. Another limitation related to this is that the imbalance ratio in the used data set is not as extreme as it can be in medical data sets.[?] To overcome the lack of generalizability, for my master thesis I will build on these results by conducting an elaborate simulation study varying the event fraction and the number of events per variable.

In conclusion, prediction modelers should be careful when applying class imbalance solutions when calibration of the prediction model is important. The case study illustrates that miscalibration effects of commonly used solution approaches can be strong, future research should elucidate the consequences of these solutions in different scenarios.

□