Title

Alex Carriero

January 10, 2022

Supervisors:

Maarten van Smeden, Utrecht Medical Center Utrecht
Kim Luijken, Utrecht Medical Center Utrecht
Ben van Calster, Catholic Univeristy of Leuven and Leiden University Medical Center

Program: Methodology and Statistics for Behavioral, BioMedical, and Social Sciences.

Host Institution: Julius Center for Health Science and Primary Care, UMC.

Candidate Journal: Statistics in Medicine.

Word Count: 2500

1. Introduction

Prediction modelling in medicine is gaining increasing attention. Clinicians are often interested in predicting a patient's risk of disease. Due to the (thankfully) rare nature of many diseases, the data available to train clinical prediction models are often heavily imbalanced (i.e., the number of patients in one class dramatically outnumbers the other)[14]. This is referred to as class imbalance. Class imbalance is seen as a major problem as it is known to degrade model performance[25]. Consequently, imbalance correction methodologies are proposed as a solution[25].

An ideal imbalance correction will improve all aspects of model performance. These criteria include: classification accuracy, discrimination and calibration. Accuracy refers to the proportion of patients that a model classifies correctly (after a risk threshold is imposed). Discrimination refers to a model's ability to yield higher risk estimates for patients in the positive class than for those in the negative class. Finally, calibration refers to the reliability of the risk predictions themselves; for instance, a poorly calibrated model may produce risk predictions that consistently over- or under-estimate reality, or produce risk estimates which are too extreme (too close of 0 or 1) or too modest[22].

In a clinical context, a model is only useful if it is well calibrated[22]. This is because in practice, the risk predictions are used by clinicians to council patients and inform treatment decisions. If risk predictions are unreliable, the personal costs to the patient may be enormous. Further, it is entirely possible for a model to exhibit excellent classification accuracy and discrimination while calibration is poor[22]. Therefore, assessing only discrimination and accuracy is insufficient.

Class imbalance is not unique to medical data sets and literature introducing imbalance correction methods arises from many disciplines. An abundance of imbalance corrections exist and are summarized by[1;12;14;6;9]. Information regarding the effect of these corrections on model calibration is sparse. Only one study has assessed the impact of imbalance corrections on model calibration. van den Goorbergh et al.[23] demonstrated that implementing imbalance corrections lead to dramatically deteriorated model calibration, to the extent that no correction was recommended[23]. In this study, models were developed using logistic regression and penalized (ridge) logistic regression[23].

Motivated by the work of van den Goorbergh et al.[23], we must ensure that the "cure" is not worse than the disease. In our research, we aim to assess the impact of imbalance corrections on model calibration for prediction models trained with a wider variety of classification algorithms including: linear classifiers, ensemble learning algorithms and algorithms specifically designed to handle class imbalance. Furthermore, we aim to answer the question: can imbalance corrections improve overall model performance without comprising model calibration?

2. Methods

- we do two things:
- pilot study – we investigate the baseline performance of the models with no class imbalance corrections. Clearly see that class imbalance has an effect on model calibration.
  – full study – same simulation study is implemented – however, this time imbalance. corrections are applied.

– We adhere to the ADEMP guidelines for the design and reporting of our simulation study[17].

Full Study

In the full study we investigate the effect of imbalance corrections on prediction model performance in 27 (3 x 3 x 3) unique scenarios. These scenarios are achieved by varying the following three characteristics of the simulated data: number of predictors, event fraction and sample size. The number of predictors will vary through the set {8,16,32} and event fraction through the set {0.5, 0.2, 0.02}. The minimum sample size for the prediction model (N) will be computed according to formulae presented in Riley et al.[19] using functions from the pmsampsize package[5]. Sample size will then vary through the set {$\frac{1}{2}$N, N and 2N}.

Pilot Study

In this study we investigate the baseline performance of the classification algorithms. No imbalance corrections will be applied. The aim in this study is to determine the baseline performance of prediction models developed with the six classification algorithms under 3 scenarios. Across these three scenarios, the only simulation factor to vary is the event fraction. Pilot study scenarios are presented in Table 1 and results are presented in Section 3 of this paper.

2.3 Simulation Study

We adhere to the ADEMP guidelines for the design and reporting of our simulation study[17].

Aim

We aim to determine the best practices for handling class imbalance when developing clinical prediction models for dichotomous risk prediction. Under a variety of scenarios, four imbalance corrections and six classification algorithms will be used to train prediction models; models will then be systematically compared based on their out-of-sample predictive performance.

We aim to identify any combination of imbalance correction and classification algorithm that, together, produce a model which outperforms it's associated control model (a model trained using the classification algorithm and no imbalance correction).

Data-Generating Mechanism

Data for each class is generated independently from two distinct multivariate normal distributions:

$$\text{Class 0: } \mathbf{X} \sim mvn(\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0}) = mvn(\mathbf{0}, \boldsymbol{\Sigma_0})$$

$$\text{Class 1: } \mathbf{X} \sim mvn(\boldsymbol{\mu_1}, \boldsymbol{\Sigma_1}) = mvn(\boldsymbol{\Delta_\mu}, \boldsymbol{\Sigma_0} - \boldsymbol{\Delta_\Sigma})$$

The parameters (mean vector and covariance matrix) of the data generating distributions are distinct between the classes. In the formulae above, $\boldsymbol{\Delta_\mu}$ refers to the vector housing the difference in predictor means between the two classes. Similarly, $\boldsymbol{\Delta_\Sigma}$ refers to the matrix housing the difference in predictor variances/covariances between the classes.

In class 0, all predictor means are fixed to zero and all variances are fixed to 1. In class 1, all means are stored in the vector $\boldsymbol{\Delta_\mu}$. There is no variation in means among predictors within a class, thus, every element in the vector $\boldsymbol{\Delta_\mu}$ is equivalent; denoted by δ_μ. Similarly, there is no variation in predictor variances within a class, so every diagonal element in $\boldsymbol{\Delta_\Sigma}$ is equivalent; diagonal elements are denoted by δ_Σ.

Finally, 80% of the predictors are allowed to covary. All correlations among predictors in each class are set to 0.2. To ensure the correlation of predictors is not stronger in one class, the correlation matrices of the two classes are fixed to be equal. This is accomplished by computing the off-diagonal elements of $\boldsymbol{\Delta_\Sigma}$ such that the correlation matrices between the two classes are equivalent. Note, the covariance matrices are *not* equivalent between the classes. For example, in scenario where we have 8 predictors:

mean and covariance structure for class 0,

$$\boldsymbol{\mu_0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \boldsymbol{\Sigma_0} = \begin{bmatrix} 1 & 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 \\ 0.2 & 1 & 0.2 & 0.2 & 0.2 & 0.2 & 0 & 0 \\ 0.2 & 0.2 & 1 & 0.2 & 0.2 & 0.2 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 1 & 0.2 & 0.2 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 1 & 0.2 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0.2 & 0.2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

mean and covariance structure for class 1,

$$\boldsymbol{\mu_1} = \begin{bmatrix} \delta_\mu \\ \delta_\mu \\ \delta_\mu \\ \delta_\mu \\ \delta_\mu \\ \delta_\mu \\ \delta_\mu \\ \delta_\mu \end{bmatrix}, \boldsymbol{\Sigma_1} = \begin{bmatrix} 1-\delta_\Sigma & z & z & z & z & z & 0 & 0 \\ z & 1-\delta_\Sigma & z & z & z & z & 0 & 0 \\ z & z & 1-\delta_\Sigma & z & z & z & 0 & 0 \\ z & z & z & 1-\delta_\Sigma & z & z & 0 & 0 \\ z & z & z & z & 1-\delta_\Sigma & z & 0 & 0 \\ z & z & z & z & z & 1-\delta_\Sigma & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1-\delta_\Sigma & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1-\delta_\Sigma \end{bmatrix}$$

.

Here, $z = \frac{(1-\delta_\Sigma)*0.2}{1}$, to ensure the correlation matrices of the two classes are equivalent.

Under the assumption of normality for all predictors (in each class), the ΔC Statistic can be expressed as a function of $\boldsymbol{\Delta_\mu}$, $\boldsymbol{\Sigma_0}$ and $\boldsymbol{\Sigma_1}$ [4]. For every scenario, the parameter values for the data generating distributions (δ_μ and δ_Σ) are selected to generate a ΔC Statistic = 0.85. Their values are computed analytically, based on the following formula from Demler et al. [4].

$$AUC = \Phi\left(\sqrt{\boldsymbol{\Delta_\mu}'\ (\boldsymbol{\Sigma_0} + \boldsymbol{\Sigma_1})^{-1}\ \boldsymbol{\Delta_\mu}}\right) \tag{1}$$

In equation (1), Φ represents the cumulative density function (cdf) of the standard normal distribution; $\boldsymbol{\Delta_\mu}$, $\boldsymbol{\Sigma_0}$ and $\boldsymbol{\Sigma_1}$ maintain their previous definitions. To ensure a unique solution δ_Σ will remain fixed at 0.3 for each scenario, while equation (1) is solved to yield the appropriate value of δ_μ in each scenario.

Finally, given that data for each class are generated independently, we have direct control over how many observations are generated under each class. The number of observations from the positive class (n_1) will be will be sampled from the binomial distribution with probability equal to the specified event fraction. The number of observations in the negative class (n_0) will then be computed as $X - n_1$, where X is the total sample size specified for the prediction model.

Table 1: Summary of simulation scenarios (Pilot Study).

No. Predictors	Event Fraction	N Level	Sample Size (X)
8	0.50	N	385
8	0.20	N	247
8	0.02	N	1797

Estimands

The focus of this study is the out-of-sample predictive performance of clinical prediction models for dichotomous risk prediction.

Methods

Imbalance Corrections

Common approaches to handing class imbalance involve data pre-processing[14;15]. The goal of this pre-processing is to arrive at an artificially balanced population of observations; this can be achieved by under sampling from the majority class, over sampling from the minority class or both (hybrid sampling). We consider four data pre-processing techniques in this paper: random under sampling (RUS), random over sampling (ROS), synthetic majority over sampling technique (SMOTE), and synthetic majority over sampling technique + edited nearest neighbors (SENN). RUS involves randomly removing observations from the majority class until a balanced population is reached. ROS artificially increases the number of observations in the minority class by adding randomly sampled values from the minority population until a balanced population is reached. SMOTE is a form of oversampling whereby "synthetic" minority observations are generated by making interpolations from the minority class[2]. Finally, SMOTE-ENN is a method of hybrid sampling in which the Wilson's edited nearest neighbors rule is applied after SMOTE to remove any observations that are misclassified by their three nearest neighbors[11;24]. These imbalance corrections and the R packages used for their implementation in the simulation are summarized in Table 1.

Classification Algorithms

The effect of imbalance corrections on prediction models trained with logistic regression has been well established by van den Goorbergh et al.[23]. In our research we train prediction models with a wider range of classification algorithms, as well as logistic regression, for the purpose of replicating the findings of van den Goorbergh et al.[23]. The two linear classifiers considered are: logistic regression and support vector machine. We include two ensemble classifiers: random forest and xgboost. Finally, we include two algorithms specifically designed to handle class imbalance: RUSBoost and EasyEnsemble. Currently, all algorithms are implemented using their default hyperparameters. All classification algorithms and the R packages used for the implementation are summarized in Table 1.

Under each scenario,2000 (full study) or 200 (pilot study) data sets will be generated. Each data set will be comprised of training and validation data. The training and validation data will be generated independently using identical data generating mechanisms; this is done to ensure a similar event fraction in the training

Table 2: Summary of imbalance corrections and classification algorithms used in simulation study.

Name	Abbreviation	R Package
Imbalance Corrections		
Random Under Sampling	RUS	ROSE[13]
Random Over Sampling	ROS	ROSE[13]
SMOTE	SMOTE	smotefamily[21]
SMOTE - ENN	SENN	iric[26]
Classification Algorithms		
Logistic Regression	LR	base R[18]
Support Vector Machine	SVM	e1071[16]
Random Forest	RF	randomForest[10]
XGBoost	XG	xgboost[3]
RUSBoost	RB	ebmc[7]
EasyEnsemmble	EE	iric[26]

and validation data. The validation data is generated to be 10x larger than the training set.

To create a fair comparison for class imbalance corrections a full factorial (5 x 6) simulation design will be implemented. For each generated data set, five imbalance corrections (four and one control) will be applied to the training set. Six prediction models will then be developed for each of the five imbalance corrected training sets. In other words, each data set will result in: 5 imbalance corrected training sets x 6 classification algorithms = 30 prediction models. All models will be trained using training data sets. Out-of-sample performance will be then be assessed using the validation data.

Performance Measures

Out-of-sample model performance will be assessed using measures of discrimination, accuracy and calibration.

Discrimination will be measured by area under the receiver operator curve (ΔC-statistic); computed using the function *auc* from `pROC`[20].

Accuracy will be measured by Brier Score; calculated using equation (2).

$$\text{Brier Score} = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2 \tag{2}$$

Here, N is the sample size, p_i represents the predicted probability for the i^{th} observation and y_i represents the true classification (0 or 1) for the i^{th} observation. Measures of accuracy which involve the selection of a decision threshold (e.g., total accuracy, sensitivity, specificity) will not be considered.

Calibration will be measured empirically in terms of calibration intercept and slope. Calibration intercept is estimated as the regression intercept (β_0) resulting from the logistic regression equation shown in (3).

$$\text{logit}(P(Y=1)) = \beta_0 + \text{logit}(p) \tag{3}$$

Calibration slope is estimated as the regression slope (β_1) resulting from the logistic regression equation shown in (4).

$$\text{logit}(P(Y=1)) = \beta_0^* + \beta_1 \text{logit}(p) \tag{4}$$

From equations (3) and (4), y and p maintain their previous definitions. To obtain estimates of calibration intercept and slope, logistic regression models will be fit using the glm function in base R[18]. Model calibration will be then be visualized via flexible calibration curves fit using loess regression. A flexible calibration curve will be fit for each algorithm for every iteration in the simulation. Within one simulation scenario, all results for each algorithm will be displayed on the same facet grid.

For the empirical measures of model performance (AUC, brier score, calibration intercept and slope), the mean over all iterations in each scenario as well as the corresponding monte carlo standard error will be reported.

Software

All analyses will be conducting using R version 4.1.2[18]. For the full study, the high performance computers at University Medical Center Utrecht will be used.

3. Results

For the pilot study, empirical performance metrics are summarized in Table 3 and model performance visualizations are displayed in Figure 1.

In a well calibrated model, predicted probabilities correspond to observed proportions[22]. In terms of calibration intercept an slope, good calibration is represented by values of 0 and 1, respectively. For a flexible calibration curve, when predicted probabilities (x-axis) correspond well to the observed proportions (y-axis), the curve follows a diagonal line ($y = x$)[22]. For approximately balanced data (event fraction = 0.5), all algorithms, except XG and EE, were well calibrated. While, on average, both XG and EE had calibration intercepts near zero, their average calibration slopes dramatically deviated from 1. We see that for XGBoost, the risk predictions above 0.5 overestimated true risk, while the risk predictions below 0.5 underestimated true risk (Figure 1(a)). In other words, the XGBoost models resulted in risk estimates which were too extreme (calibration slope = XXX). The opposite pattern is true for EasyEnsemble; EE produced risk estimates which were too moderate (calibration slope = XXX). With respect to overall performance and discrimination, an ideal model will produce brier scores close to zero ΔC statistics close to one[?] . For balanced data, SVM and LR had similar overall performance and discrimination and outperformed the other algorithm.

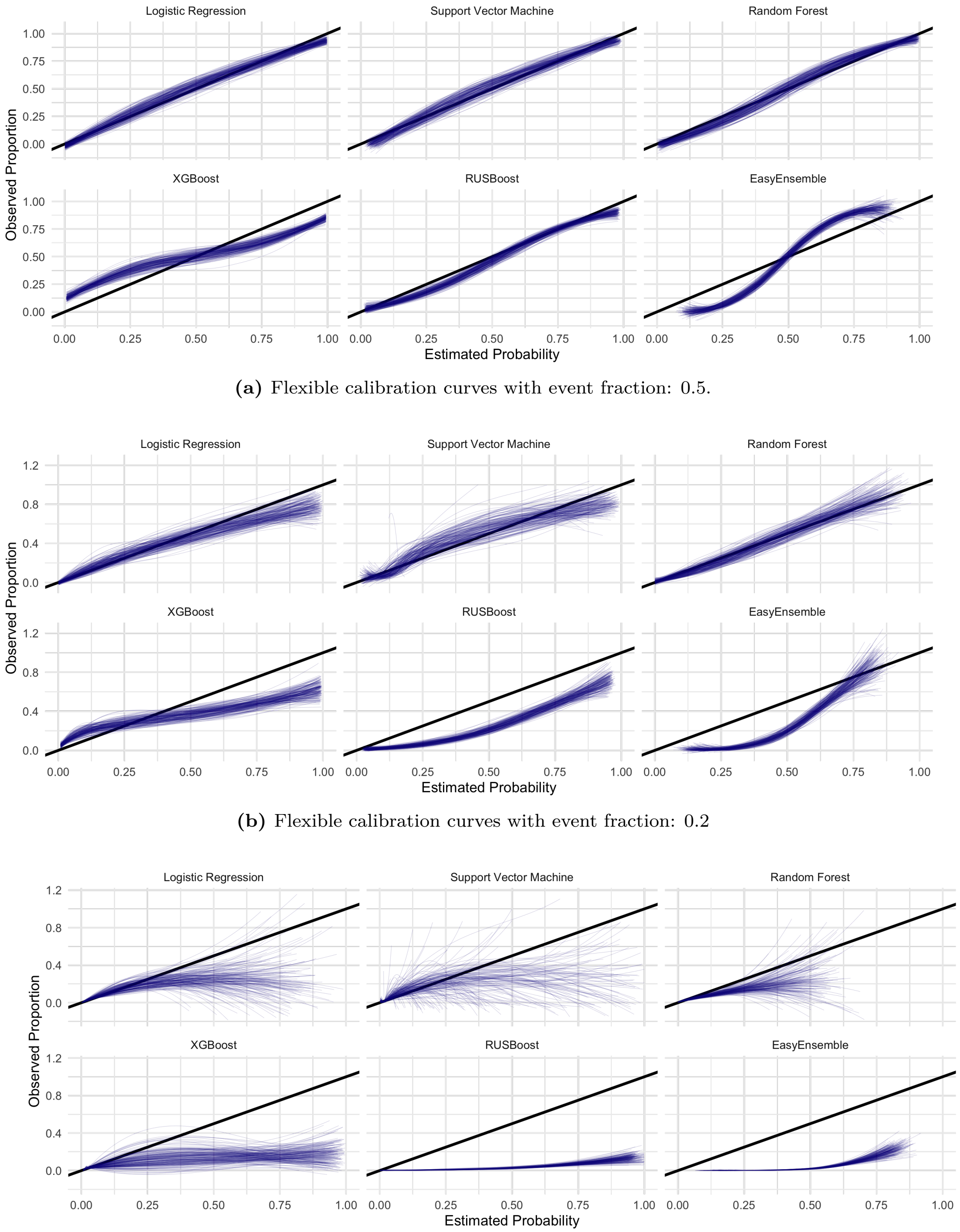
With the event fraction at 0.2, for half of the algorithms, model calibration was moderate. While the linear classifiers and random forest maintained adequate calibration in this scenario, XGBoost, RUSBoost and EasyEnsemble all, on average, producde risk predictions that dramatically over estimated true risk (Figure 1(b)). With respect to overall performance and discrimination, in this scenario, LR was the best performing algorithm.

At an event fraction of 0.02, all algorithms exhibited miscalibration. From Figure 1 (c), we see that for the linear classifiers and random forest, the calibration curves were sporadic; there is large variation in the calibration curves produced for each iteration of the simulation. Meanwhile, for XGBoost, RUSBoost, and EasyEnsemble, the calibration curves did not vary much across the iterations, rather, they exhibited a specific pattern of miscalibration: all risk predictions over estimated true risk. With respect to overall performance and discrimination, in this scenario, LR was again, the best performing algorithm.

Overall, as the imbalance between the classes was magnified, model calibration deteriorated across all algorithms. Meanwhile, discrimination maintained relatively constant. Interestingly, as the imbalance between the classes was magnified, overall performance appeared to improve, especially for the linear classifiers. This apparent improvement in overall performance is misleading and is a result of a poor choice in performance metric.

Table 3: Mean (monte carlo error) of performance metrics across 200 simulation iterations for each scenario in the Pilot Study

	Δ C Statistic	Brier Score	Calibration Int.	Calibration Slope
Event Fraction: 0.5				
Logistic Regression	0.845 (0.007)	0.161 (0.004)	-0.001 (0.14)	0.928 (0.098)
Support Vector Machine	0.849 (0.008)	0.158 (0.004)	-0.002 (0.131)	1.022 (0.12)
Random Forest	0.842 (0.008)	0.163 (0.004)	-0.016 (0.124)	1.168 (0.087)
XGBoost	0.79 (0.012)	0.205 (0.007)	-0.046 (0.217)	0.464 (0.024)
RUSBoost	0.813 (0.01)	0.178 (0.005)	-0.183 (0.081)	1.063 (0.072)
EasyEnsemble	0.826 (0.01)	0.187 (0.004)	0.001 (0.045)	2.279 (0.176)
Event Fraction: 0.2				
Logistic Regression	0.836 (0.013)	0.122 (0.005)	-0.035 (0.216)	0.86 (0.137)
Support Vector Machine	0.81 (0.026)	0.123 (0.007)	-0.025 (0.197)	1.049 (1.057)
Random Forest	0.808 (0.018)	0.125 (0.005)	-0.079 (0.179)	1.079 (0.12)
XGBoost	0.755 (0.022)	0.153 (0.008)	0.006 (0.304)	0.445 (0.04)
RUSBoost	0.796 (0.018)	0.184 (0.011)	-1.335 (0.107)	1.078 (0.115)
EasyEnsemble	0.814 (0.018)	0.196 (0.009)	-1.302 (0.064)	2.331 (0.256)
Event Fraction: 0.02				
Logistic Regression	0.839 (0.011)	0.019 (0.001)	0.014 (0.199)	0.904 (0.124)
Support Vector Machine	0.698 (0.034)	0.019 (0.001)	-0.013 (0.193)	1.255 (2.652)
Random Forest	0.756 (0.022)	0.019 (0.001)	-0.115 (0.189)	0.655 (0.083)
XGBoost	0.707 (0.028)	0.022 (0.001)	-0.331 (0.186)	0.508 (0.047)
RUSBoost	0.778 (0.027)	0.158 (0.019)	-3.696 (0.157)	0.869 (0.152)
EasyEnsemble	0.81 (0.019)	0.2 (0.014)	-3.738 (0.08)	2.295 (0.304)



(a) Flexible calibration curves with event fraction: 0.5.

(b) Flexible calibration curves with event fraction: 0.2

(c) Flexible calibration curves with event fraction: 0.02

Figure 1: Visual representation of model calibration for each simulation scenario in the Pilot Study.

4. Discussion

In this paper, the problem of class imbalance and proposed solutions are introduced. The results from the Pilot Study, investigating the baseline performance of the classification algorithms, without imbalance correction, are presented. From these results, the problem of class imbalance is clearly demonstrated. As the event fraction drops from 0.2 to 0.02, all classification algorithms exhibit miscalibration. Further, these results confirm that assessing discrimination an classification accuracy alone is insufficient. As imbalance between the classes is magnified, these metrics appear relatively unaffected.

Based on the results of the pilot study, brier score appears to be an uninformative measure of classification accuracy when class imbalance is extreme. Therefore, in the full study, it may be worth investigating another metric of classification accuracy, such as the index of prediction accuracy (IPA), that is known to be more informative in the presence of class imbalance[8].

The algorithms RUSBoost and EasyEnsemble are designed specifically to handle class imbalance. Interestingly, these algorithms exhibit the highest degree of miscalibration in the presence of class imbalance. Further, in the pilot study, they do not out perform logistic regression with respect to classification and discrimination in any scenario. In fact, these algorithms had worse classification accuracy than a trivial majority classifier at the most extreme event fraction.

With respect to the full simulation study, we look forward to seeing how the imbalance corrections will influence model calibration. However, given the results of van den Goorbergh et al.[23], we hypothesize that the imbalance corrections will worsen model calibration across all models in all scenarios.

References

[1] Paula Branco, Luis Torgo, and Rita P. Ribeiro. A survey of predictive modeling on imbalanced domains. 49(2), aug 2016. ISSN 0360-0300. doi: 10.1145/2907070. URL https://doi.org/10.1145/2907070.

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, jun 2002. doi: 10.1613/jair.953. URL https://doi.org/10.1613%2Fjair.953.

[3] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, Yutian Li, and Jiaming Yuan. *xgboost: Extreme Gradient Boosting*, 2022. URL https://CRAN.R-project.org/package=xgboost. R package version 1.6.0.1.

[4] Olga V. Demler, Michael J. Pencina, and Ralph B. D'Agostino Sr. Equivalence of improvement in area under roc curve and linear discriminant analysis coefficient under assumption of normality. *Statistics in Medicine*, 30(12):1410–1418, 2011.

[5] Joie Ensor, Emma C. Martin, and Richard D. Riley. *pmsampsize: Calculates the Minimum Sample Size Required for Developing a Multivariable Prediction Model*, 2022. URL https://CRAN.R-project.org/package=pmsampsize. R package version 1.1.2.

[6] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. Learning from class-imbalanced data: Review of methods and applications. *Expert Systems with Applications*, 73:220–239, 2017.

[7] Hsiang Hao and Chen. *ebmc: Ensemble-Based Methods for Class Imbalance Problem*, 2022. URL https://CRAN.R-project.org/package=ebmc. R package version 1.0.1.

[8] Michael W. Kattan and Thomas A. Gerds. The index of prediction accuracy: an intuitive measure useful for evaluating risk prediction models. *Diagnostic and Prognostic Research*, 2(1):7, 2018. doi: 10.1186/s41512-018-0029-2. URL https://doi.org/10.1186/s41512-018-0029-2.

[9] Bartosz Krawczyk. Learning from imbalanced data: open challenges and future directions. *Progress in Artificial Intelligence*, 5(4):221–232, 2016.

[10] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002. URL https://CRAN.R-project.org/doc/Rnews/.

[11] Victoria López, Alberto Fernández, Jose G. Moreno-Torres, and Francisco Herrera. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics. *Expert Systems with Applications*, 39(7):6585–6608, 2012.

[12] Victoria López, Alberto Fernández, Salvador Garcia, Vasile Palade, and Francisco Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information Sciences*, 250:113–141, 2013.

[13] Nicola Lunardon, Giovanna Menardi, and Nicola Torelli. ROSE: a Package for Binary Imbalanced Learning. *R Journal*, 6(1):82–92, 2014.

[14] Satyam Maheshwari, R.C. Jain, and R.S. Jadon. An insight into rare class problem: Analysis and potential solutions. *Journal of Computer Science*, 14(6):777–792, May 2018.

[15] Fadel M. Megahed, Ying-Ju Chen, Aly Megahed, Yuya Ong, Naomi Altman, and Martin Krzywinski. The class imbalance problem. *Nature Methods*, 18(11):1270–1272, 2021.

[16] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2022. URL https://CRAN.R-project.org/package=e1071. R package version 1.7-12.

[17] Tim P. Morris, Ian R. White, and Michael J. Crowther. Using simulation studies to evaluate statistical methods. *Statistics in Medicine*, 38(11):2074–2102, 2019.

[18] R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2021. URL https://www.R-project.org/.

[19] Richard D Riley, Joie Ensor, Kym I E Snell, Frank E Harrell, Glen P Martin, Johannes B Reitsma, Karel G M Moons, Gary Collins, and Maarten van Smeden. Calculating the sample size required for developing a clinical prediction model. *BMJ*, 368, 2020.

[20] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics*, 12:77, 2011.

[21] Wacharasak Siriseriwan. *smotefamily: A Collection of Oversampling Techniques for Class Imbalance Problem Based on SMOTE*, 2019. URL https://CRAN.R-project.org/package=smotefamily. R package version 1.3.1.

[22] Ben Van Calster, David J. McLernon, Maarten van Smeden, Laure Wynants, Ewout W. Steyerberg, Patrick Bossuyt, Gary S. Collins, Petra Macaskill, David J. McLernon, Karel G. M. Moons, Ewout W. Steyerberg, Andrew J. Vickers, On behalf of Topic Group 'Evaluating diagnostic tests, and prediction models'of the STRATOS initiative. Calibration: the achilles heel of predictive analytics. *BMC Medicine*, 17(1):230, 2019.

[23] Ruben van den Goorbergh, Maarten van Smeden, Dirk Timmerman, and Ben Van Calster. 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, 06 2022. ISSN 1527-974X. doi: 10.1093/jamia/ocac093. URL https://doi.org/10.1093/jamia/ocac093.

[24] Dennis L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-2(3):408–421, 1972. doi: 10.1109/TSMC.1972.4309137.

[25] Lian Yu and Nengfeng Zhou. Survey of imbalanced data methodologies. 2021. doi: 10.48550/ARXIV.2104.02240. URL https://arxiv.org/abs/2104.02240.

[26] Bing Zhu, Zihan Gao, Junkai Zhao, and Seppe K.L.M. vanden Broucke. Iric: An r library for binary imbalanced classification. *SoftwareX*, 10:100341, 2019.