# Pilot Study: Assessing the impact of class imbalance on the performance of prediction models developed for dichotomous risk prediction.

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

a nice introduction:

1. General background to set the scene, working to the specific topic of your thesis at the final sentence.

Prediction modelling in medicine is receiving increasing attention.

While an abundance of prediction models exist - few are clinically useful.

This is because of calibration is the achilles heel.

Harder to develop a good prediction model in medicine than it sounds.

---

Major obstacle to developing good prediction models in medicine is class imbalance.

Known to degrade the predictive performance of the models – less information about the class we are most interested in.

2. Summary of existing work on the topic you're addressing + exact formulation of knowledge gap.
3. Why this work will address the knowledge gap and how

In this research project, we aim to determine the best practices for handling class imbalance when developing clinical prediction models for dichotomous risk prediction. The harms of imbalance corrections have been previously demonstrated for clinical prediction models developed using logistic regression.[1]

We begin this project with a pilot study. The aim of this pilot study is to demonstrate the baseline performance of various prediction models developed in the presence of class imbalance. We do not pre-process data to correct for class imbalance. In this pilot study, we aim to answer the questions: how does class imbalance affect the performance of clinical prediction models developed using various classification algorithms?

## 2. Methods

In this paper, we implemented a simulation study to investigate the effect of class imbalance on the performance of various classification algorithms. We adhere to the ADEMP guidelines for the design and reporting of our simulation study.[2]

### Aim

The aim of this pilot study was to investigate the effect of class imbalance on the performance of six commonly used classification algorithms. In particular, we aimed to assess the out-of-sample predictive performance of prediction models trained with various classification algorithms, when the event fraction was varied across three levels and no imbalance corrections were implemented.

### Data-Generating Mechanism

Data for each class was 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 these data generating distributions were 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 were fixed to zero and all variances were fixed to 1. In class 1, all means were non-zero and are stored in the vector $\boldsymbol{\Delta_\mu}$. There was no variation in means among predictors within a class, thus, every element in the vector $\boldsymbol{\Delta_\mu}$ is equivalent; denoted by $\delta_\mu$. Similarly, there was no variation in predictor variances within a class, so every diagonal element in $\boldsymbol{\Delta_\Sigma}$ is equivalent; diagonal elements are denoted by $\delta_\Sigma$.

Finally, 80% of the predictors were allowed to covary. All correlations among predictors in each class were set to 0.2. To ensure the correlation of predictors was not stronger in one class, the correlation matrices of the two classes were fixed to be equal. This was accomplished by computing the off-diagonal elements of $\boldsymbol{\Delta_\Sigma}$ such that the correlation matrices between the two classes were equivalent. Note, the covariance matrices were *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, $z = (1 - \delta_\Sigma) * 0.2$, to ensure the correlation matrices of the two classes were equivalent.

In the study we investigated simulated data to reflect three unique scenarios. This was accomplished by varying the event fraction through the set {0.5, 0.2, 0.02}. For all scenarios, data for 8 predictors was generated and the sample size ($N$) was determined as the minimum sample size required for the prediction model using the R package `pmsampsize`.[3]

Under the assumption of normality for all predictors (in each class), the $\Delta C$ Statistic of the data 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 ($\delta_\mu$ and $\delta_\Sigma$) were selected to generate a $\Delta C$ Statistic = 0.85. Their values were computed analytically, based on the following formula:[4]

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

In equation (1), $\Phi$ 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 $\delta_\Sigma$ was fixed at 0.3 for each scenario, while equation (1) was solved to yield the appropriate value of $\delta_\mu$ in each scenario.

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

Parameter values for the data generating distributions of the simulation scenarios are presented in Table 1.

**Table 1:** Summary of the data generating parameters for each simulation scenario.

| Event Fraction | No. Predictors | N | $\delta_\mu$ | $\delta_\Sigma$ | $\Delta C$ |
|---|---|---|---|---|---|
| 0.50 | 8 | 385 | 0.6043 | 0.3 | 0.85 |
| 0.20 | 8 | 247 | 0.6043 | 0.3 | 0.85 |
| 0.02 | 8 | 1797 | 0.6043 | 0.3 | 0.85 |

**Simulation Methods**

Under each simulation scenario, 200 data sets were generated. Each data set was comprised of training and validation data. The training and validation data were generated independently using identical data generating mechanisms; this was done to ensure a similar event fraction in the training and validation data. The validation data was generated to be 10x larger than the training set (Table 1).

For each generated data set, six prediction models were developed, each using a different classification algorithm. All prediction models were trained using the training data. Out-of-sample performance was then assessed using the validation data.

Classification algorithms were selected based on a systematic review identifying common algorithms used to develop prediction models in a medical context.[5] These algorithms include: logistic regression, support vector machine, random forest and XGBoost. Additionally, based on literature summarizing common strategies to handle class imbalance,[6,7,8,9] we include two ensemble learning algorithms designed specifically to handle class imbalance: RUSBoost and and EasyEnsemble. Classification algorithms were implemented with their default hyper-parameters; no hyper-parameter tuning was conducted. All classification algorithms and the R packages used for their implementation are summarized in Table 2.

**Table 2:** Summary of classification algorithms used in simulation study.

| Classification Algorithm | Abbreviation | R Package |
|---|---|---|
| Logistic Regression | LR | base R[10] |
| Support Vector Machine | SVM | e1071[11] |
| Random Forest | RF | randomForest[12] |
| XGBoost | XG | xgboost[13] |
| RUSBoost | RB | ebmc[14] |
| EasyEnsemmble | EE | iric[15] |

**Performance Measures**

Out-of-sample model performance was assessed using measures of calibration, discrimination and overall performance. All performance metrics were computed using the validation data.

Calibration refers to the reliability of the risk predictions; it measures the agreement between the risk predictions and observed proportions in the data.[16] A calibration plot is the recommended method for evaluating model calibration.[17] For each simulation iteration, we fitted a flexible calibration curve for all models, using loess regression. In a flexible calibration curve, when risk predictions (x-axis) correspond well to the observed proportions (y-axis), the curve follows a diagonal line ($y = x$).[18] In addition to the calibration plots, calibration intercept and slope were calculated. With respect to calibration intercept and slope, good calibration is represented by values of 0 and 1, respectively.

Discrimination refers to a model's ability to distinguish between the classes.[17] The concordance statistic was used to measure model discrimination ($\Delta$C-statistic); computed using the R package `pROC`.[19] This is the most common metric to asses discrimination and for dichotomous risk prediction, it is equivalent to the area under the Receiver Operating Characteristic (ROC) curve.[16,17] A model which perfectly discriminates between the classes will have a $\Delta$C-statistic of 1; the minimum value for this statistic is 0.5.

Overall performance was measured by brier score. This metric reflects both model discrimination and calibration and is calculated according to the following formula:[16]

$$\text{BrierScore} = \frac{1}{N}\sum_{i=1}^{N}(p_i - o_i)^2 \tag{2}$$

where $N$ is the sample size, $p_i$ represents the estimated probability for the $i$th individual and $o_i$ represents the observed outcome (0 or 1) for the $i$th individual. In an ideal model, estimated probabilities approximate the observed outcome well for all individuals; ideal models produce a brier score near to zero.

For empirical measures of model performance ($\Delta C$ statistic, brier score, calibration intercept and slope), the mean over all iterations and corresponding monte carlo error are reported. No measures of classification accuracy were considered. Measures of classification accuracy require a decision threshold to be selected and in this simulation study, there is not sufficient context to motivate where to place a threshold.

**Software**

All analyses were conducting using R version 4.1.2.[10]

## 3. Results

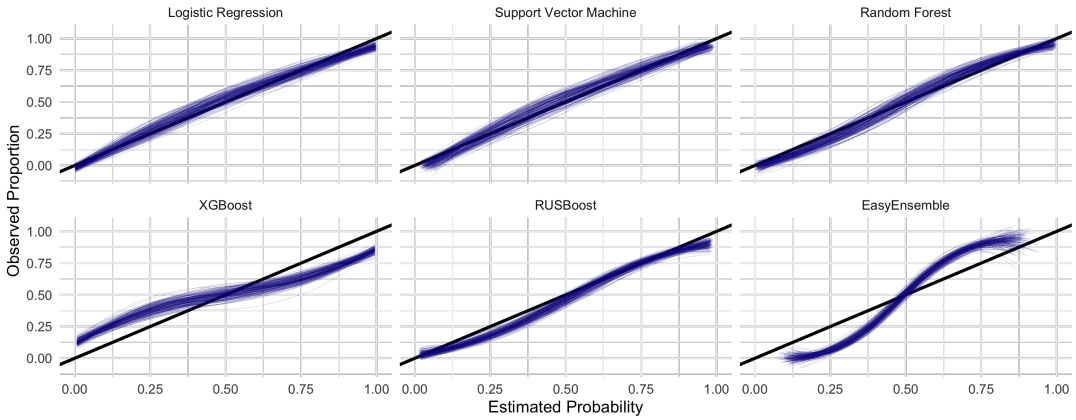
Results are summarized in Table 3 and model performance visualizations are displayed in Figure 1.

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 XG, 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 XG models resulted in risk estimates which were too extreme (calibration slope = 0.464). The opposite pattern is true for EE; EE produced risk estimates which were too moderate (calibration slope = 2.279). In the balanced data scenario, SVM and LR had similar discrimination and overall performance and both outperformed the other algorithms.

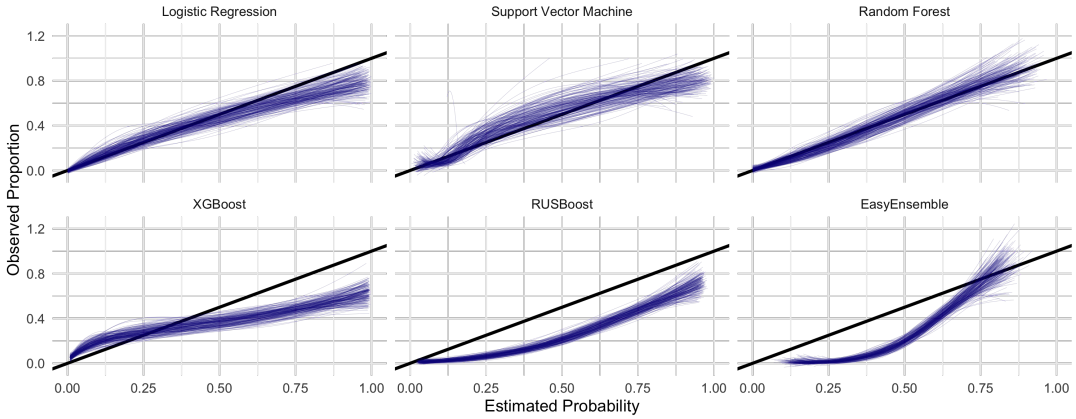
With the event fraction at 0.2, all algorithms exhibited worse calibration, on average, compared to the balanced data scenario. While the LR, SVM and RF maintained adequate calibration in this scenario, XG, RB and EE all, on average, produced risk predictions that dramatically over-estimated true risk (Figure 1(b)). With respect to discrimination and overall performance, 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 LR, SVM and RF, there was large variation in the calibration curves produced for each iteration of the simulation. Meanwhile, for XG, RB, and EE, 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 discrimination and overall performance, in this scenario, LR was again, the best performing algorithm.

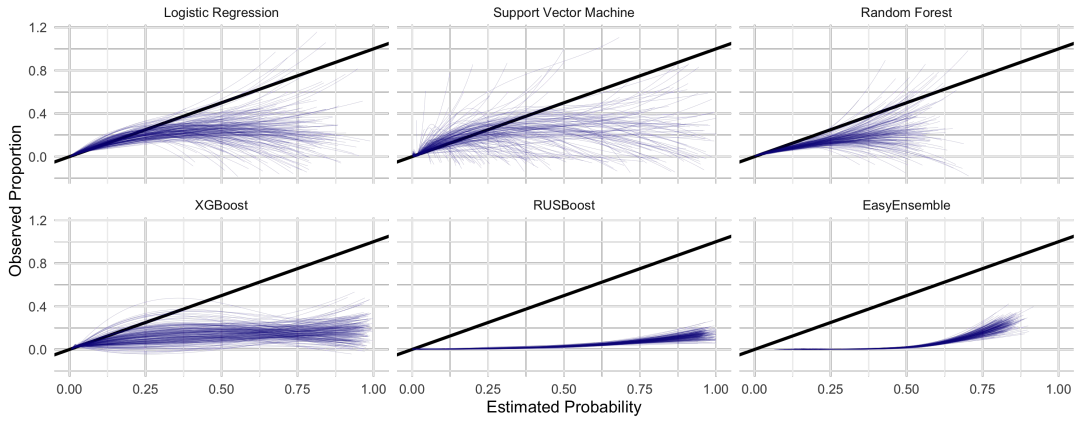
Overall, as imbalance between the classes was magnified, model calibration deteriorated across all algorithms. Meanwhile, discrimination maintained relatively constant. Interestingly, as imbalance between the classes was magnified, overall performance appeared to improve, especially for the LR and SVM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




**(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.

**Table 3:** Mean (monte carlo error) of performance metrics across 200 iterations in each simulation scenario.

| | $\Delta C$ Statistic | Brier Score | Calibration Intercept | Calibration Slope |
|---|---|---|---|---|
| **Event Fraction: 0.5** | | | | |
| LR | 0.845 (0.007) | 0.161 (0.004) | -0.001 (0.140) | 0.928 (0.098) |
| SVM | 0.849 (0.008) | 0.158 (0.004) | -0.002 (0.131) | 1.022 (0.120) |
| RF | 0.842 (0.008) | 0.163 (0.004) | -0.016 (0.124) | 1.168 (0.087) |
| XG | 0.790 (0.012) | 0.205 (0.007) | -0.046 (0.217) | 0.464 (0.024) |
| RB | 0.813 (0.010) | 0.178 (0.005) | -0.183 (0.081) | 1.063 (0.072) |
| EE | 0.826 (0.010) | 0.187 (0.004) | 0.001 (0.045) | 2.279 (0.176) |
| **Event Fraction: 0.2** | | | | |
| LR | 0.836 (0.013) | 0.122 (0.005) | -0.035 (0.216) | 0.860 (0.137) |
| SVM | 0.810 (0.026) | 0.123 (0.007) | -0.025 (0.197) | 1.049 (1.057) |
| RF | 0.808 (0.018) | 0.125 (0.005) | -0.079 (0.179) | 1.079 (0.120) |
| XG | 0.755 (0.022) | 0.153 (0.008) | 0.006 (0.304) | 0.445 (0.040) |
| RB | 0.796 (0.018) | 0.184 (0.011) | -1.335 (0.107) | 1.078 (0.115) |
| EE | 0.814 (0.018) | 0.196 (0.009) | -1.302 (0.064) | 2.331 (0.256) |
| **Event Fraction: 0.02** | | | | |
| LR | 0.839 (0.011) | 0.019 (0.001) | 0.014 (0.199) | 0.904 (0.124) |
| SVM | 0.698 (0.034) | 0.019 (0.001) | -0.013 (0.193) | 1.255 (2.652) |
| RF | 0.756 (0.022) | 0.019 (0.001) | -0.115 (0.189) | 0.655 (0.083) |
| XG | 0.707 (0.028) | 0.022 (0.001) | -0.331 (0.186) | 0.508 (0.047) |
| RB | 0.778 (0.027) | 0.158 (0.019) | -3.696 (0.157) | 0.869 (0.152) |
| EE | 0.810 (0.019) | 0.200 (0.014) | -3.738 (0.080) | 2.295 (0.304) |

## 4. Discussion

In this paper we investigated the impact of class imbalance on the performance of clinical prediction models developed using six different classification algorithms. The results of this Pilot Study illustrated the baseline performance of the prediction models; no imbalance corrections were applied. *From the results, calibration, discrimination overall performance.* Two significant limitations to this study include the performance metric chosen to measure overall model performance and the lack of hyper-parameter tuning for algorithms that typically require it.

Based on the results of the pilot study, brier score appeared to be an uninformative measure of overall performance when class imbalance is extreme. With event fraction of 0.02, a trivial majority classifier (a model that predicts everyone will belong to the majority class) would yield a brier score of 0.02. Therefore, in our future work, we will utilize another metric of overall performance, such a re-scaled brier score[16] which is known to be more informative in the presence of class imbalance.

The algorithms RUSBoost and EasyEnsemble are designed specifically to handle class imbalance. Interestingly, these algorithms exhibited the highest degree of miscalibration in the presence of class imbalance. In this pilot study, they did not out perform logistic regression with respect to classification and discrimination in any scenario. In fact, these algorithms had worse overall performance than a trivial majority classifier at the most extreme event fraction. The relatively poor performance of these algorithms may be due to the lack of hyper parameter tuning; all algorithms in this study were implemented using their software defaults.

*nice last paragraph* - introduce the full simulation - mention hyper parameter tuning, re-scaled brier – maybe clinical utility

# References


[1] Ruben van den Goorbergh et al. "The harm of class imbalance corrections for risk prediction models: illustration and simulation using logistic regression". In: *Journal of the American Medical Informatics Association* 29.9 (June 2022), pp. 1525–1534. ISSN: 1527-974X. DOI: 10.1093/jamia/ocac093. URL: https://doi.org/10.1093/jamia/ocac093.

[2] Tim P. Morris, Ian R. White, and Michael J. Crowther. "Using simulation studies to evaluate statistical methods". In: *Statistics in Medicine* 38.11 (2019), pp. 2074–2102.

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

[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". In: *Statistics in Medicine* 30.12 (2011), pp. 1410–1418.

[5] Constanza Navarro et al. "Systematic review identifies the design and methodological conduct of studies on machine learning-based prediction models". In: *Journal of Clinical Epidemiology* (Nov. 2022). DOI: 10.1016/j.jclinepi.2022.11.015.

[6] Prabhjot Kaur and Anjana Gosain. "Empirical Assessment of Ensemble based Approaches to Classify Imbalanced Data in Binary Classification". In: *International Journal of Advanced Computer Science and Applications* (2019).

[7] Victoria López et al. "Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics". In: *Expert Systems with Applications* 39.7 (2012), pp. 6585–6608.

[8] Satyam Maheshwari, R.C. Jain, and R.S. Jadon. "An Insight into Rare Class Problem: Analysis and Potential Solutions". In: *Journal of Computer Science* 14.6 (May 2018), pp. 777–792.

[9] Lian Yu and Nengfeng Zhou. *Survey of Imbalanced Data Methodologies*. 2021. DOI: 10.48550/ARXIV.2104.02240. URL: https://arxiv.org/abs/2104.02240.

[10] 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/.

[11] David Meyer et al. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*. R package version 1.7-12. 2022. URL: https://CRAN.R-project.org/package=e1071.

[12] Andy Liaw and Matthew Wiener. "Classification and Regression by randomForest". In: *R News* 2.3 (2002), pp. 18–22. URL: https://CRAN.R-project.org/doc/Rnews/.

[13] Tianqi Chen et al. *xgboost: Extreme Gradient Boosting*. R package version 1.6.0.1. 2022. URL: https://CRAN.R-project.org/package=xgboost.

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

[15] Bing Zhu et al. "IRIC: An R library for binary imbalanced classification". In: *SoftwareX* 10 (2019), p. 100341.

[16] Ewout W Steyerberg et al. "Assessing the performance of prediction models: a framework for traditional and novel measures". In: *Epidemiology (Cambridge, Mass.)* 21.1 (Jan. 2010), pp. 128–138.

[17] Fadel M. Megahed et al. "The class imbalance problem". In: *Nature Methods* 18.11 (2021), pp. 1270–1272.

[18] Ben Van Calster et al. "Calibration: the Achilles heel of predictive analytics". In: *BMC Medicine* 17.1 (2019), p. 230.

[19] Xavier Robin et al. "pROC: an open-source package for R and S+ to analyze and compare ROC curves". In: *BMC Bioinformatics* 12 (2011), p. 77.