Post-Calibration Techniques: Balancing Calibration and Score Distribution Alignment

Anonymous Author(s)
Affiliation
Address
email

Abstract

A binary scoring classifier can appear well-calibrated according to standard calibration metrics, even when the distribution of scores does not align with the distribution of the true events. In this paper, we investigate the impact of post-processing calibration on the score distribution (sometimes named "recalibration"). Using simulated data, where the true probability is known, followed by real-world datasets with prior knowledge on event distributions, we compare the performance of an XGBoost model before and after applying calibration techniques. The results show that while applying methods such as Platt scaling or isotonic regression can improve the model's calibration, they may also lead to an increase in the divergence between the score distribution and the underlying event probability distribution.

1 Introduction

When estimating a probabilistic scoring classifier, the model must not only discriminate between observations according to their class but also return scores that can be interpreted as probabilities. The distribution of scores produced by the classifier should align with the underlying event distribution. To assess whether classifiers return probabilistic scores, one must evaluate the model's calibration [4, 23, 8]. While some models, such as logistic regression (LR) when correctly specified, are known to be well-calibrated [21], others, including ensemble methods like Random Forests (RF) [13, 3] and XGBoost [10], are not inherently calibrated [27]. To assess a model's ability to provide probabilistic scores, the literature recommends evaluating its calibration using metrics like the Brier Score (BS, [4]) or the Integrated Calibration Index (ICI, [1]). When a model is not well-calibrated, post-processing calibration methods including Platt scaling [28], or isotonic regression [36] are often applied to adjust the scores [20, 11, 17]. After applying calibration techniques, these metrics generally indicate an improvement in the model's calibration relative to its initial state.

Since the true underlying probability distribution of the data is typically unobserved in practice, calibration metrics are assessed solely on the classifier's output range. [9] demonstrated with simulated data that methods such as RF and XGBoost, can appear well-calibrated according to standard calibration metrics and exhibit strong discrimination based on performance metrics, yet still fail to align the score distribution with the true event distribution. This discrepancy can arise when predicted scores from those algorithms lack the heterogeneity present in the underlying data distribution. To address this, they suggest selecting model hyperparameters based on Kullback-Leibler (KL) divergence, knowing the true event distribution in the case of simulated data. For real data, where the true distribution is unknown, the approach involves prior information about the underlying data distribution to better align it with the predicted score distribution. Their analysis only considers model evaluation to accurately interpret predicted scores as probabilities, typically through calibration metrics. However, many practitioners employ calibration techniques to ensure that output scores

Submitted to Workshop on Bayesian Decision-making and Uncertainty, 38th Conference on Neural Information Processing Systems (BDU at NeurIPS 2024). Do not distribute.

represent probabilistic estimates. In this paper, we examine how calibration techniques affect the variability of score distribution in XGBoost binary classifiers, comparing it to the true underlying data distribution using KL divergence. We find that these post-processing methods often reduce score heterogeneity. Additionally, tree-based models optimized for KL divergence, rather than calibration or performance metrics, tend to align more closely with the underlying data distribution after calibration. Using simulated data, where the true probability is known, followed by real-world datasets with prior knowledge of event distributions, we assess the accuracy of XGBoost predicted scores as probabilistic estimates before and after applying calibration techniques.

2 Calibration

We focus on the context of a binary scoring classifier. Let $Y \in \mathcal{Y} = \{0, 1\}$ be a binary response variable, and let $\boldsymbol{X} \in \mathcal{X} = \mathbb{R}^d$ denote features. The goal is to predict $s(\boldsymbol{X}) = \mathbb{P}(Y = 1|\boldsymbol{X})$, using a sample of n i.i.d. observations $(\mathbf{x}_i, y_i)_{i=1}^n$. We estimate this probability $\hat{s}(\mathbf{x}_i) \in [0, 1]$ using an XGBoost classifier, which produces a distribution of estimated scores $\hat{s}(\boldsymbol{X})$. If the score distribution is poorly calibrated, these scores cannot be interpreted as the "true underlying probabilities" [32, 18, 16]. A model $\hat{s}$ is well-calibrated for a binary variable Y when [30]:

$$\mathbb{P}(Y = 1 \mid \hat{s}(\boldsymbol{X})) = \mathbb{E}[Y \mid \hat{s}(\boldsymbol{X})] = \hat{s}(\boldsymbol{X}) \quad \text{a.s.}, \tag{1}$$

i.e., equivalently, $\mathbb{E}[Y \mid \hat{s}(\boldsymbol{X}) = p] = p, \forall p \in [0, 1]$.

2.1 Calibration Metrics

To measure calibration, the literature suggests various metrics. Here, we focus on two of them: BS and ICI. The former [12, 17, 28, 29] is often used to assess a model's calibration. It writes: $\text{BS} = n^{-1} \sum_{i=1}^n \left(\hat{s}(\mathbf{x}_i) - y_i\right)^2$ [4]. More recently, [1] introduced the ICI, a metric that relies on the calibration curve. In the binary case, the calibration curve writes $\mathrm{g} : [0, 1] \to [0, 1], \quad p \mapsto \mathrm{g}(p) := \mathbb{E}[Y \mid \hat{s}(\mathbf{X}) = p]$. For a well-calibrated model, the g function is the identity function $\mathrm{g}(p) = p$. While the calibration curve is usually estimated using bins [34, 19, 26], the ICI relies on a smoother version, based on splines. The empirical version writes $\text{ICI} = n^{-1} \sum_{i=1}^n \left|\hat{s}(\mathbf{x}_i) - \hat{\mathrm{g}}\big(\hat{s}(\mathbf{x}_i)\big)\right|$, which corresponds to computing the average of the absolute difference between the calibration curve and the 45-degree diagonal line, the latter representing perfect calibration.

2.2 Calibration Methods

When using scores generated by a model estimating the probability of a binary event, the literature advocates calibrating the model by applying the calibration curve g—which serves as a transformation function—on the scores [28, 36, 17, 19]. In this paper, we focus on two calibration methods: Platt Scaling, and isotonic regression.

Platt Scaling This methodology was initially introduced to map SVM outputs to well-calibrated posterior probabilities [28]. This parametric approach consists of fitting a LR to the binary response variable using predicted scores of a binary classifier as the unique feature. Platt scaling learns parameters, μ and s ($s > 0$ for a non-decreasing calibration map g) on a calibration set. The obtained calibrated probabilities are $\mathrm{g}\left(\hat{s}(\mathbf{x})\right) = \left(1 + \exp\left\{-\frac{1}{s}\left(\hat{s}(\mathbf{x}) - \mu\right)\right\}\right)^{-1}$. As pointed out in [17], Platt scaling is unable to learn the identity function g if the predicted scores are already calibrated.

Isotonic Regression This solution arises from a constrained optimization problem [36], solved using the Pool-Adjacent-Violators Algorithm, ensuring that corrected predicted scores remain monotonic: $\min_{\beta_1,\ldots,\beta_n} \sum_{i=1}^n (y_{(i)} - \beta_i)^2$, s.t. $\beta_1 \leq \ldots \leq \beta_n$. $y_{(i)}$ corresponds to the value in $\{y_1, \cdots, y_n\}$ associated with the i-th largest predicted score $\{\hat{s}(\mathbf{x}_1), \cdots, \hat{s}(\mathbf{x}_n)\}$. Isotonic regression, like Platt scaling, assumes the initial model's predicted scores $\hat{s}(\mathbf{x})$ are well-ordered, limiting its ability to correct non-monotonic probability distortions.

3 Score Heterogeneity

To accurately interpret predicted scores from a binary classifier as probabilistic estimates, since the true underlying probability $s(\boldsymbol{X})$ is unobservable, calibration metrics rely solely on the predicted

score range. When a binary classification model is well-calibrated, the distribution of its scores $\hat{s}(\boldsymbol{X})$, as defined by Eq. 1, should align with the actual probability of the event in the vicinity of score values. Therefore, calibration metrics cannot fully capture discrepancies between the score distribution and the true probability distribution of the response variable Y when the predicted score variability does not accurately reflect the latter.

Kullback-Leibler divergence [9] demonstrated through simulated data that scores from ensemble methods may exhibit less variability compared to the true underlying probabilities when selecting hyperparameters based on calibration (ICI) or performance (AUC) metrics. This reduced heterogeneity makes calibration metrics less reliable for interpreting output scores as probabilities of event occurrence. Instead of evaluating discrepancies solely on predicted score values, the authors suggest evaluating the model's probabilistic estimates using KL divergence between the overall score distribution, $\hat{s}(\boldsymbol{X})$, and the available information on the "true" distribution, $s(\boldsymbol{X})$. Additionally, the flexibility of tree-based methods such as XGBoost allows for the selection of model hyperparameters based on KL divergence rather than traditional performance metrics, without incurring significant losses in performance or calibration and ensuring that the predicted score distribution more closely aligns with prior knowledge.

Bayesian Framework When working with simulated data, the distribution $s(\boldsymbol{X})$ is fully known, allowing for the direct computation of KL divergence with $\hat{s}(\boldsymbol{X})$. However, with real data, the KL divergence can only be computed by relying on a prior belief about the distribution of $s(\boldsymbol{X})$, potentially informed by expert opinion, and thus assuming a prior distribution $\mathcal{B}$. In the following, as in [9], we take $s(\boldsymbol{X}) \sim \mathcal{B} = \text{Beta}(\alpha, \beta)$ as the assumed prior distribution where each true probability p_i of the i-th observation is a sample from $\mathcal{B}$. We observe a sequence of n independent (as the features $\boldsymbol{X}_i$ are considered n i.i.d. random variables) but non-identically distributed binary random variables Y_i where $Y_i|s(\boldsymbol{X}_i) = p_i \sim \text{Bernoulli}(p_i)$. In this case, instead of selecting the model with hyperparameters that minimize the empirical mean of the KL divergence across individual distributions, given by $\frac{1}{n}\sum_{i=1}^{n} \hat{s}(\mathbf{x_i}) \log \frac{\hat{s}(\mathbf{x_i})}{p_i} + (1 - \hat{s}(\mathbf{x_i})) \log \frac{1-\hat{s}(\mathbf{x_i})}{1-p_i}$, we adopt the approach proposed by [9] and minimize the distance between the prior distribution $\mathcal{B}$ and the overall distribution of $\hat{s}(\boldsymbol{X})$.

Calibration techniques We extend the work of [9] by investigating how score heterogeneity predicted by certain XGBoost algorithms is affected after applying post-calibration techniques such as Platt scaling or isotonic regression. These methods can potentially reduce score heterogeneity; for instance, isotonic regression applies a stepwise function g. Additionally, with Platt scaling, the range of calibrated predicted scores is always narrower than the range of the initial scores when the parameter $s \geq \frac{1}{4}$ (see Appendix A.1). And, due to the concavity of the sigmoid function over $[0, +\infty]$, this post-calibration method tends to reduce the range of predicted scores more significantly when the initial scores are highly concentrated.

4 Numerical Experiments

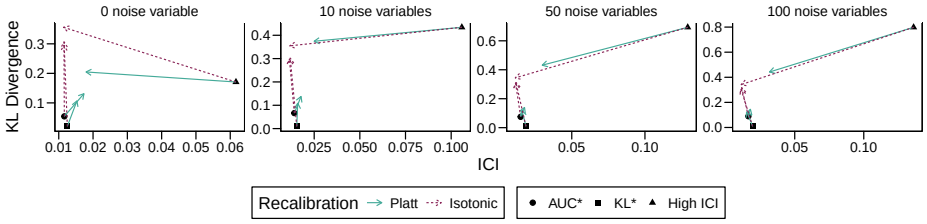
4.1 Simulated Data

We use the simulated data from [9]. We consider four data-generating processes (DGPs), all of which use a logistic link function. The first three are from [25], the fourth adds interaction terms. More details can be found in Appendix B.1. For each DGP, we generate data that include more or less noise variables: 0, 10, 50 or 100. We split the data into four samples: the train and validation samples used to train an XGBoost model and select the set of hyperparameters, the calibration sample to train a calibrator using the selected model, and lastly, a test sample to assess the performance of models on unseen data. We select the model's hyperparameters (number of boosting iterations and maximum tree depth) to optimize three different criteria on the validation set: maximizing AUC (AUC*), minimizing KL divergence (KL*), or, for illustrative purposes, producing a model that is poorly calibrated based on the ICI metric (High ICI). Once the hyperparameters are selected, a calibration technique is applied to the scores. This allows for a comparison of models on the test set, both before and after calibration, according to the chosen optimization criterion. We run the simulations on 100 replications for each configuration. The results for DGP 1 are shown in Fig. 1

(see Fig. C18 for full results and Table C3 for numerical values). The shapes represent models before calibration, and arrows indicate model performance after calibration.

The results highlight two key findings. First, when hyperparameters are optimized for AUC, isotonic regression enhances calibration if the model is initially well-calibrated, while Platt scaling often fails due to the logistic family lacking the identity function. Second, calibration techniques generally increase the KL divergence between scores and true probabilities, except when the model is poorly calibrated (High ICI) with multiple noise variables. As shown in the histograms from Figs. C2 to C17, after applying calibration techniques, the model becomes better calibrated but the score distribution moves further from the true distribution.

We conclude that optimizing hyperparameters based on KL divergence rather than AUC leads to a closer alignment between the score distribution and the true distribution, even after post-processing calibration. Additionally, calibration techniques can even increase the distance from the true probability distribution by reducing score variability.



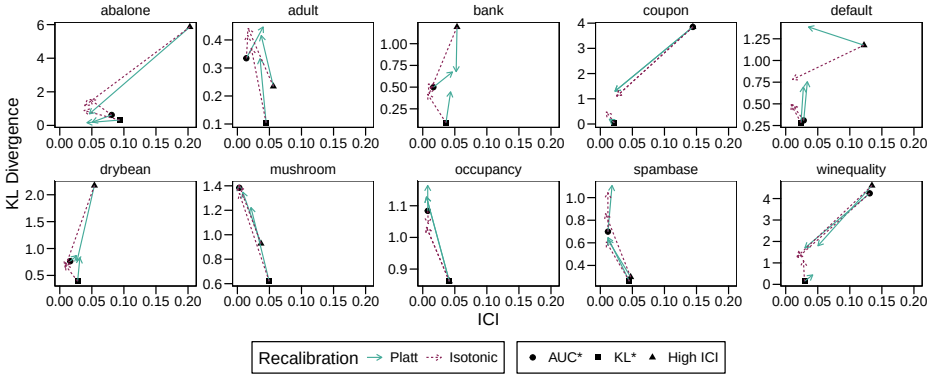
Notes: AUC*, KL*, High ICI: models selected by optimizing AUC, KL divergence, or by selecting a high ICI.

Figure 1: Average KL divergence and ICI before and after recalibration, for DGP 1.

4.2 Real Data

The 10 datasets from the UCI ML Repository used in [9] are used here (see details in Appendix B.2). For each dataset, we apply the method outlined in Section 4.1, this time calculating the KL divergence between the predicted score distribution and the prior distribution described in Section 3.[1]

The results across the 10 datasets are shown in Fig. 2, with detailed metric values in Table C4. The findings are consistent with Section 4.1: isotonic regression preserves calibration when the model is already well-calibrated, while Platt scaling often worsens it. After calibration, KL* models align best with the prior distribution and, in some cases, show better ICI than AUC* or High ICI models. For the adult, mushroom, occupancy, and spambase datasets, calibration techniques generally increase KL divergence, reducing score distribution heterogeneity.



Notes: AUC*, KL*, High ICI: models selected by optimizing AUC, KL divergence, or by selecting a high ICI.

Figure 2: Average KL divergence and ICI before and after recalibration.

[1] For illustration purposes, the parameters of the prior distribution $\mathcal{B}$ are estimated via maximum likelihood using scores from a GAMSEL model [6], where the event is regressed on the variables generating the data.

References

[1] Austin, P. C. and Steyerberg, E. W. (2019). The integrated calibration index (ici) and related metrics for quantifying the calibration of logistic regression models. *Statistics in Medicine*, 38(21):4051–4065.

[2] Becker, B. and Kohavi, R. (1996). Adult. UCI Machine Learning Repository. doi:10.24432/C5XW20.

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

[4] Brier, G. W. (1950). Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3.

[5] Candanedo, L. (2016). Occupancy Detection . UCI Machine Learning Repository. doi:10.24432/C5X01N.

[6] Chouldechova, A. and Hastie, T. (2015). Generalized additive model selection. *arXiv:1506.03850*.

[7] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., and Reis, J. (2009). Wine Quality. UCI Machine Learning Repository. doi:10.24432/C56S3T.

[8] Dawid, A. P. (1982). The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610.

[9] Fernandes Machado, A., Charpentier, A., Flachaire, E., Gallic, E., and Hu, F. (2024). Probabilistic scores of classifiers, calibration is not enough. *arXiv:2408.03421*.

[10] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232.

[11] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

[12] Gupta, K., Rahimi, A., Ajanthan, T., Sminchisescu, C., Mensink, T., and Hartley, R. I. (2021). Calibration of neural networks using splines. In *International Conference on Learning Representations (ICLR)*.

[13] Ho, T. K. (1995). Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1.

[14] Hopkins, M., Reeber, E., Forman, G., and Suermondt, J. (1999). Spambase. UCI Machine Learning Repository. doi:10.24432/C53G6X.

[15] Koklu, M. and Ali Ozkan, I. (2020). Dry Bean. UCI Machine Learning Repository. doi:10.24432/C50S4B.

[16] Konek, J. (2016). Probabilistic knowledge and cognitive ability. *Philosophical Review*, 125(4):509–587.

[17] Kull, M., Filho, T. M. S., and Flach, P. (2017). Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11(2):5052 – 5080.

[18] Kull, M. and Flach, P. A. (2014). Reliability maps: a tool to enhance probability estimates and improve classification accuracy. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*, pages 18–33. Springer.

[19] Kumar, A., Liang, P. S., and Ma, T. (2019). Verified uncertainty calibration. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

[20] Leathart, T., Frank, E., Pfahringer, B., and Holmes, G. (2019). On calibration of nested dichotomies. In *Springer-Verlag*, page 69–80.

[21] Mildenhall, S. J. (1999). A systematic relationship between minimum bias and generalized linear models. In *Journal Proceedings of the Casualty Actuarial Society*, volume 86, pages 393–487.

[22] Moro, S., Rita, P., and Cortez, P. (2012). Bank Marketing. UCI Machine Learning Repository. doi: https://doi.org/10.24432/C5K306.

[23] Murphy, A. H. (1972). Scalar and vector partitions of the probability score: Part i. two-state situation. *Journal of Applied Meteorology and Climatology*, 11(2):273–282.

[24] Nash, W., Sellers, T., Talbot, S., Cawthorn, A., and Ford, W. (1995). Abalone. UCI Machine Learning Repository. doi:10.24432/C55C7W.

[25] Ojeda, F. M., Jansen, M. L., Thiéry, A., Blankenberg, S., Weimar, C., Schmid, M., and Ziegler, A. (2023). Calibrating machine learning approaches for probability estimation: A comprehensive comparison. *Statistics in Medicine*, 42(29):5451–5478.

[26] Pakdaman Naeini, M., Cooper, G., and Hauskrecht, M. (2015). Obtaining well calibrated probabilities using bayesian binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1):2901–2907.

[27] Park, Y. and Ho, J. C. (2020). Califorest: Calibrated random forest for health data. *Proceedings of the ACM Conference on Health, Inference, and Learning 2020*, pages 40–50.

[28] Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

[29] Rahimi, A., Shaban, A., Cheng, C.-A., Hartley, R., and Boots, B. (2020). Intra order-preserving functions for calibration of multi-class neural networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13456–13467. Curran Associates, Inc.

[30] Schervish, M. J. (1989). A General Method for Comparing Probability Assessors. *The Annals of Statistics*, 17(4):1856–1879.

[31] Schlimmer, J. (1987). Mushroom. UCI Machine Learning Repository. doi:10.24432/C5959T.

[32] Van Fraassen, B. C. (1995). Fine-grained opinion, probability, and the logic of full belief. *Journal of Philosophical logic*, 24(4):349–377.

[33] Wang, T., Rudin, C., Doshi-Velez, F., Liu, Y., Klampfl, E., and MacNeille, P. (2020). In-Vehicle Coupon Recommendation. UCI Machine Learning Repository. doi:10.24432/C5GS4P.

[34] Wilks, D. S. (1990). On the combination of forecast probabilities for consecutive precipitation periods. *Weather and Forecasting*, 5(4):640–650.

[35] Yeh, I.-C. (2016). Default of Credit Card Clients. UCI Machine Learning Repository. doi:10.24432/C55S3H.

[36] Zadrozny, B. and Elkan, C. (2002). Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699.

A Platt scaling

A.1 Reduction in Score Range

Platt scaling learns parameters, μ and s ($s > 0$ for a non-decreasing calibration map g) on a calibration set. The obtained calibrated probabilities are:

$$\mathrm{g}\left(\hat{s}(\mathbf{x})\right) = \frac{1}{1 + \exp\left\{-\frac{1}{s}\left(\hat{s}(\mathbf{x}) - \mu\right)\right\}}.$$

With Platt scaling, the range of calibrated predicted scores is always narrower than the range of the initial scores when the parameter $s \geq \frac{1}{4}$. Indeed, since $\rho = \frac{1}{4}$ is the minimum value for which $\sigma(x) = \frac{1}{1+\exp -x}$ remains ρ-Lipschitz on $\mathbb{R}$, for $x_1 < x_2 \in \mathbb{R}$, we have:

$$\left|\sigma\left(\frac{x_2 - \mu}{s}\right) - \sigma\left(\frac{x_1 - \mu}{s}\right)\right| \leq \frac{1}{4}\left|\frac{x_2 - \mu}{s} - \frac{x_1 - \mu}{s}\right| \leq \frac{1}{4s}\left|x_2 - x_1\right| \text{ with } s > 0.$$

As a result, if $s \geq \frac{1}{4}$, the range of $(\hat{s}(\mathbf{x}_i))_{i=1}^n$ is larger than the range of the calibrated scores with Platt scaling $(\mathrm{g}(\hat{s}(\mathbf{x}_i)))_{i=1}^n$. Let $\hat{s}_m$ (resp. $\hat{s}_M$) denote the minimum (resp. the maximum) value of $(\hat{s}(\mathbf{x}_i))_{i=1}^n$. If $s \geq \frac{1}{4}$, we have:

$$\left|\frac{\mathrm{g}(\hat{s}_M) - \mathrm{g}(\hat{s}_m)}{\hat{s}_M - \hat{s}_m}\right| = \left|\frac{\sigma\left(\frac{\hat{s}_M - \mu}{s}\right) - \sigma\left(\frac{\hat{s}_m - \mu}{s}\right)}{\hat{s}_M - \hat{s}_m}\right| \leq 1.$$

And, due to the concavity of the sigmoid function over $[0, +\infty]$, this post-calibration method tends to reduce the range of predicted scores more significantly when the initial scores are highly concentrated.

B Data

B.1 Simulated Data

To simulate data, we consider the DGPs from [9]. The first three are from Ojeda et al. [25]. In the fourth, an interaction term between two predictors is added. Each scenario uses a logistic model to generate the outcome. Let Y_i be a binary variable following a Bernoulli distribution: $Y_i \sim B(p_i)$, where p_i is the probability of observing $Y_i = 1$. The probability p_i is defined by:

$$p_i = \mathbb{P}(Y = 1 \mid \mathbf{x}_i) = \left[1 + \exp(-\eta_i)\right]^{-1}. \tag{2}$$

For the second DGP, to introduce non-linearities, p^3 is used as true probabilities instead of p.

For all DGPs, $\eta_i = \mathbf{x}_i^\top \boldsymbol{\beta}$, where $\mathbf{x}_i$ is a vector of covariates and $\boldsymbol{\beta}$ is a vector of arbitrary scalars. The covariate vector includes two continuous predictors for DGPs 1 and 2. For DGP 3, it includes five continuous and five categorical predictors. For DGP 4, it contains three continuous variables, the square of the first variable, and an interaction term between the second and third variables. Specifically, $\eta_i = \beta_1 x_{1,i} + \beta_2 x_{2,i} + \beta_3 x_{3,i} + \beta_4 x_{1,i}^2 + \beta_5 x_{2,i} \times x_{3,i}$. Continuous predictors are drawn from $\mathcal{N}(0, 1)$. Categorical predictors consist of two variables with two categories, one with three categories, and one with five categories, all uniformly distributed. The values of coefficients $\boldsymbol{\beta}$ are reported in Table B1.

For each DGP, we generate data considering four scenarios with varying numbers of noise variables: 0, 10, 50, or 100 variables drawn from $\mathcal{N}(0, 1)$.

For the fourth DGP, to achieve a similar probability distribution to DGP 1, we perform resampling using a rejection algorithm (the algorithm is detailed in [9]).

DGP	No. Cont.	No. Cat.	No. Noise	$\boldsymbol{\beta}$	Type η
1	2	0	$\{0, 10, 50, 100\}$	$(.5, 1)$	Linear terms
2	Same as DGP 2, but with probabilities p^3				
3	5	5	$\{0, 10, 50, 100\}$	$(.1, .2, .3, .4, .5, .01, .02, .03, .04, .05)$	Linear terms
4	3	0	$\{0, 10, 50, 100\}$	$(.5, 1, .3)$	Non-linear terms

Notes: No. Cont., No. Cat. and No. Noise correspond to the number of continuous, categorical and noise variables, respectively.

Table B1: Parameters of the different scenarios.

The datasets are split into fourt parts: a training sample, a validation sample, a calibration sample, and a test sample, each containing 10,000 observations. The empirical distribution of samples of from each DGP are shown in Fig. B1.

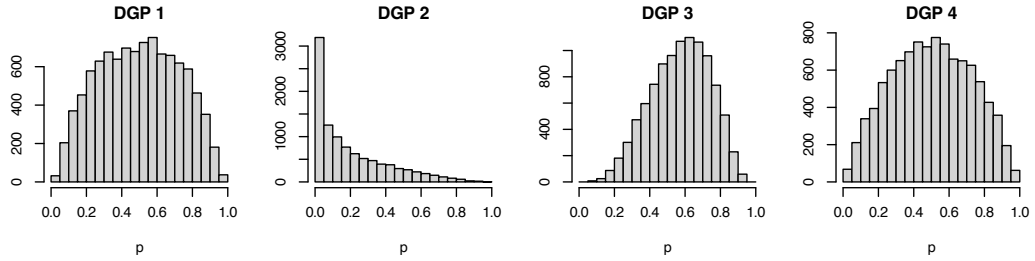


Figure B1: Distribution of the underlying probabilities in the different categories of scenarios.

B.2 Real Data

The main characteristics of the datasets are summarized in Table B2.

Most of the datasets used are associated with classification tasks. If not, they contain a binary variable suitable for classification or a variable that can be converted into a binary variable. The target variables for each dataset are as follows:

Table B2: Key characteristics of the datasets

Dataset	n	No. predictors	No. num. predictors	Prop. target = 1	Reference	License
`abalone`	4,177	8	8	0.37	Nash et al. [24]	CC BY 4.0
`adult`	32,561	14	6	0.24	Becker and Kohavi [2]	CC BY 4.0
`bank`	45,211	16	7	0.12	Moro et al. [22]	CC BY 4.0
`default`	30,000	23	14	0.22	Yeh [35]	CC BY 4.0
`drybean`	13,611	16	16	0.26	Koklu and Ali Ozkan [15]	CC BY 4.0
`coupon`	12,079	22	0	0.57	Wang et al. [33]	CC BY 4.0
`mushroom`	8,124	21	0	0.52	Schlimmer [31]	CC BY 4.0
`occupancy`	20,560	5	5	0.23	Candanedo [5]	CC BY 4.0
`winequality`	6,495	12	11	0.63	Cortez et al. [7]	CC BY 4.0
`spambase`	4,601	57	57	0.39	Hopkins et al. [14]	CC BY 4.0

Notes: n represents the number of observations, 'No. predictors' the total number of predictors, 'No. num. predictors' the number of numeric predictors, and 'Prop. target = 1' the proportion of positive observed events.

- `abalone`: gender of abalones (1 for male, 0 for female); originally used to predict the size of abalones.
- `adult`: high income (1 if income $\geq$ 50k per year).
- `bank`: subscription to a term deposit (1 if yes, 0 otherwise).
- `default`: default payment (1 if default, 0 otherwise).
- `drybean`: type of dry bean (1 if dermason, 0 otherwise); originally a multi-class variable.
- `coupon`: acceptance of a recommended coupon in different driving scenarios (1 if accepted, 0 otherwise).
- `mushroom`: mushroom classification (1 if edible, 0 otherwise).
- `occupancy`: prediction of room occupancy (1 if occupied, 0 otherwise); originally aimed at predicting the age of occupancy from physical measurements.
- `winequality`: quality of wine (1 if quality $\geq$ 6, 0 otherwise); originally a scale from 0 to 10, with 0 being bad quality and 10 being good quality.
- `spambase`: email classification (1 if spam, 0 otherwise).

C Numerical Experiments

C.1 Simulated Data

For each of the four DGPs (see Section B.1) and each configuration of the number of noise variables (0, 10, 50, or 100), we generate 100 sample replications. For each sample, we train an XGBoost model on 10,000 observation using the `xgb.train` function from the R package `xgboost`. The learning rate is set to 0.3. The tree depth (argument `max_depth`) varies according to the following values: 2, 4, 6. The number of boosting iterations (argument `nrounds`) ranges from 1 to 400. All variables (predictors and, if applicable, noise variables) are included in the model without transformation.

For each model configuration, we select the hyperparameters based on different criteria using the validation set results. Specifically, we make three model choices:

- AUC*: hyperparameters are selected to maximize the AUC.
- KL*: hyperparameters are chosen to minimize the Kullback-Leibler divergence between the scores on the validation set and the true probability distribution (observable here in the context of simulated data).
- High ICI: hyperparameters are selected to produce relatively poor calibration, as measured by the ICI. Specifically, we select the model with the smallest ICI among those with an ICI at least one standard deviation above the mean ICI obtained during grid search.

Once the hyperparameters are selected, we apply a recalibration method on an independent calibration set: either Platt scaling or isotonic regression.

The model performance is then evaluated on a test set, allowing for comparison based on: (i) the metric used to select the hyperparameters, and (ii) whether or not calibration techniques were applied to the scores.

Figs C2 to C17 display the empirical distribution of scores for a single replication (the first one) in each of the 4×4 configurations (4 DGPs and 4 different values for the number of noise variables introduced in the training data). In each figure, the first row shows the distribution of test set scores without applying any calibration technique to the selected model. The second row, in green, shows the score distributions after applying Platt scaling for calibration. The third row, in purple, displays the score distributions after applying isotonic regression. The columns correspond to the criteria used to select the hyperparameters based on the validation set results: AUC, Brier, ICI, KL, or High ICI.

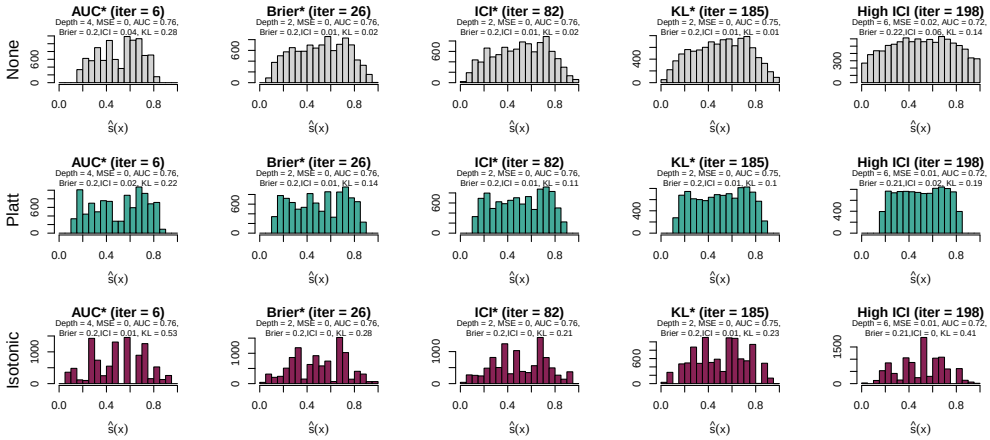


Figure C2: Distribution of estimated scores for XGB: **DGP 1**, **0 noise variable**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

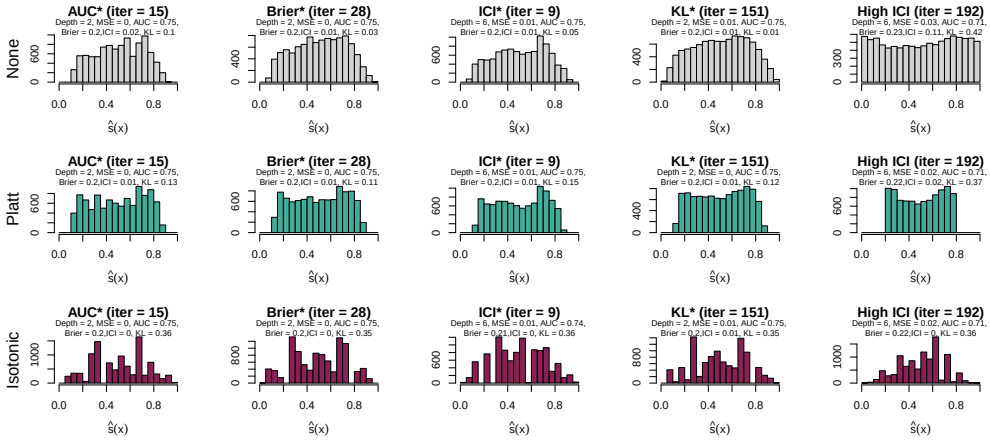


Figure C3: Distribution of estimated scores for XGB: **DGP 1**, **10 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

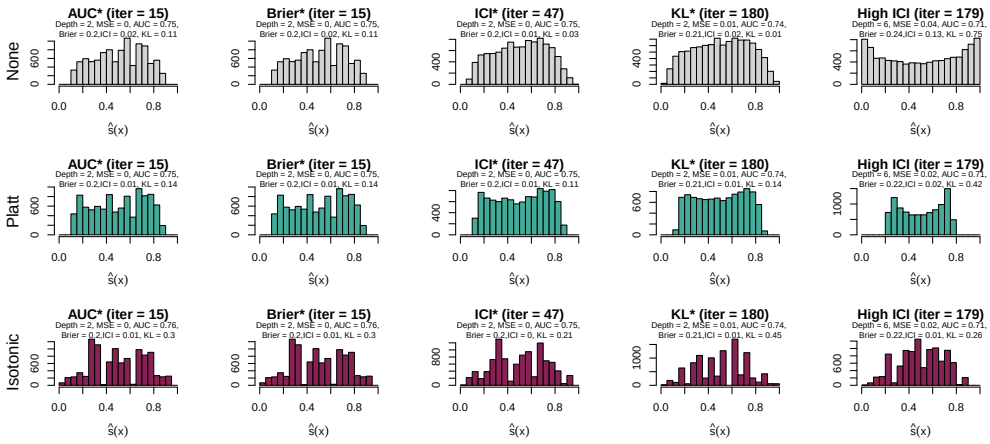


Figure C4: Distribution of estimated scores for XGB: **DGP 1**, **50 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

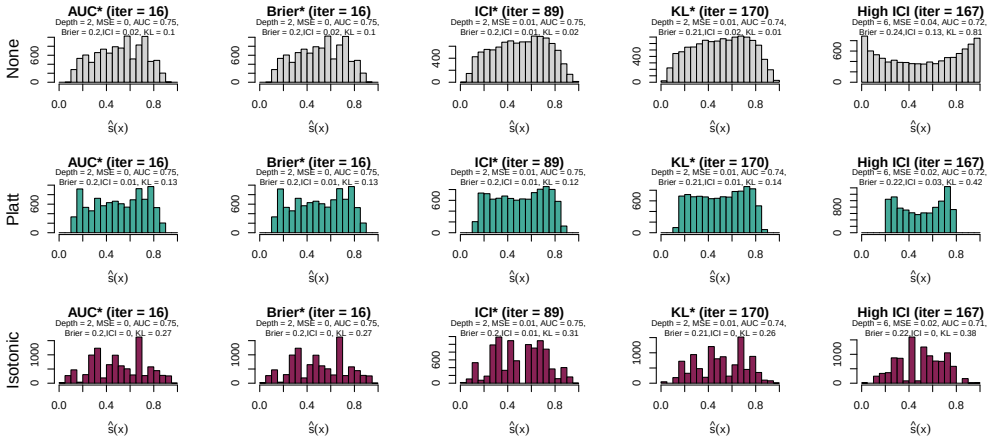


Figure C5: Distribution of estimated scores for XGB: **DGP 1**, **100 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

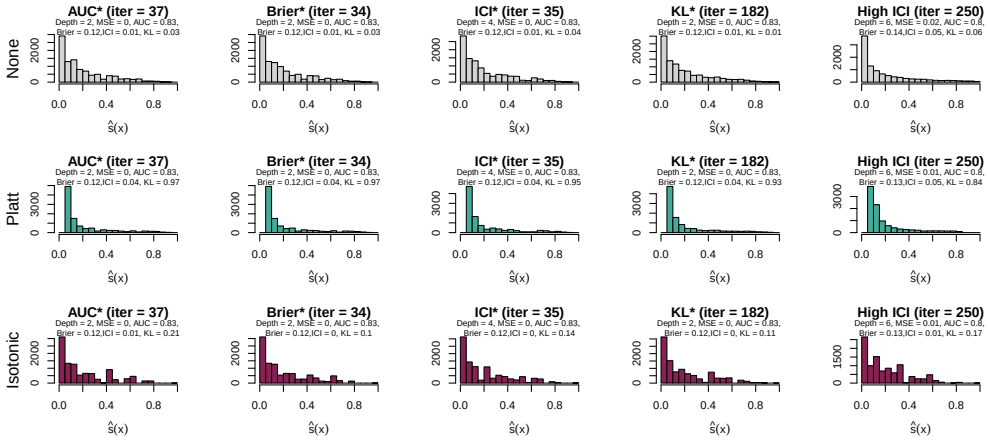


Figure C6: Distribution of estimated scores for XGB: **DGP 2**, **0 noise variable**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

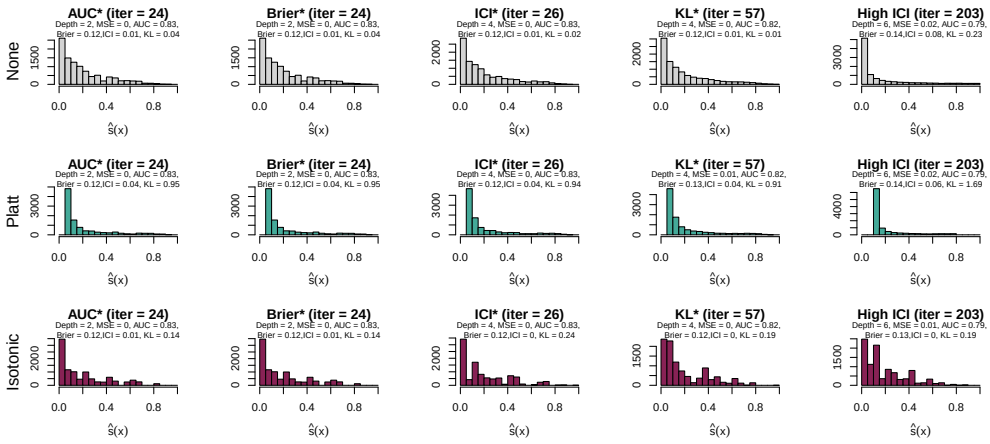


Figure C7: Distribution of estimated scores for XGB: **DGP 2**, **10 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

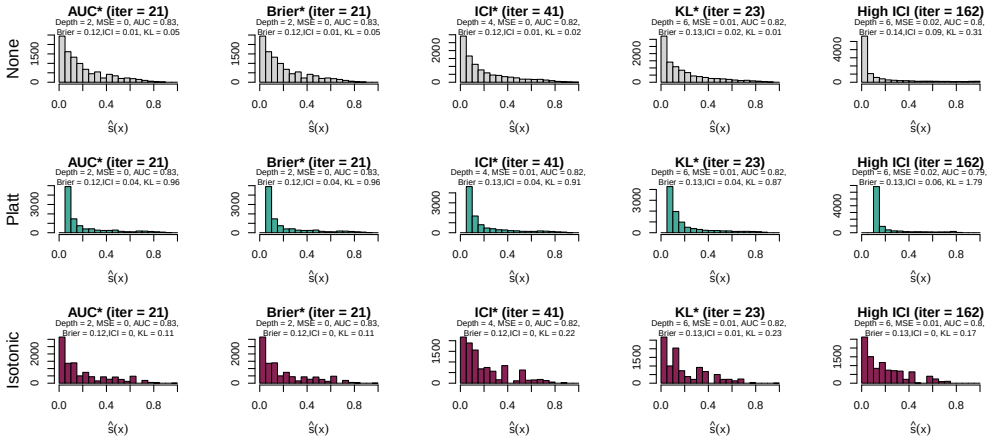


Figure C8: Distribution of estimated scores for XGB: **DGP 2**, **50 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

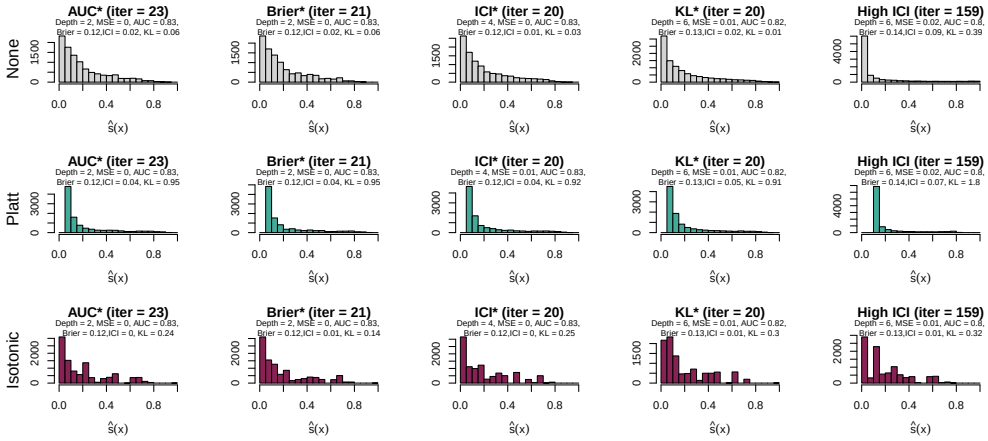


Figure C9: Distribution of estimated scores for XGB: **DGP 2**, **100 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

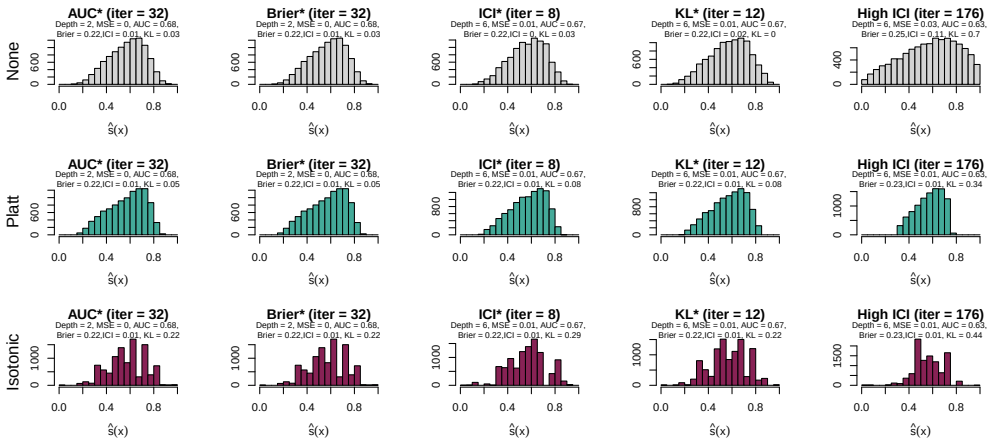


Figure C10: Distribution of estimated scores for XGB: **DGP 3**, **0 noise variable**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

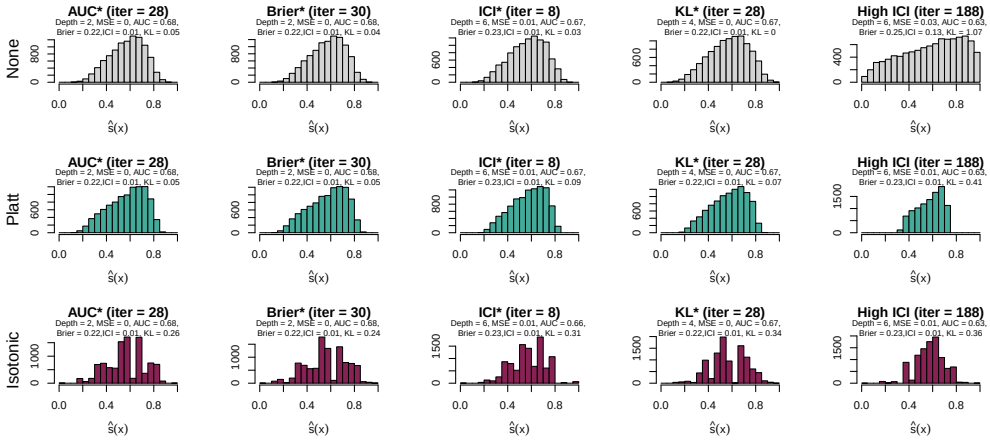


Figure C11: Distribution of estimated scores for XGB: **DGP 3**, **10 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

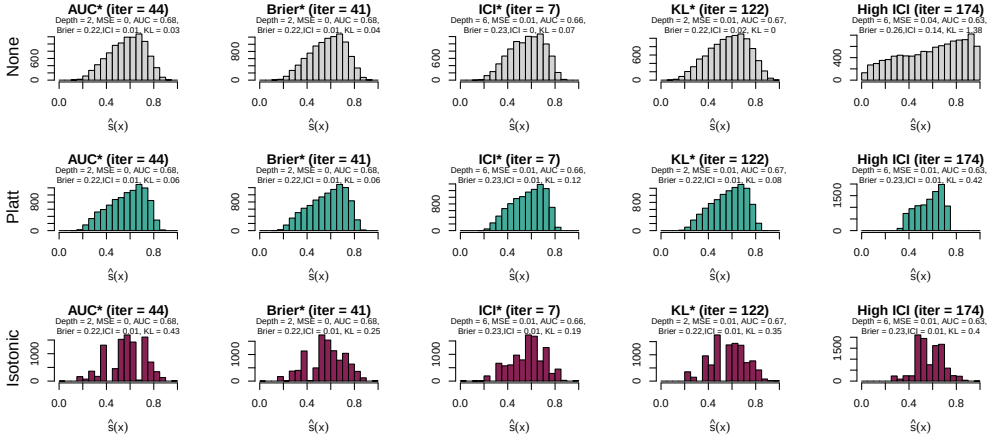


Figure C12: Distribution of estimated scores for XGB: **DGP 3**, **50 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

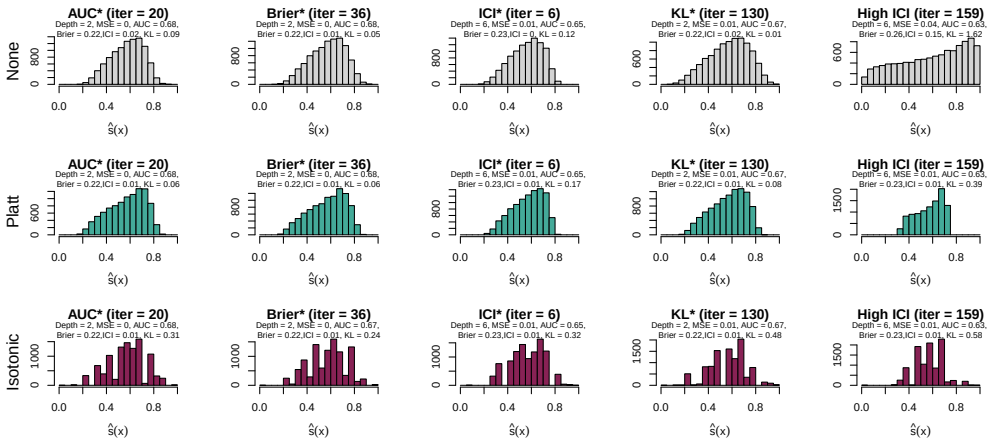


Figure C13: Distribution of estimated scores for XGB: **DGP 3**, **100 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

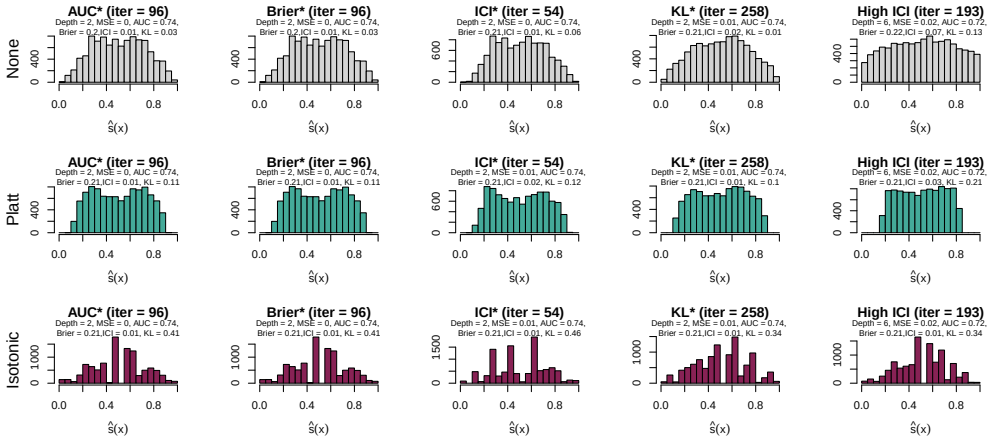


Figure C14: Distribution of estimated scores for XGB: **DGP 4**, **0 noise variable**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

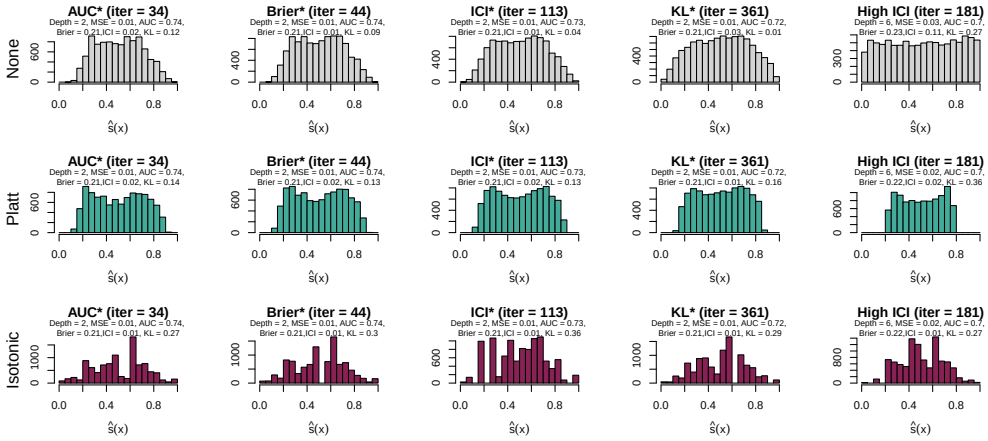


Figure C15: Distribution of estimated scores for XGB: **DGP 4**, **10 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

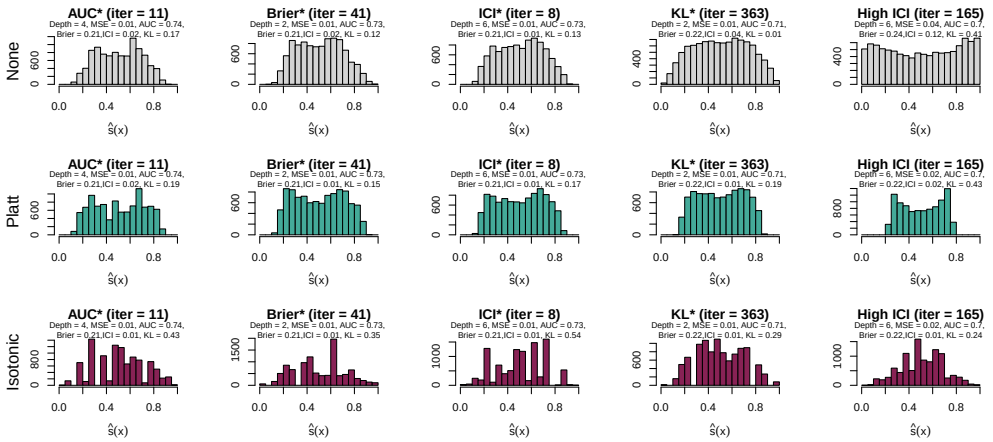


Figure C16: Distribution of estimated scores for XGB: **DGP 4**, **50 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

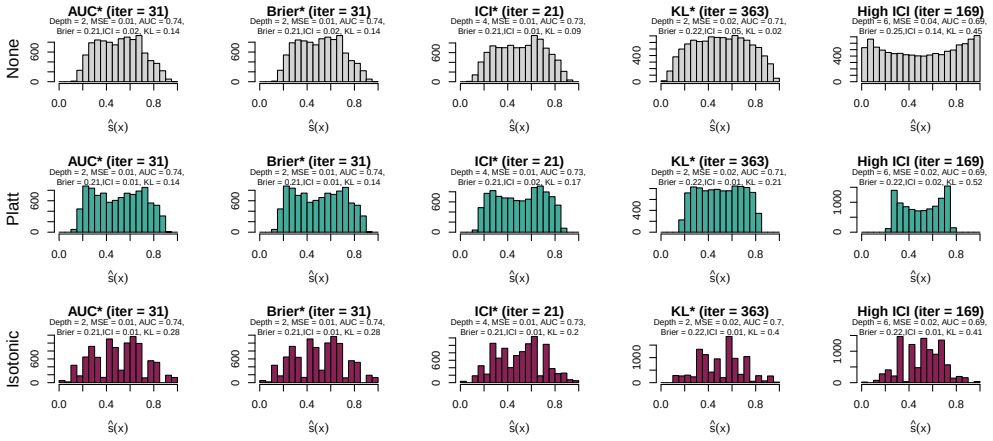


Figure C17: Distribution of estimated scores for XGB: **DGP 4**, **100 noise variables**, single replication.

Notes: AUC*, Brier*, ICI*, and KL*: models selected based on optimizing AUC, Brier score, ICI, and Kullback-Leibler divergence, resp.

Table C3 reports the average values of metrics calculated on the test set over 100 replications for each DGP and each number of noise variables in the training data. The values are presented for models selected by optimizing, on the validation set, either AUC (AUC*) or KL divergence (KL*), as well as for a model with poor calibration (High ICI). The metrics are calculated before applying any calibration method (column "None"), after applying Platt Scaling, and after applying isotonic regression calibration.

Fig. C18 shows the performance of the models, measured by the KL divergence between the test set score distribution (x-axis) and the true probability distribution (y-axis), before and after applying calibration methods. The values represent the average of these two metrics over 100 replications for each DGP (rows), based on the number of noise variables in the training set (columns). The point corresponds to the model whose hyperparameters (number of boosting iterations and tree depth) are selected to maximize AUC on the validation set. The square represents the model selected by minimizing the Kullback-Leibler divergence between the score distribution on the validation set and the true probability distribution. The triangle denotes a model with poor calibration on the test set. Solid green arrows illustrate the change in metrics after Platt scaling calibration, while dashed purple arrows indicate changes after isotonic regression calibration.

Table C3: Comparison of metrics computed on the validation set for models selected based on AUC, KL divergence, or ICI across 100 replications. Standard errors are provided in parentheses.

			None			Platt Scaling			Isotonic		
DGP	Noise	Optim.	BS	ICI	KL	BS	ICI	KL	BS	ICI	KL
1	0	AUC*	0.201 (0.002)	0.012 (0.005)	0.054 (0.038)	0.201 (0.002)	0.017 (0.005)	0.131 (0.029)	0.201 (0.002)	0.012 (0.004)	0.295 (0.099)
		KL*	0.202 (0.002)	0.012 (0.005)	0.021 (0.006)	0.202 (0.002)	0.015 (0.005)	0.107 (0.016)	0.202 (0.002)	0.012 (0.004)	0.307 (0.101)
		High ICI	0.217 (0.003)	0.062 (0.006)	0.171 (0.101)	0.212 (0.002)	0.018 (0.005)	0.205 (0.065)	0.212 (0.002)	0.011 (0.005)	0.356 (0.122)
	10	AUC*	0.201 (0.002)	0.014 (0.005)	0.066 (0.033)	0.201 (0.002)	0.018 (0.005)	0.138 (0.028)	0.201 (0.002)	0.011 (0.004)	0.298 (0.106)
		KL*	0.204 (0.002)	0.015 (0.005)	0.01 (0.004)	0.204 (0.002)	0.016 (0.005)	0.109 (0.015)	0.204 (0.002)	0.012 (0.004)	0.302 (0.11)
		High ICI	0.229 (0.003)	0.106 (0.006)	0.434 (0.179)	0.216 (0.002)	0.025 (0.005)	0.374 (0.032)	0.216 (0.002)	0.012 (0.004)	0.355 (0.117)
	50	AUC*	0.201 (0.002)	0.015 (0.005)	0.075 (0.028)	0.201 (0.002)	0.018 (0.005)	0.139 (0.028)	0.201 (0.002)	0.012 (0.004)	0.307 (0.1)
		KL*	0.205 (0.002)	0.019 (0.005)	0.009 (0.003)	0.205 (0.002)	0.016 (0.004)	0.12 (0.02)	0.205 (0.002)	0.012 (0.004)	0.312 (0.1)
		High ICI	0.235 (0.003)	0.129 (0.006)	0.693 (0.089)	0.216 (0.002)	0.03 (0.005)	0.432 (0.036)	0.215 (0.002)	0.011 (0.004)	0.344 (0.115)
	100	AUC*	0.201 (0.002)	0.016 (0.005)	0.09 (0.031)	0.201 (0.002)	0.018 (0.005)	0.145 (0.026)	0.201 (0.002)	0.012 (0.004)	0.317 (0.111)
		KL*	0.206 (0.002)	0.02 (0.005)	0.009 (0.004)	0.206 (0.002)	0.015 (0.004)	0.125 (0.022)	0.206 (0.002)	0.011 (0.004)	0.304 (0.105)
		High ICI	0.236 (0.003)	0.136 (0.006)	0.798 (0.063)	0.216 (0.002)	0.032 (0.005)	0.442 (0.028)	0.215 (0.002)	0.011 (0.004)	0.342 (0.114)
2	0	AUC*	0.118 (0.002)	0.01 (0.004)	0.031 (0.027)	0.12 (0.002)	0.038 (0.004)	0.778 (0.221)	0.118 (0.002)	0.009 (0.004)	0.213 (0.073)
		KL*	0.12 (0.002)	0.01 (0.004)	0.013 (0.005)	0.121 (0.002)	0.038 (0.004)	0.863 (0.141)	0.12 (0.002)	0.009 (0.004)	0.214 (0.072)
		High ICI	0.131 (0.003)	0.048 (0.004)	0.131 (0.131)	0.13 (0.003)	0.05 (0.005)	0.839 (0.081)	0.127 (0.003)	0.009 (0.003)	0.241 (0.073)
	10	AUC*	0.119 (0.002)	0.012 (0.005)	0.036 (0.067)	0.12 (0.002)	0.038 (0.004)	0.758 (0.226)	0.119 (0.002)	0.009 (0.004)	0.209 (0.069)
		KL*	0.12 (0.002)	0.011 (0.003)	0.007 (0.003)	0.122 (0.002)	0.04 (0.004)	0.887 (0.076)	0.121 (0.002)	0.01 (0.004)	0.205 (0.074)
		High ICI	0.137 (0.003)	0.075 (0.004)	0.294 (0.136)	0.132 (0.003)	0.058 (0.005)	1.235 (0.363)	0.128 (0.002)	0.009 (0.003)	0.263 (0.093)
	50	AUC*	0.119 (0.002)	0.013 (0.004)	0.036 (0.016)	0.12 (0.002)	0.038 (0.004)	0.727 (0.226)	0.119 (0.002)	0.009 (0.004)	0.204 (0.073)
		KL*	0.121 (0.002)	0.011 (0.003)	0.006 (0.003)	0.123 (0.002)	0.041 (0.004)	0.894 (0.045)	0.121 (0.002)	0.009 (0.003)	0.215 (0.068)
		High ICI	0.139 (0.003)	0.089 (0.004)	0.434 (0.14)	0.133 (0.003)	0.064 (0.006)	1.793 (0.165)	0.127 (0.002)	0.009 (0.003)	0.232 (0.077)
	100	AUC*	0.119 (0.002)	0.014 (0.004)	0.041 (0.02)	0.121 (0.002)	0.038 (0.004)	0.727 (0.22)	0.119 (0.002)	0.009 (0.004)	0.212 (0.068)
		KL*	0.122 (0.002)	0.012 (0.004)	0.006 (0.003)	0.124 (0.002)	0.041 (0.003)	0.89 (0.032)	0.122 (0.002)	0.009 (0.003)	0.214 (0.074)
		High ICI	0.14 (0.003)	0.094 (0.004)	0.476 (0.082)	0.133 (0.003)	0.066 (0.004)	1.852 (0.112)	0.127 (0.002)	0.009 (0.004)	0.234 (0.067)
3	0	AUC*	0.22 (0.002)	0.01 (0.004)	0.012 (0.008)	0.221 (0.002)	0.012 (0.004)	0.042 (0.013)	0.221 (0.002)	0.011 (0.004)	0.285 (0.113)
		KL*	0.221 (0.002)	0.012 (0.004)	0.005 (0.002)	0.221 (0.002)	0.011 (0.004)	0.047 (0.014)	0.222 (0.002)	0.011 (0.004)	0.286 (0.115)
		High ICI	0.246 (0.002)	0.105 (0.005)	0.63 (0.086)	0.231 (0.001)	0.014 (0.004)	0.268 (0.047)	0.231 (0.001)	0.011 (0.004)	0.375 (0.106)
	10	AUC*	0.221 (0.001)	0.01 (0.004)	0.027 (0.021)	0.221 (0.002)	0.012 (0.004)	0.046 (0.014)	0.221 (0.002)	0.011 (0.004)	0.287 (0.097)
		KL*	0.222 (0.002)	0.014 (0.005)	0.004 (0.002)	0.222 (0.002)	0.012 (0.004)	0.056 (0.019)	0.222 (0.002)	0.011 (0.004)	0.284 (0.103)
		High ICI	0.253 (0.003)	0.127 (0.005)	0.933 (0.114)	0.232 (0.001)	0.015 (0.004)	0.366 (0.035)	0.232 (0.001)	0.011 (0.004)	0.382 (0.113)
	50	AUC*	0.221 (0.001)	0.013 (0.006)	0.054 (0.032)	0.221 (0.002)	0.012 (0.004)	0.049 (0.015)	0.222 (0.002)	0.011 (0.004)	0.27 (0.122)
		KL*	0.224 (0.002)	0.018 (0.006)	0.004 (0.002)	0.224 (0.002)	0.011 (0.004)	0.075 (0.023)	0.224 (0.002)	0.011 (0.004)	0.284 (0.104)
		High ICI	0.259 (0.003)	0.145 (0.006)	1.286 (0.167)	0.233 (0.001)	0.017 (0.005)	0.403 (0.028)	0.232 (0.001)	0.011 (0.004)	0.417 (0.113)
	100	AUC*	0.222 (0.001)	0.015 (0.006)	0.067 (0.031)	0.221 (0.002)	0.012 (0.004)	0.052 (0.016)	0.222 (0.002)	0.011 (0.004)	0.291 (0.111)
		KL*	0.225 (0.002)	0.019 (0.005)	0.004 (0.002)	0.224 (0.002)	0.011 (0.004)	0.08 (0.021)	0.225 (0.002)	0.011 (0.004)	0.301 (0.122)
		High ICI	0.261 (0.003)	0.152 (0.005)	1.454 (0.181)	0.233 (0.001)	0.017 (0.004)	0.418 (0.036)	0.233 (0.001)	0.011 (0.005)	0.426 (0.111)
4	0	AUC*	0.204 (0.002)	0.011 (0.004)	0.041 (0.021)	0.205 (0.002)	0.017 (0.004)	0.132 (0.02)	0.205 (0.002)	0.011 (0.005)	0.305 (0.095)
		KL*	0.206 (0.002)	0.018 (0.005)	0.011 (0.004)	0.206 (0.002)	0.015 (0.004)	0.115 (0.012)	0.206 (0.002)	0.011 (0.004)	0.288 (0.105)
		High ICI	0.222 (0.003)	0.074 (0.005)	0.176 (0.196)	0.216 (0.002)	0.02 (0.005)	0.243 (0.128)	0.215 (0.002)	0.011 (0.004)	0.359 (0.191)
	10	AUC*	0.206 (0.002)	0.014 (0.005)	0.089 (0.026)	0.206 (0.002)	0.016 (0.005)	0.142 (0.023)	0.206 (0.002)	0.012 (0.005)	0.302 (0.108)
		KL*	0.211 (0.002)	0.028 (0.005)	0.015 (0.005)	0.21 (0.002)	0.015 (0.004)	0.156 (0.02)	0.21 (0.002)	0.012 (0.005)	0.306 (0.089)
		High ICI	0.232 (0.002)	0.106 (0.005)	0.324 (0.279)	0.219 (0.002)	0.021 (0.005)	0.396 (0.125)	0.219 (0.002)	0.011 (0.005)	0.435 (0.206)
	50	AUC*	0.207 (0.002)	0.019 (0.006)	0.127 (0.031)	0.207 (0.002)	0.016 (0.005)	0.145 (0.025)	0.207 (0.002)	0.012 (0.005)	0.292 (0.105)
		KL*	0.215 (0.002)	0.034 (0.006)	0.017 (0.004)	0.213 (0.002)	0.014 (0.004)	0.19 (0.021)	0.214 (0.002)	0.012 (0.004)	0.347 (0.1)
		High ICI	0.238 (0.003)	0.126 (0.006)	0.422 (0.048)	0.221 (0.002)	0.024 (0.005)	0.424 (0.039)	0.22 (0.002)	0.012 (0.005)	0.382 (0.108)
	100	AUC*	0.208 (0.002)	0.021 (0.006)	0.144 (0.034)	0.207 (0.002)	0.015 (0.005)	0.147 (0.024)	0.207 (0.002)	0.012 (0.005)	0.326 (0.104)
		KL*	0.216 (0.002)	0.037 (0.006)	0.017 (0.004)	0.215 (0.002)	0.014 (0.005)	0.202 (0.022)	0.215 (0.002)	0.012 (0.005)	0.368 (0.111)
		High ICI	0.241 (0.003)	0.133 (0.005)	0.486 (0.046)	0.221 (0.002)	0.025 (0.004)	0.471 (0.06)	0.22 (0.002)	0.011 (0.005)	0.399 (0.098)

Notes: AUC*, KL*, High ICI: models selected by optimizing AUC, KL divergence, or by selecting a high ICI.

C.2 Real Data

We train XGBoost models on the 10 datasets presented in Section B.2. Unlike Section C.1, the true probabilities underlying the binary events are not observable. Here, we assume that we have prior knowledge about the probability distribution, which can be considered as expert opinion. To simulate this expert opinion, we assume that the true probabilities follow a Beta distribution. The parameters of this distribution, specific to each dataset, are estimated via MLE using the scores from a GAMSEL model [6].

Using these prior distributions, it is possible to replicate the estimation procedure previously applied to the simulated data. Each dataset is split into two parts: 70% of the observations are used to train an XGBoost model (on a training set comprising 80% of these observations, with hyperparameters selected based on metrics calculated on the remaining 20% validation set), and the remaining 30%

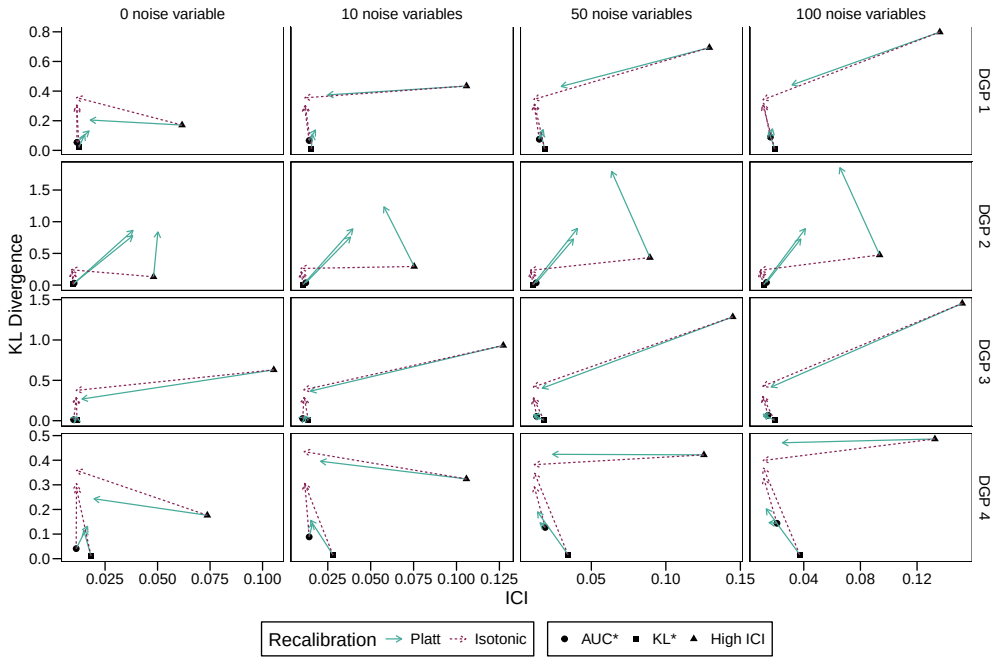


Figure C18: Average KL divergence and ICI before and after recalibration of the estimated scores.

Notes: AUC*, KL*, High ICI: models selected by optimizing AUC, KL divergence, or by selecting a high ICI.

are used for model calibration (with 80% of this subset forming the calibration set) and for testing model performance on unseen data (the remaining 20%).

Table C4 presents the metrics calculated on the test set for models selected based on their validation performance, according to AUC (AUC*), KL divergence between the score distribution and the prior distribution (KL*), or to intentionally obtain poor calibration (High ICI), both before and after applying calibration methods. For real datasets, High ICI refers to the algorithm with hyperparameters yielding the highest AUC among models with an ICI at least one standard deviation above the mean ICI observed during grid search. This table complements Fig. 2 from the main part of the article.

Table C4: Comparison of metrics computed on the validation set for models selected based on AUC, KL divergence, or ICI. Standard errors are provided in parentheses.

		None			Platt			Isotonic		
Dataset	Optim.	BS	ICI	KL	BS	ICI	KL	BS	ICI	KL
abalone	AUC*	0.218	0.081	0.623	0.208	0.051	0.197	0.208	0.047	1.541
	KL*	0.212	0.093	0.319	0.204	0.042	0.163	0.205	0.038	0.887
	High ICI	0.266	0.203	5.861	0.209	0.046	0.689	0.208	0.039	1.174
adult	AUC*	0.086	0.014	0.335	0.088	0.040	0.446	0.086	0.016	0.416
	KL*	0.097	0.045	0.105	0.096	0.035	0.333	0.096	0.022	0.370
	High ICI	0.096	0.056	0.234	0.094	0.037	0.416	0.093	0.017	0.441
bank	AUC*	0.066	0.017	0.499	0.070	0.046	0.674	0.066	0.009	0.535
	KL*	0.077	0.036	0.091	0.079	0.043	0.446	0.075	0.008	0.417
	High ICI	0.078	0.053	1.193	0.078	0.052	0.679	0.069	0.010	0.427
default	AUC*	0.132	0.028	0.309	0.133	0.034	0.756	0.132	0.010	0.490
	KL*	0.133	0.024	0.283	0.133	0.028	0.695	0.132	0.009	0.491
	High ICI	0.151	0.122	1.176	0.135	0.036	1.387	0.134	0.011	0.781
drybean	AUC*	0.034	0.016	0.764	0.037	0.026	0.865	0.033	0.008	0.721
	KL*	0.037	0.028	0.392	0.040	0.031	0.826	0.036	0.008	0.692
	High ICI	0.037	0.054	2.172	0.036	0.027	0.769	0.034	0.008	0.650
coupon	AUC*	0.202	0.144	3.849	0.180	0.023	1.300	0.178	0.026	1.106
	KL*	0.196	0.022	0.052	0.196	0.014	0.191	0.197	0.009	0.478
	High ICI	0.202	0.144	3.849	0.180	0.023	1.300	0.178	0.026	1.106
mushroom	AUC*	0.000	0.003	1.384	0.000	0.002	1.403	0.000	0.002	1.403
	KL*	0.013	0.050	0.620	0.011	0.021	1.222	0.005	0.003	1.337
	High ICI	0.006	0.037	0.928	0.001	0.008	1.348	0.000	0.003	1.403
occupancy	AUC*	0.009	0.008	1.084	0.009	0.008	1.164	0.009	0.009	1.044
	KL*	0.011	0.041	0.862	0.009	0.006	1.128	0.009	0.006	1.032
	High ICI	0.011	0.041	0.862	0.009	0.006	1.128	0.009	0.006	1.032
winequality	AUC*	0.178	0.131	4.244	0.162	0.030	1.697	0.157	0.019	1.253
	KL*	0.173	0.029	0.146	0.174	0.042	0.439	0.171	0.028	1.109
	High ICI	0.172	0.134	4.609	0.157	0.050	1.797	0.151	0.018	1.288
spambase	AUC*	0.044	0.012	0.699	0.045	0.018	1.107	0.045	0.012	1.044
	KL*	0.059	0.045	0.260	0.057	0.012	0.647	0.056	0.009	0.607
	High ICI	0.061	0.047	0.297	0.058	0.012	0.626	0.058	0.008	0.858

Notes: AUC*, KL*, High ICI: models selected by optimizing AUC, KL divergence, or by selecting a high ICI.