Decoupling Decision-Making in Fraud Prevention through Classifier Calibration for Business Logic Action

Emanuele Luzio
Mercado Libre
Montevideo, Uruguay
emanuele.luzio@mercadolibre.com

Moacir Antonelli Ponti[*]
Mercado Livre
Osasco-SP, Brazil

Christian Ramirez Arevalo
Mercado Libre Mexico
Ciudad de Mexico, Mexico
christian.arevalo@mercadolibre.com.mx

Luis Argerich
Mercado Libre
Buenos Aires, Argentina

ABSTRACT

Machine learning models typically focus on specific targets like creating classifiers, often based on known population feature distributions in a business context. However, models calculating individual features adapt over time to improve precision, introducing the concept of decoupling: shifting from point evaluation to data distribution. We use calibration strategies as strategy for decoupling machine learning (ML) classifiers from score-based actions within business logic frameworks. To evaluate these strategies, we perform a comparative analysis using a real-world business scenario and multiple ML models. Our findings highlight the trade-offs and performance implications of the approach, offering valuable insights for practitioners seeking to optimize their decoupling efforts. In particular, the Isotonic and Beta calibration methods stand out for scenarios in which there is shift between training and testing data. [2]

1 Introduction

Over the past decade, Machine Learning (ML) has played an increasingly important role in a wide range of industries, contributing to advancements in areas such as healthcare, finance, manufacturing, and marketing. As businesses continue to capitalize on the power of ML to extract valuable insights from data, integrating ML classifiers into various aspects of their operations has become the norm [Azeem et al., 2022]. Such models are often applied to make decisions, estimate values, or rank/sort observations. In particular, decisions and actions are defined by setting thresholds on the score output of the models [Narasimhan et al., 2014].

Thresholds are often defined upon model deployment by the business analytic and data science teams. However, it often leads to tightly-coupled systems, where ML models and business logic become intertwined.

In fact, machine learning models are commonly built with a specific target in mind, often aiming to create a classifier directly. In a business context, the population distribution of a particular feature is known, and decisions are made based on this information. Assuming that this distribution is static, the criteria remain constant over time until they are modified. However, models that calculate this feature for each individual do change in order to enhance their precision, and this is where the concept of decoupling arises. Essentially, it represents the transition from point evaluation to data distribution.

---

[*]M. Ponti is also with ICMC/Universidade de São Paulo, São Carlos-SP, Brazil

[2]This is the long version of the paper with the same name accepted for presentation at ACM-SAC 2024, Machine Learning and its Application Track.

Decoupling ML classifiers from score-based actions in business logic is essential to creating more modular, maintainable, and adaptable systems. With that, businesses gain the flexibility to independently update, test, and modify their ML models and actions [Sahoo et al., 2021]. This reduces the risk of unintended consequences that may arise from changes in one component affecting the other, ultimately promoting stability and efficiency within the system. Moreover, decoupling facilitates the maintainability of ML models across various applications, resulting in cost savings, reduced development time, and increased consistency.

Let us explore a practical business scenario where a bank utilizes machine learning (ML) models to assess the creditworthiness of mortgage applicants. As the bank continually improves its model with new data, it is crucial to decouple the predefined cutoff threshold, which determines mortgage approval, from the model's performance. This separation ensures that the threshold remains consistent, regardless of model updates, and is driven by factors independent of the model's performance, such as regulatory requirements or risk appetite.

Traditionally, banks rely on a combination of manual evaluation and rule-based systems to determine the eligibility of mortgage applicants [Cooper et al., 2023]. With the advent of ML, financial institutions can leverage large volumes of historical data to build predictive models that accurately assess credit risk. These models can consider numerous factors, such as income, credit score, and employment history, to generate a risk score for each applicant.

As new mortgage data becomes available, the bank's ML model may undergo regular updates to maintain or improve its accuracy. However, the cutoff threshold for mortgage approval should remain decoupled from the model's performance to ensure consistency and compliance with external factors.

Given that poorly calibrated algorithms are potentially harmful for decision-making [Van Calster et al., 2019], and the importance of decoupling for organizations seeking to leverage ML effectively, this study aims to provide an in-depth analysis of the calibration as a tool that enables the separation of ML classifiers and score-based actions in business logic. In particular we are interested in scenarios such as fraud prevention, in which data often has some degree of noise labels [Ponti et al., 2022], as well as imbalance [Wallace and Dahabreh, 2012] and may have significant concept drift between training and testing/productive data [Jesus et al., 2022].

1.1 Objectives and scope

Specifically, we aim to study techniques that delivers accurate posterior probability estimates under different conditions. For that, we use the Bank Account Fraud dataset [Jesus et al., 2022] as described later. An essential benefit of such an approach is its adaptability for post-testing threshold adjustments. This flexibility allows us to achieve strong performance across various metrics even under perturbations of the training set.

We first discuss the challenges and potential drawbacks of tightly-coupled ML classifiers and business logic, highlighting the need for effective decoupling strategies. We then present an overview of some mathematical approaches for decoupling based on calibration namely: Platt scaling [Platt et al., 1999], Isotonic calibration [Zadrozny and Elkan, 2002], Temperature Scaling [Guo et al., 2017] and Beta Calibration [Kull et al., 2017]. To assess the effectiveness of these strategies, we perform a comparative analysis using a real-world business scenario and different ML models, evaluating the trade-offs and performance implications associated with each approach. In particular, we compared CatBoost [Prokhorenkova et al., 2018], LightGBM [Ke et al., 2017] and Multi-layer Perceptron [Linnainmaa, 1976, Werbos, 2005, Mello and Ponti, 2018] in terms of their classification performances with and without calibration.

Finally, we synthesize our findings to provide practical guidance for selecting the most suitable mathematical tools for decoupling ML classifiers and score-based actions in business logic considering: (i) different classifiers, (ii) different calibration techniques, and (iii) different dataset scenarios. By fostering a more modular and adaptable landscape for ML-driven business applications, our study contributes to the ongoing evolution of efficient and innovative systems in the era of artificial intelligence.

2 Decoupling Thresholds from Evolving Models

Decoupling the mortgage approval threshold from the ML model offers several advantages:

- Consistency: By maintaining a fixed threshold, the bank ensures that mortgage approval decisions remain consistent, even as the ML model evolves.

- Compliance: A decoupled threshold allows the bank to adhere to regulatory requirements and industry standards, reducing the risk of non-compliance due to model updates.

- Flexibility: The separation between the threshold and the model allows the bank to adjust the cutoff threshold independently to accommodate changes in external factors, such as market conditions or regulatory guidelines.
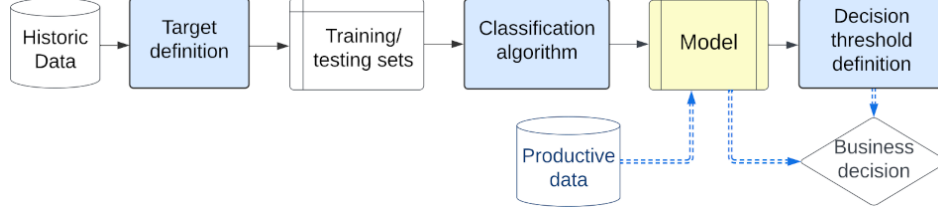
Figure 1: General approach for machine Learning model training and usage. The need
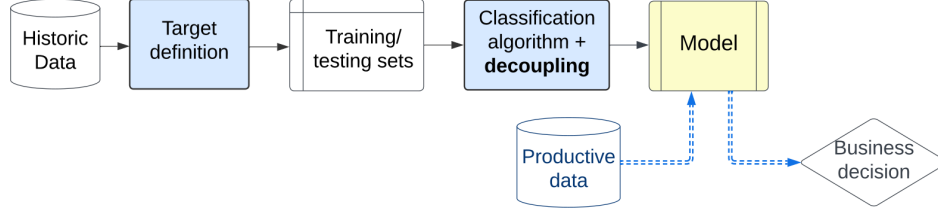

Figure 2: Decoupling business decision by calibration

- Maintainability: Decoupling the threshold from the model simplifies the maintenance of the approval process, as updates to the ML model will not necessitate changes to the predefined threshold.

Practical business scenarios where decoupling ML models from business logic is a good idea are not limited to credit scoring and loan approval. A not comprehensive list can be found in the Appendix A.2.

2.1   Calibration protocol for decoupled thresholds

Models can be seen as functions $f : X \to Y$, where $X \in R^d$ is the feature space in which observations/instances are represented by a d-dimensional vector, and Y represents the space of outputs, or labels in the case of a classification problem [Mello and Ponti, 2018]. Figure 1 shows the general approach to train and use a classification method.

Considering a binary classifier, for a given instance $\mathbf{x}_i$, the output of a model $f(\mathbf{x}_i) = s_i$ can be interpreted as a score for the positive class (fraud in this case). Such score is evaluated in order to determine a threshold for which a decision/action can be applied. This often is step separate from the modeling (see Figure 1). Although the model's output score is often a number in the $[0, 1]$ interval, those seldom represent true likelihoods in particular for imbalanced datasets [Wallace and Dahabreh, 2012]. In fact, classifiers with a large number of parameters, for which the activation function is symmetric and concave on the positive part are often overconfident on their predictions [Bai et al., 2021].

Regarding the threshold definition, it is not easy to predict which threshold value would better suit the decisions. Also, any change in productive data distribution affects the business decision (again, see Figure 1 final step), which may render the model useless.

In this paper we investigate calibration of classifiers for a fraud prevention problem as follows:

1. Train and validate a classifier using available features, optimizing the classification loss function according to the algorithm.
2. Apply the calibration as decoupling techniques (e.g., Platt Scaling or other mathematical transformations) to convert the model's output scores into probability estimates or risk levels.
3. Determine an appropriate cutoff threshold based on business objectives, which are often correlated but may not match exactly the output scores or likelihoods.
4. Use the predefined cutoff threshold to compare against the transformed risk levels or probability estimates for each instance, in our case, each bank account applicant.
5. Make approval decisions based on the comparison between the transformed risk levels and the fixed cutoff threshold.

Our hypothesis is that calibration techniques allow to decouple the approval decision threshold from different machine learning models, ensuring that approval decisions remain consistent and driven by external factors, even in scenarios of concept drift when comparing training vs testing/production data. This way, once the threshold is determined in terms

of a key performance indicator, it should be more robust to perturbations on productive data. We illustrate this concept in Figure 2, where the decision is no longer dependent on a separate step of threshold definition.

2.2 Bank Account Fraud dataset

In particular for this paper we will consider the task of identifying fraudulent online bank account opening requests within a large consumer bank [Jesus et al., 2022]. In this scenario, fraudsters employ tactics such as identity theft or inventing fictitious individuals to gain access to banking services. A positive prediction (indicating fraud) results in the rejection of the customer's bank account application as a punitive measure, whereas a negative prediction results in the approval of a new bank account and credit card as an assistive measure. Each row in the dataset represents an individual application, all of which were submitted through an online platform with the applicant's explicit consent to store and process their data.

The dataset encompasses eight months of data. The prevalence of fraud fluctuates between 0.85% and 1.5% of the instances across different months, with higher values observed in the later months. Furthermore, the distribution of applications also varies from month to month, ranging from 9.5% on the lower end to 15% on the higher end.

- **Group Size Disparity**: indicates differences in group frequencies within a dataset, often due to imbalanced populations or uneven adoption of applications among demographics. In particular groups are related to younger (age < 50) or older (50+) individuals;
- **Prevalence Disparity**: emerges when class probabilities rely on the protected group, enabling dataset creation where label probabilities are tied to these groups. It displays higher fraud rates among older age groups, possibly due to fraudsters targeting them for larger bank credit lines;
- **Separability disparity**: goes beyond prior concepts, considering both input features and labels' joint distribution. For instance, in ATM transactions, illumination and age may interact. If the 20-40 age group often uses poorly lit ATMs, it heightens the risk of card cloning. The illumination feature can detect fraud within this group but not others, illustrating separability disparity's intricacies.

There is one base dataset and 5 dataset variants with additional patterns. Each variant follows the same underlying distribution as the base dataset but has the following characteristics:

- **Variant I**: Group size disparity worsens, reducing the minority group to 10%, without prevalence disparities;
- **Variant II**: Steeper prevalence disparities (one group five times the fraud rate), with equal group sizes;
- **Variant III**: Has better separability (fraud/non-fraud) for the the majority group;
- **Variant IV**: Temporal bias introduced with prevalence disparities for the first six months only;
- **Variant V**: Temporal bias with balanced group size and prevalence, featuring feature distribution shifts over time, akin to fraudsters adapting to evade detection.

Therefore, variants IV and V present not only difference on prevalence/groups sizes and separability, but also drift with respect to training and testing data.

3 Calibration Techniques Overview

We champion calibrating ML models to produce probability estimates that not only represent the true likelihood of an instance belonging to a particular class but also remain consistent during successive retraining, allowing an effective decoupling between machine learning (ML) models and business logic

While a myriad of calibration techniques exist for binary classifiers, this paper takes in consideration four pivotal methods: Platt Scaling, Isotonic Regression, Temperature Scaling and Beta Calibration. Some critics might question the decision to limit our discussion to just two techniques amidst a sea of options. However, our rationale is twofold: simplicity and efficacy. We contend that these four methods, in their essence, are representative enough to illuminate the overarching principles of calibration, making our discourse both concise and impactful.

3.1 Platt Scaling

Platt Scaling [Platt et al., 1999] is a post-processing method that converts binary classifiers raw scores into probability estimates. Specifically, given a raw score s produced by a classifier, Platt Scaling maps this score to a probability using logistic regression:

$$P(y = 1|s) = \frac{1}{1 + \exp(A \cdot s + B)}$$

Where A and B are parameters learned from a validation dataset using maximum likelihood estimation. This approach is especially effective for SVMs and neural networks as they don't naturally output calibrated probability scores.

3.2 Isotonic Regression

Isotonic Regression [Zadrozny and Elkan, 2002] is a non-parametric technique used to map classifier scores to probabilities. Given a set of scores $S = \{s_1, s_2, \ldots, s_n\}$, the idea is to find a set of probabilities $P = \{p_1, p_2, \ldots, p_n\}$ such that p_i is monotonically increasing with s_i, while minimizing the weighted mean squared error between the probabilities and true labels. The optimization problem can be formalized as:

$$L = \min_P \sum_{i=1}^{n} w_i (y_i - p_i)^2$$

Subject to the constraint that $p_1 \le p_2 \le \ldots \le p_n$, where y_i are the true labels and w_i are instance weights.

3.3 Temperature Scaling

Temperature scaling [Guo et al., 2017] is a post-processing calibration technique that can be applied to any model producing raw scores or logits. Given scores z (pre-softmax values or analogous outputs), these scores are scaled by a learned temperature T before converting them into probabilities. For models like deep neural networks which use softmax:

$$P(y = k|z) = \frac{\exp(\frac{z_k}{T})}{\sum_j \exp(\frac{z_j}{T})}$$

Where k denotes the class. For models producing raw scores, a similar scaling can be applied:

$$P(y = 1|z) = \frac{1}{1 + \exp(\frac{-z}{T})}$$

T is optimized on a validation set to minimize the discrepancy, such as negative log likelihood, between the predicted probabilities and true labels. A temperature value of 1 implies no modification to the original scores, while $T > 1$ softens the probability distribution, leading to more calibrated probabilities [Guo et al., 2017].

3.4 Beta Calibration

While Platt scaling and isotonic regression are older and well-known methods for calibration, they have their limitations. Beta calibration [Kull et al., 2017] was introduced as a method to generalize Platt scaling and isotonic regression.

Beta calibration models the predicted probability p using the following transformation:

$$m(p; a, b, c) = \frac{p^a \cdot c}{p^a \cdot c + (1 - p)^b}$$

Here a, b, and c are parameters that are learned from the data. p is the predicted probability from the model. The parameters are learnt by minimizing the negative log likelihood $L(a, b, c)$:

$$L(a, b, c) = -\sum_{i=1}^{N} [y_i \log(m(p_i; a, b, c)) + \\ + (1 - y_i) \log(1 - m(p_i; a, b, c))]$$

We can see that Platt scaling is a special case of beta calibration where $b = 0$ and $c = 1$ [Kull et al., 2017].
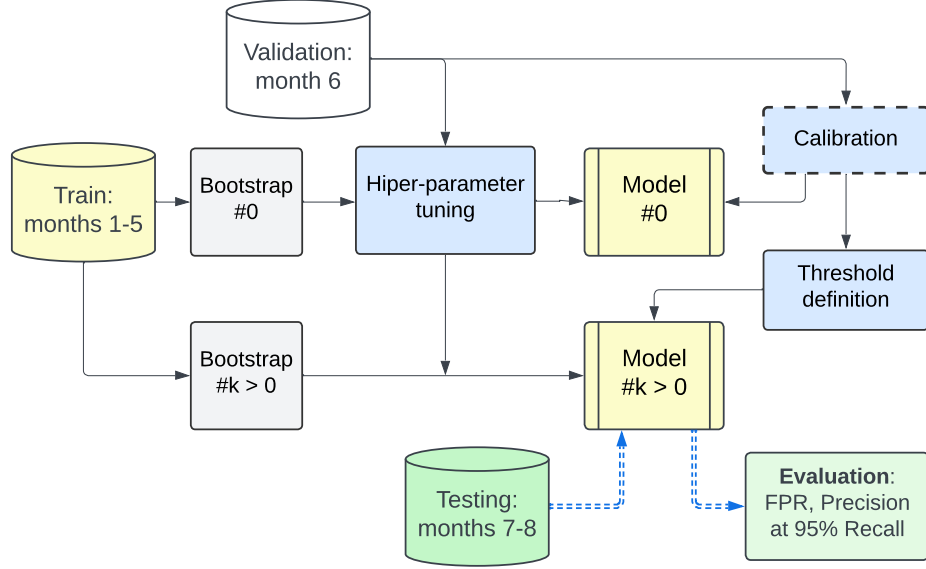
5

Figure 3: Diagram for the Experimental Setup: obtaining bootstraps of the training set to train several models tuned and calibrated using a validation set, and evaluation using a test set.

4   Experimental Design

To provide empirical evidence of decoupling advantages, we orchestrated an experiment using the BAF (Bank Account Fraud) detection dataset [Jesus et al., 2022] detailed in Section 2.2. The experiments follow a protocol encompassing the steps below, and illustrated in Figure 3:

1. **Bootstrapping:** for each iteration, a bootstrapped sample from the fraud detection training dataset is procured;
2. **Training Phase:** models are trained on the training subset drawn from the current bootstrapped sample. The hyperparameters are found using the first bootstrap as training, and applied to the following bootstraps;
3. **Calibration Phase:** after training, models may be calibrated using as reference the validation subset. We compare a non-calibrated version (skipping this step) and the calibration methods: sigmoid, isotonic, beta and temperature scaling;
4. **Threshold Setting:** in the initial bootstrap iteration (# 0), we set a decision threshold aiming for a 95% recall on the validation set. This threshold remains consistent for evaluations in all subsequent bootstrap iterations. Here is where actual decoupling happens.
5. **Evaluation Phase:** the performance of these calibrated models is then gauged on the testing set.

4.1   Statistical test

The Wilcoxon signed rank test is used to compare the result. It is a non-parametric statistical test comparing the distribution of the signed differences between the paired observations to a null hypothesis distribution of randomly signed differences. We compared the uncalibrated results against the calibrated ones, and considered significant the differences for which the test resulted in p-values ≤ 0.01.

4.2   Results

From the two tables presented, i.e. Table 1 and Table 2, it is evident that models that have been decoupled generally exhibit performance similar to and in some cases better than the not calibrated models.

The dataset Base, Variants I and II do not have concept drift between training and test sets. In such cases, process of decoupling does not significantly compromise the efficacy of the models in scenarios. For the Variants III, IV and V the methods: isotonic, beta and temperature scaling achieved superior results. This suggests that the isotonic decoupling might offer additional benefits or optimizations for specific types of data or problem contexts, making it a noteworthy

Model Name	Calibration	Base	Variant_I	Variant_II	Variant_III	Variant_IV	Variant_V
Catboost	not calibrated	13.5 ± 1.0	12.0 ± 1.0	13.3 ± 1.0	12.6 ± 1.0	8.1 ± 0.4	6.4 ± 0.6
	temperature scaling	12.4 ± 0.8	11.8 ± 0.8	12.4 ± 0.6	12.5 ± 0.6	8.1 ± 0.4	6.7 ± 0.4
	sigmoid	12.3 ± 0.8	11.6 ± 0.8	12.3 ± 0.6	12.4 ± 0.6	8.0 ± 0.4	6.6 ± 0.4
	isotonic	**13.0 ± 1.8**	**12.2 ± 1.8**	**13.0 ± 1.8**	**13.0 ± 2.0**	**8.8 ± 1.4**	**7.0 ± 0.8**
	beta	13.0 ± 1.6	11.9 ± 1.0	12.4 ± 0.6	12.6 ± 0.6	8.8 ± 1.2	6.7 ± 0.4
Light GBM	not calibrated	**13.5 ± 1.0**	11.8 ± 0.8	12.9 ± 0.8	12.5 ± 0.8	7.9 ± 0.4	6.1 ± 0.4
	temperature scaling	12.4 ± 0.6	11.7 ± 0.6	12.1 ± 0.6	12.5 ± 0.6	7.9 ± 0.4	6.4 ± 0.2
	sigmoid	12.3 ± 0.6	11.6 ± 0.6	12.0 ± 0.6	12.4 ± 0.6	7.8 ± 0.4	6.3 ± 0.2
	isotonic	13.0 ± 1.4	**12.0 ± 1.4**	**12.7 ± 1.8**	**13.4 ± 1.8**	**8.3 ± 1.2**	**6.7 ± 0.8**
	beta	12.4 ± 0.6	11.7 ± 0.6	12.2 ± 0.6	12.7 ± 0.6	8.0 ± 0.4	6.4 ± 0.2
MLP Neural Network	not calibrated	**15.0 ± 3.0**	11.0 ± 4.0	**14.0 ± 5.0**	11.0 ± 3.0	6.4 ± 1.4	5.2 ± 1.6
	temperature scaling	12.6 ± 1.2	11.2 ± 1.4	13.0 ± 3.0	**12.0 ± 3.0**	7.2 ± 1.2	**6.7 ± 1.8**
	sigmoid	12.1 ± 1.2	10.5 ± 1.4	11.4 ± 1.6	11.2 ± 1.8	6.7 ± 0.8	5.8 ± 0.8
	isotonic	12.8 ± 1.8	10.8 ± 1.8	12.0 ± 2.0	12.0 ± 2.0	**7.3 ± 1.4**	6.5 ± 1.8
	beta	12.4 ± 1.2	10.9 ± 1.4	12.1 ± 1.8	11.7 ± 1.8	7.1 ± 1.0	6.7 ± 1.4

Table 1: *Precision metric at 95% Recall. Comparison of the various models and calibration methods in the different dataset variations. Bold values indicate the higher absolute value for each dataset variant, * indicates statistical significance with p-value ≤ 0.01 when comparing calibration with non calibrated classifiers.*

Model Name	Calibration	Base	Variant_I	Variant_II	Variant_III	Variant_IV	Variant_V
Catboost	not calibrated	51.9 ± 0.4	48.1 ± 0.4	52.4 ± 0.6	69.2 ± 0.4	42.8 ± 0.6	32.3 ± 0.4
	temperature scaling	51.9 ± 0.8	48.1 ± 0.8	52.4 ± 0.8	69.2 ± 0.8	42.8 ± 1.2	32.3 ± 0.8
	sigmoid	51.9 ± 0.8	48.1 ± 0.8	52.4 ± 0.8	69.2 ± 0.8	42.8 ± 1.2	32.3 ± 0.8
	isotonic	**52.4 ± 8.3**	**50.6 ± 4.1**	**55.0 ± 3.8**	**71.7* ± 3.1**	**44.7* ± 3.0**	**35.0* ± 2.9**
	beta	52.0 ± 0.8	48.3 ± 1.2	53.3 ± 2.0	70.0 ± 2.2	44.0* ± 2.2	33.5* ± 1.9
Light GBM	not calibrated	51.5 ± 1.2	47.0 ± 0.4	50.9 ± 0.4	68.3 ± 0.3	41.4 ± 0.6	31.2 ± 0.5
	temperature scaling	51.6 ± 0.8	47.0 ± 0.4	51.0 ± 0.4	68.3 ± 0.3	41.6 ± 0.6	31.1 ± 0.5
	sigmoid	51.6 ± 0.8	47.0 ± 0.4	51.0 ± 0.4	68.3 ± 0.3	41.6 ± 0.6	31.6 ± 0.5
	isotonic	**54.3 ± 4.7**	**48.0 ± 3.1**	**52.5 ± 3.2**	**68.9 ± 9.4**	**43.9* ± 3.8**	**33.0* ± 2.9**
	beta	52.1 ± 1.8	47.0 ± 0.4	51.0 ± 0.4	68.5 ± 0.7	42.0* ± 1.7	32.4* ± 0.9
MLP Neural Network	not calibrated	46.7 ± 1.2	41.6 ± 0.9	46.5 ± 0.8	65.4 ± 0.8	38.4 ± 0.9	29.6 ± 0.9
	temperature scaling	47.3 ± 1.0	41.4 ± 1.0	46.9 ± 1.0	66.0 ± 1.0	38.9 ± 0.8	32.0* ± 1.2
	sigmoid	47.2 ± 1.1	42.4 ± 1.0	46.8 ± 0.8	65.4 ± 0.8	38.7 ± 0.9	30.0 ± 1.2
	isotonic	**49.6 ± 4.1**	**43.7 ± 3.5**	**48.0 ± 7.6**	**67.0 ± 3.0**	**42.7* ± 4.3**	**33.0* ± 3.0**
	beta	47.1 ± 1.2	42.7 ± 0.8	47.0 ± 1.0	66.0 ± 1.8	39.7 ± 2.4	32.0* ± 2.9

Table 2: *True positive rates at 95% Recall. Comparison of the various models and calibration methods in the different dataset variations. Bold values indicate the higher absolute value for each dataset variant, * indicates statistical significance with p-value ≤ 0.01 when comparing calibration with non calibrated classifiers.*

approach for those particular scenarios. This observation underscores the importance of evaluating various calibration techniques against different datasets to discern the most effective method for a given application.

Our primary assumption is unambiguous: we firmly believe that even if we set a threshold specifically to maximize a given metric from the very first iteration, the decoupled models have the capability to not just match, but potentially surpass the performance of models that have not undergone decoupling when it comes to precision or true positive rates on the test datasets. By stating this, we demonstrate a significant point: *it is indeed possible to distinguish and separate the functioning of the model from the underlying business processes and logic without compromising safety or consistency.*

5   Conclusion

In this paper, we addressed the critical question of the separation between machine learning logic and business logic. By meticulously setting up controlled experiments, our findings not only confirm the intuitive benefits of such a separation, but also shed light on some non-trivial advantages.

Our results underscore the importance of probability calibration in maintaining a clear boundary between machine learning logic, and business logic. Such separation fosters improved maintainability by enabling independent modifications and upgrades, thereby ensuring that updates to the business logic do not inadvertently affect the ML model, and vice versa.

Furthermore, scalability greatly benefits from this approach. Since each model can be independently used based on its own skill, it reduces the potential bottleneck situations where we need different logic for different use cases. Stability also witnesses a marked improvement, as any instability or bugs in one domain remain isolated, preventing cascading failures across the system.

Interestingly, while the aforementioned benefits center around system design principles, our experiments also brought to the fore an empirical observation. In some cases, the very act of separating the two logics via probability calibration not only maintained, but in fact, enhanced the model's performance. This indicates that the separation might allow each logic to focus on its core functionality, thereby optimizing its operation.

In conclusion, our research makes a compelling case for the clear demarcation between machine learning logic and business logic using probability calibration. Not only does this methodology pave the way for more robust, maintainable, and scalable systems, but it also has the potential to yield performance dividends. As the realms of machine learning and business operations continue to converge, it is our belief that such an architectural separation will become a gold standard in system design.

Acknowledgments

References

Mohd Azeem, Abid Haleem, and Mohd Javaid. Symbiotic relationship between machine learning and industry 4.0: A review. *Journal of Industrial Integration and Management*, 7(03):401–433, 2022.

Harikrishna Narasimhan, Rohit Vaish, and Shivani Agarwal. On the statistical consistency of plug-in classifiers for non-decomposable performance measures. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

Roshni Sahoo, Shengjia Zhao, Alyssa Chen, and Stefano Ermon. Reliable decisions with threshold calibration. *Advances in Neural Information Processing Systems*, 34:1831–1844, 2021.

A. Feder Cooper, Katherine Lee, Madiha Zahrah Choksi, Solon Barocas, Christopher De Sa, James Grimmelmann, Jon Kleinberg, Siddhartha Sen, and Baobao Zhang. Is my prediction arbitrary? the confounding effects of variance in fair classification benchmarks, 2023.

Ben Van Calster, David J McLernon, Maarten Van Smeden, Laure Wynants, and Ewout W Steyerberg. Calibration: the achilles heel of predictive analytics. *BMC medicine*, 17(1):1–7, 2019.

Moacir A. Ponti, Lucas de A. Oliveira, Juan M. Román, and Luis Argerich. Improving data quality with training dynamics of gradient boosting decision trees. *arXiv preprint arXiv:2210.11327*, 2022.

Byron C Wallace and Issa J Dahabreh. Class probability estimates are unreliable for imbalanced data (and how to fix them). In *2012 IEEE 12th international conference on data mining*, pages 695–704. IEEE, 2012.

Sérgio Jesus, José Pombal, Duarte Alves, André Cruz, Pedro Saleiro, Rita Ribeiro, João Gama, and Pedro Bizarro. Turning the tables: Biased, imbalanced, dynamic tabular datasets for ml evaluation. *Advances in Neural Information Processing Systems*, 35:33563–33575, 2022.

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

Bianca Zadrozny and Charles Elkan. 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, 2002.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

Meelis Kull, Telmo de Menezes e Silva Filho, and Peter A. Flach. Beta calibration: a well-founded and easily implemented improvement on logistic calibration for binary classifiers. In *International Conference on Artificial Intelligence and Statistics*, 2017. URL https://api.semanticscholar.org/CorpusID:217264968.

Liudmila Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Seppo Linnainmaa. Taylor expansion of the accumulated rounding error. *BIT Numerical Mathematics*, 16(2):146–160, 1976.

Paul J Werbos. Applications of advances in nonlinear sensitivity analysis. In *System Modeling and Optimization: Proceedings of the 10th IFIP Conference New York City, USA, August 31–September 4, 1981*, pages 762–770. Springer, 2005.

Rodrigo F. Mello and Moacir A. Ponti. *Machine Learning: A Practical Approach on the Statistical Learning Theory*. Springer, Cham, 2018.

Yu Bai, Song Mei, Huan Wang, and Caiming Xiong. Don't just blame over-parametrization for over-confidence: Theoretical analysis of calibration in binary classification. In *International Conference on Machine Learning*, pages 566–576. PMLR, 2021.

A   Appendix

A.1   Data

As dataset for out experiment we used Bank Account Fraud Dataset Suite (NeurIPS 2022), introduced at NeurIPS 2022. It encompasses a series of six distinct synthetic tabular datasets oriented towards bank account fraud detection. Unveiled as a pioneering endeavor in its realm, this suite seeks to provide a comprehensive and robust platform for the assessment of both emergent and established machine learning (ML) techniques, including those focusing on fairness in ML.

Characteristics of the BAF dataset suite are delineated as follows:

1. Realism: Derived from contemporary real-world data pertinent to fraud detection, the BAF datasets emulate the intricacies and nuances inherent in genuine transactional records.

2. Controlled Bias: Each dataset within the suite is specifically crafted to exhibit unique, predefined types of bias. This is vital for testing the resilience and adaptability of ML algorithms under varying biased conditions.

3. Class Imbalance: Mirroring the rarity of fraudulent activities in actual transactional contexts, the datasets are marked by an exceptionally low prevalence of the positive (fraudulent) class. This characteristic introduces the challenge of distinguishing rare events from a sea of legitimate transactions.

4. Temporal Dynamics: The datasets are not only chronologically structured but also exhibit observed distribution shifts over time. This facet ensures that algorithms are tested for their ability to adapt and remain efficacious in evolving scenarios.

5. Privacy Considerations: Given the paramount importance of individual privacy, especially in financial contexts, a multipronged approach has been adopted to safeguard the identity of potential account holders. This entails the integration of differential privacy measures (via noise infusion), feature encryption, and the deployment of a generative model known as CTGAN. These strategies collectively obfuscate sensitive details while preserving the overarching data structure and relationships.

The BAF dataset suite stands as a meticulously designed tool for the scientific community, bridging the gap between theoretical algorithmic advancements and their practical applications in a sensitive domain like fraud detection.

A.2   Practical Examples of Decoupling Decision Making from Model Outputs

- Fraud Detection in Financial Transactions: In the financial sector, ML models are extensively used to identify potentially fraudulent activities and suspicious transactions. Decoupling the fraud detection model from

predefined intervention thresholds is pivotal. This ensures that decisions related to flagging and investigating potential frauds remain consistent and are driven by factors independent of the model's performance, such as regulatory guidelines, risk appetites of the institution, and historical fraud patterns.

- Healthcare Patient Risk Stratification: In healthcare, ML models can help predict patient outcomes, such as the likelihood of readmission or complications after surgery. Decoupling the patient risk stratification model from the predefined risk thresholds allows healthcare providers to make consistent treatment decisions based on factors independent of the model's performance, such as clinical guidelines, resource allocation, and patient preferences.

- Customer Churn Prediction and Retention Strategy: ML models can help businesses predict which customers are likely to churn or cancel their services. Decoupling the churn prediction model from predefined intervention thresholds is essential, as it ensures that customer retention strategies remain consistent and are driven by factors independent of the model's performance, such as business objectives, budget constraints, and customer segmentation.

- Anomaly Detection in Industrial Processes: In manufacturing and industrial settings, ML models can be employed to monitor equipment, production lines, or processes to identify anomalies that may indicate potential issues, such as equipment failure, quality issues, or safety hazards. Decoupling the anomaly detection model from predefined intervention thresholds is critical, as it ensures that maintenance and safety decisions remain consistent and are driven by factors independent of the model's performance, such as company policies, safety regulations, and resource availability.

- Marketing Campaign Optimization: In the marketing domain, ML models can help predict customer responses to different marketing campaigns, enabling businesses to optimize their marketing efforts and maximize return on investment. Decoupling the marketing campaign optimization model from predefined decision thresholds is essential, as it ensures that marketing decisions remain consistent and are driven by factors independent of the model's performance, such as business objectives, budget constraints, and competitive landscape.

These examples highlight the importance of decoupling techniques in real-world applications, demonstrating their value in creating efficient and adaptable ML-driven systems.