WHITE PAPER

Detecting churn before it happens

Why a good model alone is not a retention system – heterogeneous ensembles, calibrated probabilities and the economically correct decision threshold

findic

Contents

About this white paper 3
1 · Why churn models fail in production 3
The majority class always wins 3
A 0.7 is rarely a 0.7 3
The threshold is a business economics question 4
2 · The four-layer architecture 4
3 · Why three boosting machines are better than one 5
4 · Calibration: turning scores into probabilities 6
5 · The threshold as an economic decision 7
6 · Measured results 8
7 · What the system cannot do 9
8 · The next step 9
9 · Sources and further reading 10
10 · Contact 11

About this white paper

In many churn projects, the same decision comes up sooner or later: above what predicted probability should a customer be contacted proactively? The default threshold of 0.5 is often used for this. For a retention scenario driven by business economics, however, this threshold is usually unsuitable, because it is not derived from the actual costs of the wrong decisions.

A simple example illustrates the problem: an unnecessary retention action – a call, a discount or a pause offer, for example – costs about €50 on average. A missed churner, by contrast, can cost many times that, depending on the contract value. If the false-negative error is significantly more expensive than the false-positive, a blanket threshold of 0.5 makes no sense. In our test, the cost-optimised threshold was 0.27. Compared with the default threshold, this reduced the realised misclassification costs by 40.5 per cent while identifying 73 additional churners, that is, almost half as many missed cases.

This white paper describes a churn pipeline that systematically accounts for this threshold decision. It also addresses two points that are frequently underestimated in production: the calibration of the predicted probabilities and the question of whether a single model is sufficient for stable decisions. The aim is not a complete implementation recipe, but a transparent account of the modelling and evaluation decisions taken.[1]

1 · Why churn models fail in production

When churn projects disappoint after go-live, the algorithm is rarely to blame. The cause lies in three places that look harmless in development and become expensive in production.

The majority class always wins

With a churn rate of 23 per cent, a model that simply flags no one as a churner achieves 77 per cent accuracy – and zero value. Every standard procedure without a countermeasure drifts in exactly this direction: the majority class dominates the loss function, and the cases that actually matter become statistical noise. What is more: on imbalanced data, the widely cited ROC-AUC is systematically flattering.[2]

Our answer is two-stage. First, all base models train with class weights derived from the actual ratio in the training data – in our case roughly 3.3 to 1 in favour of retained customers; none of this is assumed across the board. Second, the hyperparameter search is consistently optimised for average precision, a metric that actually takes the minority class into account.

A 0.7 is rarely a 0.7

Gradient boosting models can often rank customers very well by risk. Whether a predicted probability of 0.7 actually corresponds to an observed churn rate of around 70 per cent is, however, not a given.

---

[1] The pipeline was developed and evaluated on a B2B SaaS dataset of 8,214 customers with a churn rate of 23.4 per cent (stratified split: 60 per cent training, 20 per cent calibration, 20 per cent test).

[2] Saito, T. & Rehmsmeier, M. (2015): The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLOS ONE 10(3).

Tree-based ensembles in particular often deliver well-ordered scores whose absolute probabilities can be distorted without calibration.[3]

For operational use, this difference is decisive. A ranking helps with prioritisation; a reliable probability is needed when budgets, response times or escalation levels are tied to it. Section 5 shows the deviation of the raw scores (expected calibration error 0.081) and the result after calibration (0.027).

The threshold is a business economics question

To classify is to decide, and every decision has a price. A cost matrix makes this explicit: the false-negative error (a missed churner) costs €500, the false-positive (an unnecessary action) €50. For calibrated probabilities, the optimal threshold can be computed directly: t* = C_FP / (C_FP + C_FN) – theoretically 0.09 under our assumptions.[4]

The fact that our measured threshold, at 0.27, lies well above this theoretical value has a concrete reason, which Section 6 explains. One thing can be stated already here: anyone who leaves the threshold at 0.5 has implicitly decided that both error types are equally expensive. In the retention business, that is practically never true.

2 · The four-layer architecture

The pipeline consists of four layers. Each layer takes on a clearly delineated task and builds on the results of the previous layer.

Layer 1 – Features. Eleven raw fields (contract duration, support tickets, login behaviour, NPS, satisfaction scores) are turned into 57 features: ratios instead of absolute values, RFM-inspired proxies, interaction terms and a rule-based risk score that explicitly bundles known danger patterns. Every statistic, whether median, quantiles or maxima, is estimated exclusively on the training data and then frozen. This prevents information from validation or test data flowing into feature engineering.

Layer 2 – Models. Three gradient boosting methods, CatBoost, LightGBM and XGBoost, are trained separately and combined via out-of-fold stacking. Each customer receives their meta-features from models that did not see them during training. A logistic regression learns the final combination from these. The hyperparameters are determined via Bayesian optimisation with early termination of unpromising trials; the number of trees is limited by early stopping on a dedicated validation split.[5]

Layer 3 – Calibration. Calibration is performed in two stages: first for the outputs of the base models, then for the outputs of the meta-learner, each on its own calibration set. Isotonic regression is used, a non-parametric, monotonic method without a prescribed curve shape. Because this method requires a sufficient number of calibration cases, the pipeline includes a safeguard: below 1,000 calibration cases, it automatically switches to the more robust Platt scaling.

---

[3] Niculescu-Mizil, A. & Caruana, R. (2005): Predicting Good Probabilities with Supervised Learning. Proceedings of ICML 2005.

[4] Elkan, C. (2001): The Foundations of Cost-Sensitive Learning. Proceedings of IJCAI 2001.

[5] CatBoost: Prokhorenkova et al. (2018), NeurIPS 31. LightGBM: Ke et al. (2017), NeurIPS 30. XGBoost: Chen & Guestrin (2016), KDD. Stacking: Wolpert (1992), Neural Networks 5(2). Hyperparameter search: Optuna, Akiba et al. (2019), KDD.

Layer 4 – Decision. The calibrated probability is linked to a cost matrix. The threshold is chosen so as to minimise the expected euro costs on the calibration set. Building on this, three risk tiers with fixed response times are defined. The result is not an isolated model metric, but a prioritised work list for operational use.

3 · Why three boosting machines are better than one

CatBoost, LightGBM and XGBoost belong to the same model family. On tabular data, this family still beats considerably more elaborate deep learning approaches.[6]

The three models differ, however, in their error structure. CatBoost encodes categorical features leakage-free on the basis of the target variable, but represents rare categories less accurately. LightGBM grows trees leaf-wise and can thus capture deeper structures, but is more prone to overfitting. XGBoost applies stronger regularisation. This produces partly different misclassifications. An ensemble can exploit these differences, provided the errors of the individual models are not fully correlated.

The meta-learner's coefficients show a stronger weighting of CatBoost (2.31) and LightGBM (2.04) relative to XGBoost (1.18). This suggests that the outputs of the first two models make a larger contribution in the final combination. The measured PR-AUC is 0.817, compared with 0.772 for the best individual model and 0.793 for soft voting.

The gain from stacking is measurable but limited. PR-AUC rises by 0.045 points over the best individual model and by 0.024 points over soft voting. The practical benefit lies above all in the fact that the downstream steps (calibration and threshold logic) rest on a more stable model basis.

[6] Grinsztajn, L., Oyallon, E. & Varoquaux, G. (2022): Why do tree-based models still outperform deep learning on typical tabular data? NeurIPS 35.

4 · Calibration: turning scores into probabilities

The uncalibrated meta-learner of our pipeline recorded an expected calibration error of 0.081 on the test set.[7] After two-stage isotonic calibration, the value was 0.027. In the corresponding probability ranges, the observed churn rate thus moves closer to the predicted probabilities.

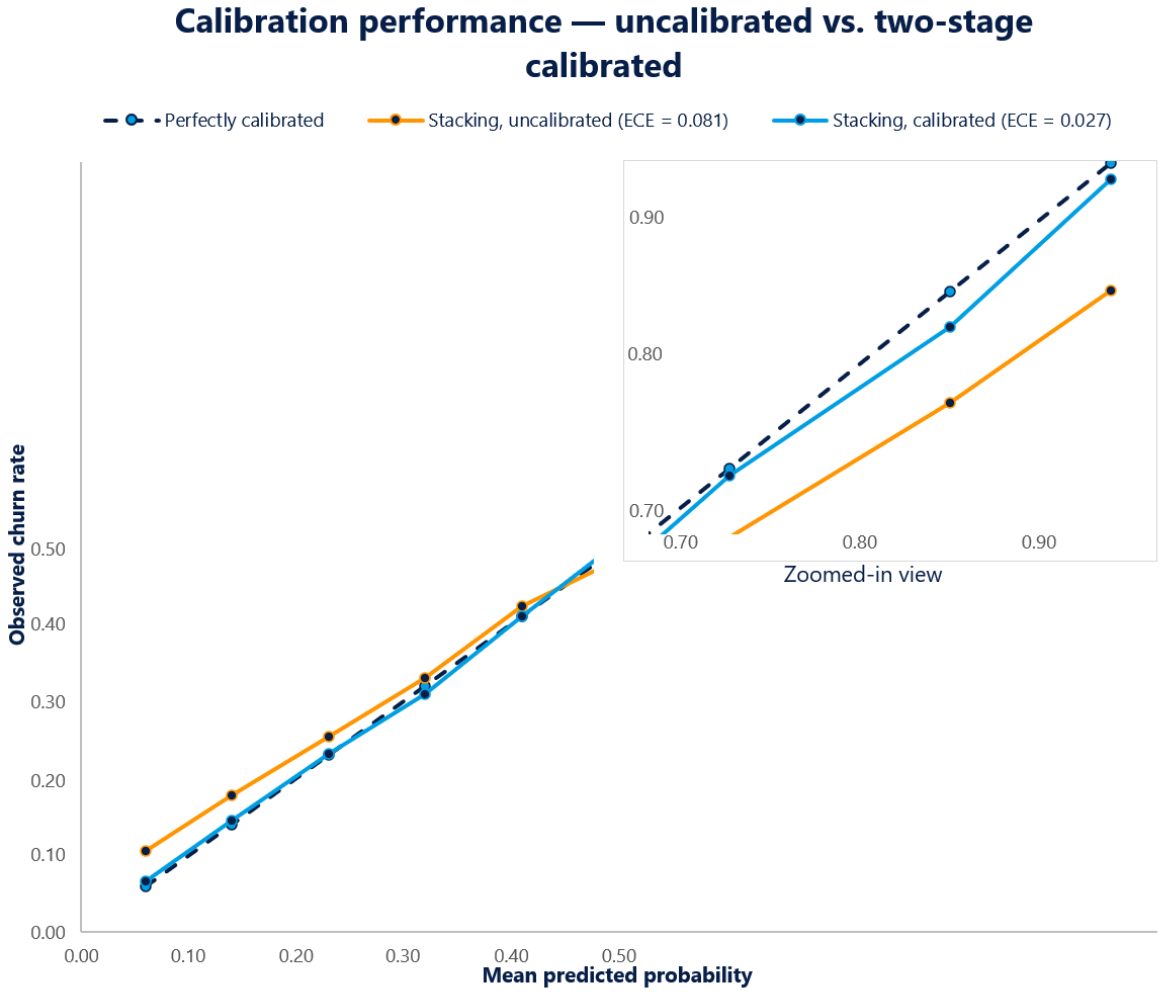

Figure 1: Reliability diagram on the test data set. The raw ensemble output (orange) deviates systematically from the diagonal; after calibration (turquoise), the points lie on the ideal line.

The two-stage calibration addresses different sources of distortion. The first stage corrects the outputs of the base models before combination. The second stage corrects possible distortions of the meta-learner, for example from class weighting. To limit overfitting, calibration is safeguarded across separate data partitions and the ECE is reported on the independent test data set.

Without calibration, the risk tiers from Section 6 would be only partially reliable. Calibration makes the model values more interpretable as probabilities, so they can be used for budgeting and prioritisation decisions.

[7] Naeini, M. P., Cooper, G. F. & Hauskrecht, M. (2015): Obtaining Well Calibrated Probabilities Using Bayesian Binning. Proceedings of AAAI 2015.

5 · The threshold as an economic decision

Elkan's formula t* = C_FP / (C_FP + C_FN) yields the theoretical value 0.09 at €50 and €500. In our case, the empirically determined threshold is 0.27. This deviation matters: the theoretical rule assumes, among other things, very well-calibrated probabilities and a directly applicable cost assumption. In practice, calibration errors, sampling noise and the actual effect of retention measures can push the empirical cost minimum higher. The threshold is therefore determined on the calibration set from the realised cost curve:

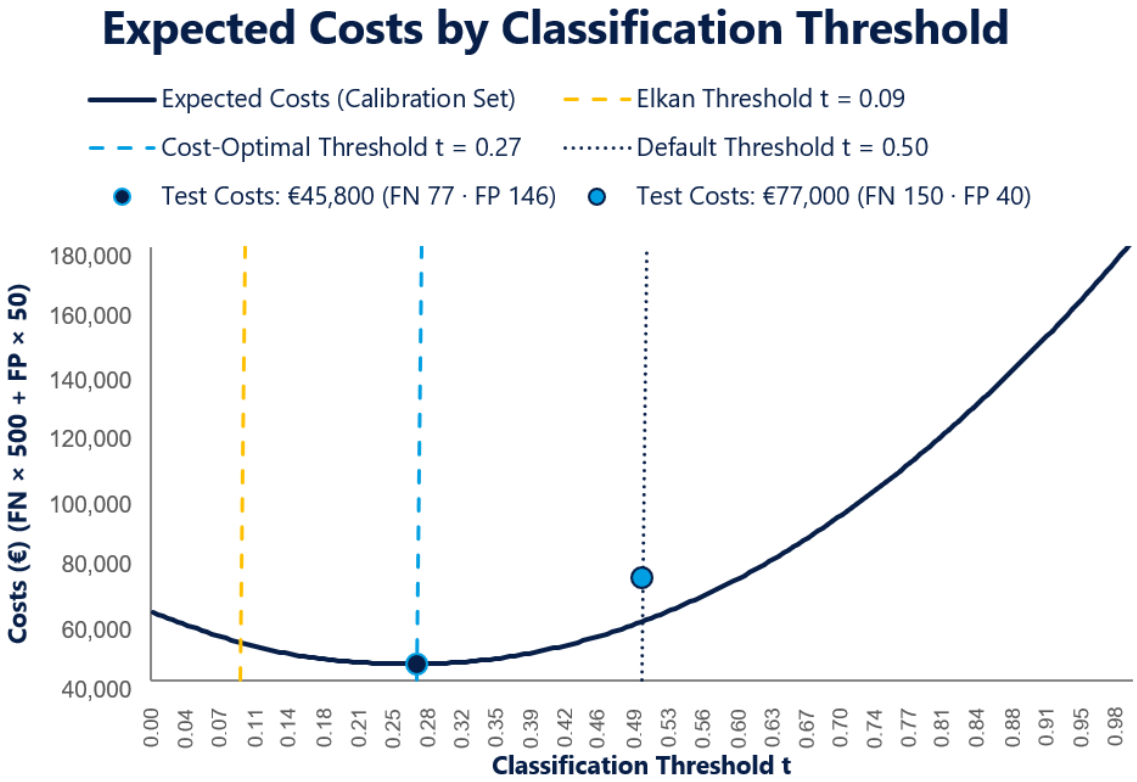

Figure 2: Expected costs over the classification threshold (calibration set). The empirical minimum lies at t = 0.27 – well above the theoretical Elkan threshold, far below the 0.5 default.

A comparison of the common strategies on the test set:

Strategy	Threshold	Missed churners (FN)	False alarms (FP)	Total costs
Standard (default)	0.50	150	40	€77,000
Youden-optimised	0.34	86	118	€48,900
F2-optimised	0.31	82	128	€47,400
Cost-optimised	0.27	77	146	€45,800

Table 1: Threshold strategies compared (test data set; cost assumptions FN = €500, FP = €50).

The last row shows the central effect: the cost-optimised threshold produces the most false alarms of all strategies (146), but also leaves the fewest churners to their own devices: only 77 go undetected, five fewer than under the F2 variant and 73 fewer than under the default threshold. This is consistent given the chosen cost assumptions: at €500, one missed churner offsets exactly ten unnecessary measures at €50 each. Compared with the default threshold, costs fall by 40.5 per cent. The effect does not arise from a different model, but from a decision layer aligned with the costs.

The operational implementation translates the calibrated probabilities into three tiers with fixed response times:

Risk tier	Probability	Response time	Measure
Low Risk	below 0.27	Asynchronous	In-app notice, information email
Medium Risk	0.27 to 0.70	3 days	Contact by Customer Success, usage coaching
High Risk	above 0.70	24 hours	Personal contact, individual offer

Table 2: Risk stratification and measure assignment.

6 · Measured results

All values come from the independent test data set (20 per cent of the 8,214 customers, stratified), which was never seen by training, calibration or threshold selection. The intervals in brackets are 95 per cent confidence intervals from 1,000 bootstrap draws.

Metric	Stacking (calibrated)	Soft voting	Best individual model (CatBoost)
ROC-AUC	0.913 [0.891–0.932]	0.905	0.894
PR-AUC	0.817 [0.774–0.856]	0.793	0.772
Brier score	0.104 [0.093–0.116]	0.121	0.138
ECE	0.027	0.052	0.096
MCC	0.647	0.601	0.562

Table 3: Test results. Confidence intervals where the bootstrap scales meaningfully.

The following points can be derived from the results. First, stacking achieves the best values across all metrics considered, though the margin over soft voting remains moderate. Second, calibration is a material quality factor: the ECE falls from 0.096 for the best individual model to 0.027, and the Brier score

improves markedly. Third, the greatest practical benefit appears in the cost-based decision layer: misclassification costs fall by 40.5 per cent because the threshold is aligned with the economic error costs.

7 · What the system cannot do

Three limitations matter for putting the results in context.

It is predictive, not causal. A high churn probability says nothing about whether a measure will retain this customer. The cancellation may long since have been decided; the customer might have stayed even without a call. Anyone wanting to optimise the effect of interventions needs uplift models or controlled experiments. Strictly speaking, our cost logic holds only if the measures work at all among the customers contacted.

It is bound to its data. RFM proxies, contract flags and satisfaction indices presuppose that such variables are collected. Transferring the approach to other industries requires adapted feature definitions, meaning specialist work rather than a retraining of the principle.

It ages. Customer behaviour changes: prices, products and market conditions do not stay constant. Monitoring and scheduled retraining must therefore be part of operations. Each model version documents its metrics and decision parameters in a machine-readable model card, so that changes between versions can be checked.

8 · The next step

A pilot operation usually requires an export of the relevant customer data – contracts, usage and support interactions, for example – a period of six to eight weeks and a specialist contact in Customer Success. At the end, there is a prioritised list of customers for whom making contact appears economically sensible, including risk tier and rationale.

What matters is not only how high the current churn rate is, but how early at-risk customers can be reliably identified and sensibly prioritised.

9 · Sources and further reading

Akiba, T., Sano, S., Yanase, T., Ohta, T. & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2623–2631. Framework for Bayesian hyperparameter search with automatic termination of unpromising trials.

Chen, T. & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. Scalable gradient boosting with strong regularisation.

Elkan, C. (2001). The Foundations of Cost-Sensitive Learning. Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI), 973–978. Foundational derivation of cost-sensitive decision rules – the optimal classification threshold follows directly from the cost ratio of the error types.

Fridrich, M. (2018). Cost-Benefit Metrics in Customer Churn Prediction: A Review. MMK 2018 – International Masaryk Conference for Ph.D. Students and Young Researchers, 178–185. Overview of business-oriented evaluation metrics and the limits of purely technical classification metrics in the retention context.

Grinsztajn, L., Oyallon, E. & Varoquaux, G. (2022). Why Do Tree-Based Models Still Outperform Deep Learning on Tabular Data? Advances in Neural Information Processing Systems 35 (NeurIPS). Extensive benchmark showing that tree-based methods remain superior to considerably more elaborate deep learning approaches on typical tabular data.

Imani, M., Joudaki, M., Beikmohammadi, A. & Arabnia, H. R. (2025). Customer Churn Prediction: A Systematic Review of Recent Advances, Trends, and Challenges in Machine Learning and Deep Learning. Machine Learning and Knowledge Extraction, 7(3), 105. Systematic overview of current methods, challenges and application areas of churn prediction.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. & Liu, T.-Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Advances in Neural Information Processing Systems 30 (NeurIPS). Histogram-based, leaf-wise gradient boosting with high training speed.

Naeini, M. P., Cooper, G. F. & Hauskrecht, M. (2015). Obtaining Well Calibrated Probabilities Using Bayesian Binning. Proceedings of the AAAI Conference on Artificial Intelligence, 29(1), 2901–2907. Introduces the expected calibration error (ECE) as a measure of the deviation between predicted and observed event rates.

Niculescu-Mizil, A. & Caruana, R. (2005). Predicting Good Probabilities with Supervised Learning. Proceedings of the 22nd International Conference on Machine Learning (ICML), 625–632. Shows that boosted trees rank excellently but their raw probabilities are systematically distorted – and that Platt scaling and isotonic regression turn them into reliable probabilities.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V. & Gulin, A. (2018). CatBoost: Unbiased Boosting with Categorical Features. Advances in Neural Information Processing Systems 31 (NeurIPS). Gradient boosting with unbiased encoding of categorical features without target-knowledge leakage.

Saito, T. & Rehmsmeier, M. (2015). The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PLOS ONE, 10(3), e0118432. Demonstrates that ROC curves give an overly optimistic picture on imbalanced data and that precision-recall curves permit the more realistic evaluation.

Wolpert, D. H. (1992). Stacked Generalization. Neural Networks, 5(2), 241–259. Foundational text on stacking: combining several models via a meta-learner on out-of-fold predictions.

10 · Contact

Udo Jacobasch
Partner
Udo.Jacobasch@findic.de
Office Berlin

Dr. Josef Dirnberger
Manager
Josef.Dirnberger@findic.de
Office München

Hamburg

Kurze Mühren 20,

D-20095 Hamburg

Phone +49 40/303740-0

E-Mail info@findic.de

Zurich

Gutenbergstrasse 1,

CH-8002 Zurich

Phone +41 44/56098-00

E-Mail info@findic.ch