Proceedings of the 8th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/2026.36208

Model Choice or Decision Threshold? A Decision-Aware Evaluation Protocol for Employee Attrition Prediction

Zheng Jin

School of Public Administration, Sichuan Agricultural University, Ya'an, China
zjin26390@gmail.com

Abstract. Research on employee attrition prediction has grown quickly, yet most studies leave two questions open: whether the metric difference used to rank models is statistically reliable on the data concerned, and whether, once a business cost is attached to the retention decision, it is the choice of model or of decision threshold that determines what an organisation pays. This study proposes a four-step, decision-aware evaluation protocol addressing both and applies it to four datasets spanning 295 to 14,999 records, three public and the fourth from a real organisation. On the two smaller benchmarks the models cannot be separated, and over ten independent splits the leading model on the IBM set changes hands four times, so any single split's winner is an accident of partitioning. Where models are tied only the threshold reduces cost, and its optimum lies well below the conventional 0.5; where they are separated by a wide margin the model determines cost instead. The real dataset bounds that rule: there the models are separable, yet the cost parameters place the optimal policy so close to intervening on everyone that no model can improve on it. A sensitivity analysis over the cost parameters leaves these conclusions unchanged. The contribution is a reusable protocol rather than a new algorithm, with evidence that the dominant cost lever depends on the data and on whether the cost-optimal policy discriminates at all.

Keywords: employee attrition, cost-sensitive learning, model selection, decision threshold, evaluation protocol

1. Introduction

Gallup estimates that employee replacement costs range from half to twice an employee's annual salary, and that over half of voluntary turnover is preventable, making attrition a costly and actionable problem for firms [1]. This has motivated a substantial body of machine-learning research on predicting which employees will leave [2].

This body of work shares two evaluation habits that deserve scrutiny. The first concerns how broadly results are tested and how models are then ranked. Existing studies report highly inconsistent model accuracy scores on the same IBM benchmark, with results ranging from approximately 85 to over 99 per cent under different preprocessing and data-splitting strategies [3, 4].

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.36208

The second habit concerns the target of evaluation. Accuracy, AUC and F1 measure prediction, not the decision the organisation must take, which is whom to approach with a costly retention measure. The two errors are not symmetric: missing an employee who then leaves forfeits a replacement cost that grows with that employee's value, whereas approaching an employee who would have stayed wastes the cost of an intervention. A model that looks strong under a cost-blind metric may therefore be weak once these asymmetric costs are introduced, and the converse can hold. Remedies are well established in the cost-sensitive and prescriptive literature (Section 2.3), yet most attrition studies stop at prediction.

This study brings these two problems together through an explicit cost. The research question is whether, once a cost is attached to the retention decision, it is the choice of model or of decision threshold that governs what an organisation pays, and how this depends on the data. The objective is a reusable, decision-aware evaluation procedure that answers this question on a given benchmark and applies across datasets of contrasting character. The procedure introduces no new learning algorithm; its contribution lies in organising established tools into a decision-first workflow, and in the evidence that workflow produces, including a direct measurement of how far the apparent best model on a benchmark is an artefact of the split it was chosen on.

2. Related work

2.1. Employee attrition prediction methods

Supervised learning dominates employee-turnover research. A review of fifty-two studies found supervised methods in almost all of them, random forests the most common and salary and overtime among the strongest predictors [2]; a review of fifty-eight reaches the same picture [5]. Which model wins is unsettled and shifts with the data: extreme gradient boosting led on a large graduate sample [6], while the surveys favour random forests. Some work adds clustering or augmentation to tailor retention to segments [7], or richer encodings such as multi-grained tabular embeddings [8]. Because leavers are a minority, oversampling by SMOTE is usual and raises reported accuracy on IBM [3], but it distorts the predicted-probability scale, which matters once probabilities meet costs. This study therefore uses no resampling or reweighting, keeping each model's native, better-calibrated probabilities under a transparent one-hot encoding.

2.2. Limitations in current evaluation paradigms

Two weaknesses in how these models are evaluated motivate this study. First, models are usually validated on a single dataset, so their behaviour on other data is rarely established [3]. Second, accuracy behaves poorly on these imbalanced sets: a random forest on the IBM benchmark reaches about 85 per cent accuracy yet recovers only around 15 per cent of leavers, an F1 near 0.24 [3]. That gap between cost-blind evaluation and the decision it is meant to inform is the core problem addressed here.

Calibration has recently begun to draw attention here. Pavithran and Vadivel apply a sigmoid correction to their strongest model before scoring, report the resulting Brier improvement, and add local explanations to make individual risk actionable [9], the step recommended in Section 4.2. What the present protocol adds sits on either side of it: whether the models can be separated at all, which precedes calibration, and whether the calibrated probabilities change what the organisation pays, which follows it.

DOI: 10.54254/2755-2721/2026.36208

2.3. Cost-sensitive learning and decision thresholds

That classification should respond to error costs rather than accuracy is long established in cost-sensitive learning, where the optimal threshold is a function of the cost ratio [10]. In HR analytics, Pessach et al. built a prescriptive framework coupling an interpretable model with a population-level optimisation, returning a course of action rather than a ranked list, and noted that turnover cost varies by employee [11]. That is the closest work in spirit, and the contrast is instructive: their framework optimises the decision and returns the interventions a budget should fund, whereas the protocol here evaluates the decision and diagnoses which design choice is worth an analyst's effort. A prescriptive layer also takes the models as given, while the first two steps here ask whether the models can be told apart and whether their probabilities bear a cost rule's weight, questions it presumes settled. Threshold adjustment has been found to beat random undersampling on cost-sensitive tasks [12], among the remedies recent surveys catalogue [13], and this study tests it directly.

Against this background the study makes three contributions: a reusable four-step protocol running from separability through calibration to a cost-based decision and a diagnosis of the lever; empirical evidence, across four datasets, that the dominant lever depends on whether the models are separable and is settled wherever the cost parameters leave the optimal policy nothing to discriminate; and a set of findings that follow from taking the decision rather than the metric as the object of evaluation, that a cost-optimal policy may need no model, that income-weighting steers effort towards the employees least likely to leave, and that oversampling distorts the probabilities without lowering cost.

3. Methodology

Nomenclature. Symbols used throughout this paper.

Symbol	Meaning
k_r	Replacement-cost multiplier, in annual salaries
f_I	Intervention cost, as a fraction of median income
r	Probability that an intervention retains a would-be leaver
t^*	Homogeneous cost-optimal decision threshold
$C_{A,i}$	Replacement (attrition) cost for employee i
$C_{I,i}$	Intervention cost for employee i
r_i	Intervention success rate for employee i
t^*_i	Per-employee cost-optimal threshold
AUC	Area under the ROC curve
ECE	Expected calibration error
Brier	Mean squared error of the predicted probabilities

DOI: 10.54254/2755-2721/2026.36208

3.1. A decision-aware evaluation protocol

The protocol consists of four steps, set out in Figure 1.

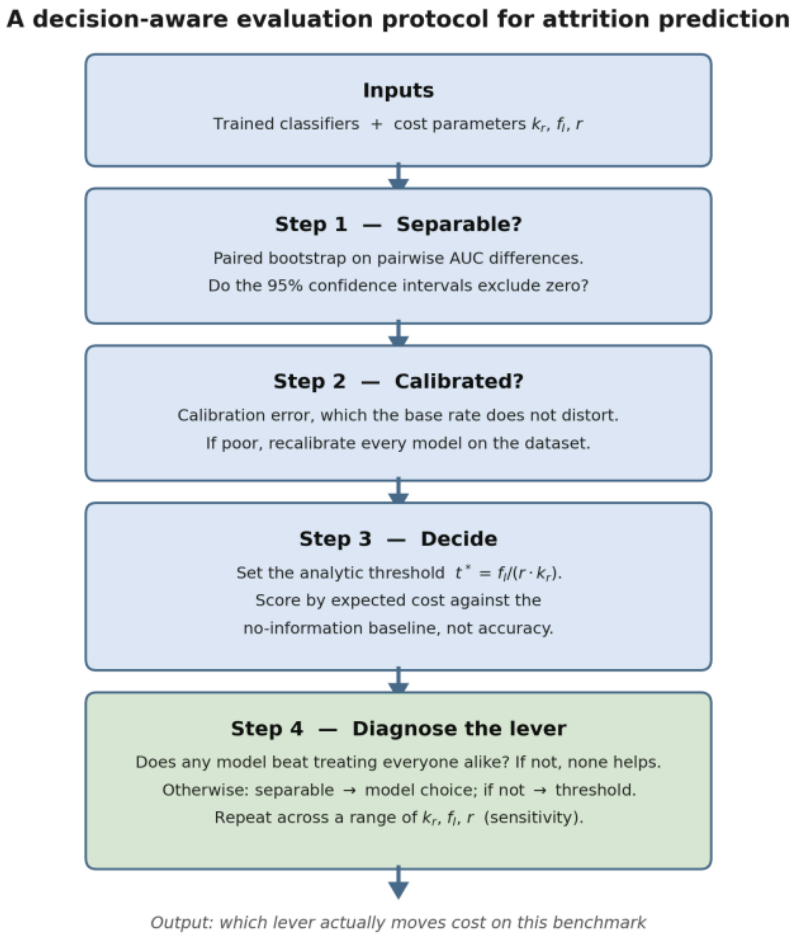


Figure 1. A decision-aware evaluation protocol for attrition prediction

Algorithm 1 states the four steps; the reasoning behind them is as follows. Separability is tested first because a ranking of models is only worth acting on if the differences behind it survive resampling. Calibration is examined next because the third step thresholds the probabilities directly, and it is measured by the expected calibration error, which unlike the Brier score is comparable across datasets of unequal prevalence; where a dataset's probabilities are poorly calibrated, every model on it is recalibrated so that the models remain mutually comparable. The threshold is then fixed analytically from the cost parameters (Section 3.2) and models are scored by expected cost against the no-information baseline rather than by accuracy. The diagnosis asks first whether the policy is genuinely selective, since a threshold under which no model beats treating everyone alike needs no model at all, and otherwise follows separability.

Algorithm 1. The protocol stated so that it can be applied to a benchmark other than those examined here. Step 2 is a precondition of Step 3 rather than a report on it, and the degeneracy test that opens Step 4 is what Section 4.8 turns on.

DOI: 10.54254/2755-2721/2026.36208

Input: trained classifiers M; test partition (X, y); cost parameters k_r, f_I, r; annual income where recorded.
Output: the lever that governs cost on this benchmark.
Step 1 Separability
for every pair (m_i, m_j) in M: draw B bootstrap resamples of the test partition, recomputing AUC(m_i) - AUC(m_j) on each; report the 2.5th and 97.5th percentiles of that difference.
separable <- at least one interval excludes zero.
Step 2 Calibration
for every m_i: compute the expected calibration error of its test probabilities. The Brier score is reported beside it but is not compared across datasets of unequal prevalence.
if the errors on this dataset are large, or too few test records make them reliable-looking by accident: refit every model on it with Platt scaling, fitted on the training partition, so that all models on it are treated alike. Step 3 uses the replacements. No resampling is applied at any point.
Step 3 Decision
t^* <- the homogeneous threshold of Equation (5)
t^*_i <- the per-employee threshold of Equation (4)
for every m_i: cost(m_i) <- expected cost at the relevant threshold, by Equation (3), scored against the cost of intervening on no one rather than against accuracy.
Step 4 Diagnosis
c_all <- cost of intervening on every employee.
if no model beats c_all by more than the models differ from one another: the policy has degenerated. It treats the workforce alike, so no model can improve on it and only the threshold moves cost.
else if separable: the model is the lever.
else: the threshold is the lever.
repeat Steps 3 and 4 across plausible ranges of k_r, f_I and r.

3.2. A transparent cost model for the retention decision

For each employee the organisation takes a binary action: intervene with a retention measure, or do nothing. Let y = 1 denote that the employee later leaves and y = 0 that the employee stays, and let d = 1 denote intervention. Three costs are defined.

Replacement cost is specific to the individual and is assumed to scale with that person's value to the firm, proxied by annual income, as in Equation (1):

$$C_{A,i} = k_r \cdot income_i \tag{1}$$

Intervention cost is treated as a single organisational constant, fixed relative to the median income so that it does not vary with any one employee, as in Equation (2):

$$C_I = f_i \cdot \text{median}(income) \tag{2}$$

DOI: 10.54254/2755-2721/2026.36208

An intervention succeeds with probability r, in which case a would-be leaver is retained. The expected cost over the evaluated population is given by Equation (3):

$$Total = \sum_{i}[(1 - d_i)y_i C_{A,i} + d_i(C_I + y_i(1 - r)C_{A,i})] \tag{3}$$

Minimising this expected cost for a single employee gives a threshold on the predicted leaving probability above which intervention is worthwhile. For one individual, this is Equation (4):

$$t_i^* = \frac{C_I}{r \cdot C_{A,i}} \tag{4}$$

Substituting the median replacement cost gives a single homogeneous threshold in which, after the median cancels, the result reduces to Equation (5):

$$t^* = \frac{f_i}{r \cdot k_r} \tag{5}$$

Equation (5) has a convenient property: the homogeneous threshold reduces to a ratio of the cost parameters and needs no income figures. This follows from the parameterisation, not the data, since intervention cost is a fraction of median income, so the median cancels; had it been fixed in absolute terms the threshold would still depend on the income distribution. The property is useful because the same threshold can then be computed for a workforce whose pay is not recorded numerically (Section 4.8) and yields a transferable decision guide (Section 4.5). An employee is selected for intervention when the predicted leaving probability exceeds the threshold.

Baseline cost parameters. All symbols are listed in the Nomenclature at the start of this section. The baseline values are $k_r = 1.0$, $f_I = 0.10$ and $r = 0.5$. A replacement multiplier of one salary lies inside the range Gallup reports [1], which the sensitivity analysis sweeps. The success rate is anchored to Gallup's finding that about fifty-two per cent of departures are preventable [1], though the two differ: preventability bounds what any measure could achieve, whereas r is the rate a particular measure achieves, which no observational dataset reports. The figure is therefore a ceiling, not an estimate, which is why r is swept from 0.3 to 0.7 (Section 4.5) and returned to in Section 5.

Intervention cost and success rate are treated as constants, which keeps the derivation transparent, though neither need be. If retaining employee i costs $C_{I,i}$ and succeeds with probability r_i, the per-employee threshold becomes $t^*_i = C_{I,i}/(r_i \cdot C_{A,i})$, so an employee costlier to replace or to retain is held to a different bar. Estimating $C_{I,i}$ and r_i needs records of past retention offers and outcomes, which the public benchmarks lack; the constant form is therefore kept for the experiments, the per-employee version being the natural extension once such data exist.

3.3. Datasets

Four datasets are used. Three are public benchmarks, chosen so that they differ in size and in class separability, and the full protocol is run on each of them; the fourth is a real operational dataset held back for external validation of the diagnosis. Their main properties are summarised in Table 1.

DOI: 10.54254/2755-2721/2026.36208

HRDataset_v14 records 311 employees with real salary figures; removing the sixteen terminated for cause leaves the 295 voluntary records used here [14]. Columns that would leak the outcome, such as the date and reason of termination and the employment-status field, are dropped, since only departed employees carry them; target leakage of exactly this kind is among the failures this work cautions against. Identifiers, high-cardinality fields and numeric duplicates of categorical columns are also removed and age is derived from date of birth, leaving thirteen predictors; with the leaking columns gone and the sample small, the discrimination genuinely available here is modest. The IBM HR Analytics set spans 35 fields and is described by its creators as a fictional demonstration dataset [15]; four constant or identifier fields are dropped, leaving 30 features, 7 categorical and 23 numerical. The Kaggle HR set spans 10 fields [16], of which department and a three-level salary band are categorical and the remaining seven numerical, among them satisfaction level, last evaluation, project count, average monthly hours and tenure.

The fourth dataset, the Employee Turnover set released by Babushkin [17], enters on a different footing: it comes from a real organisation and is near-balanced, and its fields, among them recruitment channel, commuting mode and five personality scores, have little in common with the public sets. Critically, pay is recorded only as a two-level category, so the income-weighted threshold of Equation (4) cannot be formed and only the income-independent form of Equation (5) applies; this limitation confines the dataset to testing the income-independent diagnosis, and its results are reported separately in Section 4.8. The set is released under a CC BY-NC-SA 4.0 licence and used here for non-commercial academic research only.

Table 1. The four datasets and their main properties

Dataset	Records (test)	Leaving rate	No-information accuracy	Income field
HRDataset_v14	295 (59)	29.8%	0.695	Real salary
IBM HR Analytics	1,470 (294)	16.1%	0.840	Monthly income
Kaggle HR	14,999 (3,000)	23.8%	0.762	Assigned band
Employee Turnover	1,129 (226)	50.6%	0.504	Categorical only

In Table 1 the leaving rate is computed over the full sample, and the no-information accuracy is the majority-class rate on the test partition, against which model performance is judged.

Only HRDataset_v14 records pay as a real annual figure, which makes its cost analysis the most grounded. The Kaggle set records pay only as an ordinal band, so a representative annual income is assigned to each for the cost calculation: 40,000, 70,000 and 120,000 currency units for the low, medium and high bands. How far the Kaggle conclusions depend on these figures is examined in the sensitivity analysis.

3.4. Experimental procedure

Each dataset is split 80/20 into training and test partitions, stratified by the target with a fixed seed (sensitivity to the seed is examined in Section 4.9). Categorical features are one-hot encoded and numerical features standardised, both fitted on the training partition only inside each model's pipeline, so no test information leaks. No oversampling or class reweighting is applied, to keep the probabilities calibrated, since the cost model depends on them. Four classifiers are tuned by grid search with stratified five-fold cross-validation: logistic regression, random forest, extreme gradient boosting and a support vector machine, spanning the families most used in attrition research, a linear

DOI: 10.54254/2755-2721/2026.36208

baseline, a bagged and a boosted tree ensemble, and a kernel method, so separability is posed across genuinely different inductive biases. One asymmetry bears on the calibration results: the support vector machine returns distances, not probabilities, and the implementation obtains probabilities by Platt scaling on internal folds, so it arrives already corrected by the technique Section 4.2 recommends for the others, and its calibration figures should be read in that light.

4. Results

4.1. Step one: are the models separable

Test AUC values are reported in Table 2 and their bootstrap intervals in Figure 2. On HRDataset_v14 the four models fall between 0.60 and 0.69, a low band that marks the dataset as hard. On the IBM set they fall between 0.78 and 0.82, and on the Kaggle set logistic regression reaches 0.83 while the two tree ensembles reach 0.99.

Table 2. Test AUC, Brier score and expected calibration error for each model on thethree public benchmarks

Dataset	Model	AUC	Brier	ECE
HRDataset_v14	Logistic regression	0.675	0.208	0.100
	Random forest	0.684	0.193	0.025
	Extreme gradient boosting	0.603	0.230	0.144
	Support vector machine	0.690	0.188	0.026
IBM HR Analytics	Logistic regression	0.812	0.099	0.047
	Random forest	0.808	0.109	0.059
	Extreme gradient boosting	0.784	0.110	0.070
	Support vector machine	0.819	0.100	0.023
Kaggle HR	Logistic regression	0.828	0.138	0.049
	Random forest	0.991	0.011	0.015
	Extreme gradient boosting	0.994	0.017	0.009

The paired bootstrap turns these impressions into a test (Table 3). On HRDataset_v14 and the IBM set every interval between the strongest model and the others straddles zero, so the models cannot be distinguished. On the Kaggle set each tree ensemble beats logistic regression by about 0.16 in AUC, well clear of zero, while the two ensembles are indistinguishable from each other. The boundary lies between the linear and tree families, not within the tree models (Figure 2).

That test carries two caveats. Three pairwise comparisons are made per dataset, setting the strongest model against each of the others where four models are available, and the intervals are not corrected for multiplicity; the strongest model is also picked on the same partition that then tests it. Both tilt towards declaring a difference. The two non-separability verdicts are therefore conservative, reached against a test leaning the other way, and the single separability verdict rests on an interval, [+0.146, +0.180], that no correction at this scale would carry to zero.

DOI: 10.54254/2755-2721/2026.36208

Table 3. Paired bootstrap differences in AUC with 95 per cent confidence intervals

Comparison	Dataset	ΔAUC [95% CI]	Significant
SVM − logistic regression	HRDataset_v14	+0.015 [−0.110, +0.140]	No
SVM − random forest	HRDataset_v14	+0.005 [−0.100, +0.106]	No
SVM − extreme gradient boosting	HRDataset_v14	+0.088 [−0.051, +0.221]	No
SVM − logistic regression	IBM	+0.008 [−0.050, +0.066]	No
SVM − random forest	IBM	+0.011 [−0.043, +0.066]	No
SVM − extreme gradient boosting	IBM	+0.035 [−0.025, +0.103]	No
Random forest − logistic regression	Kaggle	+0.163 [+0.146, +0.180]	Yes
Extreme gradient boosting − logistic regression	Kaggle	+0.166 [+0.150, +0.182]	Yes
Extreme gradient boosting − random forest	Kaggle	+0.002 [−0.000, +0.005]	No

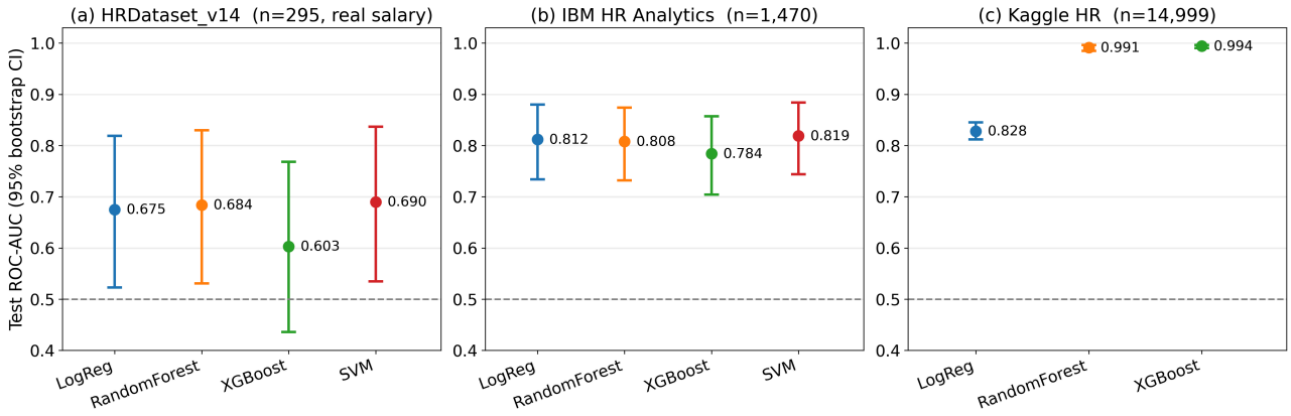


Figure 2. Test AUC with 95 per cent bootstrap intervals on the three public benchmarks, ordered by size

4.2. Step two: are the probabilities calibrated

The calibration check is carried by Table 2. The tree ensembles are well calibrated on the two larger benchmarks and logistic regression is poorer. HRDataset_v14 is the exception and the warning: extreme gradient boosting is badly miscalibrated there, at an error of 0.144, and with only 59 test records over four usable bins its figures are indicative rather than firm, so its cost-threshold results

DOI: 10.54254/2755-2721/2026.36208

are read with more caution than the rest. When calibration is this poor the third step is preceded by recalibration; for a sample this small, Platt (sigmoid) scaling is used rather than isotonic, which needs far more points [18]. Fitting a sigmoid correction here lowers the Brier score of logistic regression from 0.208 to 0.197 and of extreme gradient boosting from 0.230 to 0.202, while the random forest and support vector machine are essentially unchanged. The support vector machine is unchanged for a reason worth naming: it was already Platt-scaled when its probabilities were produced (Section 3.4), so its favourable calibration reflects that internal step, not the algorithm.

Because the protocol treats that correction as a precondition of the third step rather than a comment on the second, the HRDataset_v14 figures in Section 4.3 are computed from the corrected probabilities. Before correction, moving to the cost-optimal threshold saved between 3.0 and 18.6 per cent across the four models; afterwards it saves between 6.8 and 16.5. The correction lifts the two whose probabilities were least honest, logistic regression from 3.0 to 6.8 per cent and extreme gradient boosting from 6.5 to 16.4, and costs the random forest a little, which had least to gain. The four converge, as expected if the cost rule consumes calibrated probabilities rather than rankings: once the probabilities are comparable, so is what they cost.

4.3. Step three: the accuracy illusion and the cost-optimal decision

On the imbalanced benchmarks the conventional metrics mislead. On the IBM set the majority-class baseline, predicting that everyone stays, attains 0.840 accuracy (Table 1), and the random forest at the default 0.5 threshold fails to beat it, reaching 0.837 with an F1 of only 0.20. The cost consequence is immediate: at that threshold the random forest costs almost exactly what intervening on no one would (Table 4). A classifier that looks respectable by accuracy is in effect doing nothing, because high accuracy on an imbalanced set is bought by predicting the majority. The illusion carries to the probability scale too: on HRDataset_v14 extreme gradient boosting returns a negative Brier skill against the base rate, a skill of −0.08, its probabilities worse than a constant guess, a failure no headline metric reports. Nor is this an artefact of the split; Nawaz et al. report the same pattern on the IBM set, high accuracy with recall near 15 per cent [3].

The cost analysis exposes a sharp difference across the three public benchmarks (Table 4). For each dataset the expected cost is reported at the default threshold and at the homogeneous analytic threshold, $t^* = 0.20$ under the baseline parameters, with the resulting saving.

Table 4. Expected cost at the default and cost-optimal thresholds on the three publicbenchmarks, under the baseline parameters. A negative saving indicates the defaultlies nearer the optimum

Dataset (do-nothing cost)	Model	Cost at 0.5	Cost at $t^* = 0.20$	Saving
HRDataset_v14 (1,159,901)	Logistic regression	1,059,652	987,732	6.8%
	Random forest	1,137,318	950,165	16.5%
	Extreme gradient boosting	1,137,318	951,119	16.4%
	Support vector machine	1,113,745	939,475	15.6%
IBM HR Analytics (2,570,448)	Logistic regression	2,341,268	2,296,808	1.9%
	Random forest	2,569,453	2,265,487	11.8%
	Extreme gradient boosting	2,380,759	2,361,049	0.8%
	Support vector machine	2,313,993	2,326,743	−0.6%

DOI: 10.54254/2755-2721/2026.36208

Table 4. (continued)

Kaggle HR (37,470,000)	Logistic regression	34,369,000	31,838,000	7.4%
	Random forest	24,219,000	24,322,000	−0.4%
	Extreme gradient boosting	24,670,000	24,660,000	0.0%

On the two smaller benchmarks, where the models are statistically tied, the choice among them is not the lever that helps. What helps is moving the threshold away from 0.5 towards its cost-optimal value, saving between 0.8 and 16.5 per cent for seven of the eight models on those two sets; the exception is the IBM support vector machine, already near its own optimum at 0.5, which becomes 0.6 per cent dearer. The cost-optimal threshold of 0.20 lies well below the default, reflecting that a missed leaver costs more than a wasted intervention. Figure 3 plots expected cost against threshold for the IBM models and confirms the minimum lies well left of 0.5. On the Kaggle set the result reverses. The gap between logistic regression and the tree ensembles is about ten million currency units, roughly thirty per cent of cost when moving from the linear model to a tree. Moving the threshold helps the weaker linear model by about seven per cent but leaves it far above the trees, and does nothing for the already strong ensembles, so here the model, not the threshold, governs cost.

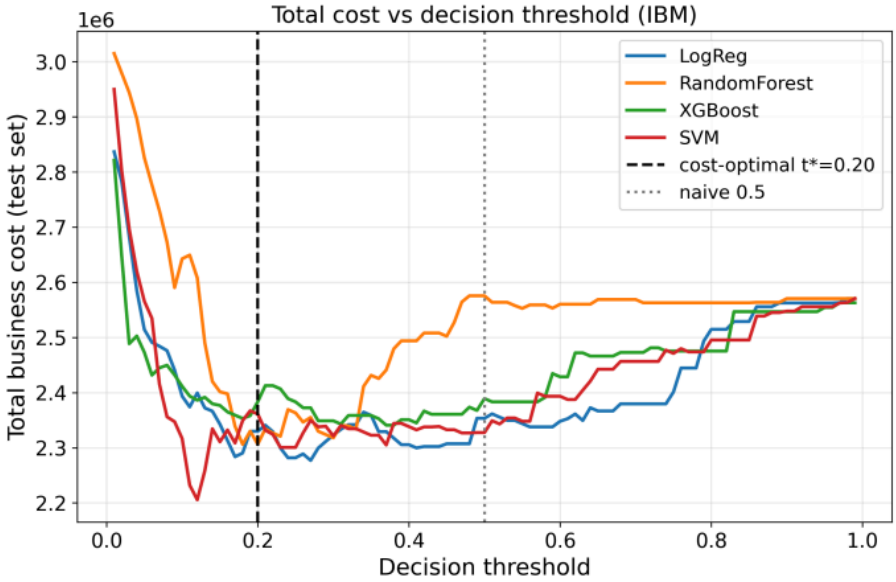


Figure 3. Expected cost against the decision threshold on the IBM set

4.4. Step four: which lever matters

Steps one to three combine into a diagnosis that differs by dataset: where the models are tied the threshold is the lever and delivers material savings, and where they are decisively separated the model fixes the cost while threshold tuning is negligible. On all three public benchmarks the policy at t* is selective, so the diagnosis rests on separability alone. Section 4.8 examines a case where it cannot.

That diagnosis is only worth as much as the choices behind it, and the protocol forces several. The sections that follow test each in turn: the cost parameters that fix the threshold (Section 4.5), the decision to leave the probabilities unresampled (Section 4.7), the datasets themselves (Section 4.8), and the split on which every figure so far has rested (Section 4.9).

DOI: 10.54254/2755-2721/2026.36208

4.5. Sensitivity and a transferable decision guide

Because the cost parameters are uncertain, the conclusions are recomputed over a range of them; two sweeps cover three parameters, since only the ratio of k_r to f_I matters and r is the one remaining dimension. As k_r runs from 0.5 to 2.0 on the IBM set the cheapest model keeps changing, yet the spread between best and worst stays around four per cent, so no model is reliably ahead. On the Kaggle set a tree ensemble is always cheapest, the linear model always dearest by a wide margin, and threshold tuning stays within about one per cent. The qualitative verdict therefore holds across the cost assumptions on both kinds of data.

The Kaggle costs also rest on incomes assigned to three ordinal bands, and no conclusion should turn on the exact figures; by the cancellation noted in Section 3.2, only the ratio between bands matters. Three schemes were tried, compressed, stretched, and a flat one at a single income for every band that drops the weighting entirely. The cost ordering, random forest then extreme gradient boosting then logistic regression, is identical under all four and survives even the flat scheme. The Kaggle verdict rests on the models, not the assigned incomes.

Table 5 evaluates the homogeneous threshold of Equation (5) over plausible ranges, as a small decision guide an analyst can apply directly. Across those ranges the cost-optimal threshold lies below the default 0.5 in every case but one, a low replacement cost paired with a low success rate. The conventional 0.5 cutoff is thus almost never cost-optimal for attrition, and is usually far too high.

Table 5. The homogeneous cost-optimal threshold of Equation (5), over ranges ofthe replacement multiplier and the intervention success rate

$k_r \setminus r$	r = 0.3	r = 0.5	r = 0.7
$k_r = 0.5$	0.667	0.400	0.286
$k_r = 1.0$	0.333	0.200	0.143
$k_r = 2.0$	0.167	0.100	0.071

Note: k_r is the replacement-cost multiplier in annual salaries, f_I the intervention cost as a fraction of median income, and r the probability that an intervention retains a would-be leaver; here $f_I = 0.10$.

4.6. An efficiency-equity tension in cost-weighted targeting

Equation (4) sets a separate threshold for each employee, and because that threshold falls as replacement cost rises, higher earners are alerted at lower predicted probabilities than lower-paid colleagues. Income quartiles on the IBM test set make the effect concrete (Table 6): the median per-employee threshold falls from 0.42 in the lowest quartile to 0.08 in the highest, so the share flagged rises from 19 to 57 per cent. Attrition runs the other way, concentrated in the lowest quartile, with income correlating with leaving at −0.160. Precision among those flagged reaches 0.86 in the bottom quartile but only 0.14 in the top, so most of the policy's attention falls on high earners who go on to stay.

This is not a model artefact but a direct consequence of weighting the decision by the value at stake: minimising expected cost lowers the bar for the employees whose replacement would cost most. Read as policy, the same rule is regressive, steering scarce retention effort towards the best paid while the lower-paid, who are likelier to leave, receive least. A firm might therefore cap the

DOI: 10.54254/2755-2721/2026.36208

income weighting or set a floor on coverage of lower-paid staff. The protocol makes this tension visible rather than creating it, since it is inherent in cost-sensitive targeting.

Table 6. The per-employee threshold and its consequences by income quartile on theIBM test set

Income quartile	Median threshold t*_i	Share flagged	Actual attrition	Precision among flagged
Q1 (lowest paid)	0.42	19%	30%	0.86
Q2	0.27	27%	10%	0.25
Q3	0.18	26%	14%	0.37
Q4 (highest paid)	0.08	57%	11%	0.14

4.7. Oversampling distorts the probabilities without lowering cost

Because the protocol thresholds native probabilities and applies no resampling (Section 3.4), it is worth checking this is more than a convenience. Each model was retrained in an identical pipeline with SMOTE [19] on the training folds only, and the resampled and native versions compared on Brier calibration and on cost at each pipeline's own cost-minimising threshold, the most favourable footing for resampling (Table 7).

The effect is one-directional. Oversampling raises the Brier score of the linear and support-vector models on every dataset where they appear, while leaving the already-calibrated tree ensembles almost untouched (Table 7). Cost, by contrast, barely moves: with each pipeline at its own best threshold, cost changes by under one and a half per cent everywhere, the sole exception being the IBM support vector machine at roughly seven per cent dearer.

Oversampling thus distorts the probabilities the cost-optimal threshold depends on while leaving cost essentially unchanged. This is the empirical case for absorbing class imbalance into the decision threshold rather than the training data, and it matches both Leevy et al. [12], who find threshold tuning without undersampling the better cost-sensitive choice, and the broader cost-sensitive literature [13].

Table 7. Resampled versus native pipelines on the three public benchmarks, each atits own cost-minimising threshold.

Dataset	Model	Brier (native)	Brier (SMOTE)	Δ cost at own optimum (%)
IBM	Logistic regression	0.099	0.151	+1.1
	Random forest	0.109	0.111	+1.2
	Extreme gradient boosting	0.110	0.107	+0.4
	Support vector machine	0.100	0.130	+7.1
Kaggle	Logistic regression	0.138	0.167	−1.2
	Random forest	0.011	0.012	+0.1
	Extreme gradient boosting	0.017	0.016	−0.6

DOI: 10.54254/2755-2721/2026.36208

Table 7. (continued)

HRDataset_v14	Logistic regression	0.208	0.238	+0.1
	Random forest	0.193	0.197	−1.4
	Extreme gradient boosting	0.230	0.236	−0.7
	Support vector machine	0.188	0.204	+0.1

4.8. External validation on a real workforce dataset

The three public benchmarks are teaching or simulated rather than operational (Section 5), and the Employee Turnover dataset of Section 3.3 supplies the external check. Because it records pay only as a category, the per-employee threshold of Equation (4) is unavailable and the analysis uses the income-independent threshold of Equation (5), the regime for which it was derived; the first three steps are otherwise unchanged. Its fields barely resemble the public sets, so this is a robustness check, not a replication.

Here the models are not tied, though their discrimination is only moderate. The random forest leads at an AUC of 0.77 and logistic regression trails at 0.65, a paired-bootstrap difference of 0.12 clearly above zero (Table 8). Their probabilities are of ordinary quality, with calibration errors overlapping the IBM band (Table 8). By the first two steps this dataset resembles a smaller Kaggle: the models can be told apart and their probabilities are usable.

Separability nonetheless buys almost nothing, and the reason is the cost parameters, not the models. With a leaving rate of one half, an intervention costing a tenth of a salary, and a half success rate, the analytic threshold of 0.20 sits so far below the prevailing risk that it flags nearly everyone. Intervening on the whole workforce costs 79.6 on this partition against 114.0 for doing nothing, and the best model at its own optimum saves only 2.3 per cent on that. A policy that treats everyone alike needs no model, which is why the 0.12 AUC advantage is worth just 1.7 per cent in cost while moving the threshold from 0.5 to 0.20 is worth 6.6 to 12.1 per cent (Table 8).

The case therefore marks a boundary of the protocol's fourth step rather than a counterexample to it. Separability establishes whether a ranking is warranted; it does not promise that a better ranking will be worth anything, because the cost parameters may already have settled the policy. Asking whether the policy at t^* is genuinely selective, before asking which lever to pull, costs nothing and would have flagged this dataset at once, and the check has been folded into the fourth step accordingly (Section 3.1). The modest sample and categorical pay field mean the finding extends the earlier results rather than overturning them.

Table 8. External validation on the Employee Turnover dataset (n = 1,129)

Model	AUC [95% CI]	Brier	ECE	Cost saving from optimal threshold (%)
Logistic regression	0.65 [0.58, 0.72]	0.234	0.065	12.1
Random forest	0.77 [0.71, 0.83]	0.195	0.044	6.6
Extreme gradient boosting	0.70 [0.62, 0.77]	0.231	0.116	10.7
Support vector machine	0.72 [0.65, 0.79]	0.213	0.057	9.7

DOI: 10.54254/2755-2721/2026.36208

4.9. Robustness to the choice of split

The results so far rest on one stratified split, and the intervals of Section 4.1 capture variation within that partition, not variation from the split itself. With two datasets small, that distinction matters: a non-separability verdict could be an accident of the partition. The whole procedure, grid search included, was repeated on ten independent stratified splits (Table 9).

Where the first step calls the models tied, repetition sharpens the verdict. On the IBM set the top AUC changes hands four times across the ten splits, and on HRDataset_v14 three times, so the apparent winner is close to arbitrary: reporting one split's leader as the better model reports an accident of partitioning. The spread between best and worst stays small and stable throughout (Table 9).

On HRDataset_v14 extreme gradient boosting never leads, its mean well below logistic regression, so repetition sets it apart from the rest; it cannot say which of the other three is best, as the lead passes among them from split to split. Repetition can thus rule a model out here without ruling one in, less than a ranking and as much as the data will bear.

Where the first step calls the models separable, the verdict is stable. On the Kaggle set the two tree ensembles lead on every split, and on the Employee Turnover set the random forest leads on all ten, with margins many times their standard deviation (Table 9), consistent with Section 4.8. The first step therefore reproduces across splits: what it calls tied stays tied, and what it calls separable stays separable.

Two qualifications belong here. Split variation on HRDataset_v14 is large in absolute terms, with per-model standard deviations reaching 0.074 and mean AUCs running some four to six points above the single split in Table 2, so that benchmark is best read as one draw rather than a stable characterisation. And the repetition varies the split alone: the cost parameters are held at baseline throughout, their sensitivity having been examined in Section 4.5.

Table 9. Ten independent stratified splits: models leading and splits led, mean bestworst AUC gap, and the largest single-model standard deviation

Dataset	Models leading, with splits led	Best − worst AUC, mean ± sd	Largest single-model sd
HRDataset_v14	Logistic regression (6), random forest (2), support vector machine (2)	0.087 ± 0.026	0.074
IBM HR Analytics	Support vector machine (5), logistic regression (3), random forest (1), extreme gradient boosting (1)	0.042 ± 0.014	0.043
Kaggle HR	Random forest (5), extreme gradient boosting (5)	0.171 ± 0.006	0.005
Employee Turnover	Random forest (10)	0.134 ± 0.032	0.037

5. Discussion and conclusion

Once separability is taken into account, the four datasets point to one conclusion: the governing lever is a property of the data, not of the modelling, and the protocol makes the question explicit rather than assuming an answer. The wide spread of published IBM accuracies is what this would predict [4], and the spread is telling: a recent study reports a random forest at 0.974 AUC on this

DOI: 10.54254/2755-2721/2026.36208

benchmark [9], a cross-validated score on SMOTE-balanced data, against the 0.808 held-out score here with no resampling. The two are not comparable, and the gap is a matter of protocol, not algorithms, which is why the first step asks whether a difference is real before any model is preferred to another.

Several limitations bound these claims. Four datasets show that the dominant lever varies but cannot characterise how it varies in general; separating the effect of separability from that of sample size would need a large but low-signal dataset carrying income, and none is publicly available. The three public benchmarks are teaching or simulated rather than one firm's records, the IBM set being described by its creators as fictional [15], while the Employee Turnover set is operational but records pay only as a category. HRDataset_v14 is small enough that its figures carry little weight, its calibration errors resting on 59 test records and a single model's AUC moving by as much as 0.07 between splits. The cost model is stylised, and as Section 4.6 shows its parameterisation is regressive in effect, so an organisation adopting it may wish to cap the income weighting or set a floor on coverage of lower-paid staff.

A deeper limitation bounds what a protocol of this kind can claim at all. The cost model treats the success rate r as constant, assuming a retention measure works equally well whatever an employee's risk of leaving. The uplift literature gives reason to doubt this, since the employees a risk model ranks highest are often those whose decision is already made. None of the four datasets can settle it, because in none has anyone been intervened upon, and estimating r_i would need records of retention offers and their outcomes that observational data does not contain. Verifying the probabilities is therefore necessary for a decision-aware evaluation and, on data of this kind, not sufficient for a decision.

The wider message is that attrition models should be separated by differences tested for significance rather than raw metric gaps, that evaluation should be tied to the cost of the decision rather than to a cost-blind metric, and that the probabilities the decision rests on should be verified rather than assumed. The limitations mark the directions for further work: applying the protocol to a large but weakly separable workforce would disentangle sample size from signal strength, and estimating the cost parameters, and a varying success rate, from an organisation's own intervention records would tie the evaluation more closely to the decision it serves.

References

[1] Gallup. (2019). This fixable problem costs U.S. businesses $1 trillion. *Gallup Workplace*. https://www.gallup.com/workplace/247391/fixable-problem-costs-businesses-trillion.aspx

[2] Al Akasheh, M., Malik, E. F., Hujran, O., & Zaki, N. (2024). A decade of research on machine learning techniques for predicting employee turnover: A systematic literature review. *Expert Systems with Applications, 238*, 121794. https://doi.org/10.1016/j.eswa.2023.121794

[3] Nawaz, M. S., Nawaz, M. Z., Fournier-Viger, P., & Luna, J. M. (2024). Analysis and classification of employee attrition and absenteeism in industry: A sequential pattern mining-based methodology. *Computers in Industry, 159–160*, 104106. https://doi.org/10.1016/j.compind.2024.104106

[4] Benabou, A., & Touhami, F. (2026). Optimising HRM practices in call centres: Predicting and explaining employee turnover intention using classification and regression trees. *International Journal of Organizational Analysis, 34*(3), 870–891. https://doi.org/10.1108/IJOA-12-2024-5117

[5] Talebi, H., Khatibi Bardsiri, A., & Khatibi Bardsiri, V. K. (2025). Machine learning approaches for predicting employee turnover: A systematic review. *Engineering Reports, 7*(8), e70298. https://doi.org/10.1002/eng2.70298

[6] Park, J., Feng, Y., & Jeong, S.-P. (2024). Developing an advanced prediction model for new employee turnover intention utilizing machine learning techniques. *Scientific Reports, 14*, 1221. https://doi.org/10.1038/s41598-023-50593-4

[7] Shafie, M. R., Khosravi, H., Farhadpour, S., Das, S., & Ahmed, I. (2024). A cluster-based human resources analytics for predicting employee turnover using optimized Artificial Neural Networks and data augmentation.

DOI: 10.54254/2755-2721/2026.36208

Decision Analytics Journal, 11, 100461. https://doi.org/10.1016/j.dajour.2024.100461

[8] Liu, H., Qiu, Q., & Zhang, Q. (2024). End-to-end approach of multi-grained embedding of categorical features in tabular data. *Information Processing and Management, 61*, 103645.

[9] Pavithran, M. S., & Vadivel, S. M. (2026). Explainable attrition risk scoring for managerial retention decisions in human resource analytics. *Frontiers in Big Data, 8*, 1699561. https://doi.org/10.3389/fdata.2025.1699561

[10] Elkan, C. (2001). The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 973–978).

[11] Pessach, D., Singer, G., Avrahami, D., Chalutz Ben-Gal, H., Shmueli, E., & Ben-Gal, I. (2020). Employees recruitment: A prescriptive analytics approach via machine learning and mathematical programming. *Decision Support Systems, 134*, 113290. https://doi.org/10.1016/j.dss.2020.113290

[12] Leevy, J. L., Johnson, J. M., Hancock, J., & Khoshgoftaar, T. M. (2023). Threshold optimization and random undersampling for imbalanced credit card data. *Journal of Big Data, 10*(1), 58.

[13] Araf, I., Idri, A., & Chairi, I. (2024). Cost-sensitive learning for imbalanced data: A review. *Artificial Intelligence Review, 57*, 80.

[14] Huebner, R., & Patalano, C. (2020). Human resources data set [HRDataset_v14.csv]. *Kaggle*. https://www.kaggle.com/datasets/rhuebner/human-resources-data-set

[15] Pavansubhash. (2017). IBM HR analytics employee attrition & performance [Data set]. *Kaggle*. https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

[16] Benistant, L. (2016). Human resources analytics [Data set; HR_comma_sep.csv]. *Kaggle*. https://www.kaggle.com/ludobenistant/hr-analytics

[17] Babushkin, E. (2020). Employee turnover [Data set; turnover.csv, shared by davinwijaya]. *Kaggle*. https://www.kaggle.com/datasets/davinwijaya/employee-turnover

[18] Niculescu-Mizil, A., & Caruana, R. (2005). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning* (pp. 625–632). Association for Computing Machinery.

[19] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357.