Highlights

Predict-then-optimize or predict-and-optimize? An empirical evaluation of cost-sensitive learning strategies

Toon Vanderschueren,Tim Verdonck,Bart Baesens,Wouter Verbeke

- We review the literature on cost-sensitive learning and differentiate between two key approaches: cost-sensitive training of models and cost-sensitive decision-making.
- We conduct an extensive empirical analysis to evaluate and compare different cost-sensitive learning strategies using nine real-world datasets from different application areas.
- The decision-making strategy is generally found to be more important in terms of costs than the objective function that is used to train a classifier.

Predict-then-optimize or predict-and-optimize? An empirical evaluation of cost-sensitive learning strategies

Toon Vanderschueren[a,b,*], Tim Verdonck[b,c], Bart Baesens[a,d] and Wouter Verbeke[a]

[a] *Decision Sciences and Information Management, KU Leuven, Belgium*
[b] *Department of Mathematics, University of Antwerp, Belgium*
[c] *Department of Mathematics, KU Leuven, Belgium*
[d] *Southampton Business School, University of Southampton, United Kingdom*

ARTICLE INFO

Keywords:
Cost-sensitive learning
Instance-dependent costs
Classification
Supervised learning

ABSTRACT

Predictive models are increasingly being used to optimize decision-making and minimize costs. A conventional approach is *predict-then-optimize*: first, a predictive model is built; then, this model is used to optimize decision-making. A drawback of this approach, however, is that it only incorporates costs in the second stage. Conversely, the *predict-and-optimize* approach proposes learning a predictive model by directly minimizing the cost of the downstream decision-making task. This is achieved by using a task-specific loss function incorporating the costs of different outcomes in the first stage, with the eventual aim of obtaining more cost-effective decisions in the second stage. This work compares both approaches in the context of cost-sensitive classification. Conceptually, we use the two-stage framework to categorize existing cost-sensitive learning methodologies by differentiating between methodologies for cost-sensitive model training and decision-making. Empirically, we compare and evaluate both approaches using different cost-sensitive training and decision-making methodologies, as well as both class-dependent and instance-dependent cost-sensitive methods. This is achieved using real-world data from a range of application areas and a combination of cost-sensitive and cost-insensitive performance measures. The key finding is that the decision-making strategy is generally found to be more effective than training with a task-specific loss or their combination.

1. Introduction

Predictive models are increasingly being used to optimize decision-making. In many applications, the goal is to minimize the cost incurred through decisions. A conventional approach is to *predict-then-optimize*: in the first stage, a predictive model is built to maximize its predictive power; then, in the second stage, decisions are made based on the model's predictions and the costs associated with decisions. However, a drawback of this approach is that it only considers costs in the second decision-making stage. Conversely, several recent works proposed an alternative, integrated *predict-and-optimize* approach [14, 45]. This approach works by integrating costs within the learning objective of the predictive model in the first stage. Thus, model learning is decision-focused: the quality of the predictions on downstream decision-making is directly considered [45]. The goal of this approach is to make more cost-effective decisions. Therefore, the model's predictions need only be accurate insofar as this contributes to optimal decision-making in the second stage.

We use the predict-and-optimize approach to analyze an earlier line of work on cost-sensitive machine learning [19, 16]. Although predict-and-optimize has typically been applied to problems such as stochastic programming and combinatorial optimization [14, 45], the goal of cost-sensitive methodologies is similar to the one in predict-and-optimize in the sense that both aim to obtain better decisions by aligning the predictive model with the decision-making context. Even though a variety of cost-sensitive learning methodologies have been proposed to more effectively deal with classification tasks where different decisions have different costs associated with them, it is not clear which of these approaches work best and how they relate to each other. The lack of understanding of these methods is due to a combination of reasons. First, novel approaches are often only compared to their cost-insensitive counterparts. Second,

[*] Corresponding author.

toon.vanderschueren@kuleuven.be (T. Vanderschueren); tim.verdonck@uantwerpen.be (T. Verdonck); bart.baesens@kuleuven.be (B. Baesens); wouter.verbeke@kuleuven.be (W. Verbeke)

ORCID(s): 0000-0001-7202-1589 (T. Vanderschueren); 0000-0003-1105-2028 (T. Verdonck); 0000-0002-5831-5668 (B. Baesens); 0000-0002-8438-0535 (W. Verbeke)

a variety of different metrics are used to judge these methodologies. Third, a limited number of datasets are typically used. These are often also proprietary, making it impossible to replicate findings.

Using the two-stage framework, we categorize existing techniques as either learning cost-sensitive models in the first stage or making cost-sensitive decisions in the second stage. Thus, we can empirically compare the predict-then-optimize and predict-and-optimize approaches for cost-sensitive classification. Our main contributions are as follows:

- Conceptually, we review the literature on cost-sensitive learning and differentiate between two general approaches using the two-stage framework: cost-sensitive training of models and cost-sensitive decision-making.
- Empirically, we conduct an extensive evaluation to compare predict-then-optimize and predict-and-optimize using nine real-world datasets from different application areas. Moreover, we analyze different methods of incorporating costs during training and during decision-making, as well as their combinations. We also look at the effect of incorporating costs at an instance level as opposed to a class level.
- To facilitate replication of the presented results and encourage further research on instance-dependent cost-sensitive learning, the full experimental code is made publicly available at `https://github.com/toonvds/CostSensitiveLearning`.

2. Related work

Before applying the two-stage framework to cost-sensitive classification, we summarize existing work based on two criteria (see Table 1): 1) whether the costs are class- or instance-dependent (see section 2.1) and 2) whether costs are integrated before, during or after the training of a classification model (see section 2.2). Before training, instances can be preprocessed, i.e., they can be sampled, weighted, or relabeled (e.g., MetaCost [13]). During training, costs can be incorporated in the learning algorithm, e.g., with custom decision tree splitting criteria or through a cost-sensitive objective function. After training, the decision threshold can be made cost-sensitive.

There are other cost-sensitive strategies that are not covered by these criteria and outside the scope of this work. Several methodologies look at cost-sensitive feature [28] or model selection [26]. A recent, dedicated framework and overview of cost-sensitive ensemble methods is presented in [34]. Moreover, whereas this work focuses on cost-sensitive learning in the context of supervised learning, other work has focused on cost-sensitive semi-supervised [44] and positive-unlabeled learning [8]. Finally, a related line of work in regression considers asymmetric objectives to more closely align a regression model's learning objective with the decision-making task [19].

2.1. Types of costs

In classification, costs can be formalized with a cost matrix [16]. Similar to how the confusion matrix in Table 2a differentiates between outcomes depending on the actual and predicted class, a cost matrix associates a cost to these different outcomes. In Table 2b, a cost matrix is shown for the setting with class-dependent costs. When costs are instance-dependent, each instance will have a different cost matrix, denoted by the index i in Table 2c. Note that this framework also allows the inclusion of benefits or profits in the form of negative costs.

2.1.1. Class-dependent costs

Various cost-sensitive machine learning techniques have been proposed for dealing with class-dependent costs. In this setting, one class is more important in terms of costs, and because of that, a cost-sensitive model should focus more on correctly classifying this class compared to a cost-insensitive model. In the simple case of a linear decision boundary, class-dependent costs result in a parallel shift away from the more costly class (see Figure 1).

Even though no general benchmarking studies exist, two works analyze class-dependent cost-sensitive boosting specifically and find cost-sensitive decision-making to be the most effective strategy [49, 31]. Finally, note that the literature on class-dependent cost-sensitive learning is intertwined with the literature on learning with class imbalance, and by using the appropriate costs, similar techniques can be used. For a recent survey on class imbalance, we refer the reader to [24].

2.1.2. Instance-dependent costs

Conceptually, many of the techniques for dealing with class-dependent costs can and have been transferred to the instance-dependent setting. However, instance-dependent costs create an additional degree of complexity, as they depend not only on the class but also on characteristics of the instance (e.g., the transaction's amount in fraud detection). For a simple linear classifier, class-dependent costs result in a parallel shift of the cost-insensitive optimal decision

Table 1: **Cost-sensitive learning overview.** We present an overview of various cost-sensitive learning methods in terms of the type of costs, place with respect to model training and classifier(s) used when applicable.

Reference	Costs		Place with respect to training			Classifier(s)
	CD	ID	Before	During	After	
[32]	✓	✗	✗	✓	✓	DR
[41]	✓	✗	✗	✓	✗	DT
[18]	✓	✗	✗	✗	✓	DR, NN
[23]	✓	✗	✓	✓	✓	NN
[36]	✓	✗	✗	✓	✗	BO
[13]	✓	✗	✓	✗	✗	-
[43]	✓	✗	✗	✓	✗	SVM
[15]	✓	✗	✗	✓	✗	DT
[6]	✓	✗	✗	✓	✗	NB
[40]	✓	✗	✓	✗	✗	DT
[50]	✓	✗	✓	✗	✓	NN
[37]	✓	✗	✗	✗	✓	-
[39]	✓	✗	✗	✓	✗	BO
[7]	✓	✗	✓	✗	✗	-
[12]	✓	✗	✓	✗	✗	-
[27]	✓	✗	✗	✓	✗	SVM
[49]	✓	✗	✗	✓	✓	BO
[22]	✓	✗	✗	✓	✗	BO
[38]	✓	✗	✗	✓	✓	LR
[21]	✓	✗	✗	✓	✓	DT
[17]	✗	✓	✗	✓	✗	BO
[47]	✗	✓	✗	✗	✓	-
[48]	✗	✓	✓	✗	✗	BO
[5]	✗	✓	✗	✓	✗	SVM
[35]	✗	✓	✗	✓	✗	DT
[1]	✗	✓	✗	✓	✓	LR
[4]	✗	✓	✗	✗	✓	-
[2]	✗	✓	✗	✓	✗	DT
[20]	✗	✓	✗	✓	✓	BO, LR

Costs CD: class-dependent, ID: instance-dependent
Classifier BO: boosting, DR: decision rule, DT: decision tree, LR: logistic regression, NB: Naive Bayes, NN: neural network, SVM: support vector machine, -: classifier-agnostic

Table 2: **Cost matrix.** Extending the confusion matrix (2a) to a class- (2b) and instance-dependent cost matrix (2c).

(a) Confusion matrix

		Actual	
		0	1
Predicted	0	TN	FN
	1	FP	TP

(b) Class-dependent cost matrix

		Actual	
		0	1
Predicted	0	c^{TN}	c^{FN}
	1	c^{FP}	c^{TP}

(c) Instance-dependent cost matrix

		Actual	
		0	1
Predicted	0	c_i^{TN}	c_i^{FN}
	1	c_i^{FP}	c_i^{TP}

boundary, whereas instance-dependent costs can additionally result in a rotation of this boundary (see Figure 1). This illustrates that when costs are instance-dependent, the learner needs to consider both the class distribution (explicitly) and cost distribution (implicitly). In theory, including instance-dependent costs in decision-making can lead to lower

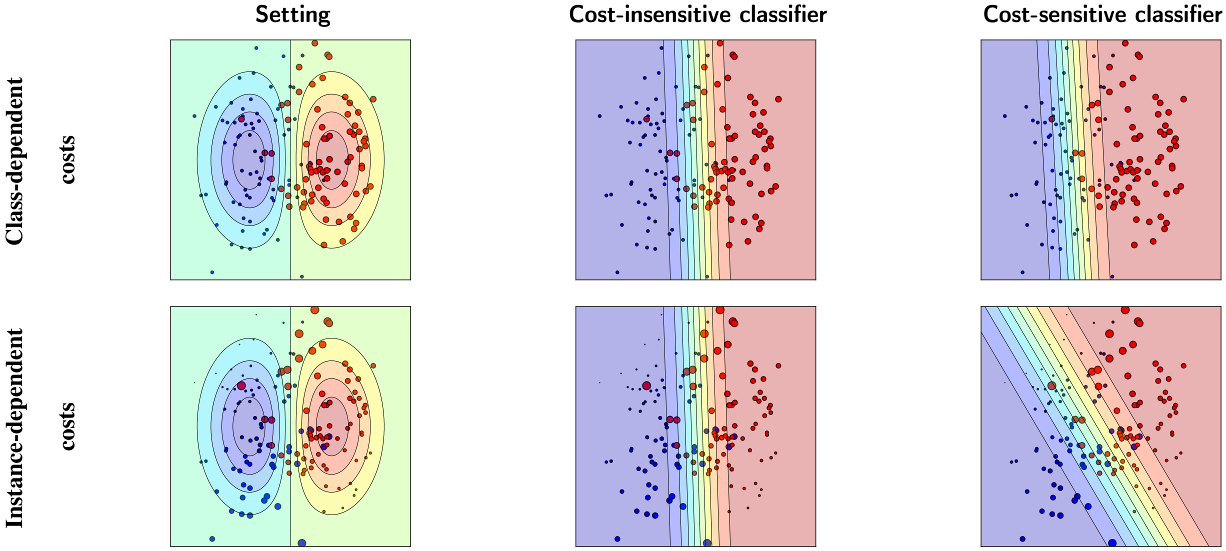


Figure 1: Toy example with class-dependent (top) and instance-dependent costs (bottom). (Left) Two classes and the probability distribution are shown, with the instance size proportional to its misclassification cost. (Middle) The resulting decision boundary for a *cost-insensitive* classifier mimics the underlying probability distribution. (Right) For a *cost-sensitive* classifier, the decision boundary lies further from the more costly class when costs are class-dependent. With instance-dependent costs, the decision boundary is not only related to the probability distribution, but also the cost distribution.

overall costs [5]. However, despite the conceptual differences, the benefits and drawbacks of using instance- rather than class-dependent costs on the performance of the learning algorithms have not yet been examined empirically.

2.2. Cost-sensitive classification in the predict-and-optimize framework

Machine learning models are increasingly being used to support and optimize decision-making. The conventional two-stage *predict-then-optimize* approach builds a predictive model with the aim of maximizing its accuracy in the first stage and then uses this model to optimize decision-making in the second stage. Conversely, *predict-and-optimize* is a recent paradigm that directly optimizes a predictive model by using a task-specific loss function in the first stage to optimize decision-making in the second stage [45]. The benefit of an integrated approach is that it directly learns a model to minimize the cost of the eventual decisions. The model in the predict-then-optimize approach might produce more accurate predictions overall, but the model in the predict-and-optimize is decision-focused instead of prediction-focused: it learns to accurately predict only insofar as it impacts the decision-making in the second stage, and as such, the resulting decisions are of higher quality [14].

We can apply this two-stage framework to cost-sensitive classification: in the first stage, a predictive model (i.e., a classifier) is built; in the second stage, this model is used to assign class labels to instances in order to minimize the resulting cost. Thus, we can classify existing cost-sensitive learning methodologies as either learning a predictive model in the first stage or optimizing decisions in the second stage. This distinction is based on whether costs are integrated before, during or after the training of a model (see Table 1). The first category, *cost-sensitive training of models*, consists of techniques that are applied before or during training to build a classifier, whereas the second, *cost-sensitive decision-making*, consists of thresholding techniques that are applied after training to make decisions. Note that several approaches are possible in each stage – several of these are described in the following.

2.2.1. Cost-sensitive training of classification models

In the first stage, a predictive model is learned. A traditional approach learns a model by maximizing its likelihood – independent of how predictions are used in the downstream task. Alternatively, learning can be done with a task-specific loss function to align the model with the objective of the downstream task and obtain a cost-sensitive model. Thus, the quality of the predictions on the resulting solution of the downstream task is directly considered [14].

Training with a traditional classification objective also leads to tradeoffs in the resulting model's accuracy for different regions of the input-output space. However, in contrast to the decision-focused approach, this tradeoff might not be optimal for the downstream task [14]. An illustration of the different tradeoffs for a cost-insensitive and a cost-sensitive linear model can be seen in Figure 1.

In general, machine learning algorithms can be understood in terms of risk minimization [42]. In this framework, the goal of a learning algorithm is to find the classifier that minimizes the risk. Formally, for a distribution $p(\mathbf{x}, y)$ and a classifier $f_\theta : \mathbf{X} \rightarrow [0, 1] : \mathbf{x} \mapsto f_\theta(\mathbf{x})$ defined by parameters $\theta \in \Theta$, the risk to be minimized is:

$$R(\theta) = \int \int \mathcal{L}(y, \mathbf{x}, \theta) p(\mathbf{x}, y) d\mathbf{x} dy,$$

where $\mathcal{L}(y, \mathbf{x}, \theta)$ represents the loss or objective function for a classifier $f_\theta(\mathbf{x})$ and data $(\mathbf{x}, y)$ [12]. In reality, the true joint probability distribution $p(\mathbf{x}, y)$ is unknown. Consequently, the learner relies on the empirical density to minimize the risk given the available training data. This is the principle of empirical risk minimization (ERM) [42]. For a dataset $(\mathbf{x}_i, y_i) \in \mathcal{D}$ with $i \in \{1, ..., N\}$, the empirical risk is defined as:

$$R_{emp}(\theta) = \mathop{\mathbb{E}}_{x,y \sim D} \left[\mathcal{L}(y_i, \mathbf{x}_i, \theta) \right] = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(y_i, \mathbf{x}_i, \theta).$$

Clearly, it is essential to choose an appropriate loss function $\mathcal{L}$. A first and straightforward candidate is the zero-one loss comparing the actual y and predicted label $\hat{y}$: $\mathcal{L}^{0/1}(y, \hat{y}) = I(y \neq \hat{y})$, although it is common to use a convex surrogate. A popular choice is the **cross-entropy** loss, which is equivalent to the maximum likelihood (ML) method [42]. In binary classification, we have $\mathcal{L}^{CE}(y_i, \mathbf{x}_i, \theta)$:

$$y_i \log f_\theta(\mathbf{x}_i) + (1 - y_i) \log\big(1 - f_\theta(\mathbf{x}_i)\big). \tag{1}$$

However, as argued above, a disadvantage of the maximum likelihood approach is that it does not take into account the costs of different decisions. Consequently, using this loss function, the empirical risk fails to reflect the true risk of the downstream task. To solve this issue, the ERM framework can be extended to include costs: given a dataset $(\mathbf{x}_i, y_i, \mathbf{c}_i) \in \mathcal{D}$ for $i \in \{1, ..., N\}$ with an instance's cost matrix $\mathbf{c}_i$, a cost-sensitive loss function $\mathcal{L}(y, \mathbf{x}, \mathbf{c}, \theta)$ can be defined [12]. In this way, the empirical risk can be made cost-sensitive, and a task-specific loss can be used.

A first approach for a task-specific loss is to weight the training examples by their misclassification cost [16, 48]. This can be formulated in terms of a **weighted cross-entropy** loss function $\mathcal{L}^{wCE}(y_i, \mathbf{x}_i, \mathbf{c}_i, \theta)$ [12]:

$$c_i^{FN} y_i \log f_\theta(\mathbf{x}_i) + c_i^{FP} (1 - y_i) \log\big(1 - f_\theta(\mathbf{x}_i)\big). \tag{2}$$

Note that this approach is equivalent to oversampling proportional to misclassification costs [12].

A second task-specific approach builds on the idea that the optimal cost-sensitive prediction minimizes the expected cost [16]. Using this, an alternative loss function can be defined that equals the expected cost [1, 20]. The corresponding empirical risk is the **average expected cost** $\mathcal{L}^{AEC}(y_i, \mathbf{x}_i, \mathbf{c}_i, \theta)$:

$$y_i \Big(f_\theta(\mathbf{x}_i) c_i^{TP} + \big(1 - f_\theta(\mathbf{x}_i)\big) c_i^{FN} \Big) + (1 - y_i) \Big(f_\theta(\mathbf{x}_i) c_i^{FP} + \big(1 - f_\theta(\mathbf{x}_i)\big) c_i^{TN} \Big). \tag{3}$$

2.2.2. Cost-sensitive decision-making

The predictive model learned in the first stage is used to make decisions in the second stage. In the case of cost-sensitive classification, the predicted posterior probabilities are used to classify instances with the aim of minimizing the resulting cost. This is achieved by applying an appropriate decision threshold. There are several policies that can be used to optimize decision-making in the second stage.

The first and most natural candidate is the instance-dependent cost-sensitive threshold, which predicts the class with the minimal expected risk. Because this risk depends not only on the posterior probabilities but also on the associated costs, an instance's optimal classification should consider both its posterior probability and its cost related to the different outcomes [16]. Formally, for an instance i, a prediction $\hat{y}_i$ has a certain risk $R(\hat{y}_i|\mathbf{x}_i, y)$ associated with it depending on its posterior probability and cost matrix:

$$R(\hat{y}_i|\mathbf{x}_i, y_i) = \begin{cases} p(y_i = 0|\mathbf{x}_i) c_i^{TN} + p(y_i = 1|\mathbf{x}_i) c_i^{FN} \text{ if } \hat{y}_i = 0 \\ p(y_i = 0|\mathbf{x}_i) c_i^{FP} + p(y_i = 1|\mathbf{x}_i) c_i^{TP} \text{ if } \hat{y}_i = 1 \end{cases}$$

The optimal decision $\hat{y}^*$ minimizes this risk, i.e., $\hat{y}_i^* = 1$ if $R(\hat{y}_i = 1|\mathbf{x}_i) < R(\hat{y}_i = 0|\mathbf{x}_i)$. Using this, the optimal decision threshold t_i^* for an instance can be found: $\hat{y}_i^* = 1$ if $p(y_i = 1|\mathbf{x}_i) > t_i^*$, with

$$t_i^* = \frac{c_i^{FP} - c_i^{TN}}{c_i^{FP} - c_i^{TN} + c_i^{FN} - c_i^{TP}}. \tag{4}$$

For a given classifier θ, the score $f_\theta(\mathbf{x}_i)$ can be used as an estimate of the posterior probability $p(y_i = 1|\mathbf{x}_i)$. However, it is important to note that this requires the model to produce calibrated probabilities or that some calibration method is first applied to the model's output.

In addition to the instance-dependent cost-sensitive threshold, several alternative decision-making strategies are possible. For example, by using the average cost matrix, a single class-dependent cost-sensitive threshold can be used for all instances. Furthermore, instead of the theoretically motivated optimal thresholds, several alternatives are possible. Empirical thresholding searches for the threshold that gives the lowest cost on a validation set [37]. Moreover, a common heuristic is to use the class imbalance threshold, which uses the prior probability of the minority class as a threshold $t^{CI} = P(Y = 1)$. The idea is that this will compensate for the lack of focus on this class, which is often more important in terms of costs.

3. Methodology

The goal of this work is to empirically analyze different instance-dependent cost-sensitive learning approaches on the resulting classification performance in terms of both costs and errors. Therefore, following the presented analysis of the literature, we formulate three key research questions to study the effect of cost-sensitive training using task-specific loss (RQ1), cost-sensitive decision-making (RQ2) and their combination (RQ3). Moreover, we look at the effect of considering costs at an instance level (RQ4). For each question, several hypotheses are proposed.

RQ1. Does instance-dependent cost-sensitive training result in improved performance compared to training without costs?

- $\mathcal{H}1.1$: In terms of *costs*, cost-sensitive training results in better performance compared to training without costs.
- $\mathcal{H}1.2$: In terms of *errors*, cost-insensitive training results in better performance compared to training with costs.

RQ2. Does instance-dependent cost-sensitive thresholding result in improved performance compared to class-dependent thresholding?

- $\mathcal{H}2.1$: In terms of *costs*, instance-dependent cost-sensitive thresholding results in improved performance compared to class-dependent thresholding.
- $\mathcal{H}2.2$: In terms of *costs*, calibrating probabilities results in more effective thresholding.
- $\mathcal{H}2.3$: In terms of *errors*, instance-dependent cost-sensitive thresholding results in improved performance compared to class-dependent thresholding.
- $\mathcal{H}2.4$: In terms of *errors*, calibrating probabilities results in more effective thresholding.

RQ3. Does combining cost-sensitive training and cost-sensitive thresholding result in improved performance compared to either method separately or completely cost-insensitive classification?

- $\mathcal{H}3.1$: In terms of *costs*, combining cost-sensitive training and cost-sensitive thresholding results in improved performance compared to either method separately or completely cost-insensitive classification.
- $\mathcal{H}3.2$: In terms of *errors*, combining cost-sensitive training and cost-sensitive thresholding results in improved cost performance compared to either method separately or completely cost-insensitive classification.

RQ4. Is it beneficial to train with instance-dependent costs instead of class-dependent costs?

- $\mathcal{H}4.1$: In terms of *costs*, using instance-dependent costs results in better performance compared to class-dependent costs.
- $\mathcal{H}4.2$: In terms of *errors*, using instance-dependent costs results in better performance compared to class-dependent costs.

3.1. Experimental design

In this section, we describe the experimental design that is used to answer the proposed research questions empirically. We analyze the effect of different factors in the decision-focused learning framework (see Figure 2 for

Table 3: **Overview of the different models.** These are obtained by combining the different objective functions with the different types of classifiers.

	Logistic regression	**Neural network**	**Gradient boosting**
$\mathcal{L}^{CE}$	logit	net	boost
$\mathcal{L}^{wCE}$	wlogit	wnet	wboost
$\mathcal{L}^{AEC}$	cslogit	csnet	csboost

an overview): cost-sensitive training of models in the first stage, cost-sensitive decision-making in the second stage and the combination of both. Finally, to look at the effect of training with instance-dependent costs, we also compare these models with models trained with class-dependent costs in terms of both the scores and decisions.

3.1.1. Cost-sensitive training

To compare different approaches to learn a predictive model in the first stage, we will compare a traditional, cost-insensitive approach (cross-entropy $\mathcal{L}^{CE}$) with two cost-sensitive task-specific objective functions: an indirect, weighted approach (weighted cross-entropy $\mathcal{L}^{wCE}$) and a direct approach (average expected cost $\mathcal{L}^{AEC}$) (see equations 1, 2 and 3). These are implemented using three different types of classifiers: logistic regression, neural network and gradient boosting. For the neural network, we use a multilayer perceptron with one hidden layer and hyperbolic tangent as activation function. This results in a total of 9 models (see Table 3). For neural networks and gradient boosting, hyperparameter selection is based on the best value of the objective function on a validation set.

These classifiers are frequently adopted in both science and industry, and can be considered as representative of prominent and diverse types of machine learning techniques: they span both linear and nonlinear models, both tree-based and neural-based models, as well as both ensembles and single classifiers. This selection is further motivated by strong performance reported across various benchmarking studies [e.g., 25]. Finally, as all three methodologies optimize an objective function, they allow for a direct and fair comparison of general cost-sensitive learning strategies.

3.1.2. Cost-sensitive decision-making

To consider the effect of cost-sensitive decision-making in the second stage, we compare a range of nine different thresholding strategies: the theoretically optimal instance-dependent cost-sensitive threshold (IDCS) (see equation 4) or the equivalent with calibrated probabilities (IDCS*), as well as their class-dependent variants (CDCS and CDCS*). Calibration is performed with the nonparametric isotonic regression, which has been shown to achieve good results when enough data are available [30]. Furthermore, we include different types of empirical thresholding techniques by finding the best threshold in terms of instance-dependent costs, class-dependent costs and F1 score on a validation set. Finally, we also include using the class-imbalance (CI) ratio $P(Y = 1)$ and the default threshold for binary classification ($t = 0.5$).

To summarize, instance-dependent cost-sensitive learning is analyzed by comparing different objective functions for different classifiers in the first stage and decision-making strategies in the second stage. This allows us to compare

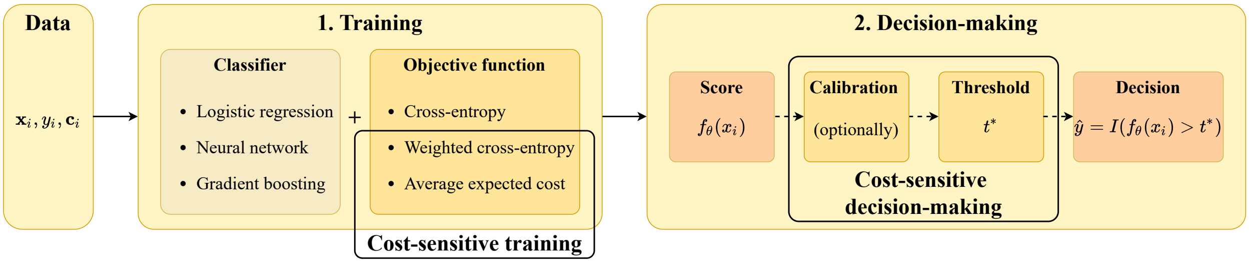


Figure 2: Overview of the experimental design using the two-staged framework. In the first stage, a predictive model is built by training a type of classifier with an objective function, which can be both cost-sensitive or cost-insensitive. In the second stage, the predictions of this model are used to make decisions. Both the scores and decisions are evaluated.

the predict-then-optimize and predict-and-optimize approaches, as well as to analyze different cost-sensitive techniques in each stage. Moreover, we also look at the effect of using instance-dependent costs as opposed to using class-dependent costs.

3.2. Experimental procedure and evaluation metrics

For the empirical evaluation, a 2×5-fold stratified cross-validation procedure is used (see Algorithm 1). This is repeated for each dataset. Using this framework, we conduct two experiments for each model: one with instance-dependent costs and one with class-dependent costs. The full experimental procedure is available in the code.

Algorithm 1: Experimental procedure per dataset

```
Result: Evaluation metrics
Load data;
Initialize cost matrix;
Split data into 5 stratified folds;
for each fold $i \in 1 : 5$ do
    for each repetition $j \in 1 : 2$ do
        Test data = fold $i$;
        Training data = 75% of remaining data;
        Validation data = 25% remaining data;

        # Preprocess data:
        Convert categorical features (using WoE encoding);
        Standardize data: $z = \frac{x-\mu}{\sigma}$;
        if training with class-dependent costs then
            Average cost matrix for training set;
            Average cost matrix for validation set;
        end

        # Train and evaluate models:
        Train models;
        Set decision thresholds;
        Evaluate model outputs and predictions for different thresholds;
    end
end
Summarize evaluation metrics over all folds;
```

We use a variety of metrics to evaluate the models. These can be categorized based on two criteria: whether these incorporate costs (cost sensitivity) and whether they look at probabilities or decisions (threshold dependency). To assess the importance of costs during training independently from the thresholding strategy, we rely on threshold-independent metrics. To compare the different thresholding strategies, we use threshold-dependent metrics.

Several cost-insensitive metrics are used to assess the models' ability to accurately classify instances. First, two threshold-independent metrics are the area under the ROC curve (AUROC) and average precision (AP), which summarize the ROC and precision-recall curves, respectively. The latter may be more informative given the high degree of class imbalance that is typically encountered in cost-sensitive applications [10]. Moreover, the Brier score is used to assess whether the model's outputs are calibrated probabilities. Finally, to evaluate the impact of the decision-making threshold, we use the F1-score.

Moreover, performance is also judged in terms of costs. Again, several threshold-independent metrics are applicable. First, the average expected cost (AEC, see equation 3) is used. Second, Spearman's rank correlation coefficient ρ is used to look at the correlation between probabilities and costs for positive instances. This metric analyzes whether cost-sensitive models prioritize correctly classifying costlier instances. Finally, one cost-sensitive, threshold-dependent metric is also used: cost savings. These compare the total costs incurred by a model to classify by predicting all instances as the cheapest default class (either 0 or 1) [1]:

$$\text{Savings} = \frac{\text{Cost}(f_\theta(\mathbf{x})) - \min\{\text{Cost}(f_0(\mathbf{x})), \text{Cost}(f_1(\mathbf{x}))\}}{\text{Cost}(f_\theta(\mathbf{x}))} \tag{5}$$

The domain of this ratio is $[-\infty, 1]$, where 1 is the perfect model, but when the model does better than predicting the default class, we obtain savings in $]0, 1]$.

To test the statistical significance of the results, we use two types of tests depending on whether we are performing multiple or pairwise comparisons [11]. In the case of multiple comparisons, Friedman tests with Nemenyi post hoc correction are used. These are visualized using critical difference diagrams that show the average rankings (where a lower rank is better). Models that are not connected in this diagram have significantly different mean ranks. For pairwise comparison, Wilcoxon signed-rank tests are used. A significance level of 5% is used primarily, except where both 5% and 10% are used when indicated.

4. Empirical results

In this section, the empirical results are presented. First, the data and corresponding cost matrices are described. Second, the results are presented, and these findings are used to answer the proposed research questions.

4.1. Data

The data are from a diverse set of classification tasks where costs are instance-dependent: fraud detection, direct marketing, customer churn and credit scoring (see Table 4). All datasets are publicly available (see appendix A). In each dataset, there is some degree of class imbalance with the positive class being the minority, though some cases are more extreme than others. The cost matrices depend on the application area and are adopted from earlier work (for an overview, see Table 5). The idea behind these is provided below.

Table 4: **Overview of the datasets**. Size (N), dimensionality (D) and degree of class imbalance (% Pos) are shown.

Application	**Dataset**	**Abbr.**	N	D	**% Pos**
Fraud detection	Kaggle Credit Card Fraud	*KCCF*	282,982	29	0.16
	Kaggle IEEE Fraud Detection	*KIFD*	590,540	431	3.50
Direct marketing	KDD Cup 1998	*KDD*	191,779	22	5.07
	UCI Bank Marketing	*UBM*	45,211	15	11.70
Churn prediction	Kaggle Telco Customer Churn	*KTCC*	7,032	19	26.58
	TV Subscription Churn	*TSC*	9,379	46	4.79
Credit scoring	Kaggle Give Me Some Credit	*GMSC*	112,915	10	6.74
	UCI Default of Credit Card Clients	*DCCC*	30,000	23	22.12
	VUB Credit Scoring	*VCS*	18,917	16	16.95

Fraud detection In fraud detection, a positive prediction triggers an investigation that has a fixed cost c_f, while a missed fraudulent transaction incurs a cost equal to its amount A_i (see Table 5a). For both datasets, c_f is set to 10 following [20].

Direct marketing A similar reasoning applies here: any direct marketing action results in a fixed cost c_f, and missing a potential success incurs an instance-dependent cost (see Table 5b). Whereas *KDD* uses the amount A_i and $c_f = 0.68$ following both [47] and [34], *UBM* instead uses the expected interest given A_i and $c_f = 1$, following [2].

Customer churn For customer churn prediction, c_i^{FP} and c_i^{FN} are set at 2 and 12 times the monthly amount A_i for *KTCC*, respectively, following [34] (see Table 5c). For *TSC*, the cost matrix provided with the dataset is used (not shown here, see [3]).

Credit scoring Finally, for credit scoring, the costs of a FP and FN are calculated following [1] with both a function of the loan amount A_i.

Table 5: **Cost matrices for the different application areas.** For each application, we present the different costs associated with different outcomes. A_i, Int_i, c_i^{FN} and c_i^{FP} represent instance-dependent costs, and c_f is a fixed cost.

(a) Fraud detection

		y	
		0	1
$\hat{y}$	0	0	A_i
	1	c_f	c_f

(b) Direct marketing

		y	
		0	1
$\hat{y}$	0	0	A_i/Int_i
	1	c_f	c_f

(c) Customer churn

		y	
		0	1
$\hat{y}$	0	0	$12A_i$
	1	$2A_i$	0

(d) Credit scoring

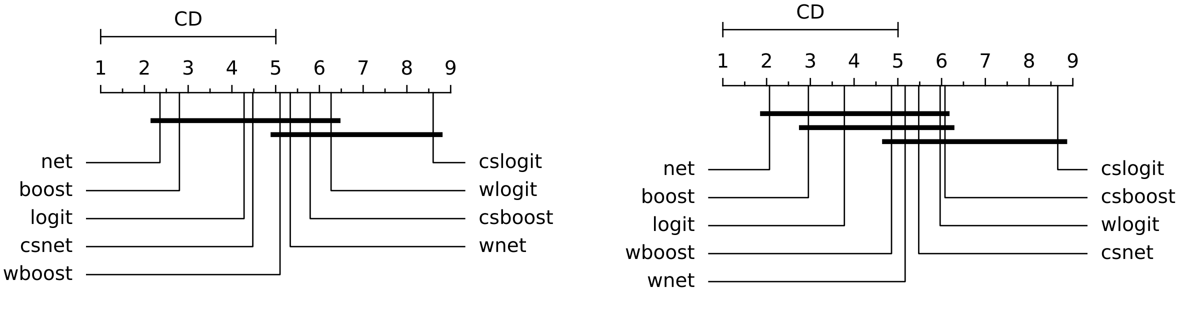
		y	
		0	1
$\hat{y}$	0	0	c_i^{FN}
	1	c_i^{FP}	0

4.2. Results

In this section, we report the results of the experiments and discuss the implications for the research questions of this study. First, we compare training with the three different objective functions with threshold-independent metrics. Second, we use threshold-dependent metrics to analyze the different thresholding strategies for the different models. Third, we compare the results of this analysis by training with class-dependent costs. Complete results on the different experiments can be found in the digital appendix.

4.2.1. Cost-sensitive training

We start by looking at two traditional evaluation metrics: the area under the ROC curve (AUROC) and the average precision (AP) (see Figure 3). The cost-insensitive methodologies (net, boost, logit) have the best scores for both of these metrics, although only the difference with the worst classifier, cslogit, is significant at a 5% level.



(a) Area under the ROC curve (AUROC)

(b) Average precision (AP)

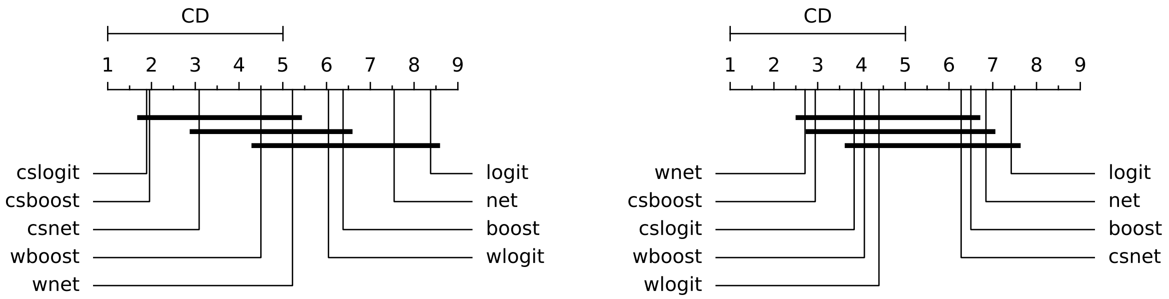
Figure 3: Cost-insensitive metrics: critical difference diagrams for the AUROC and AP

The AUROC and AP do not consider costs. Therefore, the next metric is the average expected cost (AEC) (see Figure 4a). Unsurprisingly, the best performing classifiers are those directly optimizing this expected cost. In almost all cases, the differences with the cost-insensitive models are statistically significant at the 5% level. Models trained with a cost-weighted objective function perform worse but still better than the cost-insensitive classifiers.

Similarly, Spearman's rank correlation coefficient ρ is used to compare the correlation between the predicted probabilities and costs for the positive instances (see Figure 4b). The cost-sensitive classifiers perform better on average. For this metric, there does not seem to be a substantial difference between training with weighted cross-entropy and the average expected cost.

The tradeoff between minimizing costs or errors seems to more strongly affect the least flexible classifier, logistic regression. Cslogit has the worst performance for the cost-insensitive metrics but the best performance in terms of AEC. In contrast, logit performs well for AUROC and AP but is the worst in terms of the cost-sensitive metrics. This indicates that there is a larger tradeoff between minimizing costs and errors for a more inflexible, linear model compared to the neural networks and gradient boosting.

In conclusion, cost-sensitive models perform worse on average for traditional, cost-insensitive evaluation metrics but better in terms of cost-sensitive metrics. This indicates that minimizing errors or minimizing costs are two fundamentally different objectives. Moreover, the type of objective function seems to be more important than the type



(a) Average expected cost (AEC)

(b) Spearman's rank correlation coefficient ρ

Figure 4: Cost-sensitive metrics: critical difference diagrams for the AEC and Spearman ρ.

of classifiers, as neither logistic regression, neural networks nor gradient boosting consistently outperform another category.

4.2.2. Cost-sensitive decision-making

To analyze the different approaches to cost-sensitive decision-making, we first compare the savings (see Table 6) and then the F1 scores (see Table 8) for each model and thresholding strategy averaged across all datasets. This also allows us to analyze the effect of using a cost-sensitive objective function in the first stage on the quality of the decisions in the second stage.

Table 6: **Savings: comparison of the different thresholding strategies (averaged across all datasets).** Best and second-best result for each model are denoted in **bold** and *italic*.

	IDCS	IDCS*	CDCS	CDCS*		Empirical		CI	0.5
					ID	CD	F1		
logit	**0.36**	*0.35*	0.29	0.30	0.30	0.30	0.24	0.09	0.06
wlogit	0.14	*0.36*	0.06	**0.37**	**0.37**	**0.37**	0.34	-1.33	**0.37**
cslogit	**0.38**	*0.37*	**0.38**	**0.38**	**0.38**	**0.38**	*0.37*	**0.38**	**0.38**
net	**0.41**	*0.40*	0.35	0.35	0.35	0.35	0.29	0.20	0.12
wnet	0.13	0.36	0.13	*0.39*	*0.39*	*0.39*	0.35	-0.66	**0.40**
csnet	*0.36*	**0.39**	0.34	0.35	0.35	0.34	0.29	0.34	0.34
boost	**0.41**	*0.40*	0.35	0.36	0.36	0.36	0.29	0.32	0.13
wboost	0.30	**0.37**	0.25	*0.36*	*0.36*	*0.36*	0.30	0.23	0.32
csboost	**0.39**	0.36	**0.39**	**0.39**	**0.39**	**0.39**	0.34	**0.39**	*0.38*

In terms of savings (see Table 6), the importance of the decision-making strategy is strongly related to the objective function that is used to train a classifier. For the cost-insensitive models (trained with cross-entropy), it is absolutely crucial to not use the default threshold 0.5 and instead use the instance-dependent cost-sensitive threshold. When a cost-weighted objective function is used, good results can be obtained either when $t = 0.5$, when probabilities are calibrated and a cost-sensitive threshold is used, or when the threshold is tuned empirically. Conversely, the models trained with AEC achieve relatively stable savings across thresholding strategies. In other words, using a task-specific loss function is related to the performance of different decision-making strategies, with the direct approach giving the most consistent results across strategies.

Moreover, the type of decision-making strategy that is used in the first stage, i.e., the threshold, is more important than the type of objective function used to train the predictive model in the first stage. In fact, given that an appropriate threshold is used, it is only beneficial in terms of savings to use a cost-sensitive objective function for the simplest model: logistic regression. For neural networks and gradient boosting, the cost-insensitive models also achieve good results given that the optimal threshold is used. The best savings overall are obtained when a cost-insensitive model is combined with instance-dependent cost-sensitive thresholding.

Calibrating probabilities achieve better results only for either the weighted cross-entropy or for class-dependent cost-sensitive thresholds. In fact, the two best savings are obtained without calibration. For models trained with a normal cross-entropy loss, calibration does not result in a higher Brier score, suggesting that these probabilities were already calibrated (see Table 7). Although the largest improvement of calibration is observed for the models trained with AEC, this only leads to an improvement in terms of savings for csnet. Only the models trained with weighted cross-entropy have a much better performance after calibration.

In terms of savings, it is clearly beneficial to consider costs during decision-making: empirical thresholding with the F1-score, the class imbalance heuristic or $t = 0.5$ can obtain bad results (depending on the objective function). In general, thresholding on an instance level also seems to be favorable to class-dependent thresholding. Finally, both theoretical and empirical thresholding can achieve good results.

Table 7: **Brier score before and after calibration for the different models (averaged across all datasets).** The Brier score of models trained with a cost-sensitive objective function improves considerably, whereas it is stable for the models trained with a cross-entropy loss.

Calibration	logit	wlogit	cslogit	net	wnet	csnet	boost	wboost	csboost
Before	0.07	0.16	0.24	0.07	0.16	0.23	0.07	0.13	0.19
After	0.07	0.07	0.08	0.07	0.07	0.07	0.07	0.08	0.08
Difference	0.00	-0.09	-0.17	0.00	-0.09	-0.16	0.00	-0.05	-0.11

The best thresholding strategies in terms of F1 scores do not necessarily achieve the lowest costs (see Table 8). This emphasizes that there is also a clear difference between minimizing errors and costs in the decision-making stage. The best results in terms of the F1 score are obtained when the threshold is tuned empirically to maximize this metric. Again, calibrating probabilities is only beneficial for the models trained with weighted cross-entropy. For these models, however, empirical thresholding is more effective than theoretical thresholding. Finally, note how using $t = 0.5$ achieves relatively good results in terms of the F1 score, even though it does not result in large cost savings.

Table 8: **F1 Score: comparison of the different thresholding strategies (averaged across all datasets).** Best and second-best result for each model are denoted in **bold** and *italic*.

	IDCS	IDCS*	CDCS	CDCS*		Empirical		CI	0.5
					ID	CD	F1		
logit	0.33	0.33	*0.39*	*0.39*	*0.39*	*0.39*	**0.42**	0.30	0.31
wlogit	0.23	0.30	0.26	*0.39*	0.38	0.38	**0.41**	0.21	*0.39*
cslogit	0.36	0.31	**0.39**	**0.39**	*0.38*	*0.38*	**0.39**	**0.39**	**0.39**
net	0.36	0.36	*0.42*	*0.42*	*0.42*	*0.42*	**0.47**	0.34	0.37
wnet	0.23	0.29	0.26	*0.39*	0.38	*0.39*	**0.43**	0.22	*0.39*
csnet	0.39	0.35	0.41	0.42	0.41	0.42	**0.45**	0.41	*0.43*
boost	0.39	0.36	*0.45*	0.44	0.43	0.44	**0.48**	0.40	0.40
wboost	0.31	0.32	0.35	0.41	0.41	0.41	**0.45**	0.33	*0.43*
csboost	0.35	0.28	0.36	0.36	0.36	0.36	**0.40**	0.36	*0.38*

4.2.3. Is it beneficial to train with instance-dependent costs instead of class-dependent costs?

First, we look at the effect of using instance-dependent costs during training as opposed to training with class-dependent costs in terms of cost-insensitive metrics (see Tables 9 and B1). Although the results are fairly similar for the two settings, training with class-dependent costs achieves better results for these metrics for almost all cases. Based on this observation, it can be concluded that training with instance-dependent costs may be disadvantageous in terms of errors.

Table 9: **Instance-dependent or class-dependent costs: cost-insensitive metrics per model.** Significantly better results are denoted in **bold** (5%) and *italic* (10%).

Metric	Costs	wlogit	cslogit	wnet	csnet	wboost	csboost
AUROC	ID	0.76	0.72	0.76	0.77	0.77	0.76
	CD	**0.77**	*0.73*	**0.78**	0.77	**0.78**	**0.79**
AP	ID	0.38	0.27	0.40	*0.38*	0.42	0.38
	CD	**0.42**	0.27	**0.45**	0.36	0.45	**0.44**
F1 IDCS	ID	0.23	0.36	0.23	0.39	0.31	0.35
	CD	0.24	0.37	0.25	0.39	0.32	**0.38**
F1 IDCS*	ID	0.30	0.31	0.29	0.35	0.32	0.28
	CD	**0.34**	0.32	**0.35**	0.35	0.35	**0.35**
F1 CDCS	ID	0.26	0.39	0.26	0.41	0.35	0.36
	CD	0.27	0.39	0.27	**0.42**	0.34	*0.41*
F1 CDCS*	ID	0.39	0.39	0.39	0.42	0.41	0.36
	CD	**0.41**	0.39	**0.42**	**0.42**	0.43	**0.43**
F1 Emp ID	ID	0.38	0.38	0.38	0.41	0.41	0.36
	CD	**0.41**	0.39	**0.42**	0.42	0.43	**0.43**
F1 Emp CD	ID	0.38	0.38	0.39	0.42	0.41	0.36
	CD	**0.41**	0.39	**0.42**	*0.42*	0.43	**0.43**
F1 Emp F1	ID	0.41	0.39	0.43	0.45	0.45	0.40
	CD	**0.44**	0.39	**0.46**	0.45	0.46	**0.47**
F1 CI	ID	0.21	0.39	*0.22*	0.41	0.33	0.36
	CD	0.21	0.39	0.21	*0.42*	0.33	*0.40*
F1 0.5	ID	0.39	0.39	0.39	0.43	0.43	0.38
	CD	0.41	0.39	**0.42**	0.43	0.45	**0.44**

Next, we consider cost-sensitive metrics (see Tables 10 and B2). Here, training with instance-dependent costs achieves comparatively better results. Using instance-dependent costs consistently leads to lower average expected costs (though the difference is not always significant). Additionally, in terms of Spearman's ρ, it is better for all models, and this difference is significant except for csnet. In terms of savings, instance-dependent costs are better on average, although not consistently.

5. Discussion

In this section, we draw upon the results of the empirical evaluation to answer the four key research questions that were previously proposed. An overview of findings per research question can be found in Table 11.

Does cost-sensitive training result in improved performance compared to training without costs? Cost-sensitive training achieves better performance in terms of cost-sensitive, but performs worse in terms of cost-insensitive metrics. Cost-sensitive objectives result in a lower expected cost and learn to prioritize costly instances based on the Spearman correlation between model outputs and costs for positive instances. This is observed for both cost-sensitive objective functions: the indirect, weighted approach (weighted cross-entropy) and the direct approach (average expected cost). These findings illustrate that there is a tradeoff between minimizing costs or minimizing errors during training, indicating that these are two fundamentally different objectives.

Does instance-dependent cost-sensitive thresholding result in improved performance compared to class-dependent thresholding? In terms of costs, cost-sensitive thresholding at an instance level was observed to be the most successful decision-making strategy, outperforming all other decision-making thresholds. Calibrating probabilities

Table 10: **Instance-dependent or class-dependent costs: cost-sensitive metrics per model.** Significantly better results are denoted in **bold** (5%) and *italic* (10%). AEC is normalized between 0 and 1 per dataset (lower is better).

Metric	Costs	wlogit	cslogit	wnet	csnet	wboost	csboost
AEC	ID	*0.56*	**0.07**	*0.47*	0.18	*0.41*	**0.06**
	CD	0.68	0.25	0.59	0.18	0.48	0.21
Spearman's ρ	ID	**0.09**	**0.11**	**0.16**	-0.06	**0.13**	**0.23**
	CD	-0.10	-0.05	-0.10	-0.07	-0.07	-0.10
Savings IDCS	ID	0.14	**0.38**	0.13	0.36	0.30	0.39
	CD	0.14	0.32	0.17	0.36	0.30	0.38
Savings IDCS*	ID	0.36	*0.37*	0.36	0.39	0.37	0.36
	CD	*0.38*	0.36	**0.40**	0.39	0.38	*0.39*
Savings CDCS	ID	0.06	**0.38**	0.13	0.34	0.25	**0.39**
	CD	0.03	0.31	0.08	0.34	0.22	0.35
Savings CDCS*	ID	**0.37**	**0.38**	**0.39**	0.35	0.36	0.39
	CD	0.32	0.31	0.34	0.35	0.34	0.35
Savings Emp ID	ID	**0.37**	**0.38**	**0.39**	0.35	0.36	**0.39**
	CD	0.32	0.31	0.34	0.34	0.34	0.35
Savings Emp CD	ID	**0.37**	**0.38**	**0.39**	0.34	0.36	0.39
	CD	0.32	0.31	0.34	0.34	0.34	0.35
Savings Emp F1	ID	**0.34**	**0.37**	**0.35**	0.29	**0.30**	**0.34**
	CD	0.27	0.31	0.27	0.29	0.26	0.27
Savings CI	ID	-1.33	**0.38**	**-0.66**	0.34	0.23	**0.39**
	CD	-1.59	0.31	-0.81	0.34	0.19	0.34
Savings 0.5	ID	**0.37**	**0.38**	**0.40**	0.34	0.32	*0.38*
	CD	0.33	0.31	0.35	0.34	0.29	0.34

was only beneficial when the weighted cross-entropy or a class-dependent threshold was used. The differences in best-performing thresholds when optimizing for savings or F1 score illustrate that minimizing errors and costs are also two different objectives in the decision-making stage.

Does combining cost-sensitive training and cost-sensitive thresholding result in improved performance compared to either method separately or completely cost-insensitive classification? Combining cost-sensitive training and decision-making did not necessarily achieve better results. In fact, the best savings were obtained by training with a cost-insensitive objective function and using the instance-dependent cost-sensitive threshold. This illustrates that the type of thresholding is more important than the type of objective function in terms of costs.

Is it beneficial to train with instance-dependent costs instead of class-dependent costs? In terms of both training and thresholding, using instance-dependent instead of class-dependent costs was observed to achieve better results for cost-sensitive metrics, but worse results for traditional cost-insensitive metrics. Specifically, not using costs at all is preferential for minimizing errors, using instance-dependent costs is optimal for minimizing costs, and using class-dependent costs lies somewhere between these two.

6. Conclusion

In this paper, we presented a focused review and empirical analysis of instance-dependent cost-sensitive classification. Conceptually, we reviewed cost-sensitive classification through the lens of predict-and-optimize and differentiated between different methods for both cost-sensitive training and decision-making. Several key methodologies were implemented for different classifiers, and the resulting models were compared empirically on nine datasets from

Table 11: **Summary of the key findings.** We present a summary of the results per research question and hypothesis. Performance is judged in terms of costs and errors. Each question is answered with yes (✓), no (✗) or inconclusive (?).

Research question	Costs	Errors
1. Does instance-dependent cost-sensitive training result in improved performance compared to training without costs?		
Instance-dependent cost-sensitive training results in better performance compared to training without costs.	✓	✗
2. Does instance-dependent cost-sensitive thresholding result in improved performance compared to class-dependent thresholding?		
Instance-dependent cost-sensitive thresholding results in improved performance compared to class-dependent thresholding.	✓	✗
Calibrating probabilities results in more effective thresholding.	?	?
3. Does combining cost-sensitive training and cost-sensitive thresholding result in improved performance compared to either method separately or completely cost-insensitive classification?		
Combining cost-sensitive training and cost-sensitive thresholding results in improved performance compared to either method separately or completely cost-insensitive classification.	✗	✗
4. Is it beneficial to train with instance-dependent costs instead of class-dependent costs?		
Using instance-dependent costs results in better performance compared to class-dependent costs.	✓	✗

different application areas. Based on the experimental results obtained from this large-scale benchmarking experiment, we answered four research questions (see Table 11 for an overview).

These findings stress the importance of considering the right objective for an application. Optimizing for accuracy can be detrimental to a classifier's performance when the actual objective is to minimize costs, which is the case in a large variety of business applications. For this, it is especially important to consider the right type of thresholding strategy. Overall, a conceptually simple yet well-performing strategy is to first train a cost-insensitive model and only introduce costs in a second stage through instance-dependent thresholding. In other words, using a task-specific loss in the first stage does not result in better decisions in the second stage, given that the optimal decision-making policy is used.

These results correspond with empirical research in the class-dependent setting: two works compared cost-sensitive boosting algorithms with cost-sensitive thresholding and found the latter to be the more effective strategy [49, 31]. Nevertheless, theoretical results in the class-dependent setting suggest that cost-sensitive training can be optimal under certain conditions. For example, under model misspecification, a cost-sensitive objective function [12] can be preferential to theoretical thresholding. Consequently, a direction for future research is to extend the theoretical analysis from the class-dependent setting toward instance-dependent costs. Additionally, it will be interesting to investigate the influence of the characteristics of the cost distribution and cost matrix on the performance of instance-dependent cost-sensitive training and decision-making methods. By sharing our code, we hope to encourage and facilitate further research on instance-dependent cost-sensitive learning.

Acknowledgments

This work was supported by the BNP Paribas Fortis Chair in Fraud Analytics and FWO research project G015020N. The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation – Flanders (FWO) and the Flemish Government – department EWI.

A. Data

The data sets that are used in the experiments presented in this paper are publicly available online (names are clickable links):

- Kaggle Credit Card Fraud [9]
- Kaggle IEEE Fraud Detection
- UCI KDD98 Direct Mailing
- UCI Bank Marketing [29]
- Kaggle Telco Customer Churn
- TV Subscription Churn [3]
- Kaggle Give Me Some Credit
- UCI Default of Credit Card Clients [46]
- VUB Credit Scoring [33]

B. Training with instance-dependent or class-dependent costs: results per dataset

Detailed results comparing training with instance-dependent and class-dependent costs per dataset can be found in Tables B1 and B2.

Table B1: **Instance-dependent or class-dependent costs: cost-insensitive metrics per dataset.** Significantly better results are denoted in **bold** (5%) and *italic* (10%).

Metric	Costs	KCCF	GMSC	KIFD	KTCC	KDD	TSC	UBM	DCCC	VCS
AUC	ID	0.96	0.81	0.89	0.82	0.51	0.61	0.73	0.72	0.76
	CD	0.96	**0.81**	*0.90*	0.82	**0.53**	0.62	**0.76**	*0.75*	*0.77*
AP	ID	0.72	0.30	0.45	0.61	0.05	0.08	0.29	0.46	0.38
	CD	0.77	0.31	*0.51*	0.60	**0.06**	0.08	*0.37*	0.49	0.39
Brier score	ID	0.00	0.16	0.05	0.25	0.40	0.19	0.17	0.21	0.24
	CD	0.00	**0.17**	0.06	0.25	**0.42**	0.19	0.16	**0.23**	0.25
F1 IDCS	ID	0.41	0.22	0.27	0.54	0.10	0.12	0.31	0.43	0.40
	CD	*0.46*	0.22	0.28	*0.55*	0.10	**0.12**	0.33	*0.44*	0.40
F1 IDCS*	ID	0.40	0.30	0.28	0.57	0.06	0.12	0.28	0.37	0.40
	CD	*0.49*	*0.31*	**0.38**	0.57	0.06	0.12	*0.31*	*0.44*	**0.43**
F1 CDCS	ID	0.63	0.22	0.27	0.54	0.10	0.12	0.32	0.44	0.40
	CD	0.74	0.22	0.27	*0.55*	0.10	*0.12*	0.32	0.45	0.39
F1 CDCS*	ID	0.75	0.31	0.40	0.57	0.10	0.13	0.36	0.48	0.43
	CD	0.81	0.31	0.45	0.57	**0.10**	0.14	**0.42**	**0.51**	**0.44**
F1 Emp ID	ID	0.70	*0.32*	0.40	0.56	0.10	0.13	0.36	0.49	0.44
	CD	**0.81**	0.31	0.45	0.57	**0.10**	*0.14*	**0.42**	*0.51*	0.44
F1 Emp CD	ID	0.75	0.31	0.40	0.57	0.10	0.13	0.36	0.48	0.43
	CD	0.81	0.31	0.45	0.57	**0.10**	*0.14*	**0.42**	**0.51**	*0.44*
F1 Emp F1	ID	0.77	0.39	0.49	0.61	0.10	0.13	0.37	0.49	0.44
	CD	0.82	0.39	0.54	0.61	*0.10*	0.13	**0.43**	*0.53*	**0.46**
F1 CI	ID	0.50	0.22	0.24	0.56	0.10	0.12	0.31	0.45	0.40
	CD	0.53	0.22	0.23	*0.57*	0.10	**0.12**	0.31	0.45	0.40
F1 0.5	ID	0.74	0.33	0.43	0.59	0.10	0.13	0.37	0.49	0.44
	CD	0.81	0.33	0.47	0.59	*0.10*	0.14	*0.41*	*0.51*	*0.45*

Table B2: **Instance-dependent or class-dependent costs: cost-sensitive metrics per dataset.** Significantly better results are denoted in **bold** (5%) and *italic* (10%).

Metric	Costs	KCCF	GMSC	KIFD	KTCC	KDD	TSC	UBM	DCCC	VCS
AEC	ID	0.08	458.90	*2.53*	82.05	*0.72*	60.22	**0.52**	**15674.65**	*0.08*
	CD	0.08	460.81	3.05	81.32	0.72	60.42	0.67	16724.90	0.09
Spearman's ρ	ID	**0.17**	**-0.04**	*0.09*	0.12	**0.03**	-0.35	*0.55*	*0.05*	**0.36**
	CD	-0.07	-0.15	-0.17	0.12	-0.14	**-0.30**	0.18	-0.30	0.11
Savings IDCS	ID	0.54	*0.24*	0.47	0.21	-0.01	-0.09	**0.57**	**0.29**	*0.36*
	CD	0.68	0.23	0.42	0.22	-0.01	-0.09	0.51	0.21	0.34
Savings IDCS*	ID	0.66	0.47	0.60	0.26	*-0.08*	0.07	0.61	0.32	0.42
	CD	*0.70*	0.48	0.61	0.26	-0.09	0.06	0.62	0.36	*0.43*
Savings CDCS	ID	0.62	0.23	0.38	0.21	-0.01	-0.10	**0.45**	**0.22**	*0.34*
	CD	0.65	0.22	0.28	0.22	-0.01	-0.10	0.26	0.16	0.30
Savings CDCS*	ID	0.67	*0.47*	*0.59*	0.26	*-0.01*	0.06	**0.55**	*0.35*	*0.41*
	CD	0.67	0.47	0.50	0.26	-0.01	0.06	0.42	0.29	0.38
Savings Emp ID	ID	0.67	*0.47*	**0.59**	0.26	**-0.01**	0.06	**0.56**	**0.35**	**0.42**
	CD	0.67	0.47	0.50	0.26	-0.02	0.06	0.42	0.29	0.38
Savings Emp CD	ID	0.67	0.47	*0.59*	0.26	*-0.01*	0.06	**0.55**	*0.35*	*0.41*
	CD	0.67	0.47	0.50	0.26	-0.02	0.06	0.42	0.29	0.38
Savings Emp F1	ID	0.66	0.40	*0.52*	0.10	**-0.03**	**0.05**	*0.54*	**0.35**	**0.39**
	CD	0.64	0.39	0.41	0.10	-0.08	0.05	0.38	0.30	0.33
Savings CI	ID	-2.58	0.23	0.24	0.22	-0.01	-0.10	**0.43**	**0.25**	*0.35*
	CD	-3.05	0.23	0.14	0.23	-0.02	-0.10	0.24	0.18	0.31
Savings 0.5	ID	0.66	*0.47*	*0.59*	0.20	*-0.03*	0.06	**0.55**	**0.35**	**0.41**
	CD	0.66	0.47	0.49	0.21	-0.05	0.06	0.43	0.30	0.37

References

[1] Bahnsen, A.C., Aouada, D., Ottersten, B., 2014a. Example-dependent cost-sensitive logistic regression for credit scoring, in: 2014 13th International Conference on Machine Learning and Applications, IEEE. pp. 263–269.

[2] Bahnsen, A.C., Aouada, D., Ottersten, B., 2015a. Example-dependent cost-sensitive decision trees. Expert Systems with Applications 42, 6609–6619.

[3] Bahnsen, A.C., Aouada, D., Ottersten, B., 2015b. A novel cost-sensitive framework for customer churn predictive modeling. Decision Analytics 2, 1–15.

[4] Bahnsen, A.C., Stojanovic, A., Aouada, D., Ottersten, B., 2014b. Improving credit card fraud detection with calibrated probabilities, in: Proceedings of the 2014 SIAM International Conference on Data Mining, SIAM. pp. 677–685.

[5] Brefeld, U., Geibel, P., Wysotzki, F., 2003. Support vector machines with example dependent costs, in: European Conference on Machine Learning, Springer. pp. 23–34.

[6] Chai, X., Deng, L., Yang, Q., Ling, C.X., 2004. Test-cost sensitive naive bayes classification, in: Fourth IEEE International Conference on Data Mining (ICDM'04), IEEE. pp. 51–58.

[7] Chawla, N.V., Cieslak, D.A., Hall, L.O., Joshi, A., 2008. Automatically countering imbalance and its empirical relationship to cost. Data Mining and Knowledge Discovery 17, 225–252.

[8] Chen, X., Gong, C., Yang, J., 2021. Cost-sensitive positive and unlabeled learning. Information Sciences 558, 229–245.

[9] Dal Pozzolo, A., Caelen, O., Johnson, R.A., Bontempi, G., 2015. Calibrating probability with undersampling for unbalanced classification, in: 2015 IEEE Symposium Series on Computational Intelligence, IEEE. pp. 159–166.

[10] Davis, J., Goadrich, M., 2006. The relationship between precision-recall and roc curves, in: Proceedings of the 23rd international conference on Machine learning, pp. 233–240.

[11] Demšar, J., 2006. Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research 7, 1–30.

[12] Dmochowski, J.P., Sajda, P., Parra, L.C., 2010. Maximum likelihood in cost-sensitive learning: Model specification, approximations, and upper bounds. Journal of Machine Learning Research 11.

[13] Domingos, P., 1999. Metacost: A general method for making classifiers cost-sensitive, in: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 155–164.

[14] Donti, P., Amos, B., Kolter, J.Z., 2017. Task-based end-to-end model learning in stochastic optimization, in: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc.

[15] Drummond, C., Holte, R.C., 2000. Exploiting the cost (in) sensitivity of decision tree splitting criteria, in: ICML.

[16] Elkan, C., 2001. The foundations of cost-sensitive learning, in: International joint conference on artificial intelligence, Lawrence Erlbaum Associates Ltd. pp. 973–978.

[17] Fan, W., Stolfo, S.J., Zhang, J., Chan, P.K., 1999. Adacost: misclassification cost-sensitive boosting, in: Icml, Citeseer. pp. 97–105.

[18] Fawcett, T., Provost, F., 1997. Adaptive fraud detection. Data mining and knowledge discovery 1, 291–316.

[19] Granger, C.W., 1969. Prediction with a generalized cost of error function. Journal of the Operational Research Society 20, 199–207.

[20] Höppner, S., Baesens, B., Verbeke, W., Verdonck, T., 2022. Instance-dependent cost-sensitive learning for detecting transfer fraud. European Journal of Operational Research 297, 291–300.

[21] Höppner, S., Stripling, E., Baesens, B., vanden Broucke, S., Verdonck, T., 2020. Profit driven decision trees for churn prediction. European Journal of Operational Research 284, 920–933.

[22] Krawczyk, B., Woźniak, M., Schaefer, G., 2014. Cost-sensitive decision tree ensembles for effective imbalanced classification. Applied Soft Computing 14, 554–562.

[23] Kukar, M., Kononenko, I., et al., 1998. Cost-sensitive learning with neural networks., in: ECAI, Citeseer. pp. 88–94.

[24] Leevy, J.L., Khoshgoftaar, T.M., Bauder, R.A., Seliya, N., 2018. A survey on addressing high-class imbalance in big data. Journal of Big Data 5, 1–30.

[25] Lessmann, S., Baesens, B., Seow, H.V., Thomas, L.C., 2015. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research 247, 124–136.

[26] Lessmann, S., Haupt, J., Coussement, K., De Bock, K.W., 2019. Targeting customers for profit: An ensemble learning framework to support marketing decision-making. Information Sciences .

[27] Li, Y.F., Kwok, J., Zhou, Z.H., 2010. Cost-sensitive semi-supervised support vector machine, in: Proceedings of the AAAI Conference on Artificial Intelligence.

[28] Maldonado, S., Flores, Á., Verbraken, T., Baesens, B., Weber, R., 2015. Profit-based feature selection using support vector machines–general framework and an application for customer retention. Applied Soft Computing 35, 740–748.

[29] Moro, S., Cortez, P., Rita, P., 2014. A data-driven approach to predict the success of bank telemarketing. Decision Support Systems 62, 22–31.

[30] 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.

[31] Nikolaou, N., Edakunni, N., Kull, M., Flach, P., Brown, G., 2016. Cost-sensitive boosting algorithms: Do we really need them? Machine Learning 104, 359–384.

[32] Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., Brunk, C., 1994. Reducing misclassification costs, in: Machine Learning Proceedings 1994. Elsevier, pp. 217–225.

[33] Petrides, G., Moldovan, D., Coenen, L., Guns, T., Verbeke, W., 2020. Cost-sensitive learning for profit-driven credit scoring. Journal of the Operational Research Society , 1–13.

[34] Petrides, G., Verbeke, W., 2021. Cost-sensitive ensemble learning: a unifying framework. Data Mining and Knowledge Discovery , 1–28.

[35] Sahin, Y., Bulkan, S., Duman, E., 2013. A cost-sensitive decision tree approach for fraud detection. Expert Systems with Applications 40, 5916–5923.

[36] Shawe-Taylor, G.K.J., Karakoulas, G., 1999. Optimizing classifiers for imbalanced training sets. Advances in neural information processing systems 11, 253.

[37] Sheng, V.S., Ling, C.X., 2006. Thresholding for making classifiers cost-sensitive, in: AAAI, pp. 476–481.

[38] Stripling, E., vanden Broucke, S., Antonio, K., Baesens, B., Snoeck, M., 2018. Profit maximizing logistic model for customer churn prediction using genetic algorithms. Swarm and Evolutionary Computation 40, 116–130.

[39] Sun, Y., Kamel, M.S., Wong, A.K., Wang, Y., 2007. Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition 40, 3358–3378.

[40] Ting, K.M., 2002. An instance-weighting method to induce cost-sensitive trees. IEEE Transactions on Knowledge and Data Engineering 14, 659–665.

[41] Turney, P.D., 1994. Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. Journal of artificial intelligence research 2, 369–409.

[42] Vapnik, V., 1999. The nature of statistical learning theory. Springer science & business media.

[43] Veropoulos, K., Campbell, C., Cristianini, N., et al., 1999. Controlling the sensitivity of support vector machines, in: Proceedings of the international joint conference on AI, Stockholm. p. 60.

[44] Wang, T., Qin, Z., Zhang, S., Zhang, C., 2012. Cost-sensitive classification with inadequate labeled data. Information Systems 37, 508–516.

[45] Wilder, B., Dilkina, B., Tambe, M., 2019. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 1658–1665.

[46] Yeh, I.C., Lien, C.h., 2009. The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications 36, 2473–2480.

[47] Zadrozny, B., Elkan, C., 2001. Learning and making decisions when costs and probabilities are both unknown, in: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 204–213.

[48] Zadrozny, B., Langford, J., Abe, N., 2003. Cost-sensitive learning by cost-proportionate example weighting, in: Third IEEE international conference on data mining, IEEE. pp. 435–442.

[49] Zheng, J., 2010. Cost-sensitive boosting neural networks for software defect prediction. Expert Systems with Applications 37, 4537–4543.

[50] Zhou, Z.H., Liu, X.Y., 2005. Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on knowledge and data engineering 18, 63–77.