A new perspective on classification: optimally allocating limited resources to uncertain tasks

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

[a] *Department of Decision Sciences and Information Management, KU Leuven, Naamsestraat 69 - bus 3555, Leuven, 3000, Vlaams-Brabant, Belgium*
[b] *Applied Mathematics, University of Antwerp, Middelheimlaan 1, Antwerpen, 2020, Antwerpen, Belgium*

Abstract

A central problem in business concerns the optimal allocation of limited resources to a set of available tasks, where the payoff of these tasks is inherently uncertain. Typically, such problems are solved using a classification framework, where task outcomes are predicted given a set of characteristics. Then, resources are allocated to the tasks predicted to be the most likely to succeed. We argue, however, that using classification to address task uncertainty is inherently suboptimal as it does not take into account the available capacity. We present a novel solution that directly optimizes the assignment's expected profit given limited, stochastic capacity. This is achieved by optimizing a specific instance of the net discounted cumulative gain, a commonly used class of metrics in learning to rank. We demonstrate that our new method achieves higher expected profit and expected precision compared to a classification approach for a wide variety of application areas.
Keywords: Machine learning, Optimal resource allocation, Classification,

*Corresponding author
Email address: toon.vanderschueren@kuleuven.be (Toon Vanderschueren)

1. Introduction

Optimally allocating limited resources is a central problem in economics [1] and operations research [2, 3, 4]. It is often complicated further by uncertainty inherent to the considered problem. On the one hand, future resource capacity may be limited and not known exactly in advance. On the other hand, the tasks that require resources might have uncertain payoff. This situation is commonly encountered in various real-world applications. As a running example, consider the case of credit card fraud detection. Fraud analysts can only investigate a limited number of transactions each day. A priori, it is not known whether investigating a transaction will uncover a fraudulent case. The general challenge is how to optimally allocate limited resources to maximize business pay-off, e.g., how to optimally allocate fraud investigators to suspicious transactions to minimize losses due to fraud. By learning from historical data, machine learning models can assist decision-makers by predicting the most relevant tasks (e.g., transactions) based on their characteristics.

Prior work addresses the problem of uncertain task outcomes via classification [e.g., 5, 6, 7, 8, 9, 10, 11]. The most promising tasks can be identified by estimating the probability of success for each task. The problem of allocating stochastic, limited capacity could then be addressed separately in a second stage, when assignment decisions are made by prioritizing tasks based on the estimated probabilities to result in a successful outcome. In our running example, this strategy would correspond to first predicting which

instances are most likely to be fraudulent, and then investigating the most suspicious transactions. This strategy is commonly used as a decision support tool for fraud detection, but also other domains where similar problems arise, such as direct marketing, churn prediction, or credit scoring. In this article, however, we argue and demonstrate that this approach based on classification models is suboptimal when resources are limited, because a classification model does not take capacity limitations into account. Hence, although only the most promising tasks can be executed, the model focuses equally on accurately predicting probabilities for tasks that are highly unlikely to be successful and, consequently, to be executed.

To tackle this challenge, we propose a novel approach based on learning to rank that simultaneously accounts for both resource and task uncertainty. When resources are limited, we demonstrate that this approach is superior to allocation based on classification. First, we show theoretically how learning to rank can directly optimize the assignment's expected profit given limited, stochastic capacity. By considering the available capacity during optimization, the model focuses on correctly ranking the most promising tasks, proportional to their likelihood of being processed under limited capacity. Second, while instances are processed individually in classification, learning to rank explicitly considers a task's relevance in comparison to the other available tasks. The benefit of this approach is that we only care about relative positions in the ranking, corresponding to the need to prioritize tasks relative to each other.

Our contributions are threefold. First, we formalize the problem of allocating limited, stochastic resources to uncertain tasks by framing it as an

assignment problem. Second, we propose a novel, integrated predict-and-optimize approach to solve this problem based on learning to rank. We contrast our approach with a two-stage predict-then-optimize framework that first uses a classification model to predict task outcomes and then solves the assignment problem using the predicted task probabilities. Third, we compare both methods empirically using various real life data sets from different application areas.

2. Related work

The proposed solution in this paper relates to prior work on uncertainty in assignment problems, predict-and-optimize, classification, and learning to rank. In this section, we briefly introduce each line of work, describe its relationship to our contribution, and clarify the remaining research gap that our work aims to address.

2.1. Uncertainty in assignment problems

Optimal allocation of resources and decision-making under uncertainty are key problems in operations research [2, 3]. In this work, we consider an assignment problem. This is a general problem formulation in which the goal is to find an optimal matching of workers and tasks subject to certain constraints. This type of problem has been analyzed extensively [12] and applied to a diverse range of tasks [e.g., 13, 14]. Moreover, various extensions consider different sources of uncertainty: uncertain worker capacity, uncertain task presence (i.e., outcomes), or uncertain task-worker profits [15, 16, 17]. This work focuses on a specific type of linear assignment problem, in which we simultaneously address two sources of uncertainty: uncertain capacity

and uncertain task success. However, instead of assuming that task success follows a probability distribution, we use a predictive model to estimate it. Although our aim is similar to Johari et al. [18], they consider an online setting, where workers arrive and depart over time with uncertainty, with a focus on trading-off exploration and exploitation. In contrast, we assume that the worker capacity follows a known, static probability distribution. Moreover, they consider fixed job types with certain outcomes, while we learn these outcomes using a predictive model. In general, our work is different from most work in this category as we aim to simultaneously tackle the prediction of task success as well as the optimization of the assignment problem, while most work in this category is limited to the optimization.

2.2. Predict-and-optimize

The intersection of operations research and machine learning has increasingly drawn the attention of researchers from both fields [19, 20]. In particular, recent work on predict-and-optimize is relevant [21, 22, 23]. The central aim in predict-and-optimize is to align a predictive model more closely with the downstream decision-making context [24]. This is achieved by fusing the prediction and optimization phases and training the model in an end-to-end manner, with the aim of obtaining higher quality decisions [25]. Ranking specifically has been studied in this context. Demirović et al. [26] use ranking to solve a ranking problem with uncertainty in the objective function–similar to task uncertainty in our work. However, in contrast to this work, they do not account for uncertainty in the constraint. Moreover, their method is limited to pairwise ranking, whereas we optimize a listwise objective, allowing us to consider the stochastic capacity in the optimization of the model.

Demirović et al. [27] are limited to linear predictive models. In contrast, our method is compatible with a variety of linear and non-linear machine learning algorithms. Their analysis considers more general optimization problems with uncertainty in the objective function. Conversely, our proposed solution is tailor-made to this specific problem setting, allowing us to use the problem structure in our solution. In general, most work in predict-and-optimize does not account for uncertainty in the constraints or optimization problem [28].

2.3. Classification

Classification is a task in machine learning where the goal is to predict the class of an instance given its characteristics. For instance, classifying a task as either successful or not is a binary classification problem. Existing work typically considers the applications in this paper as classification problems, e.g., fraud detection [10, 11], credit scoring [6, 9], direct marketing [5] and customer churn prediction [7, 8]. Moreover, to align the models more closely with the decision-making context, cost-sensitive classification has been used [29, 30, 31, 32]. Cost-sensitive methodologies incorporate the costs of different decisions into the optimization or use of predictive models [33, 34, 35]. Cost-sensitive variants have been proposed for different classification models, such as logistic regression and gradient boosting [29, 32]. Nevertheless, these consider a different setting: classify instances. Conversely, our work aims to prioritize instances, to process given limited worker capacity. The output of a classification model is often used to rank instances, reflected by widely used evaluation metrics that analyze this ranking, such as the receiver operating characteristics curve and precision–recall curve [36]. However, in contrast to our work, these approaches do not consider the available capac-

ity during optimization of the models. Although limited capacity has been acknowledged in the literature (e.g., in fraud detection [37], direct marketing [38] or churn prediction [39]), no existing solution explicitly addresses this issue. Shifman et al. [40] consider a cost-sensitive classification problem with resource constraints. However, in contrast to our work, they consider misclassification costs to be unknown and do not consider uncertainty in the capacity constraint.

2.4. Learning to rank

In learning to rank, the goal is to predict the order of instances relative to each other, based on their characteristics. Although learning to rank originated in the field of information retrieval, it is a general framework that has been applied to a variety of problems that have traditionally been solved with classification models, such as software defect prediction [41], credit scoring [42] and uplift modeling [43]. Moreover, similar to cost-sensitive classification, the learning to rank framework has been extended to incorporate costs of instances to align the optimization of the model more closely with the resulting decisions [44]. However, our approach is the first to explicitly consider the available capacity during the optimization of the ranking model.

3. Problem formulation

This work addresses the problem of optimally assigning limited and stochastic resources to tasks with uncertain outcomes to maximize the expected profit. In our running example of fraud detection, the goal would be to uncover fraudulent transactions by having fraud investigators look at them, with the aim of minimizing that day's losses due to fraud. On the one hand,

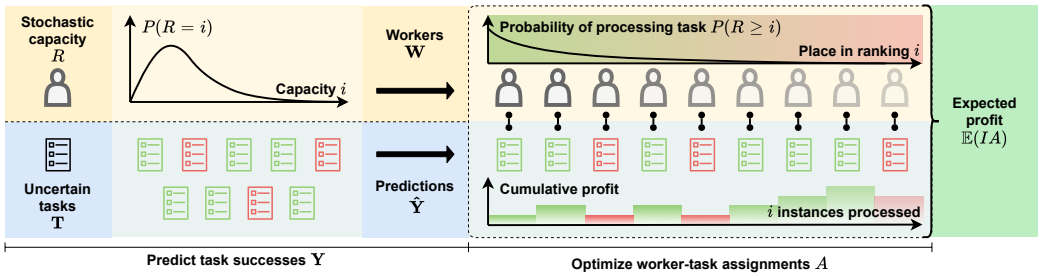


Figure 1: **Problem overview.** Our setting concerns a type of linear assignment problem with two sources of uncertainty: stochastic worker capacity and uncertain task outcomes. To account for stochastic capacity in the assignment problem, the capacity distribution is converted to workers with decreasing processing probabilities. Task outcomes are also uncertain and need to be predicted. The key objective is to assign workers to tasks to maximize the resulting expected profit.

there is task uncertainty. Before investigating a transaction, the outcome of the investigation is uncertain–though this could be estimated based on historical data. On the other hand, there is an uncertain resource constraint. The availability of fraud investigators is uncertain, as well as their productivity on that day. Using historical data, we assume that a worker capacity distribution can be estimated. In the following, we formalize this problem as a general optimization problem.

In this section, we formalize this setting as a linear assignment problem, in which the goal is to optimally assign workers to tasks, where both workers and tasks are sources of uncertainty. The exact number of workers is uncertain at the time when resources need to be allocated, but we assume it is governed by a known probability distribution. In practice, this distribution can be estimated from historical data on the available resources or based on domain knowledge. Alternatively, a deterministic capacity can be con-

sidered. Second, task outcomes are also uncertain and need to be predicted using historical data on similar tasks. A graphical overview of the problem is shown in Figure 1. In the following, we introduce and formally define each element of the assignment problem.

Stochastic capacity

The available resources or number of workers W is a discrete random variable described by a known probability distribution: $W \sim Dist$. In this work, we consider a common situation where the expected capacity is smaller than the number of available tasks: $\mathbb{E}(W) \ll N$. In expectation, the stochastic capacity can be converted to a sequence of N workers with monotonically decreasing expected success rates. Each rate w_j equals the worker's probability of being available given $W \sim Dist$ and is described by the complementary cumulative probability distribution function: $w_j = P(W \geq j) = 1 - F_W(j)$. This yields a monotonically decreasing sequence of N worker success rates $\mathbf{W} = \begin{pmatrix} w_1 & \dots & w_N \end{pmatrix} = \{1 - F_W(j)\}_{j=1}^{N}$ with $w_1 \geq \dots \geq w_N$. Given $\mathbb{E}(W) \ll N$, we expect that most tasks will not be executed and most w_j will be (close to) zero. This formulation will allow us to optimize the expected objective in section 4.

Uncertain tasks

There is also uncertainty regarding task outcomes. To address this uncertainty, we predict it using historical data on similar tasks. Let $\mathcal{T} = (\mathcal{X}, \mathcal{Y}, \mathcal{V})$ be the domain of all possible tasks $t_i = (\mathbf{x}_i, y_i, \mathbf{v}_i)$, where $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$ is a set of characteristics and $y_i \in \mathcal{Y} = \{0, 1\}$ is a binary label equal to 1 if the task is successful and 0 otherwise. Moreover, $\mathbf{v}_i = \{v_i^+, v_i^-\} \in \mathcal{V} \subset \mathbb{R}^2$ denotes

Symbol	Definition (with an example for fraud detection)
W	Stochastic worker capacity (number of tasks processed by fraud specialists)
$\mathbf{W}$	Vector of worker probabilities w_j with $w_j = P(W \geq j)$
T	Number of tasks (transactions considered)
R	Task rewards r_i (transaction payoff, i.e., fraud amount intercepted - processing cost)
Y	Task outcome y_i (fraudulent or legitimate)
A	Assignment matrix a_{ij} (which transactions fraud specialists should investigate)
$\mathbf{v}$	Payoff when executing a task
$\mathbf{c}$	Cost matrix
x	Task characteristics (time and place where transaction was made)
f_θ	Predictive model
π	Permutation of instances, i.e., a ranking

Table 1: **Notation table.** We give an overview of the notation used in this work. For each symbol, we give both the general name and its role in our running example of fraud detection.

the payoff if the task is executed, with v_i^+ if task i was successful ($y_i = 1$) and v_i^- otherwise. A task's reward is defined as $r_i = y_i v_i^+ + (1 - y_i)v_i^-$. We have N available tasks to be allocated $\mathbf{T} = \{(\mathbf{x}_i, y_i, \mathbf{v}_i) : i = 1, \ldots, N\}$, although y_i is unknown when resources need to be allocated. Given historical data, a (deterministic) predictive model can estimate task outcomes y_i resulting in N predictions.

Matching workers and tasks

Workers and tasks can then be combined in an expected profit matrix $P = \left(p_{ij}\right)$, where $p_{ij} = r_i w_j$ is the profit of assigning task i to worker j for $i, j = 1, \ldots, N$. Given P, the goal is to find the optimal assignment matrix $A = \left(a_{ij}\right)$, where $a_{ij} = 1$ if worker i is assigned to task j and 0 otherwise, for $i, j = 1, \ldots, N$. This results in the following linear assignment problem:

$$\text{maximize} \sum_{i=1}^{N} \sum_{j=1}^{W} a_{ij} r_i \tag{1}$$

$$\text{subject to} \sum_{i=1}^{N} a_{ij} \leq 1 \qquad i = 1, \ldots, N; \tag{2}$$

$$\sum_{j=1}^{W} a_{ij} = 1 \qquad j = 1, \ldots, W; \tag{3}$$

$$a_{ij} \in \{0, 1\} \quad i = 1, \ldots, N;\ j = 1, \ldots, W; \tag{4}$$

$$W \sim Dist \tag{5}$$

where conditions 2 and 3 specify that each task is assigned to exactly one worker and vice versa; condition 4 imposes absolute assignments by restricting a_{ij} to 0 or 1. Condition 5 specifies that the resource capacity or number of workers is described by a known probability distribution *Dist*.

4. Methodology

We present two approaches for the problem presented in Section 3. On the one hand, a two-stage predict-then-optimize framework can be used. In the first stage, we predict the task successes $\hat{\mathbf{Y}}$. Here, we show how different types of classification objectives can be used to predict task success. In the second stage, we optimize the assignment of tasks to workers to obtain an assignment matrix A. For this, we provide an analytical solution and prove its optimality. On the other hand, we present an integrated predict-and-optimize framework for prediction and optimization by leveraging learning to rank techniques.

4.1. Two-stage predict-then-optimize

This section presents a conventional two-stage approach for solving the problem. In the first stage, a classification model predicts each task's probability of success. Existing approaches in classification can be used to optimize this model for either accuracy or profit [32]. In the second stage, tasks are assigned to workers based on these predicted probabilities. We present a straightforward procedure for this assignment and prove its optimality.

4.1.1. Predicting task outcomes using classification

To handle the uncertainty regarding task outcomes, we train a classification model to predict whether a task will be successful. Given historical data $\mathcal{D}_{\text{Train}}$, the goal is to predict y_i using a classifier $f_\theta : \mathcal{X} \to [0,1] : \mathbf{x} \mapsto f_\theta(\mathbf{x})$ defined by parameters $\theta \in \Theta$ that predicts the probability of a task being successful. Classifier training can be accomplished with different objective

functions. We present two alternatives: one that focuses optimization on accuracy and one that optimizes the classification cost.

The conventional approach is to train the classifier with the aim of maximizing accuracy. This can be achieved using the maximum likelihood approach or, equivalently, by minimizing the cross-entropy loss function:

$$\mathcal{L}^{\mathrm{CE}} = y_i \log\, f_\theta(\mathbf{x}_i) + (1 - y_i)\log\big(1 - f_\theta(\mathbf{x}_i)\big). \tag{6}$$

A drawback of this approach is that the solution ignores some of the problem specifications. Some tasks are more important to classify correctly than others, depending on their cost (or profit) when executed. Therefore, in cost-sensitive learning, these costs are incorporated into the training of a model. In classification, the cost of a decision depends on whether it was classified correctly and on the task itself. These costs are formalized with the concept of a cost matrix $\mathbf{c}_i$ [33]:

$$\begin{array}{cc} & \textbf{Actual class } y_i \\ & \begin{array}{cc} 0 & 1 \end{array} \\ \textbf{Predicted class } \hat{y}_i \begin{array}{c} 0 \\ \\ 1 \end{array} & \begin{pmatrix} c_i^{\mathrm{TN}} & c_i^{\mathrm{FN}} \\ c_i^{\mathrm{FP}} & c_i^{\mathrm{TP}} \end{pmatrix} \end{array} \tag{7}$$

This way, we can directly minimize the average expected cost of predictions, as an alternative to the cross-entropy loss [29, 32]:

$$\begin{aligned} \mathcal{L}^{\mathrm{AEC}} = {} & y_i\Big(f_\theta(\mathbf{x}_i)c_i^{\mathrm{TP}} + \big(1 - f_\theta(\mathbf{x}_i)\big)c_i^{\mathrm{FN}}\Big) \\ & + (1 - y_i)\Big(f_\theta(\mathbf{x}_i)c_i^{\mathrm{FP}} + \big(1 - f_\theta(\mathbf{x}_i)\big)c_i^{\mathrm{TN}}\Big). \end{aligned} \tag{8}$$

$\mathcal{L}^{\mathrm{AEC}}$ is a semidirect predict-and-optimize method: it incorporates some information of the downstream decision-making task, but learning is still separated from optimization [26, 27].

4.1.2. Optimizing worker–task assignments

Given task predictions $\hat{\mathbf{Y}}$, we can optimize the task–worker assignments. Although various general algorithms have been proposed to solve assignment problems, our formulation can be solved analytically. Here, we present this solution and prove its optimality. The key insight is that, in expectation, the worker capacity can be seen as a sequence of workers with decreasing success rates, with each success rate the probability of that working existing given $W \sim Dist$. In other words, this probability is given by the complementary cumulative probability distribution function: $w_j = P(W \geq j) = 1 - F_W(j)$. Based on this, we can sort the tasks in terms of expected reward and the workers in terms of expected probability. Matching these two sortings then optimizes the assignment problem, where the most promising tasks are assigned to the most likely workers.

Theorem 1. $\mathbf{W} = \{w_i\}_{i=1}^{N}$ *is a sequence of monotonically decreasing worker success rates such that* $w_1 \geq \cdots \geq w_N$ *with* $w_i \in [0, 1]$ *for* $i = 1, \ldots, N$. $\hat{\mathbf{R}} = \begin{pmatrix} \hat{r}_1 & \ldots & \hat{r}_N \end{pmatrix}$ *are the predicted task rewards arranged in decreasing order such that* $\hat{r}_1 \geq \ldots \geq \hat{r}_N$. *For the resulting expected profit matrix* $P = \begin{pmatrix} p_{ij} \end{pmatrix}$ *with* $p_{ij} = w_i \hat{r}_j$, *the optimal assignment is* $A^* = I_N$.

Proof. Proof of Theorem 1.

$A^* = I_N$ is a feasible solution: it is straightforward to verify that the identity matrix satisfies constraints 2, 3 and 4 of the assignment problem. Moreover, the solution is the result of a greedy strategy: at each step m, we assign worker w with probability w_m to the highest remaining task m with payoff $\hat{r}_m$. To prove the optimality of this strategy, we show that it does not deviate from the optimal solution at each step up until the final solution is

obtained.

First, the best single worker–task assignment is selected: the highest profit p_{ij} is $p_{11} = w_1\hat{r}_1$; no other higher profit exists as no higher w_i or $\hat{r}_j$ exist. Next, we continue this strategy of selecting the best remaining worker–task assignment until there are no tasks left. We can show that, at each step, no other assignment matrix leads to a larger profit than this one. At step m, the profit obtained given assignment matrix A^* equals $p_{11} + p_{22} + \ldots + p_{mm} = w_1\hat{r}_1 + w_2\hat{r}_2 + \ldots + w_m\hat{r}_m$.

Deviating from A^* at a certain step means that at least one worker must be assigned to another task. We prove that no alternative assignment leads to a higher profit. Consider switching the assignments of tasks i and j with $i < j$. In the case that task j has already been assigned to a worker, we have:

$$\begin{aligned} & p_{ii} + p_{jj} \geq p_{ij} + p_{ji} \\ \iff \quad & w_i\hat{r}_i + w_j\hat{r}_j \geq w_i\hat{r}_j + w_j\hat{r}_i \\ \iff \quad & w_i(\hat{r}_i - \hat{r}_j) \geq w_j(\hat{r}_i - \hat{r}_j) \\ \iff \quad & w_i \geq w_j \text{ and } \hat{r}_i - \hat{r}_j \geq 0. \end{aligned}$$

In the case that task j has not yet been assigned, we have:

$$\begin{aligned} & p_{ii} \geq p_{ij} \\ \iff \quad & w_i\hat{r}_i \geq w_i\hat{r}_j \\ \iff \quad & w_i \geq 0 \text{ and } \hat{r}_i \geq \hat{r}_j \end{aligned}$$

In both cases, the final statements follow from $\mathbf{W}$ and $\hat{\mathbf{R}}$ being monotonically decreasing and $i < j$, or from $w_i \in [0, 1]$. □

4.2. Integrated predict-and-optimize using learning to rank

In this section, we present a novel integrated approach for solving the assignment problem in Section 3. Previously, we showed how the optimal assignment is $A^* = I_N$ if $\mathbf{W}$ and $\hat{\mathbf{R}}$ are arranged in decreasing order. Given that $\mathbf{W}$ is defined as a decreasing sequence, the challenge of optimizing the assignment can also be seen as correctly predicting the order of expected task rewards $\hat{\mathbf{R}}$. This formulation is equivalent to an alternative interpretation of the assignment problem as finding the optimal assignments by permuting the rows and columns of the profit matrix P such that the resulting sum of the elements on the diagonal is maximized, or formally [16]:

$$\max_{\pi \in \Pi_n} \sum_{i=1}^{N} p_{i,\pi(i)} \tag{9}$$

for $\pi \in \Pi_N$ with Π_N the set of all permutations of the indices $\{1, \ldots, N\}$, i.e., $\pi : \{1, \ldots, N\} \mapsto \{1, \ldots, N\}$. In our case, we need to find the optimal permutation of available tasks $\pi(\mathbf{T})$.

In this formulation, the assignment problem can be seen as predicting the optimal permutation $\pi(\mathbf{T})$ based on characteristics of the available tasks. Formally, let $g_\theta : \mathcal{X} \to \mathbb{R} : \mathbf{x} \mapsto g_\theta(\mathbf{x})$ be a ranking model. The goal is to find parameters $\theta \in \Theta$ such that the ordering of the mapping of tasks $g_\theta(x_1) \geq \ldots \geq g_\theta(x_n)$ corresponds to the ordering of their rewards $r_1 \geq \ldots \geq r_N$. A ranking based on g_θ can be seen as a permutation π of the indices $\{1, \ldots, n\}$.

The expected profit of a permutation $\pi(\mathbf{T})$ given a capacity W can be optimized directly using learning to rank. The key insight is that for a given permutation π of tasks $\mathbf{T}$, the expected profit $\sum_{i=1}^{N} w_i \hat{r}_{\pi(i)}$ of a ranking is equivalent to its discounted cumulative gain (DCG), which is a commonly

used class of metrics in learning to rank [45]. Typically, the DCG is defined with discount $\frac{1}{\log_2(i+1)}$ and gain $2^{t_i} - 1$ for $i \in \{1, \ldots, n\}$. However, to match the expected profit, our formulation uses discount $\{w_i\}_{i=1}^{N}$ corresponding to the capacity distribution, gain equal to 1 for all i, and relevance $\hat{r}_i$. By dividing the DCG by its ideal value (IDCG), the normalized DCG (NDCG) is obtained: $\text{NDCG} = \frac{\text{DCG}}{\text{IDCG}}$ with $\text{NDCG} \in [0, 1]$.

Optimizing the NDCG (or equivalently, the expected profit) directly is challenging as it depends on the predicted relative positions of instances instead of the model's outputs $g_\theta(\mathbf{x_i})$. Nevertheless, various algorithms have been proposed for this task in the literature on learning to rank. In this work, we use the widely used LambdaMART [46], which uses a combination of the LambdaRank loss and gradient boosting of decision trees to construct the ranking model. In this way, we can train a ranking model g_θ to optimize the NDCG or expected profit of the assignments directly.

Finally, we need to specify each task's relevance, which serves as a label according to which the ranking would ideally be constructed. Because the ranking corresponds to the priority that should be given to tasks, it should respect the ordering in terms of both outcome y_i and task payoffs $\mathbf{v_i}$. In other words, successful tasks should be more relevant than unsuccessful tasks, and a more profitable task should be more relevant. Therefore, we use a task's reward r_i as a cost-sensitive relevance, as it uses an instance's class label y_i and its cost matrix $\mathbf{c}_i$ (see Equation (7)). By means of this approach, a positive task's relevance is the profit (or equivalently, the negative cost) obtained by classifying it positively minus the profit obtained by classifying it negatively; vice versa for negative tasks. Thus, we obtain the relevance or

reward r_i as follows:

$$r_i = y_i v_i^+ + (1 - y_i) v_i^- = y_i \left(c_i^{\text{FN}} - c_i^{\text{TP}} \right) + (1 - y_i) \left(c_i^{\text{TN}} - c_i^{\text{FP}} \right).$$

Alternatively, if the goal is to optimize for accuracy rather than cost, we can use class label y_i as the relevance of instance i.

5. Empirical results

In this section, we empirically evaluate and compare the two-stage and the integrated approach for a variety of tasks. We use publicly available data from a variety of application areas. For each application, the goal is to optimally allocate resources to optimize the expected cost given stochastic capacity. All code for the experimental analysis will be made available online upon publication of this paper.

To compare the different approaches, we use gradient boosting to train the predictive models. Four different objectives are compared, depending on the task (classification or learning to rank) and on whether they aim to maximize precision or profit. First, xgboost and csboost are conventional approaches based on classification. More specifically, xgboost denotes a conventional classification model using the cross-entropy loss $\mathcal{L}^{\text{CE}}$ (see Equation (6)), while csboost uses a cost-sensitive objective function $\mathcal{L}^{\text{AEC}}$ (see Equation (8)). Second, LambdaMART and csLambdaMART are integrated predict-and-optimize approaches based on learning to rank. LambdaMART uses the binary class label y_i, whereas csLambdaMART uses task payoffs r_i as relevance. All models are implemented in Python using the `xgboost` package [47]. Gradient boosting is a popular methodology for both classification

and ranking that has great predictive performance, as illustrated by recent benchmarking studies [9, 48].

5.1. Data

The data sets are enlisted in Table 2 and stem from different application areas: customer churn prediction, credit scoring and direct marketing. They all concern binary classification where tasks are either successful or unsuccessful. Resources are limited and stochastic: we assume a lognormal capacity distribution $W \sim \mathcal{LN}(\mu = \log(100), \sigma = 1)$.

The cost matrices are taken from earlier work on cost-sensitive classification (see Table 3). In churn prediction, we have c_i^{FP} and c_i^{FN} as, respectively, 2 and 12 times the monthly amount A_i for KTCC following Petrides and Verbeke [34]; whereas we follow the cost matrix given with the data set for TSC [49]. For credit scoring, we calculate the instance-dependent costs c_i^{FP} and c_i^{FN} as a function of the loan amount A_i following Bahnsen et al. [29]. In direct marketing, a positive classification incurs a fixed cost $c_f = 1$, while missing a potential success incurs an instance-dependent cost equal to the expected interest given A_i, following Bahnsen et al. [50]. Similarly, in fraud detection, a positive prediction leads to an investigation that entails a fixed cost c_f, and missing a fraudulent transaction leads to a cost equal to its amount A_i. We use $c_f = 10$, following Höppner et al. [32].

5.2. Results

We present the results using various performance metrics to compare the different models. The main metric of interest is either the expected precision or the expected profit given the stochastic capacity distribution

Application	**Abbr.**	N	% Pos	**Ref.**
Churn prediction	KTCC	7,032	26.58	IBM Sample Data Sets [51]
	TSC	9,379	4.79	Bahnsen et al. [49]
Credit scoring	HMEQ	1,986	19.95	Baesens et al. [52]
	BN1	3,123	33.33	Lessmann et al. [9]
	BN2	7,190	30.00	Lessmann et al. [9]
	VCS	18,917	16.95	Petrides et al. [53]
	UK	30,000	4.00	Lessmann et al. [9]
	DCCC	30,000	22.12	Yeh and Lien [54]
	GMSC	112,915	6.74	/
Direct marketing	UBM	45,211	11.70	Moro et al. [55]
	KDD	191,779	5.07	/
Fraud detection	KCCF	282,982	0.16	Dal Pozzolo et al. [56]
	KIFD	590,540	3.50	/
	ACCF	3,639,323	0.65	Van Vlasselaer et al. [57]

Table 2: **Data sets overview.** For each data set, we present the application area, abbreviation, number of instances (N), class imbalance in terms of proportion of positive instances (% Pos), and corresponding reference.

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

(a) Churn prediction

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

(b) Credit scoring

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

(c) Direct marketing

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

(d) Fraud detection

Table 3: **Cost matrices for the different application areas.** For each application, we present the costs for all outcomes in terms of predicted ($\hat{y}$) and actual (y) labels. A_i, c_i^{FN}, c_i^{FP} and Int_i represent instance-dependent costs and c_f is a fixed cost.

W, depending on whether accuracy or profit is the objective. Furthermore, we present several additional classification and ranking metrics to gain more insight into the differences between the methodologies. For each metric, we present the average over all data sets and test whether the best performance is significantly different from the others using a Friedman test on the rankings with Bonferroni–Dunn post hoc correction [58, 59, 60] (see Table 4).

5.2.1. Expected precision and expected profit

In terms of expected precision, LambdaMART is the best performing model. Two models optimize for accuracy: LambdaMART and xgboost. The ranking model, LambdaMART, outperforms the classification model, xgboost. In terms of expected profit, the cost-sensitive ranking model, csLambdaMart, performs best. Of the two models optimizing for accuracy, xgboost and LambdaMART, the ranking model again achieves better results, although this difference is not statistically significant. This increase in performance of the rankings models compared to classification models illustrates

	Expected precision	Expected profit	Average precision	Spearman correlation	AUCPC
xgboost	0.4956 ± 0.08	0.2115 ± 0.05	**0.9423 ± 0.01**	−0.0382 ± 0.03	0.5548 ± 0.07
csboost	0.5865 ± 0.06	0.2940 ± 0.05	0.9075 ± 0.02	0.2258 ± 0.07	0.5657 ± 0.07
LambdaMART	**0.6555 ± 0.07**	0.2471 ± 0.05	0.9366 ± 0.01	−0.0302 ± 0.04	0.5363 ± 0.06
csLambdaMART	0.6089 ± 0.07	**0.3587 ± 0.05**	0.9336 ± 0.01	**0.3829 ± 0.08**	**0.5999 ± 0.06**

Table 4: **Evaluation metrics overview.** We present an overview of the evaluation metrics. The average and standard deviation over all data sets are shown, with the best result denoted in **bold**. Results that are not significantly different from the best result are underlined (α = 0.05). This is based on a Friedman test on the rankings with Bonferroni–Dunn post hoc correction. For both expected precision and profit, the ranking models perform best in their respective category. For the classification metric, average precision, the cost-insensitive classifier, xgboost, performs best. Conversely, for the ranking metrics, namely, Spearman correlation and the area under the cumulative profit curve, the ranking models outperform their classifying counterparts.

the potential benefit of our integrated ranking approach when capacity is constrained. We further compare the trade-off between profit and precision in Figure 2 by plotting the rankings for each data set. To get an idea of the densities for the different models, we estimate it using a Gaussian kernel and show it for probabilities greater than 0.5. Although the densities overlap, the ranking models outperform their classifying counterparts in their respective category. Again, this demonstrates the benefit of integrating the capacity constraint in the optimization.

5.2.2. Average precision, Spearman's ρ and AUCPC

These metrics weight all instances in the ranking equally, as opposed to the previous metrics that weighted instances depending on their probability

of being processed given the capacity distribution [36]. On the one hand, we consider a classification metric: given the high degree of class imbalance for some data sets, we use the average precision. On the other hand, we consider two ranking metrics: the area under the cumulative profit curve and Spearman's rank correlation coefficient ρ.

First, we assess the quality of the model's predictions with a standard classification metric: average precision (AP). This metric summarizes the precision-recall curve and looks at the trade-off between precision and recall at different thresholds. As expected, the cost-insensitive classification model, xgboost, performs best. This result is no surprise, given that xgboost is a classification model that optimizes for accuracy. However, this conventional classification metric has only weak correlation with the expected precision, suggesting that it is not a good indicator of performance. Therefore, this results gives rise to an important insight: when there is limited capacity to act on predictions, traditional classification metrics are not a good indicator of performance.

We also adopt two ranking metrics. First, we use Spearman's rank correlation coefficient to quantify the correlation between the ranking of the predictions and the ranking of the task payoffs. csLambdaMart is the best performing model, outperforming csboost. Moreover, both cost-insensitive models have a correlation of approximately 0. This is as expected, as these models do not take payoff into account in their optimization. Second, the cumulative profit curve plots the profit that is realized as a function of the number k of first ranked instances, with $k \in [1, N]$. We compare the area under this curve with the area of a random ranking and one of the optimal

ranking to obtain a value between 0 and 1. csLambdaMART performs best, though neither the difference with xgboost nor csboost is statistically significant. Compared to the classification metric, these results are more aligned with the expected precision and profit.

These findings indicate that metrics for evaluating the ranking quality, such as Spearman's ρ or the AUCPC, are more suitable than classification metrics, such as the average precision, for evaluating a model's performance under limited capacity. Moreover, our results suggest that ranking as a solution more closely aligns with the problem of allocating limited resources to uncertain tasks than classification, which is also confirmed by the superior performance of ranking models compared to classification models in terms of expected precision and expected profit. This represents an important insight, given the abundance of existing work using classification models for these application areas where capacity constraints are commonly encountered.

5.2.3. Top k metrics

Finally, we also consider metrics focusing solely on the top of the ranking. Given limited capacity, these are the instances that will be prioritized. We can evaluate this critical part of the ranking by looking at the precision and profit of the ranking for the first k instances for different values of k (see Figure 3). The ranking model optimizing for accuracy, LambdaMART, performs best in terms of precicision@k, while the ranking model optimizing for profit, csLambdaMART, has the best performance in terms of profit@k. Again, these findings suggest that ranking models perform better given limited worker capacity, due to their ability to better prioritize the most important tasks at the top of the ranking. Indeed, given limited capacity, these

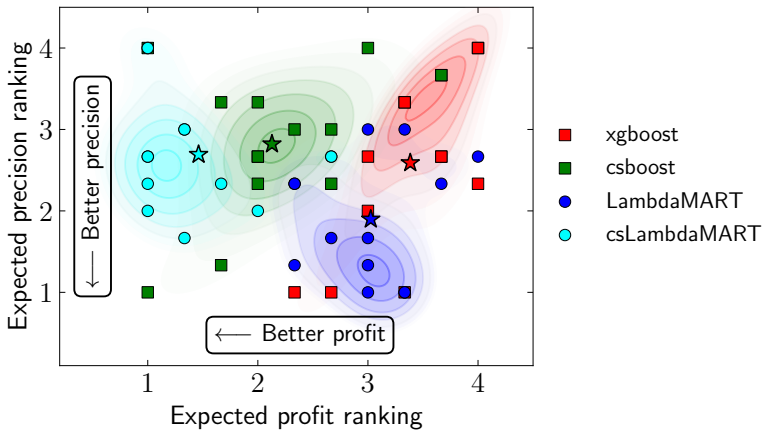


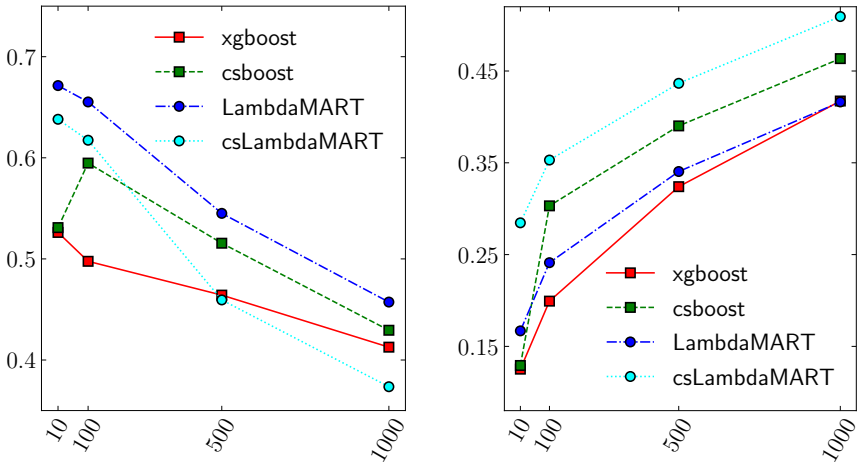
Figure 2: **Comparing the methodologies in terms of expected precision and profit.** We plot each methodologies' ranking in terms of expected profit and expected precision on each data set. For each method, the average ranking is shown with a star (☆). Moreover, the ranking density is fitted with a Gaussian kernel; for visual clarity, only probabilities greater than 0.5 are shown. On average, csLambdaMART performs best in terms of expected profit, while LambdaMART performs best in terms of expected precision.

are the tasks that will be executed.

6. Conclusion

In this work, we formally introduced and defined a commonly encountered problem: how to optimally allocate limited, stochastic resource capacity to tasks with uncertain payoff to maximize the expected profit. Moreover, we contribute by proposing a novel integrated solution using learning to rank and empirically comparing it with a more conventional predict-then-optimize approach using a classification model.

Our findings illustrate the benefit of approaching this problem as a rank-



(a) **Precision@k** (b) **Profit@k**

Figure 3: **Evaluating the top k ranked instances.** Precision **(a)** and profit **(b)** for obtained by the top k instances in the ranking for the different models averaged over all data sets. The ranking models outperform the classifiers in the metric they optimize for: LambdaMART is the best in terms of precision; csLambdaMART has the best profit.

ing problem, which allows us to consider the availability of limited and stochastic resources. Theoretically, we show how the expected profit for a given capacity distribution can be optimized directly using learning to rank with a specific formulation of the net discounted cumulative gain as the objective. Empirical results for a variety of applications show that ranking models achieve better performance in terms of expected profit or expected precision, depending on the objective. Moreover, good results in terms of ranking metrics are more indicative of good performance in terms of expected profit compared to conventional classification metrics. This illustrates how ranking is more closely aligned with the problem at hand compared to classifying. In summary, in the common scenario where decision-makers are constrained

by limited resources, deciding upon resource allocation using classification models is inferior to using learning to rank. These findings have important implications for practitioners in a variety of application areas.

Managerial implications. Our findings have significant implications for practitioners that use predictive models for decision support in applications where resource capacity to act upon predictions is limited. This situation is commonly encountered in applications such as fraud detection, credit scoring, churn prediction, and direct marketing. Our work shows that, when decision-makers are faced with the challenge of optimally allocating limited, stochastic resource capacity to tasks with uncertain payoffs, they should consider adopting a ranking-based approach. We demonstrated that optimizing the expected precision or profit with a ranking model leads to improved decision-making compared to a commonly used approach using classification models. Similarly, we showed that ranking metrics provide a more accurate assessment of performance than classification metrics in settings where resources are constrained. Our results underscore the importance of embracing learning to rank over traditional classification methods in resource allocation decisions, which has important implications for practitioners seeking to maximize profitability and efficiency in applications with resource constraints.

Our work opens several promising directions for future research. For example, it would be interesting to consider a temporal variant of the assignment problem with tasks arriving sequentially in time. Although this problem has been studied extensively for stochastic or random arrival rates [61, 62, 63], future work could consider the addition of a predictive ranking model to address uncertainty regarding task outcomes. Another possible ex-

tension would be to consider tasks that require varying degrees of resources. For example, in credit scoring, loans with a large principal require more resources.

References

[1] P. A. Samuelson, W. D. Nordhaus, Economics, 19 ed., McGraw-Hill/Irwin, 2010.

[2] L. Ward Jr, On the optimal allocation of limited resources, Operations Research 5 (1957) 815–819.

[3] H. Everett III, Generalized lagrange multiplier method for solving problems of optimum allocation of resources, Operations research 11 (1963) 399–417.

[4] A. Calma, W. Ho, L. Shao, H. Li, Operations research: topics, impact, and trends from 1952–2019, Operations Research 69 (2021) 1487–1508.

[5] B. Baesens, S. Viaene, D. Van den Poel, J. Vanthienen, G. Dedene, Bayesian neural network learning for repeat purchase modelling in direct marketing, European Journal of Operational Research 138 (2002) 191–211.

[6] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, J. Vanthienen, Benchmarking state-of-the-art classification algorithms for credit scoring, Journal of the operational research society 54 (2003) 627–635.

[7] W. Verbeke, D. Martens, C. Mues, B. Baesens, Building comprehensible customer churn prediction models with advanced rule induction techniques, Expert systems with applications 38 (2011) 2354–2364.

[8] W. Verbeke, K. Dejaeger, D. Martens, J. Hur, B. Baesens, New insights into churn prediction in the telecommunication sector: A profit driven data mining approach, European journal of operational research 218 (2012) 211–229.

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

[10] V. Van Vlasselaer, T. Eliassi-Rad, L. Akoglu, M. Snoeck, B. Baesens, Gotcha! network-based fraud detection for social security fraud, Management Science 63 (2017) 3090–3110.

[11] A. Cerioli, L. Barabesi, A. Cerasa, M. Menegatti, D. Perrotta, Newcomb–benford law and the detection of frauds in international trade, Proceedings of the National Academy of Sciences 116 (2019) 106–115.

[12] R. Burkard, M. Dell'Amico, S. Martello, Assignment problems: revised reprint, SIAM, 2012.

[13] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, D. Rus, On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment, Proceedings of the National Academy of Sciences 114 (2017) 462–467.

[14] D. Bertsimas, A. Delarue, S. Martin, Optimizing schools' start time and bus routes, Proceedings of the National Academy of Sciences 116 (2019) 5943–5948.

[15] B. Toktas, J. W. Yen, Z. B. Zabinsky, Addressing capacity uncertainty in resource-constrained assignment problems, Computers & operations research 33 (2006) 724–745.

[16] P. A. Krokhmal, P. M. Pardalos, Random assignment problems, European Journal of Operational Research 194 (2009) 1–17.

[17] J. Li, B. Xin, P. M. Pardalos, J. Chen, Solving bi-objective uncertain stochastic resource allocation problems by the cvar-based risk measure and decomposition-based multi-objective evolutionary algorithms, Annals of Operations Research 296 (2021) 639–666.

[18] R. Johari, V. Kamble, Y. Kanoria, Matching while learning, Operations Research 69 (2021) 655–681.

[19] A. Lodi, G. Zarpellon, On learning and branching: a survey, Top 25 (2017) 207–236.

[20] Y. Bengio, A. Lodi, A. Prouvost, Machine learning for combinatorial optimization: a methodological tour d'horizon, European Journal of Operational Research 290 (2021) 405–421.

[21] P. Donti, B. Amos, J. Z. Kolter, Task-based end-to-end model learning in stochastic optimization, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances

in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/3fc2c60b5782f641f76bcefc39fb2392-Paper.pdf.

[22] B. Wilder, B. Dilkina, M. Tambe, Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 1658–1665.

[23] A. N. Elmachtoub, P. Grigas, Smart "predict, then optimize", Management Science (2021).

[24] J. Mandi, P. J. Stuckey, T. Guns, et al., Smart predict-and-optimize for hard combinatorial optimization problems, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 2020, pp. 1603–1610.

[25] J. Kotary, F. Fioretto, P. Van Hentenryck, B. Wilder, End-to-end constrained optimization learning: A survey, arXiv preprint arXiv:2103.16378 (2021).

[26] E. Demirović, P. J. Stuckey, J. Bailey, J. Chan, C. Leckie, K. Ramamohanarao, T. Guns, An investigation into prediction+ optimisation for the knapsack problem, in: International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research, Springer, 2019, pp. 241–257.

[27] E. Demirović, P. J Stuckey, J. Bailey, J. Chan, C. Leckie, K. Ramamohanarao, T. Guns, Predict+ optimise with ranking objectives: Exhaustively learning linear functions, in: Proceedings of the Twenty-Eighth

International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, International Joint Conferences on Artificial Intelligence, 2019, pp. 1078–1085.

[28] J. Mandi, J. Kotary, S. Berden, M. Mulamba, V. Bucarey, T. Guns, F. Fioretto, Decision-focused learning: Foundations, state of the art, benchmark and future opportunities, arXiv preprint arXiv:2307.13565 (2023).

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

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

[31] S. Höppner, E. Stripling, B. Baesens, S. vanden Broucke, T. Verdonck, Profit driven decision trees for churn prediction, European journal of operational research 284 (2020) 920–933.

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

[33] C. Elkan, The foundations of cost-sensitive learning, in: International joint conference on artificial intelligence, volume 17, Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.

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

[35] T. Vanderschueren, T. Verdonck, B. Baesens, W. Verbeke, Predict-then-optimize or predict-and-optimize? an empirical evaluation of cost-sensitive learning strategies, Information Sciences 594 (2022) 400–415.

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

[37] A. Dal Pozzolo, G. Boracchi, O. Caelen, C. Alippi, G. Bontempi, Credit card fraud detection: a realistic modeling and a novel learning strategy, IEEE transactions on neural networks and learning systems 29 (2017) 3784–3797.

[38] I. Bose, X. Chen, Quantitative models for direct marketing: A review from systems perspective, European Journal of Operational Research 195 (2009) 1–16.

[39] J. Hadden, A. Tiwari, R. Roy, D. Ruta, Computer assisted customer churn management: State-of-the-art and future trends, Computers & Operations Research 34 (2007) 2902–2917.

[40] D. A. Shifman, I. Cohen, K. Huang, X. Xian, G. Singer, An adaptive machine learning algorithm for the resource-constrained classification problem, Engineering Applications of Artificial Intelligence 119 (2023) 105741.

[41] X. Yang, K. Tang, X. Yao, A learning-to-rank approach to software defect prediction, IEEE Transactions on Reliability 64 (2014) 234–246.

[42] L. Coenen, W. Verbeke, T. Guns, Machine learning methods for short-term probability of default: A comparison of classification, regression and ranking methods, Journal of the Operational Research Society (2020) 1–16.

[43] F. Devriendt, J. Van Belle, T. Guns, W. Verbeke, Learning to rank for uplift modeling, IEEE Transactions on Knowledge and Data Engineering (2020).

[44] R. McBride, K. Wang, Z. Ren, W. Li, Cost-sensitive learning to rank, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, 2019, pp. 4570–4577.

[45] Y. Wang, L. Wang, Y. Li, D. He, W. Chen, T.-Y. Liu, A theoretical analysis of ndcg ranking measures, in: Proceedings of the 26th annual conference on learning theory (COLT 2013), volume 8, Citeseer, 2013, p. 6.

[46] Q. Wu, C. J. Burges, K. M. Svore, J. Gao, Ranking, boosting, and model adaptation, Technical Report, Technical report, Microsoft Research, 2008.

[47] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, et al., Xgboost: extreme gradient boosting, R package version 0.4-2 1 (2015) 1–4.

[48] B. R. Gunnarsson, S. Vanden Broucke, B. Baesens, M. Óskarsdóttir, W. Lemahieu, Deep learning for credit scoring: Do or don't?, European Journal of Operational Research 295 (2021) 292–305.

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

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

[51] IBM Sample Data Sets, Telco customer churn, version 1, 2017. Retrieved October 10, 2021 from https://www.kaggle.com/blastchar/telco-customer-churn/version/1.

[52] B. Baesens, D. Roesch, H. Scheule, Credit risk analytics: Measurement techniques, applications, and examples in SAS, John Wiley & Sons, 2016.

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

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

[55] S. Moro, P. Cortez, P. Rita, A data-driven approach to predict the

success of bank telemarketing, Decision Support Systems 62 (2014) 22–31.

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

[57] V. Van Vlasselaer, C. Bravo, O. Caelen, T. Eliassi-Rad, L. Akoglu, M. Snoeck, B. Baesens, Apate: A novel approach for automated credit card transaction fraud detection using network-based extensions, Decision Support Systems 75 (2015) 38–48.

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

[59] S. Garcia, F. Herrera, An extension on" statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons., Journal of machine learning research 9 (2008).

[60] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, Information sciences 180 (2010) 2044–2064.

[61] C. Derman, G. J. Lieberman, S. M. Ross, A sequential stochastic assignment problem, Management Science 18 (1972) 349–355.

[62] C. Albright, C. Derman, Asymptotic optimal policies for the stochastic sequential assignment problem, Management Science 19 (1972) 46–51.

[63] S. C. Albright, Optimal sequential assignments with random arrival times, Management Science 21 (1974) 60–67.