Artificial Intelligence Review (2025) 58:238
https://doi.org/10.1007/s10462-025-11242-6

Rethinking and recomputing the value of machine learning models

Burcu Sayin[1] · Jie Yang[2] · Xinyue Chen[2] · Andrea Passerini[1] · Fabio Casati[1,3]

Accepted: 17 April 2025 / Published online: 8 May 2025

Abstract
In this paper, we argue that the prevailing approach to training and evaluating machine learning models often fails to consider their real-world application within organizational or societal contexts, where they are intended to create beneficial value for people. We propose a shift in perspective, redefining model assessment and selection to emphasize integration into workflows that combine machine predictions with human expertise, particularly in scenarios requiring human intervention for low-confidence predictions. Traditional metrics like accuracy and f-score fail to capture the beneficial value of models in such hybrid settings. To address this, we introduce a simple yet theoretically sound "value" metric that incorporates task-specific costs for correct predictions, errors, and rejections, offering a practical framework for real-world evaluation. Through extensive experiments, we show that existing metrics fail to capture real-world needs, often leading to suboptimal choices in terms of value when used to rank classifiers. Furthermore, we emphasize the critical role of calibration in determining model value, showing that simple, well-calibrated models can often outperform more complex models that are challenging to calibrate.

Keywords Machine learning · Hybrid intelligence · Selective classification · Cost-sensitive learning

✉ Burcu Sayin
burcu.sayin@unitn.it

Jie Yang
j.yang-3@tudelft.nl

Xinyue Chen
xinyuechen223@gmail.com

Andrea Passerini
andrea.passerini@unitn.it

Fabio Casati
fabio.casati@servicenow.com

[1] Department of Information Engineering and Computer Science, University of Trento, Via Sommarive 9, Povo, 38123 Trento, Italy

[2] Department of Software Technology, TU Delft, Mekelweg 5, 2628 XE Delft, The Netherlands

[3] Servicenow, Zurich, Switzerland

1 Introduction

Recently, a few position papers (Casati et al. 2021; Sayin et al. 2021a, 2021b; Gunel 2022; Sayin et al. 2022, 2023a, 2023b) have challenged the underlying assumptions of quality in Machine Learning (ML), particularly the overemphasis on accuracy-based metrics and various measures of calibration errors (i.e. the difference between a model's predicted probabilities and the actual likelihood of its predictions being correct). At the heart of this stance, there are two observations: (i) ML models are almost always applied in hybrid human–machine settings, where the model can abstain or its prediction be rejected for insufficient confidence (i.e. the model's estimation of the correctness of its prediction) as in Fig. 1, and (ii) the beneficial value of correct inferences, as well as the detrimental value of incorrect inferences and rejections, is determined by the use case, not by the model.

In our experience (see Sect. 4.3), we have found that the majority of AI deployments in the enterprise consist of selective models or *selective classifiers* (Geifman and El-Yaniv 2017), which is more of a rule than an exception. An example where this commonly occurs is in customer support requests, where the goal is to identify the customer's intent to trigger an automated request processing workflow if possible. Failing to comprehend the customer's intent and resorting to human agents is not ideal. However, it's even more problematic to misinterpret the customer's intent and guide them down the wrong path toward a resolution. This is why intent classifications are filtered based on prediction confidence. How "good" or "useful" a model is therefore depends on the beneficial value it brings when inserted in ML solution workflows (Fig. 1). This beneficial value depends on how often the workflow rejects the predictions, on the correctness patterns of the predictions that are not rejected, and on the detrimental value of errors vs benefits of correct predictions. While in this paper we only marginally discuss the plethora of Large Language Models (LLMs), the problem is exactly the same if not worse: With generative AI the question of whether to show an answer or to withhold it is crucial and there are many things that can be wrong in an answer, the most common being hallucination. We add that the fact that some APIs do not reveal likelihood/confidence level makes model evaluation more difficult.

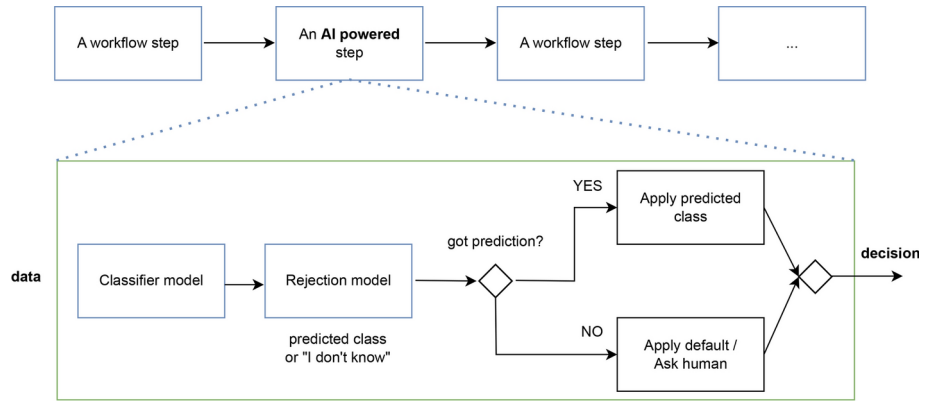

Fig. 1 A typical implementation of ML models into an ML solution workflow involves using a rejection function that filters predictions based on a confidence threshold. This approach generally assumes that the classifier is trained independently of the rejection logic. However, this is not a necessity-the classifier can be designed to be aware of the associated costs, which may make it less "general" but more tailored to specific needs

To some extent, all this is trivial. There is no inherent difficulty in developing use case-based value functions, selecting the best model from a set of well-performing models based on the value function, or evaluating a model's performance across multiple value functions. Moreover, one could contend that accuracy metrics are a sufficient substitute for evaluating model improvements in data science, or for selecting models to deploy in an AI platform designed to meet specific use cases. Thus, the practical approach would be to choose the model with the best accuracy or F1 score and enable users to filter out predictions with a confidence level lower than a set threshold. Accuracy and similar metrics are easy to comprehend and do not require us to determine parameters such as the "cost of errors", which can be difficult to estimate, especially when considering the use case.

In this paper, we show that this reasoning is wrong. If we accept that classifiers are mostly applied as selective models, then the method we use to measure, compare, and even train models must change. The implications of models being almost always applied as selective classifiers are often neglected in the literature, and this is also reflected in model leaderboards. We also show that the simplicity of not having to choose a cost parameter is an illusion: when we use accuracy to compare models, i) we do implicitly choose a cost parameter, often without realizing it, and ii) this implicitly selected cost is probably one of the worst choices possible: that of setting the relative cost of errors to zero. Despite being counter-intuitive, we show that accuracy is a quality metric that may be selected when the consequences of model errors are not critical. When a model is likely to be used across multiple use cases, relying solely on accuracy-based metrics can have significant implications. Overall, we show that:

- Universal metrics used for model evaluation are poor indicators of model value, potentially leading to incorrect decisions such as choosing models with *negative* value;
- Metrics designed to account for cost-sensitive errors are also inappropriate as they fail to consider the reject option;
- Lack of calibration substantially affects model value, and poorly calibrated complex models can be outperformed by simple, decades-old models that are easier to calibrate;
- Operating in an out-of-distribution setting further reduces the reliability of standard performance metrics.

It is worth underlining that the notion of value we introduce in this paper is not a radically different metric, but rather a combination of existing metrics, such as accuracy, detrimental value of errors and rejection rate, into a single measure accounting for the "value" of the predictor for a user. Importantly, the metric is normalized in such a way that a value of zero indicates a classifier that is completely useless, a negative value a classifier that is harmful (with respect to always ignoring it and resorting to the default path) and any value larger than zero indicates the gain that is obtained by using the classifier.

The remainder of this paper is structured as follows: in Sect. 2, we review related works on our concept of model value. Then, in Sect. 3, we formalize this notion and introduce the rejection threshold maximizing value, along with its extension to the cost-sensitive setting where different errors have different costs. Section 4 presents our experimental analysis comparing our value metric with standard performance measures, while Sect. 5 offers our conclusions.

2 Related work

Selective classification. Mimicking the typical use of ML models in many practical applications, a number of approaches rely on the combination of an ML model making an initial prediction and a human annotator taking over when the model's confidence is not high enough (Callaghan et al. 2018). Selective classifiers are specifically conceived for this use, by including a rejection mechanism to decide when to abstain from making a prediction. The literature on selective classifiers is extensive, encompassing a broad range of learning algorithms, including nearest-neighbor classifiers (Hellman 1970), SVM (Fumera and Roli 2002), and neural networks (Cordella et al. 1995; De Stefano et al. 2000; Geifman and El-Yaniv 2017) (see Hendrickx et al. Hendrickx et al. 2021 for a recent survey). The effectiveness of this solution is, however, heavily dependent on the reliability of machine confidence, which has shown to be very poor, especially for deep learning (Balda et al. 2020; Guo et al. 2017).

Classifier confidence. To effectively use a classifier (Jiang et al. 2018), it is important to understand its properties and have confidence in its individual predictions. The literature proposes various confidence-based methods, including measuring the entropy of the softmax predictions (Teerapittayanon et al. 2017), calculating trust scores based on the distance of samples to a calibration set (Jiang et al. 2018), determining a confidence threshold (via Shannon entropy Shannon 1948, Gini coefficient Bendel et al. 1989, or norm-based methods Ng 2004) that maximizes coverage for a given accuracy (Bukowski et al. 2021), and using semantics-preserving data transformation to estimate confidence (Bahat and Shakhnarovich 2020). Post-hoc recalibration is a popular strategy for improving classifier confidence, with techniques ranging from temperature scaling (Guo et al. 2017) to Dirichlet calibration (Kull et al. 2019 (see a recent survey by Filho et al. de Menezes e Silva Filho et al. 2021). However, as we will show in our experimental evaluation (see Sect. 4.2.3), it's essential to complement these solutions with a proper value metric to assess the classifier's beneficial value in real-world applications.

Cost-sensitive learning addresses the challenge of training classifiers by considering the varying costs associated with different types of errors, particularly in scenarios with significant class imbalance (Elkan 2001; Ling and Sheng 2010; Thai-Nghe et al. 2010; Tu and Lin 2020; Charoenphakdee et al. 2021). Existing work includes (Tu and Lin 2020): (i) data-level approaches Ting (1998); Zadrozny et al. (2003) where the class distribution of training data is balanced via sampling methods, and (ii) algorithm-level approaches, that use a thresholding scheme Chai et al. (2004); Domingos (1999); Elkan (2001); Ling and Sheng (2010); Sayin et al. (2021); Sheng and Ling (2006); Suri (2022) to improve the prediction performance on the minority class (e.g. in binary classification, the threshold is set such that the prediction is 1 only if the expected cost associated with this prediction is lower than or equal to that of predicting 0). Although this line of work is closely related to our setting as it also considers the impact of errors on the downstream pipeline, it assumes that the classifier provides a prediction for every instance without any rejection mechanism. This assumption can significantly impact the evaluation of the resulting classifier's quality, as our experimental evaluation will demonstrate (see Sect. 4.2.2). Finally, Charoenphakdee et al. (2021) introduces a novel approach to classification with rejection option by training an ensemble of cost-sensitive classifiers. In contrast, our goal is not to develop a novel cost-sensitive classifier. Instead, we aim to introduce a metric designed to evaluate such classifiers.

Hybrid Human-AI systems aim at solving classification problems with humans and machines (Dellermann et al. 2019a, 2019b; Raghu et al. 2019; Wilder et al. 2021), but effectively combining human and machine intelligence has many challenges. For example, *trust in humans* requires a deep understanding of how to design crowdsourcing tasks and model their complexity (Gadiraju et al. 2017; Qarout et al. 2018; Wu and Quinn 2017; Yang et al. 2016), test and filter crowd workers (Bragg et al. 2016), aggregate results into a decision (Han et al. 2020; Kamar et al. 2012; Krivosheev et al. 2018; Li 2013; Liu et al. 2013; Whitehill et al. 2009; Zhou et al. 2012), improve the engagement (Han et al. 2019, 2021); Qiu et al. 2020), or leverage crowds to learn features of ML models (Cheng and Bernstein 2015; Rodriguez et al. 2014). Furthermore, *the effective aggregation of human and machine decisions* (Nagar and Malone 2011, 2012; Nuñez 2022) depends on many factors, such as training, explaining, sustaining, interacting, and amplifying. The value metric is defined in the context of hybrid human-AI systems, where humans intervene whenever the AI defers a decision due to low confidence in its prediction. This metric accounts for the value of deferral, along with the impact of both correct and incorrect machine predictions, in assessing the overall value of the system. We believe that defining appropriate measures of the beneficial value of the joint human–machine system is a major prerequisite to keep research in the field on the right course.

3 Measuring model "value"

In this section, we formally define the notion of model "value", and show how threshold-based selective classifiers, by far the most popular class of classifiers in practical ML workflows, can be adjusted to maximize value.

3.1 The setting

Selective classifiers are ML models that generate output only when they are sufficiently confident in their prediction accuracy; otherwise, they abstain from making a decision, guided by a predefined rejection function. Selective classifiers can be implemented as follows:

(a) We take a model f that outputs a prediction y and a *confidence* c_y (or a vector c of confidence for a set of possible answers). Then, we filter the predictions to take only those above a certain confidence threshold (Fig. 2a).
(b) The model f outputs predictions and confidence, but we apply a selector model s that decides whether to accept the prediction or not, based on features of the input x (Fig. 2b).
(c) A hybrid of the two above cases is where the selector is a recalibrator r that can either take as input the prediction and confidence measure (*feature-agnostic* calibrator) or also the input features of x and adjust the confidence vector (*feature-aware* calibrator), typically applying threshold-based selection on the resulting confidence (Fig. 2c).
(d) The model f is already trained to only output predictions that are "good enough" and includes an "I don't know" class (Fig. 2d).

The first case is the most common, at least in our experience (see Sect. 4.3). The second case is an extension and generalization of the first case in two ways: it can take features

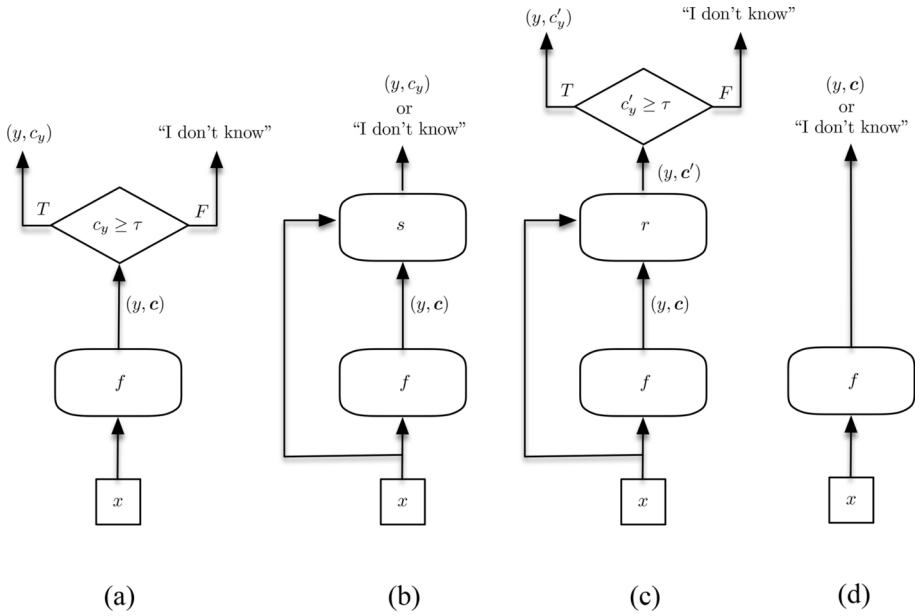

Fig. 2 Common approaches to selectivity in classification: **a** filtering predictions based on a confidence threshold, **b** employing an input-based selector model to decide on prediction acceptance, **c** using a confidence recalibrator followed by threshold-based filtering, or **d** incorporating built-in abstention with an 'I don't know' class

as input (s can be trained as opposed to "just" being a formula), and it can filter based on any formula. It however requires some form of "training" or machine teaching, which is highly non-trivial. The recalibrator also typically requires some form of training. However, a feature-agnostic calibrator can be easily set up by post-hoc calibration strategies de Menezes e (Silva Filho et al. 2021), the most common being temperature scaling (Guo et al. 2017). Finally, the last case is what is being addressed by the recent literature on learning to reject (Hendrickx et al. 2021), which is currently confined to the academic world, but could greatly benefit from incorporating the notion of value that we introduce here. In this paper, we focus on classifiers that can integrate threshold-based filtering mechanisms, enabling the use of the "value" metric with any model capable of providing confidence scores alongside its predictions.

In formalizing "value", we will progressively make a few assumptions that (i) allow to simplify the presentation of the problem without altering the essence of the concepts, (ii) are reasonable in many if not most use cases, and (iii) make the definition of the value function easier to understand and interpret for the users who eventually have to deploy ML into their companies. We scope the conversation on classification problems as it makes it easy to ground the examples and terminology, and because it is easier to define a notion of accuracy. This is important: people understand accuracy because it is simple, and that has beneficial value even if accuracy is "inaccurate" as a metric, and most users will not be able to express complex value functions. Note however that our results also apply to other performance measures, like F1-score, as we will show in our experimental evaluation.

3.2 Definition of value

We have a classifier g that operates on test examples $x \in \mathcal{D}$ and returns either a predicted class $y \in \mathcal{Y}$ or a special label y_r, denoting "rejection" of the prediction. Then, we can compute the average value per the prediction of applying a model g over $\mathcal{D}$ as follows:

$$V(g, \mathcal{D}) = \rho V_r + (1 - \rho)(\alpha V_c + (1 - \alpha)V_w) \tag{1}$$

where ρ is the proportion of items in $\mathcal{D}$ that are rejected by g (classified as y_r). The term α denotes the accuracy of predictions that exceed the threshold. V_r refers to the value associated with rejecting an item, independently of the correctness of its prediction, and thus resorting to a default path, typically involving a human expert. V_c is the value of correctly classifying an item, which is only granted for non-rejected items. Finally, V_w is the value of an incorrect classification, which again is only granted (or rather, paid) for non-rejected items. Although these values can be expressed in monetary terms, such as dollars, we focus on their relative values to facilitate comparison between different models and learning strategies.

We define the baseline scenario as one in ML is not utilized, or equivalently, where all predictions are rejected. We set this baseline value to 0 ($V_r = 0$), which simplifies the process of evaluating a model by determining (i) whether it improves upon the baseline, and (ii) whether adopting AI is beneficial for the specific problem at hand.

$$V(g, \mathcal{D}) = (1 - \rho)(\alpha V_c + (1 - \alpha)V_w) \tag{2}$$

We also express V_w in terms of V_c, as in $V_w = -kV_c$, where k is a constant telling us how bad is an error with respect to getting the correct prediction:

$$V(g, \mathcal{D}) = V_c(1 - \rho)(\alpha - k(1 - \alpha)) \tag{3}$$

In the value formula, V_c acts as a scaling factor. When evaluating an AI-powered solution workflow, the specific magnitude of this factor is less critical. Instead, we consider the value relative to a unit of V_c dollars, effectively normalizing V_c to focus primarily on value. Thus, we can discuss value in terms of "value per dollar unit of rejection cost (detrimental value)" denoted as $V' = V/V_c$. To simplify further without deviating from the equations, we set $V_c = 1$. Therefore, we obtain:

$$V(g, \mathcal{D}) = (1 - \rho)(\alpha - k(1 - \alpha)) \tag{4}$$

Eq. 4 embodies the same concepts as Equation 1, streamlining our presentation.

3.3 Filtering by threshold

We now focus on the most common situation observed in practice; the model selectivity is applied by thresholding confidence values and rejecting predictions that have confidence c_y less than a threshold τ (case (a) in Fig. 2). We are given a model m that processes items $x \in \mathcal{D}$ and returns a vector of confidences (one per class). This is the output of a softmax;

for each x, we consider the pair y, c_y corresponding to the top-level prediction of $m(x)$ and the confidence associated with the prediction. Given a threshold τ, we define a function s:

$$s(y, c_y, \tau) = \begin{cases} y & \text{if } c_y \geq \tau, \\ y_r & \text{otherwise.} \end{cases}$$

where y_r is the special class label denoting "rejection" of the prediction. Our classifier g is therefore now expressed in terms of m and τ. This means that we can express the value as a function of $m, \mathcal{D}, \tau$. In a given use case, when we are given m and have knowledge of k, we select the threshold $\tau \in [0, 1]$ that optimizes $V(g, \mathcal{D})$ (We assume τ is unique or we randomly pick one if not). Thus, we can express the value of our classification logic as a function of $(m, \mathcal{D}, k)$:

$$V(m, \mathcal{D}, k) = (1 - \rho_\tau)(\alpha_\tau - k(1 - \alpha_\tau)) \tag{5}$$

Notice that τ can be set empirically on some tuning dataset $\mathcal{D}$ (it depends on $m, \mathcal{D}, k$), and ρ_τ and α_τ reflect the proportions ρ and α given τ. However, if we are aware of the properties of confidence vectors, we can set τ regardless of $\mathcal{D}$. For example, if we assume perfect calibration (where the expected accuracy for a prediction of confidence c is c) de Menezes e (Silva Filho et al. 2021), then we know that the threshold is at the point where the value of accepting a prediction is greater than zero, and $\alpha_\tau = \tau$. This means that to have $V(m, \mathcal{D}, k) > 0$ we need $\tau - k + k\tau > 0$, which means

$$\tau > k/(k + 1) \tag{6}$$

This conforms to intuition: if k is large, it never makes sense to predict, better go with the default. If k=0 (no cost for errors), we might always predict since there is no penalty for applying inaccurate predictions. Perhaps paradoxically, this case where inaccurate predictions are harmless is when *accuracy* is the metric we want to use. If k=1 (errors are the mirror image of correct predictions), then our threshold is 0.5. Figure 3a shows how a simple threshold-based selector can be adapted to maximize model value. In most real-world settings, especially for complex models, the available classifier will not be perfectly calibrated. In these cases, the threshold can be chosen by either recalibrating the model first using existing recalibration approaches de Menezes e Silva Filho et al. (2021) and then applying Eq. (6), or directly maximizing Eq. (5) over a separate validation set before testing the classifier. We will evaluate both strategies in our experimental evaluation (see Sect. 4.2.3).

In deriving the threshold, we initially assumed that all errors incur equal costs. However, we will next demonstrate how this derivation can be readily adapted to cost-sensitive settings.

3.4 Cost-sensitive value and thresholds

In this section, we extend the discussion on the value and optimal threshold to the setting in which different errors have different costs (and possibly, different correct predictions have different beneficial values). We focus on the binary classification setting for simplicity, but the reasoning can be easily generalized to multiclass classification. In cost-sensitive learn-

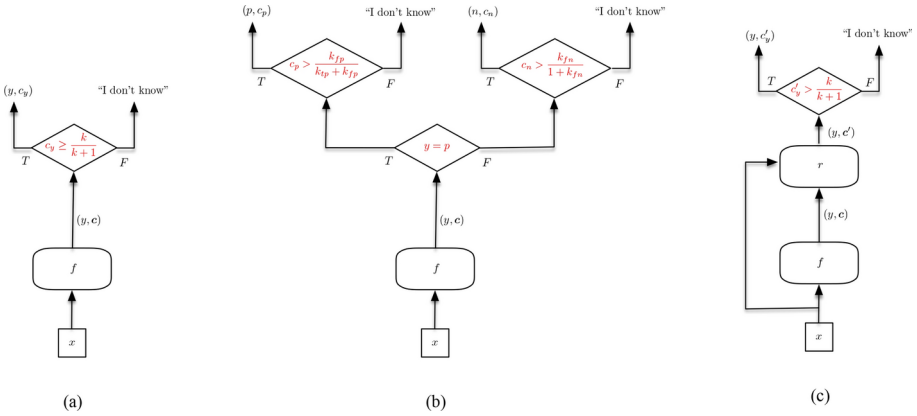

Fig. 3 Adapting selective classifiers to maximize value: **a** threshold-based selector, **b** cost-sensitive threshold-based selector; **c** recalibrator + threshold-based selector. Changes with respect to standard counterparts are highlighted in red

ing, the standard approach is that of giving a specific cost to each type of error and correct prediction (in which case the "cost" is the benefit Ling and Sheng (2010). We adapt this strategy to the value case, by providing a specific value for each possible type of error and correct prediction. The cumulative value of a selective classifier g on a dataset $\mathcal{D}$ can be written as (setting $V_r = 0$ as in the cost-insensitive case):

$$V(g, \mathcal{D}) = (1 - \rho)(N_{tp}V_{tp} + N_{tn}V_{tn} + N_{fp}V_{fp} + N_{fn}V_{fn})$$

where $N_{tp}, N_{tn}, N_{fp}, N_{fn}$ are the numbers of true positives, true negatives, false positives, and false negatives in $\mathcal{D}$, and $V_{tp}, V_{tn}, V_{fp}, V_{fn}$ are the values associated to the corresponding predictions. Let V_c be the base cost for a correct prediction. This is typically associated with a correctly predicted negative instance, i.e., $V_{tn} = V_c$. We can define the other values as multiples of this base cost as follows:

$$V_{tp} = k_{tp}V_c, \quad V_{fp} = -k_{fp}V_c, \quad V_{fn} = -k_{fn}V_c$$

for some user-defined and application-specific constants k_{tp}, k_{fp}, k_{fn}. The cumulative value simplifies as:

$$\begin{aligned} V(g, \mathcal{D}) &= (1 - \rho)(N_{tp}k_{tp}V_c + N_{tn}V_c - N_{fp}k_{fp}V_c - N_{fn}k_{fn}V_c) \\ &= (1 - \rho)V_c(k_{tp}N_{tp} + N_{tn} - k_{fp}N_{fp} - k_{fn}N_{fn}) \end{aligned}$$

Setting $V_c = 1$ (unit of value) as in the cost-insensitive case, we get:

$$V(g, \mathcal{D}) = (1 - \rho)(k_{tp}N_{tp} + N_{tn} - k_{fp}N_{fp} - k_{fn}N_{fn})$$

Let's now focus on the standard setting of a classifier rejecting by threshold. Note that we need to set class-specific thresholds τ_p and τ_n for positive and negative predictions respec-

tively to account for the different costs. Consider an instance x predicted as positive by the classifier. Its expected value (according to the predictions in $\mathcal{D}$) is given by:

$$\begin{aligned} V(g,x) &= (1-\rho)(k_{tp}N_{tp}/N_p - k_{fp}(N_{fp}/N_p) \\ &= (1-\rho)(k_{tp}N_{tp}/N_p - k_{fp}(1-N_{tp}/N_p)) \\ &= (1-\rho)(N_{tp}/N_p(k_{tp}+k_{fp}) - k_{fp}) \end{aligned}$$

where we normalized N_{tp} and N_{fp} by N_p, the number of positive instances in $\mathcal{D}$, to turn them into probabilities, and we removed the terms containing N_{tn} and N_{fn} as their corresponding probabilities are zero if the instance is predicted as positive.

If the classifier is perfectly calibrated, we know that $N_{tp}/N_p = \tau_p$. A positive value for the instance is thus achieved by setting τ_p as:

$$\tau_p > \frac{k_{fp}}{k_{tp}+k_{fp}} \tag{7}$$

Similarly, if x is predicted as negative by the classifier, it is expected value is given by:

$$\begin{aligned} V(g,x) &= (1-\rho)(N_{tn}/N_n - k_{fn}N_{fn}/N_n) \\ &= (1-\rho)(N_{tn}/N_n - k_{fn}(1-N_{tn}/N_n)) \\ &= (1-\rho)(N_{tn}/N_n(1+k_{fn}) - k_{fn}) \end{aligned}$$

where N_n is the number of negative instances in the training set. If the classifier is perfectly calibrated, we know that $N_{tn}/N_n = \tau_n$. A positive value for the instance is thus achieved by setting τ_n as:

$$\tau_n > \frac{k_{fn}}{1+k_{fn}} \tag{8}$$

Figure 3b shows how to adjust a threshold-based selector to maximize value in a cost-sensitive setting. We assumed a binary classification setting for simplicity, but the derivation can be easily extended to account for class-specific thresholds in multiclass classification.

4 Experiments

We now explore how adopting a value-oriented perspective influences model evaluation and application. Specifically, we aim to address the following questions:

Q1 Is model accuracy (or F1-score) a sensible indicator of the value of a model?

Q2 Is cost-sensitive error a sensible indicator of the value of a model in cost-sensitive settings?

Q3 How does calibration affect the value of a model?

Q4 How does predicting in an out-of-distribution setting affect the value of a model?

Our experimental evaluation is focused on NLP classification tasks, for which we analyze the behavior of simple as well as state-of-the-art models over various datasets, models, and text encoders. This choice stems from the broad diffusion of NLP models in companies, and

from our experience (see Sect. 4.3) in industrial use cases that were all NLP-based. However, the concept of value can be applied to any ML model deployed in a practical application, and we believe that the main results of our experimental evaluation hold for many other domains. We refer the reader to our GitHub repo[1] for the companion code.

4.1 Experimental Setup

Datasets and Tasks Table 1 presents a summary of the characteristics of the datasets we employed and their corresponding classification tasks. Additional information is provided in the following.

- *Hate-speech detection on Twitter.* We replicated the original tests from Arango et al. (2019) where we analyzed two widely used models (Agrawal and Awekar 2018; Badjatiya et al. 2017) and tested them on the Waseem et al. Waseem and Hovy (2016) dataset. However, we could only recover 9668 of the tweets as of October 2021 (the dataset size is 14949 in the original paper).
- *Clickbait detection.* The *Clickbait Challenge* on the *Webis Clickbait Corpus 2017*[2] was classifying Twitter posts as a clickbait or not. Both training and test sets are publicly available[3], while each team was free to choose a subset of the training set for validation (we followed the "blobfish" team).
- *Multi-Domain Sentiment Analysis - and Dataset (MDS).* Sentiment analysis based on a dataset for domain adaptation.[4] The data includes four categories of Amazon products (DVD, Books, Electronics, and Kitchen). The task is to learn sentiment from one of these domains and test it on the others.

Models and text encoders. For each task in our experiments, we use different models (see Table 2 and the accompanying code repository for details). Since we do not train models and use the validation set only to determine the optimal threshold, we do not perform standard cross-validation. The optimal threshold is selected by evaluating the model

[1] https://github.com/burcusayin/value-of-ml-models/

Table 1 Statistics of the datasets used in the experiments

Task	Dataset	Train/Val/Test size
Classifying tweets as "hate", and "non-hate" (binary)	Hate Speech	7734/967/967
Classify Twitter posts to detect clickbait (binary)	Clickbait	17600/4395/18979
Sentiment analysis on Amazon product reviews (3-class; positive, negative, and neutral)	MDS Electronics	2000/200/3386
	MDS DVD	2000/200/4265
	MDS Books	2000/200/5481
	MDS Kitchen	2000/200/5745

[2] https://webis.de/data/webis-clickbait-17.html.

[3] https://zenodo.org/record/5530410#.YWcFtC8RrRV.

[4] http://nlpprogress.com/english/domain_adaptation.html.

Table 2 Models used in the experiments

Dataset	Models	Model details
Hate-speech detection	Badjatiya et al. (2017), Agrawal and Awekar (2018)	Leader-board models
Clickbait detection	fullnetconc, weNet, lingNet, fullNet	Leader-board models
MDS	mttri (Ruder and Plank 2018)	Leader-board
	Google's T5-base	Fine-tuned for sentiment analysis
	SieBERT	Fine-tuned RoBERTa-large
	LogR, MLP1, MLP4	From scikit-learn library
	GPT-3	Fine-tuned for sentiment analysis

on the validation set across a range of candidate thresholds and choosing the one that maximizes performance.

- For the hate-speech dataset, we test the following leaderboard models: (i) Badjatiya et al. (2017) which uses an RNN to construct word embeddings and then classify them with Gradient-Boosted Decision Tree. In the original paper, test accuracy is measured as the average of the ten folds in cross-validation; however, in our reproduction, we separated validation and test set before cross-validation, and they are used for evaluation only after training. (ii) one model from Agrawal and Awekar (2018) which is composed of an embedding layer followed by a Bidirectional LSTM and a fully connected layer with softmax activation.
- For the clickbait detection dataset, we test 4 models from one leaderboard team on clickbait challenge: *fullnetconc*, *weNet*, *lingNet*, and *fullNet* which are published on Github.[5] This team modified the task into binary classification - they categorized items with a score under 0.5 into "non-clickbaiting", and vice versa.
- For the MDS dataset, we referred to the leaderboard for the sentiment analysis task of Domain adaptation[6] and tested the best-performing leader-board model, *Multi-task tri-training (mttri)* by Ruder and Plank (2018), that leverages multi-task learning strategies to improve the performance of tri-training. As the source code of other competing approaches was not publicly available, we compared mttri with three baseline models from the scikit-learn library[7]: (i) a simple Logistic Regression model (*LogR*); (ii) a basic MLP with a single hidden layer (*MLP1*); (iii) an MLP with four hidden layers (*MLP4*). All models where tested with a simple *TF-IDF* encoding.

[5] https://github.com/clickbait-challenge/blobfish.

[6] nlpprogress.com/english/domain_adaptation.html.

[7] https://scikit-learn.org/

4.2 Results

4.2.1 Q1: Accuracy and F1-score are poor indicators of model value

We first investigate whether standard performance metrics, like accuracy and F1-score, are sensible indicators of the value of the model, and how this depends on the magnitude of the cost factor k. Following the simplification in Sect. 3.2, we set $V_r = 0$ and $V_c = 1$, and use the threshold in Eq. 6 to decide whether to accept or reject each prediction given a certain k.

Table 3 report results in terms of accuracy, F1-score, and value for different values of $k \in [0, 10]$. As expected, the value of a model decreases substantially with the increase of the cost factor, with many models achieving *negative* value for larger values of k. Note that a model is useful only if its value exceeds 0; otherwise, it is deemed unnecessary, and the system can proceed without it. We want to stress that the cost factors we considered are fairly small and definitely realistic. For instance, setting $k = 4$ means that "being wrong is 4 times as bad" with respect to the advantage of being right. Many scenarios have values of k way more extreme (e.g., in medical decision support systems Sutton et al. 2020). Notice that accuracy corresponds to the case where we do not reject any predictions, which corresponds to setting $k = 0$, a rather unrealistic scenario.

Another major finding is that accuracy is a quite poor proxy of value even in relative terms. Boldface numbers indicate the best performing model in terms of the different met-

Table 3 Accuracy and F1-score results compared with value computed for increasing values of the cost factor k

Task	Model	Accuracy	F1	Value					
				$k = 0$	$k = 1$	$k = 2$	$k = 4$	$k = 8$	$k = 10$
Hate Speech	Badj et al.	**0.822**	**0.626**	**0.822**	**0.644**	**0.51**	**0.362**	**0.272**	**0.217**
	Agr et al	0.732	0.621	0.732	0.464	0.22	− 0.213	− 1.081	−1.499
Clickbait	Fullnetconc	**0.857**	**0.684**	**0.857**	**0.715**	0.564	0.286	0.041	0.013
	weNet	0.852	0.672	0.852	0.703	0.561	0.306	0.04	0.011
	LingNet	0.82	0.565	0.82	0.64	0.442	0.079	0.0	0.0
	FullNet	0.856	0.663	0.856	0.713	**0.588**	**0.367**	**0.061**	**0.015**
MDS Electronics	LogReg	0.762	0.736	0.762	0.524	0.339	0.162	0.053	0.033
	MLP1	0.749	0.711	0.749	0.497	0.327	**0.18**	**0.081**	**0.062**
	MLP4	0.735	0.713	0.735	0.47	0.24	− 0.143	− 0.78	− 1.06
	mttri	**0.808**	**0.786**	**0.808**	**0.616**	**0.441**	0.148	− 0.354	− 0.58
MDS DVD	LogReg	0.74	**0.739**	0.74	0.48	**0.283**	0.122	0.038	0.027
	MLP1	0.728	0.732	0.728	0.457	0.274	**0.133**	**0.054**	**0.038**
	MLP4	0.72	0.724	0.72	0.439	0.202	− 0.158	− 0.737	−0.981
	mttri	**0.753**	0.725	**0.753**	**0.506**	0.28	− 0.123	− 0.84	−1.166
MDS Books	LogReg	0.704	0.678	0.704	0.408	0.228	**0.102**	**0.022**	**0.015**
	MLP1	0.691	0.662	0.691	0.382	0.134	0.013	− 0.017	−0.013
	MLP4	0.696	0.681	0.696	0.393	0.154	− 0.171	− 0.666	− 0.86
	mttri	**0.742**	**0.712**	**0.742**	**0.484**	**0.254**	− 0.16	− 0.869	−1.215
MDS Kitchen	LogReg	0.782	0.771	0.782	0.565	0.374	0.176	0.06	0.034
	MLP1	0.765	0.752	0.765	0.53	0.337	0.164	**0.07**	**0.044**
	MLP4	0.761	0.758	0.761	0.521	0.312	0.003	− 0.478	−0.685
	mttri	**0.821**	**0.832**	**0.821**	**0.642**	**0.489**	**0.235**	− 0.192	−0.384

For each dataset and metric, the best performance is highlighted in bold

rics. It is clear that the best performing model is largely dependent on the cost factor, and that accuracy quickly becomes totally unreliable as a metric to identify the most appropriate model to employ. Replacing accuracy with F1-score does not change much. While we do observe substantially lower values for the unbalanced datasets (Hate Speech and Clickbait), the best performing model is unchanged almost everywhere.

4.2.2 Q2: Cost-sensitive error is a poor indicator of model value in cost-sensitive settings

The previous evaluation assumed equal cost for the different types of error. This is however rarely the case in practical applications, where false negative errors (e.g., undiagnosed diseases) can be far more costly than false positive ones (i.e., false alarms). Section 3.4 shows how to adapt value to this cost-sensitive setting, and how to determine cost-sensitive thresholds that are specific for each predicted class. In the following we evaluate the value of models in this cost-sensitive setting. We replace accuracy and F1, which are clearly inappropriate in this setting, with cost-sensitive error Elkan (2001) a popular performance measure in the cost-sensitive learning literature. Cost-sensitive error is obtained by computing the weighted sum of errors, with the weights given by the corresponding cost, i.e. $(N_{fn}k_{fn} + N_{fp}k_{fp})/|\mathcal{D}|$, where we divide by $|\mathcal{D}|$ to remove the dependency on the size of the dataset. For simplicity, and consistently with common practice in the literature, we set $k_{fp} = 1$ and vary $k_{fn} \in [1, 10]$. Results are shown in Table 4. While cost-sensitive error identifies different best performing models for different values of the cost, in only one case (MDS Kitchen) it consistently agrees with value across the spectrum of costs. What is worse, for large values of k_{fn} it often detects as best performing models that actually achieve *negative* value, making it a poor overall indicator of model value. The problem is not how it treats the costs of different errors, but in the fact that it does not assume a selective classifier and a corresponding cost-sensitive rejection threshold, which is the main practical contribution of our definition of value. This also implies that cost sensitive learning (He and Ma 2013), that aims at training classifiers to minimize (a certain notion of) cost-sensitive error, should be coupled with learning to reject mechanisms Hendrickx et al. (2021) in order to be fully effective in optimizing the value of the learned models.

4.2.3 Q3: Lack of calibration substantially affects model value

The threshold in Eq. 6 assumes that models are perfectly calibrated, which is often far from being true for trained models, and deep learning models in particular (Guo et al. 2017). In order to evaluate the role of calibration in determining value of a model, we apply temperature scaling (Guo et al. 2017), a simple yet effective recalibration technique, to each model before applying the threshold (the resulting selector is shown in Fig. 3c). Table 5 reports the results in exactly the same setting as Table 3, but using recalibrated models. Notice that accuracy and F1-score are unchanged, as temperature scaling affects the confidence in the prediction but not how classes are being ranked. In terms of value, however, we observe an overall improvement, quite substantial for larger values of k. Note that the effectiveness of calibration in improving the model's "value" depends on the accuracy of the calibrated model. The degenerate behaviour of models with negative values is almost completely eliminated, with "useless" models receiving a value of zero, as expected. These results suggest

Table 4 Comparison between cost-sensitive error and value for different values of $k = k_{fn}$ (with $k_{fp} = 1$)

Task	Model	Cost-sensitive error					Value				
		$k=1$	$k=2$	$k=4$	$k=8$	$k=10$	$k=1$	$k=2$	$k=4$	$k=8$	$k=10$
Hate Speech	Badj et al.	**0.178**	**0.297**	0.535	1.01	1.248	**0.644**	**0.545**	**0.389**	**0.315**	**0.278**
	Agr et al.	0.268	0.322	**0.429**	**0.644**	**0.752**	0.464	0.405	0.32	0.157	0.098
Clickbait	fullnetconc	**0.143**	**0.221**	**0.377**	**0.689**	**0.845**	**0.715**	0.608	0.368	0.131	**0.103**
	WeNet	0.148	0.228	0.388	0.707	0.867	0.703	0.604	0.381	0.124	0.094
	LingNet	0.18	0.295	0.524	0.983	1.213	0.64	0.467	0.125	0.052	0.052
	FullNet	0.144	0.234	0.416	0.779	0.961	0.713	**0.631**	**0.446**	**0.15**	**0.103**
MDS Electronics	LogReg	0.238	0.406	0.742	1.413	1.749	0.524	0.442	**0.355**	**0.293**	**0.282**
	MLP1	0.259	0.436	0.791	1.5	1.854	0.497	0.413	0.338	0.284	0.274
	MLP4	0.254	0.418	0.745	1.4	1.727	0.47	0.33	0.09	−0.313	−0.492
	mttri	**0.192**	**0.33**	**0.607**	**1.159**	**1.436**	**0.616**	**0.495**	0.286	−0.085	−0.245
MDS DVD	LogReg	0.26	0.394	0.663	1.201	1.47	0.48	**0.375**	0.295	0.255	**0.251**
	MLP1	0.271	0.404	0.67	1.203	1.469	0.457	0.36	**0.298**	**0.26**	**0.251**
	MLP4	0.278	**0.392**	**0.62**	**1.075**	**1.303**	0.439	0.327	0.16	−0.089	−0.193
	mttri	**0.247**	0.412	0.744	1.406	1.737	**0.506**	0.352	0.072	−0.431	−0.663
MDS Books	LogReg	0.296	0.489	0.874	1.645	2.03	0.408	**0.332**	**0.269**	**0.222**	**0.219**
	MLP1	0.303	0.492	0.87	1.627	2.005	0.382	0.272	0.197	0.18	0.183
	MLP4	0.312	0.486	**0.832**	**1.525**	**1.871**	0.393	0.258	0.081	−0.183	−0.283
	mttri	**0.258**	**0.45**	0.834	1.603	1.987	**0.484**	0.32	0.018	−0.52	−0.789
MDS Kitchen	LogReg	0.218	0.345	0.599	1.108	1.363	0.565	0.466	0.365	0.306	0.295
	MLP1	0.242	0.375	0.64	1.171	1.436	0.53	0.433	0.339	0.292	0.279
	MLP4	0.248	0.387	0.665	1.22	1.498	0.521	0.416	0.263	0.026	−0.076
	mttri	**0.179**	**0.238**	**0.355**	**0.59**	**0.708**	**0.642**	**0.589**	**0.503**	**0.376**	**0.31**

For each dataset and metric, the best performance is highlighted in bold

that learning models should always be recalibrated before being incorporated in practical workflows. This does not mean that one can then resort on standard accuracy or F1-score to choose which model to employ. The best performing model is still largely dependent on the cost factor. Notice that in the domain adaptation scenarios (MDS tasks), simple logistic regression (LogReg) consistently outperforms all other models for large values of k. This result should not be unexpected. Logistic regression is known to be a well-calibrated model per-se Kull et al. (2017), and temperature scaling likely further improves this behaviour, while more complex models struggle to achieve comparable calibration with simple recalibration strategies. The lively research area of calibration in machine learning and especially deep learning can provide useful solutions to this problem de Menezes e (Silva Filho et al. 2021).

Table 5 Comparison between accuracy, F1-score and value for recalibrated models

Task	Model	Accuracy	F1	Value					
				$k=0$	$k=1$	$k=2$	$k=4$	$k=8$	$k=10$
Hate Speech	Badj et al.	**0.822**	**0.626**	**0.822**	**0.644**	**0.513**	**0.359**	**0.268**	**0.218**
	Agr et al.	0.732	0.621	0.732	0.464	0.207	0.0	0.0	0.0
Clickbait	Fullnetconc	**0.857**	**0.684**	**0.857**	**0.715**	**0.608**	**0.488**	**0.374**	0.331
	WeNet	0.852	0.672	0.852	0.703	0.597	0.472	0.357	0.326
	LingNet	0.82	0.565	0.82	0.64	0.499	0.348	0.173	0.115
	FullNet	0.856	0.663	0.856	0.713	0.6	**0.488**	0.372	**0.335**
MDS Electronics	LogReg	0.762	0.736	0.762	0.524	0.362	**0.226**	**0.119**	**0.098**
	MLP1	0.745	0.711	0.745	0.491	0.33	0.174	0.096	0.062
	MLP4	0.745	0.713	0.745	0.491	0.291	0.11	0.0	0.0
	mttri	**0.808**	**0.786**	**0.808**	**0.616**	**0.447**	0.192	0.112	0.0
MDS DVD	LogReg	0.74	**0.739**	0.74	0.48	**0.315**	**0.17**	**0.09**	**0.062**
	MLP1	0.729	0.732	0.729	0.459	0.28	0.148	0.037	0.023
	MLP4	0.722	0.724	0.722	0.443	0.235	0.056	0.0	0.0
	mttri	**0.753**	0.725	**0.753**	**0.506**	0.292	0.08	0.0	0.0
MDS Books	LogReg	0.704	0.678	0.704	0.408	0.234	**0.111**	**0.01**	**0.001**
	MLP1	0.697	0.662	0.697	0.395	0.199	0.002	0.0	0.0
	MLP4	0.688	0.681	0.688	0.375	0.095	0.0	0.0	0.0
	mttri	**0.742**	**0.712**	**0.742**	**0.484**	**0.264**	− 0.011	0.0	0.0
MDS Kitchen	LogReg	0.782	0.771	0.782	0.565	0.41	**0.267**	**0.153**	**0.127**
	MLP1	0.758	0.752	0.758	0.515	0.345	0.197	0.096	0.011
	MLP4	0.752	0.758	0.752	0.504	0.305	0.122	0.0	0.0
	mttri	**0.821**	**0.832**	**0.821**	**0.642**	**0.493**	0.227	0.102	0.0

Comparison between accuracy, F1-score and value for recalibrated models. For each dataset and metric, the best performance is highlighted inbold

4.2.4 Q4: Operating in an out-of-distribution setting substantially affects model value

The lack of calibration in machine learning models is known to be particularly harmful when the model operates in an out-of-distribution (OOD) setting (Tomani and Buettner 2019; Wu et al. 2022), and the results on the domain adaptation tasks in Table 5 confirm this issue. To better understand the role of the OOD setting in determining the value of models, we thus focused on the MDS tasks and complemented the set of models presented in Table 5 with some state-of-the-art transformer models, which should be less affected by the problem given the huge corpora on which they are trained. The transformer models that we employed are the following:

- Google's T5-base[8] Raffel et al. (2020) (12-layers, 768-hidden-state, 3072 feed-forward hidden-state, 12-heads, 220 M parameters) fine-tuned on IMDB dataset[9] Maas et al. (2011) for sentiment analysis task.
- SieBERT[10]Heitmann et al. (2020): a fine-tuned version of RoBERTa-large[11] model Liu

[8] https://tinyurl.com/t5-base-finetuned-sentiment.

[9] https://huggingface.co/datasets/stanfordnlp/imdb.

[10] https://tinyurl.com/SieBERT-sentiment.

[11] https://huggingface.co/FacebookAI/roberta-large.

et al. (2019) (24-layer, 1024-hidden-state, 16-heads, 355 M parameters) for sentiment analysis task that is fine-tuned and evaluated on 15 diverse text sources.

- GPT-3 Brown et al. (2020). Since it is producing human-like text for a given input, we fine-tuned it using the OpenAI API[12]. First, we prepared the MDS dataset for GPT-3; we cleaned sentences that have more than 2049 tokens, and renamed the text column as "*prompt*" and the ground truth column as "*completion*". Then, we used OpenAI API to fine-tune GPT-3 separately on each of the 4 domains (DVD, books, electronics, and kitchen). We specified "$classification\_n\_classes$" parameter as 2 and $classification\_positive\_class$ as '1' so that the API tunes GPT-3 for binary sentiment analysis. Fine-tuning 4 models on the MDS dataset costs a total of $7.15. In order to test the fine-tuned models on different target domains, we specified the *prompt* in the format of "sentence + -> " because the API itself uses " ->" sign to teach GPT-3 that the sentiment for a *prompt* is (' ->') the *completion*. Thus, fine-tuned GPT-3 models produce either 0 or 1 for the given input. Testing each fine-tuned model on the other 3 domains (so, 12 cases in total) costs $43.89. We provide our source code on Github[13] to show every step of using GPT-3 in our experiments.

Table 6 reports the results of all models on the MDS tasks. As expected, large pre-trained language models tend to perform well across the board. This can be due to two reasons (besides the models being very powerful): (i) we know that very large models with very large train datasets are reasonably well calibrated (e.g. Jiang et al. 2021), and (ii) when the training data is so large, fewer examples are out of distribution in terms of language. For example, GPT-3 Brown et al. (2020) is trained on about 45TB of text data from various datasets, and the vocabulary of the MDS datasets is most likely already present in its training set.

Notice however that even for these models, accuracy is a poor proxy of value when k is large. Indeed, SieBERT slightly outperforms GPT-3 in terms of both accuracy and F1 in all tasks. However, the situation is reversed for large values of k, with SieBERT reaching negative values in most cases, most likely because of a poorer calibration with respect to GPT-3. Finally, simple linear models occasionally outperform these powerful (and very expensive to employ) large-language models for the largest values of k, again confirming the importance of value in determining the most appropriate model for the situation at hand.

4.3 Key takeaways for AI-assisted decision-making

While our experiments are conducted in a controlled setting, they are designed to reflect realistic decision-making scenarios relevant to enterprise environments, such as those in ServiceNow. Consider, for instance, an application that assesses or explains risk levels (e.g., the risk of applying a system patch). The utility of AI outputs in this context depends on the nature of potential errors:

- *Correct Assessments:* AI provides accurate risk evaluations, aiding decision-makers (e.g., Change Approvers) in making informed choices.
- *Low-Value Outputs:* AI offers insights that, while accurate, do not significantly aid de-

[12] https://openai.com/api/

[13] https://github.com/burcusayin/value-of-ml-models/

Table 6 Comparison between accuracy, F1-score and value in an OOD setting. LogRef, MLP1, ML4 and mttri are trained to perform domain adaptation and thus operate in a OOD setting, while transformer models (T5, SieBERT, GPT-3) are pre-trained or large corpora and thus likely operate in-distribution

Task	Model	Accuracy	F1	Value				
				$k=1$	$k=2$	$k=4$	$k=8$	$k=10$
MDS Electronics	LogReg	0.762	0.736	0.524	0.339	0.162	0.053	0.033
	MLP1	0.745	0.711	0.497	0.327	0.18	0.081	**0.062**
	MLP4	0.745	0.713	0.47	0.24	− 0.143	− 0.78	− 1.06
	mttri	0.808	0.786	0.616	0.441	0.148	− 0.354	− 0.58
	T5	0.784	0.765	0.568	0.352	− 0.08	− 0.944	−1.376
	SieBERT	**0.842**	**0.831**	**0.685**	**0.527**	0.217	− 0.397	−0.705
	GPT-3	0.82	0.803	0.641	0.499	**0.322**	**0.127**	0.051
MDS DVD	LogReg	0.74	0.739	0.48	0.283	0.122	0.038	0.027
	MLP1	0.729	0.732	0.457	0.274	0.133	0.054	0.038
	MLP4	0.722	0.724	0.439	0.202	− 0.158	− 0.737	−0.981
	mttri	0.753	0.725	0.506	0.28	− 0.123	− 0.84	−1.166
	T5	0.789	0.788	0.578	0.367	− 0.056	− 0.9	−1.323
	SieBERT	**0.836**	**0.832**	**0.672**	0.508	0.193	− 0.436	−0.747
	GPT-3	0.832	0.825	0.664	**0.534**	**0.367**	**0.164**	**0.089**
MDS Books	LogReg	0.704	0.678	0.408	0.228	0.102	0.022	**0.015**
	MLP1	0.697	0.662	0.382	0.134	0.013	− 0.017	−0.013
	MLP4	0.688	0.681	0.393	0.154	− 0.171	− 0.666	− 0.86
	mttri	0.742	0.712	0.484	0.254	− 0.16	− 0.869	−1.215
	T5	0.77	0.791	0.541	0.311	− 0.148	− 1.066	−1.525
	SieBERT	**0.826**	**0.827**	**0.652**	**0.479**	0.136	− 0.547	−0.879
	GPT-3	0.806	0.808	0.613	0.46	**0.272**	**0.077**	0.004
MDS Kitchen	LogReg	0.782	0.771	0.565	0.374	0.176	0.06	0.034
	MLP1	0.758	0.752	0.53	0.337	0.164	0.07	0.044
	MLP4	0.752	0.758	0.521	0.312	0.003	− 0.478	−0.685
	mttri	0.821	0.832	0.642	0.489	0.235	− 0.192	−0.384
	T5	0.777	0.768	0.555	0.332	− 0.113	− 1.004	−1.449
	SieBERT	**0.865**	**0.859**	**0.73**	0.595	0.328	− 0.195	−0.454
	GPT-3	0.853	0.851	0.706	**0.599**	**0.464**	**0.308**	**0.251**

For each dataset and metric, the best performance is highlighted in bold

cision-making.

- *Erroneous Assessments:* AI produces misleading risk evaluations (e.g., downplaying a high-risk change), potentially leading to poor decisions.

To mitigate the impact of errors, we apply a cost-based evaluation framework that assigns heavily negative weights to erroneous assessments-especially those that underestimate risks-relative to the positive weights for correct assessments. This reflects a deliberate design principle: it is preferable to provide no assistance than to offer misleading guidance.

We determine whether to deploy a model by setting penalties such that a positive overall score indicates a net beneficial impact. While this introduces cost as an additional parameter, it aligns with standard model evaluation practices, where accuracy and utility thresholds guide deployment decisions. Importantly, this framework prioritizes the decision-maker's

needs, resulting in more instances of model rejection (when thresholds are unmet) rather than erroneous inferences.

Notably, we have yet to encounter a use case where correct and erroneous assessments are assigned equal absolute weights by product managers. Similarly, non-inference (a model opting out) is rarely considered as detrimental as providing incorrect guidance. These observations suggest that our evaluation framework aligns with real-world utility considerations.

5 Limitations and conclusion

In this paper, we investigated whether (i) model accuracy or F1-score serves as a reliable proxy for evaluating the true value of ML models, (ii) cost-sensitive error provides a meaningful measure of model value in cost-sensitive scenarios, (iii) calibration influences the value of ML models, and (iv) predictions in out-of-distribution settings impact model value. Our study focused on binary and multi-class classification tasks, employing various models under different cost settings. The findings revealed that (i) accuracy and F1-score are poor indicators of model value, (ii) cost-sensitive error is also an inadequate measure of model value, (iii) poor calibration significantly diminishes model value, and (iv) operating in out-of-distribution settings considerably undermines model value.

The takeaway from our experiments is that using accuracy-oriented metrics (that is, metrics that assume models are applied without rejection) is as a minimum a risky proposition - and this is true even for models widely acknowledged as "leaders". We should always assess models over a range of cost factors, and at least for reasonable cost factors we expect based on the set of application use cases we are targeting. $k = 0$ (accuracy) is almost never a reasonable one. We also saw how applying models without thresholding can lead to a negative value, and that threshold tuning seems to perform better than calibration. We also hypothesize and have obtained some support for identifying complexity and out-of-distribution as factors that may lead to rapid model quality degradation for higher cost factors.

This being said, we see this work more as providing evidence of a problem and outlining the research needs: more studies (especially with large models and in vs out of distribution datasets) are needed to validate the hypothesis and a deeper understanding of how calibration, confidence distribution, and size of validation set affect model value.

Author contributions Burcu Sayin: Conceptualization, Implementation, Experimental Evaluation, Writing—original draft, Writing—review & editing. Jie Yang: Conceptualization, Supervision, Writing - review & editing. Xinyue Chen: Experimental Evaluation, Writing—original draft. Andrea Passerini: Conceptualization, Funding acquisition, Supervision, Writing—original draft, Writing—review & editing. Fabio Casati: Conceptualization, Funding acquisition, Supervision, Writing - original draft, Writing - review & editing.

Funding Open access funding provided by Università degli Studi di Trento within the CRUI-CARE Agreement. Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Health and Digital Executive Agency (HaDEA). Neither the European Union nor the granting authority can be held responsible for them. Grant Agreement no. 101120763 - TANGO. Grant Agreement No. 952215 - TAILOR. The work of Burcu Sayin was partially supported by the project AI@Trento (FBK-Unitn). AP also acknowledges the support of the MUR PNRR project FAIR - Future AI Research (PE00000013) funded by the NextGenerationEU.

Declarations

Conflict of interest The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential Conflict of interest.

References

Agrawal S, Awekar A (2018) Deep learning for detecting cyberbullying across multiple social media platforms. In: Pasi G, Piwowarski B, Azzopardi L, Hanbury A (eds) Advances in information retrieval. Springer, Cham, pp 141–153

Arango A, Pérez J, Poblete B (2019) Hate speech detection is not as easy as you may think: A closer look at model validation. In: Proceedings of the 42nd International ACM SIGIR Conference on research and development in information retrieval. SIGIR'19, pp. 45–54. Association for Computing Machinery New York, NY, USA.https://doi.org/10.1145/3331184.3331262

Badjatiya P, Gupta S, Gupta M, Varma V (2017) Deep learning for hate speech detection in tweets. In: Proceedings of the 26th International Conference on World Wide Web Companion. WWW '17 Companion, pp. 759–760. International World Wide Web Conferences Steering Committee Republic and Canton of Geneva, CHE.https://doi.org/10.1145/3041021.3054223

Bahat Y, Shakhnarovich G (2020) Classification confidence estimation with test-time data-augmentation. ArXiv **abs/2006.16705**. https://doi.org/10.48550/ARXIV.2006.16705

Balda E, Behboodi A, Mathar R (2020) Adversarial examples in deep neural networks: An overview. In: Deep learning: algorithms and applications, pp. 31–65. https://doi.org/10.1007/978-3-030-31760-7_2

Bendel R, Higgins S, Teberg J, Pyke D (1989) Comparison of skewness coefficient, coefficient of variation, and Gini coefficient as inequality measures within populations. Oecologia 78:394–400. https://doi.org/10.1007/BF00379115

Bragg J, Mausam Weld D.S (2016) Optimal testing for crowd workers. In: Proceedings of the 2016 international conference on autonomous agents & multiagent systems. AAMAS '16, pp. 966–974. International foundation for autonomous agents and multiagent systems Richland, SC

Brown T.B, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler D.M, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. ArXiv **abs/2005.14165**. https://doi.org/10.48550/ARXIV.2005.14165

Bukowski M, Kurek J, Antoniuk I, Jegorowa A (2021) Decision confidence assessment in multi-class classification. Sensors 21:3834. https://doi.org/10.3390/s21113834

Callaghan W, Goh J, Mohareb M, Lim A, Law E (2018) Mechanicalheart: a human-machine framework for the classification of phonocardiograms. In: CSCW'18, vol. 2, pp. 28–12817. https://doi.org/10.1145/3274297

Casati F, Noel P, Yang J (2021) On the value of ml models. In: Neurips workshop on human decisions. https://doi.org/10.48550/ARXIV.2112.06775

Chai X, Deng L, Yang Q, Ling CX (2004) Test-cost sensitive naive bayes classification. In: Fourth IEEE international conference on data mining (ICDM'04), pp. 51–58. https://doi.org/10.1109/ICDM.2004.10092

Charoenphakdee N, Cui Z, Zhang Y, Sugiyama M (2021) Classification with rejection based on cost-sensitive classification. In: Proceedings of the 38th international conference on machine learning, vol. 139, pp. 1507–1517. https://proceedings.mlr.press/v139/charoenphakdee21a.html

Cheng J, Bernstein M.S (2015) Flock: Hybrid crowd-machine learning classifiers. In: Proceedings of the 18th Acm conference on computer supported cooperative work & social computing. https://doi.org/10.1145/2675133.2675214

Cordella LP, De Stefano C, Tortorella F, Vento M (1995) A method for improving classification reliability of multilayer perceptrons. IEEE Trans Neural Netw 6(5):1140–1147. https://doi.org/10.1109/72.410358
De Stefano C, Sansone C, Vento M (2000) To reject or not to reject: that is the question-an answer in case of neural classifiers. IEEE Trans Syst Man Cybern 30(1):84–94. https://doi.org/10.1109/5326.827457
Dellermann D, Ebel P, Söllner M, Leimeister JM (2019) Hybrid intelligence. Business Inform Syst Eng 61:637–643. https://doi.org/10.1007/s12599-019-00595-2
Dellermann D, Calma A, Lipusch N, Weber T, Weigel S, Ebel PA (2019) The future of human-ai collaboration: a taxonomy of design knowledge for hybrid intelligence systems. ArXiv **abs/2105.03354**
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. KDD '99, pp. 155–164. Association for computing machinery New York, NY, USA. https://doi.org/10.1145/312129.312220
Elkan C (2001) The foundations of cost-sensitive learning. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 2. IJCAI'01, pp. 973–978. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA
Elkan C (2001) The foundations of cost-sensitive learning. In: Proceedings of the 17th international joint conference on artificial intelligence, pp. 973–978
Fumera G, Roli F (2002) Support vector machines with embedded reject option. In: Proceedings of the first international workshop on pattern recognition with support vector machines. SVM '02, pp. 68–82. Springer Berlin, Heidelberg. https://doi.org/10.1007/3-540-45665-1_6
Gadiraju U, Yang J, Bozzon A (2017) Clarity is a worthwhile quality: On the role of task clarity in microtask crowdsourcing. In: Proceedings of the 28th ACM conference on hypertext and social media. HT '17, pp. 5–14. Association for computing machinery New York, NY, USA. https://doi.org/10.1145/3078714.3078715
Geifman Y, El-Yaniv R (2017) Selective classification for deep neural networks. Adv Neural Inform Proc Syst. https://doi.org/10.48550/ARXIV.1705.08500
Gunel B.S (2022) Towards reliable hybrid human-machine classifiers. PhD thesis at University of Trento. https://hdl.handle.net/11572/349843
Guo C, Pleiss G, Sun Y, Weinberger K.Q (2017) On calibration of modern neural networks. In: proceedings of the 34th international conference on machine learning - Volume 70. ICML'17, pp. 1321–1330.https://doi.org/10.48550/ARXIV.1706.04599
Han L, Roitero K, Gadiraju U, Sarasua C, Checco A, Maddalena E, Demartini G (2021) The impact of task abandonment in crowdsourcing. IEEE Trans Knowl Data Eng 33(5):2266–2279. https://doi.org/10.1109/TKDE.2019.2948168
Han L, Maddalena E, Checco A, Sarasua C, Gadiraju U, Roitero K, Demartini G (2020) Crowd worker strategies in relevance judgment tasks. In: Proceedings of the 13th International conference on web search and data mining. WSDM '20, pp. 241–249. Association for computing machinery New York, NY, USA. https://doi.org/10.1145/3336191.3371857
Han L, Roitero K, Gadiraju U, Sarasua C, Checco A, Maddalena E, Demartini G (2019) All those wasted hours: On task abandonment in crowdsourcing. In: Proceedings of the twelfth ACM international conference on web search and data mining. WSDM '19, pp. 321–329. Association for Computing Machinery New York, NY, USA. https://doi.org/10.1145/3289600.3291035
Heitmann M, Siebert C, Hartmann J, Schamp C (2020) More than a feeling: benchmarks for sentiment analysis accuracy. In: Communication & Computational Methods eJournal
Hellman ME (1970) The nearest neighbor classification rule with a reject option. IEEE Trans Syst Sci Cybern 6(3):179–185. https://doi.org/10.1109/TSSC.1970.300339
He H, Ma Y (2013) Imbalanced learning: foundations, algorithms, and applications
Hendrickx K, Perini L, Plas D, Meert W, Davis J (2021) Machine learning with a reject option: a survey. arXiv
Jiang Z, Araki J, Ding H, Neubig G (2021) How can we know when language models know? On the calibration of language models for question answering. Trans Assoc Comput Linguist 9:962–977. https://doi.org/10.1162/tacl_a_00407
Jiang H, Kim B, Guan M.Y, Gupta M (2018) To trust or not to trust a classifier. In: Proceedings of the 32nd international conference on neural information processing systems. NIPS'18, pp. 5546–5557. Curran Associates Inc. Red Hook, NY, USA. https://doi.org/10.48550/ARXIV.1805.11783
Kamar E, Hacker S, Horvitz E (2012) Combining human and machine intelligence in large-scale crowdsourcing. In: AAMAS'12 - Volume 1, pp. 467–474
Krivosheev E, Casati F, Benatallah B (2018) Crowd-based multi-predicate screening of papers in literature reviews. In: Proceedings of the 2018 World wide web conference. WWW '18, pp. 55–64. International world wide web conferences steering committee republic and canton of Geneva, CHE. https://doi.org/10.1145/3178876.3186036

Kull M, Silva Filho T, Flach PA (2017) Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. Electr J Stat 11:5052–5080

Kull M, Perello Nieto M, Kängsepp M, Silva Filho T, Song H, Flach P.: Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In: Wallach H, Larochelle H, Beygelzimer A, Alché-Buc F, Fox E, Garnett R. (eds.) (2019) Advances in neural information processing systems, vol. 32. https://proceedings.neurips.cc/paper/2019/file/8ca01ea920679a0fe3728441494041b9-Paper.pdf

Li H (2013) Error rate analysis of labeling by crowdsourcing. In: International conference on machine learning (ICML2013), Workshop on machine learning meets crowdsourcing

Ling C, Sheng V (2010) Cost-sensitive learning and the class imbalance problem. Encycl Mach Learn

Liu Q, Ihler A.T, Steyvers M (2013) Scoring workers in crowdsourcing: How many control questions are enough? In: Burges C.J, Bottou L, Welling M, Ghahramani Z, Weinberger K.Q. (eds.) Advances in neural information processing systems, vol. 26. https://proceedings.neurips.cc/paper/2013/file/cc1aa436277138f61cda703991069eaf-Paper.pdf

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, Stoyanov V (2019) Roberta: a robustly optimized bert pretraining approach. ArXiv **abs/1907.11692**. https://doi.org/10.48550/ARXIV.1907.11692

Maas A.L, Daly R.E, Pham P.T, Huang D, Ng A.Y, Potts C (2011) Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual meeting of the association for computational linguistics: human language technologies, pp. 142–150. Association for computational linguistics Portland, Oregon, USA. https://aclanthology.org/P11-1015

Nagar Y, Malone TW (2011) Making business predictions by combining human and machine intelligence in prediction markets. In: International conference on interaction sciences

Nagar Y, Malone T.W (2012) Improving predictions with hybrid markets. In: AAAI fall symposium: machine aggregation of human judgment

Ng AY (2004) Feature selection, l1 vs. l2 regularization, and rotational invariance. In: Proceedings of the Twenty-First International Conference on Machine Learning. ICML '04, p. 78. Association for Computing Machinery New York, NY, USA. https://doi.org/10.1145/1015330.1015435

Nuñez A.C (2022) Combining diverse forms of human and machine intelligence. In: PhD thesis at Massachusetts institute of technology

Qarout R, Checco A, Bontcheva K (2018) Investigating stability and reliability of crowdsourcing output. In: Proceedings of the 1st workshop on disentangling the relation between crowdsourcing and bias management (CrowdBias 2018) Co-located the 6th AAAI conference on human computation and crowdsourcing (HCOMP 2018)

Qiu S, Gadiraju U, Bozzon A (2020) Improving worker engagement through conversational microtask crowdsourcing. In: Proceedings of the 2020 CHI conference on human factors in computing systems. CHI '20, pp. 1–12. Association for computing machinery New York, NY, USA. https://doi.org/10.1145/3313831.3376403

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learn Res. https://doi.org/10.48550/ARXIV.1910.10683

Raghu M, Blumer K, Corrado G, Kleinberg J.M, Obermeyer Z, Mullainathan S (2019) The algorithmic automation problem: prediction, triage, and human effort. CoRR **abs/1903.12220**. arXiv:1903.12220

Rodriguez C, Daniel F, Casati F (2014) Crowd-based mining of reusable process model patterns. In: Business process management, pp. 51–66. https://doi.org/10.1007/978-3-319-10172-9_4

Ruder S, Plank B (2018) Strong baselines for neural semi-supervised learning under domain shift. In: The 56th annual meeting of the association for computational linguistics (ACL 2018), pp. 1044–1054. https://doi.org/10.18653/v1/P18-1096

Sayin B, Krivosheev E, Passerini JYA, Casati F (2021) A review and experimental analysis of active learning over crowdsourced data. Artif Intel Rev 54:5283–5305. https://doi.org/10.1007/s10462-021-10021-3

Sayin B, Casati F, Passerini A, Yang J, Chen X (2022) Rethinking and recomputing the value of ml models. arXiv preprint arXiv:2209.15157

Sayin B, Krivosheev E, Ramírez J, Casati F, Taran E, Malanina V, Yang J (2021) Crowd-powered hybrid classification services: Calibration is all you need. In: 2021 IEEE International conference on web services (ICWS), pp. 42–50. https://doi.org/10.1109/ICWS53863.2021.00019

Sayin B, Yang J, Passerini A, Casati F (2021) The science of rejection: a research area for human computation. In: The 9th AAAI conference on human computation and crowdsourcing. HCOMP 2021. https://doi.org/10.48550/ARXIV.2111.06736

Sayin B, Yang J, Passerini A, Casati F (2023) Value-aware active learning. In: Frontiers in artificial intelligence and applications. Volume 368: HHAI 2023: Augmenting Human Intellect, pp. 215–223

Sayin B, Yang J, Passerini A, Casati F (2023) Value-based hybrid intelligence. In: Frontiers in artificial intelligence and applications. Volume 368: HHAI 2023: Augmenting Human Intellect, pp. 366–370

Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27(3):379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

Sheng V.S, Ling C.X (2006) Thresholding for making classifiers cost-sensitive. In: Proceedings of the 21st National conference on artificial intelligence - Volume 1. AAAI'06, pp. 476–481

Silva Filho T, Song H, Perelló-Nieto M, Santos-Rodríguez R, Kull M, Flach P.A (2021) Classifier calibration: How to assess and improve predicted class probabilities: a survey. CoRR **abs/2112.10327**

Suri M (2022) PiCkLe at SemEval-2022 task 4: Boosting pre-trained language models with task specific metadata and cost sensitive learning. In: Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pp. 464–472. Association for Computational Linguistics Seattle, United States. https://doi.org/10.18653/v1/2022.semeval-1.63

Sutton R.T, Pincock D, Baumgart D.C, Sadowski D, Fedorak R, Kroeker K (2020) An overview of clinical decision support systems: benefits, risks, and strategies for success. npj Digital Medicine **3**

Teerapittayanon S, McDanel B, Kung H.T (2017) Branchynet: fast inference via early exiting from deep neural networks. ArXiv **abs/1709.01686**. https://doi.org/10.48550/ARXIV.1709.01686

Thai-Nghe N, Gantner Z, Schmidt-Thieme L (2010) Cost-sensitive learning methods for imbalanced data. In: The 2010 international joint conference on neural networks (IJCNN), pp. 1–8. https://doi.org/10.1109/IJCNN.2010.5596486

Ting KM (1998) Inducing cost-sensitive trees via instance weighting. In: Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery. PKDD '98, pp. 139–147. Springer Berlin, Heidelberg. https://doi.org/10.1007/BFb0094814

Tomani C, Buettner F (2019) Towards trustworthy predictions from deep neural networks with fast adversarial calibration. In: AAAI conference on artificial intelligence

Tu C.-Y, Lin H.-T (2020) Cost learning network for imbalanced classification. In: 2020 international conference on technologies and applications of artificial intelligence (TAAI), pp. 47–51. https://doi.org/10.1109/TAAI51410.2020.00017

Waseem Z, Hovy D (2016) Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In: Proceedings of the NAACL Student Research Workshop, pp. 88–93. Association for Computational Linguistics San Diego, California. https://doi.org/10.18653/v1/N16-2013. https://aclanthology.org/N16-2013

Whitehill J, Wu T.-f, Bergsma J, Movellan J, Ruvolo P (2009) Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: Bengio Y, Schuurmans D, Lafferty J, Williams C, Culotta A. (eds.) Advances in neural information processing systems, vol. 22. https://proceedings.neurips.cc/paper/2009/file/f899139df5e1059396431415e770c6dd-Paper.pdf

Wilder B, Horvitz E, Kamar E (2021) Learning to complement humans. In: Proceedings of the Twenty-Ninth international joint conference on artificial intelligence. IJCAI'20. https://doi.org/10.48550/ARXIV.2005.00582

Wu M.-H, Quinn A.J (2017) Confusing the crowd: Task instruction quality on amazon mechanical turk. In: AAAI Conference on human computation & crowdsourcing

Wu Y, Zeng Z, He K, Mou Y, Wang P, Xu W (2022) Distribution calibration for out-of-domain detection with Bayesian approximation. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 608–615. International committee on computational linguistics Gyeongju, Republic of Korea. https://aclanthology.org/2022.coling-1.50

Yang J, Redi J, Demartini G, Bozzon A (2016) Modeling task complexity in crowdsourcing. In: AAAI conference on human computation & crowdsourcing

Zadrozny B, Langford J, Abe N (2003) Cost-sensitive learning by cost-proportionate example weighting. In: Proceedings of the Third IEEE international conference on data mining. ICDM '03, p. 435. IEEE Computer Society USA. https://doi.org/10.1109/ICDM.2003.1250950

Zhou D, Basu S, Mao Y, Platt J (2012) Learning from the wisdom of crowds by minimax entropy. In: Pereira F, Burges C.J, Bottou L, Weinberger K.Q. (eds.) Advances in neural information processing systems, vol. 25. https://proceedings.neurips.cc/paper/2012/file/46489c17893dfdcf028883202cefd6d1-Paper.pdf