AUC-based Selective Classification

Andrea Pugnana[†] and Salvatore Ruggieri[*]

[†] Scuola Normale Superiore, Pisa, Italy
andrea.pugnana@sns.it
[*] University of Pisa, Pisa, Italy
salvatore.ruggieri@unipi.it

With the increasing usage of machine learning models, final users must trust their predictions. A possible way to enhance trust is to allow a model to abstain from predicting if not confident enough. This requires adding a selection mechanism to the model, which decides when a prediction will be provided. Such a framework is generally called learning to reject (3) or selective classification (10).

This is especially relevant in socially-sensitive scenarios, such as healthcare or finance, where we often deal with imbalanced (binary) classes (22). In these settings, traditional metrics to evaluate selective classification, such as error rate, can be misleading, and other measures should be preferred. For instance, one of the most common metrics for imbalanced scenarios is the Area Under the ROC Curve (AUC) (42), which measures the classifier's ability to rank instances from minority and majority classes correctly.

This paper - accepted at AISTATS 2023 - presents a novel methodology to perform selective classification specific to AUC.

The work makes several contributions to the current literature on selective classification as:

- we study the problem of improving AUC once we allow for rejection from a theoretical perspective;
- we present two methods, i.e. PlugInAUC and AUCross that guarantee improvements in terms of AUC;
- we empirically show that PlugInAUC and AUCross succeed in increasing AUC in imbalanced contexts, while all existing state-of-the-art methods fail;
- we hint that this is due to a disparity in rejection rate across different classes, as traditional methods tend to reject minority class instances more often.

Abstract. Selective classification (or classification with a reject option) pairs a classifier with a selection function to determine whether or not a prediction should be accepted. This framework trades off coverage (probability of accepting a prediction) with predictive performance, typically measured by distributive loss functions. In many application scenarios, such as credit scoring, performance is instead measured by ranking metrics, such as the Area Under the ROC Curve (AUC). We propose a model-agnostic approach to associate a selection function to a given probabilistic binary classifier. The approach is specifically targeted at optimizing the AUC. We provide both theoretical justifications and a novel algorithm, called AUCROSS, to achieve such a goal. Experiments show that our method succeeds in trading-off coverage for AUC, improving over existing selective classification methods targeted at optimizing accuracy.

Keywords: Selective classification · Learning with a reject option · Trustworthy AI

1 Introduction

The predictive performance of a classifier is typically not homogeneous over the data distribution. Identifying sub-populations with low performance is helpful, e.g. for debugging and monitoring purposes. In many socially sensitive application scenarios, we can consider not predicting at all for these sub-populations as an alternative to poor or even harmful predictions. Such applications include credit scoring, curriculum screening, access to public benefits, and medical diagnoses. *Selective classification* (or *classification with a reject option*) (3; 34) pairs a classifier with a selection function to determine whether a prediction should be accepted or the classifier should abstain. The selection function assesses the trustworthiness of a prediction. The literature on selective classification has mainly considered distributive loss functions. However, in many real-world scenarios, such as credit scoring (11), the quality of a binary probabilistic classifier is concerned with the discriminative power of the ranking induced by its scores (25). This is measured by the Area Under the ROC Curve (AUC) (12) or, equivalently, by the Gini coefficient (41). Algorithms designed to minimize, e.g., the error rate may not lead to the best possible AUC values (6).

In this paper, we first introduce the problem of *AUC-based selective classification*, which consists of inducing a selective classifier which optimizes the AUC over the sub-population of accepted predictions while guaranteeing a minimum probability mass (called *coverage*) of such sub-population. We then focus on an instance of the AUC-selective classification problem, where we assume that: *(i)* the classification algorithm is given, including its hyper-parameters, and *(ii)* the selection function abstains on a range of the scores (*score-based selective functions*). Assumption *(i)* allows lifting an existing classifier to a selective classifier. In this respect, our approach is model-agnostic, as we do not make assumptions about the classification algorithm. Assumption *(ii)* is widespread in selective classification, supported by theoretical results (10). Such an assumption allows for a theoretical analysis of the AUC-based selective classification problem. We devise

AUCross, a model-agnostic algorithm for estimating the bounds of score-based selective functions that maximize the AUC for target coverage. The approach is based on a cross-fitting strategy over the training set. Bound estimates are supported by a theoretical analysis of (sufficient) conditions for abstaining that lead to an increase in the AUC. Experiments show that the approach performs very close to an oracle score-based selective function (which has access to the true class of instances), and it outperforms existing selective classification approaches targeted at optimizing accuracy.

The paper is organized as follows. Section 2 briefly surveys related work. Section 3 recalls basic concepts of selective classification and it introduces the AUC-based selective classification problem. Our theoretical approach is presented in Section 4, while we introduce AUCross algorithm in Section 5. Section 6 reports experimental results. Finally, we draw conclusions and outline possible extensions.

2 Related Work

The two main models of selective classification are the cost model (3) and the the bounded-improvement model (34). In the former, the goal is to minimize the expected cost, assuming a cost for misclassification and a cost for abstention, or a more refined cost based on the confusion matrix (39). In the Bayes optimal selective classifier, the selection function abstains when the posterior probability of the predicted class is below some threshold. Posterior probabilities can be estimated on a validation set by the *plug-in* rule (24). In the bounded-improvement model, the selection function is evaluated based on the probability mass of the accepted region (*coverage*) and the expected loss over such a region (*selective risk*). One can maximize coverage for a maximum target risk or minimize risk for a minimum target coverage (17). (15) establish the equivalence of cost-based and bounded-improvement models.

Most approaches for selective classification are model-specific in that they build the classifier and the selection function concurrently. Examples include methods for Support Vector Machines (SVM) (16), boosted decision trees (5), Deep Neural Networks (DNNs) (17; 18; 33; 26). See (23) for a complete overview of existing methods. Moreover, the definitions of cost or risk in selective classification have been provided using distributive loss functions, where the loss is defined for every prediction in isolation. AUC is a metric about the ranking induced by a classifier, for which the loss is determined for pairs of instances. To the best of our knowledge, the only work directly addressing AUC selective classification is (37). However, the selection function is used here to accept or to abstain from ranking *pairs* of instances. The AUC to be optimized is defined as the mean correct order of accepted pairs (an extension of the Mann-Whitney U-statistics): a same instance may appear in both an accepted and a rejected pair. We target, instead, either accepting or rejecting predictions for each single instance. The AUC we optimize is the mean correct order of any pair of accepted cases (i.e., the Mann-Whitney U-statistics over the accepted region).

In this paper, we adhere to the bounded-improvement model, with the AUC as the metric to optimize by abstaining on *single instances*, and we take a model-agnostic view of the problem. Notice that AUC-based selective classification is orthogonal to the many approaches for optimizing AUC in supervised learning (43).

3 Background

Consider random variables $(\mathbf{X}, Y) \in \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is a feature space and $\mathcal{Y} = \{0, 1, \dots, n_\mathcal{Y}\}$ a finite label space. The joint distribution of $(\mathbf{X}, Y) \sim \mathcal{D}$ is unknown, but we can observe one or more datasets of i.i.d. realizations. A *classifier* is a function $h : \mathcal{X} \to \mathcal{Y}$ that maps features to classes, computed from an hypothesis space and a dataset (training set). The expected loss over the distribution $R(h) = \int_{\mathcal{X}\times\mathcal{Y}} l(h(\mathbf{x}), y) d\mathcal{D}(\mathbf{x}, y) = \mathbb{E}_\mathcal{D}[l(h(\mathbf{X}), Y)]$ is called the *risk*, where $l : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a loss function. The risk can be estimated starting from a test set $S_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ through the *empirical risk* $\hat{r}(h|S_n) = \frac{1}{n}\sum_{i=1}^n l(h(\mathbf{x}_i), y_i)$. Such a canonical setting is extended to model situations where predictions of classifiers are not sufficiently reliable, and the option of abstaining is preferable. A *selective classifier* is a pair (h, g), where h is a classifier and $g : \mathcal{X} \to \{0, 1\}$ is a *selection function*, which determines when to accept/abstain from using h:

$$(h, g)(\mathbf{x}) = \begin{cases} h(\mathbf{x}) & \text{if } g(\mathbf{x}) = 1 \\ \text{abstain} & \text{otherwise} \end{cases}$$

A soft selection approach (17) consists of defining g in terms of a *confidence function* $k_h : \mathcal{X} \to [0, 1]$ (the subscript highlights that k_h depends on h) and a threshold θ of minimum confidence for accepting:

$$g(\mathbf{x}) = \mathbb{1}(k_h(\mathbf{x}) > \theta) \tag{1}$$

A good confidence function should order instances based on descending loss, i.e., if $k_h(\mathbf{x}_i) \leq k_h(\mathbf{x}_j)$ then $l(h(\mathbf{x}_i), y_i) \geq l(h(\mathbf{x}_j), y_j)$. The *coverage* of a selective classifier is $\phi(g) = E_\mathcal{D}[g(\mathbf{X})]$, i.e., the expected probability mass of the accepted region. The *selective risk* is risk normalized by coverage:

$$R(h, g) = \frac{\mathbb{E}_\mathcal{D}[l(h(\mathbf{X}), Y)g(\mathbf{X})]}{\phi(g)} = \mathbb{E}_\mathcal{D}[l(h(\mathbf{X}), Y)|g(\mathbf{X})]$$

Empirical coverage and empirical selective risk are respectively defined as follows:

$$\hat{\phi}(g|S_n) = \frac{\sum_{i=1}^n g(\mathbf{x}_i)}{n}$$

$$\hat{r}(h, g|S_n) = \frac{\frac{1}{n}\sum_{i=1}^n l(h(\mathbf{x}_i), y_i)g(\mathbf{x}_i)}{\hat{\phi}(g|S_n)}$$

By defining $S_m^g = \{(\mathbf{x}_i, y_i) \in S_n \mid g(\mathbf{x}_i) = 1\}$, the empirical coverage is $\hat{\phi}(g|S_n) = |S_m^g|/n = m/n$ and the empirical selective risk reduces to $\hat{r}(h, g|S_n) = \hat{r}(h|S_m^g)$, namely to the empirical risk over the accepted instances. The inherent trade-off between risk and coverage is summarized by the *risk-coverage curve* (10). The selective classification problem can be framed by fixing an upper bound to the selective risk and looking for a selective classifier that maximizes coverage. (17)

show how to convert this framing into an alternative one, where a lower bound c for coverage is fixed (*target coverage*), and then we look for a selective classifier that minimizes selective risk. We adhere to such a formulation. Called θ the parameter(s) defining h and g (e.g., as in the soft selection approach), the *selective classification problem* is stated as:

$$\min_{\theta} R(h, g) \quad \text{s.t.} \quad \phi(g) \geq c$$

Let us now extend the framework to the AUC metric. We consider binary classes (0/1, negatives/positives) and probabilistic classifiers, where $h(\mathbf{x}) \in [0, 1]$ is an estimate of the probability that $\mathbf{x}$ is positive. Let $\mathcal{D}_1$ and $\mathcal{D}_0$ be the conditional distributions of positives and negatives, respectively. The AUC can be defined as the probability that a randomly drawn positive receives a higher score than a randomly drawn negative, conditioned to the fact that both are selected according to g, i.e.

$$\begin{aligned} AUC(h, g) = \mathbb{E}_{\mathbf{X}_0 \sim \mathcal{D}_0, \mathbf{X}_1 \sim \mathcal{D}_1}[\mathbb{1}(h(\mathbf{X}_1) > h(\mathbf{X}_0))| \\ g(\mathbf{X}_0) = 1, g(\mathbf{X}_1) = 1] \end{aligned}$$

The *AUC-selective classification problem* can be stated as:

$$\max_{\theta} AUC(h, g) \quad \text{s.t.} \quad \phi(g) \geq c \tag{2}$$

A good confidence function should order instances based on ascending contribution to the AUC while allowing for controlling the target coverage. Empirical AUC given S_n can be defined by restricting to the set $S^g_m = \{(\mathbf{x}_i, y_i) \in S_n \mid g(\mathbf{x}_i) = 1\}$ of accepted instances directly from the definition above by resorting to the Mann-Whitney U-statistic: $\widehat{AUC}(h, g|S_n) = \widehat{AUC}(h|S^g_m) = \frac{1}{m^+} \frac{1}{m^-} \sum_{(\mathbf{x}_i,1) \in S^g_m} \sum_{(\mathbf{x}_i,0) \in S^g_m} \mathbb{1}(h(\mathbf{x}_i) > h(\mathbf{x}_j))$, where m^+ is the number of positives in S^g_m, and m^- is the number of negatives in S^g_m. An alternative calculation of $\widehat{AUC}$ is the Area Under the Receiver Operating Characteristic (ROC) Curve (12).

4 AUC Selective Classification

We consider the AUC-selective classification problem for binary probabilistic classifiers. We make the further assumption that the parameters of h are fixed, i.e., θ in (2) includes only the parameters of the selection function g. Let us denote by H a binary probabilistic classifier induction algorithm whose hyper-parameters are fixed. The algorithm induces the classifier h starting from a training set. $h(\mathbf{x})$ is the score assigned by h to an instance $\mathbf{x}$. We aim to lift h to a selective classifier (h, g) by calculating a selection function g from the following hypothesis space (*score-bounded selection functions*):

$$g(\mathbf{x}) = \begin{cases} 0 & \text{if } \theta_l \leq h(\mathbf{x}) \leq \theta_u \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

Intuitively, the selective classifier abstains if the score is within the bounds θ_l and θ_u.

To identify the bounds of g, we consider the following scenario, where we have a sample $S_n = \{\mathbf{x}_i, y_i\}_{i=1}^n$ and a classifier h. Our objective is to estimate a score-bounded selection function g such that: *(i)* $\hat{\phi}(g|S_n) \geq c$; and *(ii)*: $\widehat{AUC}(h, g|S_n)$ is as large as possible. Let us define $n = n^+ + n^-$, with n^+ (resp., n^-) the number of positive (resp., negative) instances in S_n. Let $p^+ = n^+/n$ be the fraction of positives and let $p^- = 1 - p^+$ be the fraction of negatives. Without loss of generality, we can assume that $\mathbf{x}_1, \ldots, \mathbf{x}_n$ is ordered by non-ascending scores, i.e., $h(\mathbf{x}_i) \geq h(\mathbf{x}_{i+1})$ for $i \in [1, n-1]$. The rank of an instance is $r(\mathbf{x}_i) = i$, and its relative rank is $r(\mathbf{x}_i)/n$. We write $t(\mathbf{x}_i)$ to indicate the number of positive instances occurring in ranks from 1 to i. The true positive rate (TPR) at $\mathbf{x}_i$ is the ratio $tpr(\mathbf{x}_i) = t(\mathbf{x}_i)/n^+$. We depart from reasoning directly on the definition of the AUC. Instead, we consider the linearly related definition of the Gini coefficient (11):

$$G = 2 \cdot AUC - 1 \tag{4}$$

for which the maximization problem is equivalent. The Gini coefficient (also known as Accuracy Ratio) is defined starting from the Cumulative Accuracy Profile (CAP), which maps relative rank to TPR. The Gini coefficient can be written as $G = A/(A + B)$ where A is the area of the CAP between the diagonal and the line of the classifier h, and B is the area between the line of h and the line of a perfect classifier $h^\star$. A perfect classifier assigns a score of 1 to positives and a score of 0 to negatives; hence it ranks all positives first, then all negatives afterwards. The diagonal line represents the performance of a random classifier. In summary, the Gini coefficient measures the discriminative power of a probabilistic classifier as a fraction of the difference in power between a perfect classifier and a random classifier. Using the CAP plot and the Gini coefficient will be crucial in proving our results. Figure 1 shows a sample CAP plot where the axes are ranks and true positives instead of their relative counterparts. The Gini coefficient can equivalently be computed as ratio $An^+n/(An^+n + Bn^+n)$. We observe that $(A + B)n^+n$ is $n^+n^+/2$ (the area of $h^\star$ for positives) plus n^-n^+ (the area of $h^\star$ for negatives) minus $n^+n/2$ (the area under the diagonal). Hence $A + B = n^-/(2n) = p^-/2$.

Let us define $r'(\mathbf{x}_i) = n - r(\mathbf{x}_i) + 1$, i.e. $r'(\mathbf{x}_i)$ is the rank over the non-descending scores. We show next a (sufficient) condition to improve $\widehat{AUC}(h, g|S_n)$ by abstaining on positive instances.

Proposition 1. *For any (number of) positive instance* $\mathbf{x}_i$ *such that* $g(\mathbf{x}_i) = 1$ *and:*

$$\frac{r'(\mathbf{x}_i)}{n} \leq \widehat{AUC}(h, g|S_n) \cdot p^- \tag{5}$$

we have: $\widehat{AUC}(h, g|S_n) \leq \widehat{AUC}(h, g'|S_n)$, *where* $g'(\mathbf{x}_i) = 0$ *and* $g'(\mathbf{x}) = g(\mathbf{x})$ *otherwise. The inequality is strict if at least one such* $\mathbf{x}_i$ *exists in* S_n.

Proof (Proof). By (4), we can equivalently show the result for the Gini coefficient. Let $G = A/(A+B)$, and let $\bar{G} = \bar{A}/(\bar{A}+\bar{B})$ be the Gini coefficient after removing

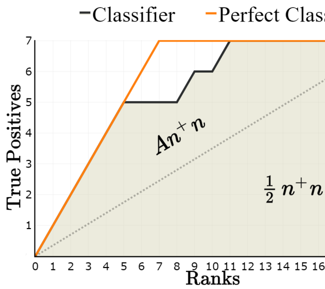

Fig. 1: Sample CAP plot with $n = 20$ and $n^+ = 7$.

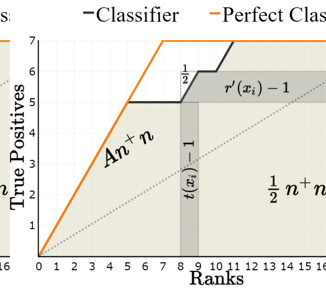

Fig. 2: CAP plot before removing a positive instance.

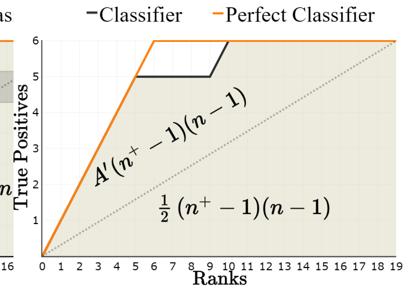

Fig. 3: CAP plot after removing a positive instance.

(a.k.a., abstaining on) one positive instance $\mathbf{x}_i$. We have $A + B = n^-/(2n)$ and $\bar{A} + \bar{B} = n^-/(2(n-1))$. Then $\bar{G} > G$ iff:

$$\bar{A}(n-1) > An \tag{6}$$

As $r'(\mathbf{x}_i) = n - r(\mathbf{x}_i) + 1$, we have that $\bar{A}$ is related to A as follows:

$$\bar{A} + \tfrac{1}{2} = (A + \tfrac{1}{2} - \left[\tfrac{(r'(\mathbf{x}_i)-1)+(t(\mathbf{x}_i)-1)+1/2}{n^+n}\right])\tfrac{n^+n}{(n^+-1)(n-1)} \tag{7}$$

Using such equality, the condition in (6) can be simplified to (see Appendix for full derivation):

$$\frac{r'(\mathbf{x}_i)}{n} \leq A + \frac{p^-}{2} + \frac{(n^+ - t(\mathbf{x}_i))}{n}$$

Since $n^+ \geq t(\mathbf{x}_i)$, the inequality above is satisfied if:

$$\frac{r'(\mathbf{x}_i)}{n} \leq A + \frac{p^-}{2} \tag{8}$$

By (4) and $G = A/(A+B) = 2A/p^-$, we have $A + {}^{p^-}\!/_2 = \widehat{AUC}(h, g|S_n) \cdot p^-$, and thus the conclusion (5) holds after removing *one* positive instance. Let us then consider the case when we remove $(1-\alpha)n^+$ positive instances, with $\alpha \in [0, 1[$. Inequality (8) becomes:

$$\frac{r'(\mathbf{x}_i)}{n} \leq A_\alpha \frac{(n^- + \alpha n^+)}{n} + \frac{p^-}{2} \tag{9}$$

where A_α is the numerator of the Gini fraction after removing $(1-\alpha)n^+$ positives. By repeatedly removing one positive instance at a time, (6) implies that $A_\alpha(n^- + \alpha n^+) > An$. Therefore, if (5) holds, then (9) holds for any α. Therefore, by removing as many as possible positive instances $\mathbf{x}_i$ such that (5) holds, we increase the Gini coefficient, and, a fortiori, the $\widehat{AUC}(h, g'|S_n)$.

A key step in the proof is equation (7). Let us give some intuitions. Abstaining on a positive instance means removing some areas in the CAP plot, as shown in grey in Figure 2. The vertical grey band consists of $t(\mathbf{x}_i) - 1$ cells in the nn^+ grid. The horizontal grey band consists of $r'(\mathbf{x}_i) - 1$ cells. In addition, an area of $1/2$ is removed from the cell with the increase of the classifier line. Finally, after removing those areas, the grid is rescaled from nn^+ to $(n^+ - 1)(n - 1)$, which provides the rescaling factor in (7). Figure 3 shows the CAP plot after removing a positive instance.

A similar result holds when removing negative examples. The condition is based on the TPR of the removed instance.

Proposition 2. *For any (number of) negative instance* $\mathbf{x}_i$ *such that* $g(\mathbf{x}_i) = 1$ *and:*

$$\frac{t(\mathbf{x}_i)}{n^+} \leq \widehat{AUC}(h, g|S_n) - \frac{1}{n^+} \tag{10}$$

we have: $\widehat{AUC}(h, g|S_n) \leq \widehat{AUC}(h, g'|S_n)$*, where* $g'(\mathbf{x}_i) = 0$ *and* $g'(\mathbf{x}) = g(\mathbf{x})$ *otherwise. The inequality is strict if at least one such* $\mathbf{x}_i$ *exists in* S_n*.*

In general, the selection functions g' in Prop. 1 and Prop. 2 are not score-bounded. Let us assume that $g(\mathbf{x}) = 1$ for all $\mathbf{x}$, hence $\widehat{AUC}(h, g|S_n) = \widehat{AUC}(h|S_n)$. We can easily lift condition (5) to scores as follows:

$$h(\mathbf{x}_i) \leq \hat{q}_n(\widehat{AUC}(h|S_n) \cdot p^-) = \theta_u$$

where $\hat{q}_n$ is the empirical quantile function over the scores $\{h(\mathbf{x}_i) \mid (\mathbf{x}_i, y_i) \in S_n\}$. As a consequence, g' is score bounded as it boils down to $g'(\mathbf{x}) = 0$ iff $0 \leq h(\mathbf{x}) \leq \theta_u$. Regarding (10), since TPR is anti-monotonic with the scores, we can restate the condition as follows:

$$\theta_l = \hat{q}_{n^+}(\widehat{AUC}(h|S_n) - {}^1\!/_{n^+}) \leq h(\mathbf{x}_i)$$

where $\hat{q}_{n^+}$ is the empirical quantile function over the scores of the positives $\{h(\mathbf{x}_i) \mid (\mathbf{x}_i, 1) \in S_n\}$. In this case, g' is score bounded as $g'(\mathbf{x}) = 0$ iff $\theta_l \leq h(\mathbf{x}) \leq 1$.

In summary, if $h(\mathbf{x}_i)$ is within the bounds θ_l and θ_u, then, irrespective whether $\mathbf{x}_i$ is positive or negative, by abstaining on $\mathbf{x}_i$ we obtain an increase in $\widehat{AUC}$. In summary, we have the following result.

Proposition 3. *Called* $\theta_l = \hat{q}_{n^+}(\widehat{AUC}(h|S_n) - {}^1\!/_{n^+})$ *and* $\theta_u = \hat{q}_n(\widehat{AUC}(h|S_n) \cdot p^-)$*, we define* $g(x)$ *as:*

$$g(\mathbf{x}) = \begin{cases} 0 & \text{if } \theta_l \leq h(\mathbf{x}) \leq \theta_u \\ 1 & \text{otherwise.} \end{cases}$$

Then we have: $\widehat{AUC}(h|S_n) \leq \widehat{AUC}(h, g|S_n)$*. The inequality is strict if at least one* $\mathbf{x}_i$ *exists in* S_n *such that* $g(\mathbf{x}_i) = 0$*.*

Algorithm 1: EstimateThetasAUC()

Input : $(\mathbf{s}, \mathbf{y})$ - scores and true class labels
Output : (θ_l, θ_u) - bounds for AUC-based selective classification

$n, n^+, p^- \leftarrow |\mathbf{y}|, |\mathbf{y} == 1|, 1 - n^+/n$
$\widehat{AUC} \leftarrow AUC.ROC(\mathbf{s}, \mathbf{y})$ // compute empirical AUC
$u \leftarrow \lfloor \widehat{AUC} \cdot p^- \cdot n \rfloor$ // θ_u position
$\mathbf{tpr} \leftarrow 1 - cum.sum(\mathbf{y}[order(\mathbf{s})])/n^+$ // compute true positive rates
$l \leftarrow search.sorted(\mathbf{tpr}, \widehat{AUC} - 1/n^+)$ // θ_l position
$\mathbf{ss} \leftarrow sort(\mathbf{s})$ // sort scores ascending
$\theta_l, \theta_u \leftarrow \mathbf{ss}[l], \mathbf{ss}[u]$ // bounds of g
return (θ_l, θ_u)

5 The AUCROSS algorithm

Alg. 1 shows a procedure to calculate the bounds stated in Prop. 3 starting from empirical scores and true class labels.

We devise an approach for lifting H to an AUC-selective classifier from this procedure. We call this induction algorithm AUCross. The approach differs from existing methods in several aspects. First, it aims to determine thresholds θ_l and θ_u specifically designed for the AUC selective classification problem (2). Second, we prevent setting apart a validation set from the training set (as done in state-of-the-art methods) to be able to fit the classifier h on the whole available training set. Since the goal is to estimate quantiles over the (unknown) population of scores of the classifier h, we approximate sampling from such a population by using a cross-fitting strategy, as in (35)[1]. Third, existing state-of-the-art methods calibrate the selection function by computing the empirical quantile of scores over a validation set. We adopt a quantile estimator based on subsamples, which improves over the variance of a full-sample quantile estimator. The theoretical backbone of our approach is based on the results by (31), reported next for completeness.

Theorem 4 ((31), Theorem 3). *Given a random sample distributed according to F, satisfying some regularity conditions, and K non-overlapping subsamples of it, let $\hat{q}(\alpha)$ be the empirical α-quantile estimator of F over the whole sample, and $\bar{q}(\alpha)$ a weighted average of the empirical quantile estimators of F over the subsamples. For $t \in [0, 1]$, let us define the linear combination $\tilde{q}(\alpha) = t\hat{q}(\alpha) + (1 - t)\bar{q}(\alpha)$. The variance of $\tilde{q}(\alpha)$ is minimized for $t = 1/\sqrt{2}$, $K = 2$ and equally sized subsamples.*

The sample quantile $\hat{q}(\alpha)$ is known to be asymptotically normal. The weighted mean of subsample quantiles $\bar{q}(\alpha)$ is first-order equivalent to $\hat{q}(\alpha)$. The above

[1] Exploiting cross-fitting is a common practice also in other fields, such as the double ML approach (2) for causal inference and the cross-conformal prediction algorithm (40).

theorem states the conditions for minimizing the variance of (the second-order term of) any linear combination of the two estimators.

We report the pseudo-code of AUCross in Alg. 2.

Algorithm 2: AUCross.fit()

Input : $(\mathbf{X}, \mathbf{y})$ - training set,
H - binary probabilistic classifier,
c - target coverage,
K - number of folds

Output : (h, g) - selective classifier

1 $Lbs, Ubs \leftarrow [], []$ // empty lists of bounds
2 $S \leftarrow \text{StratifiedKFold}((\mathbf{X}, \mathbf{y}), K)$ // stratified K-fold partition
3 **for** $\mathbf{X}_k, \mathbf{y}_k \in S$ **do** // for each fold
4 $\quad (\mathbf{X}'_k, \mathbf{y}'_k) = (\mathbf{X} - \mathbf{X}_k, \mathbf{y} - \mathbf{y}_k)$ // training data
5 $\quad h_k \leftarrow H.fit(\mathbf{X}'_k, \mathbf{y}'_k)$ // train k^{th} classifier
6 $\quad \mathbf{s}_k \leftarrow h_k.score(\mathbf{X}_k)$ // score test data
7 $\mathbf{s} \leftarrow \cup_{k=1}^{K} \mathbf{s}_k$ // store all the scores
8 $n \leftarrow |\mathbf{y}|$ // number of instances
9 $\theta_l, \theta_u \leftarrow EstimateThetasAUC(\mathbf{s}, \mathbf{y})$ // bounds for scores over whole training set
10 $J \leftarrow KFold((\mathbf{s}, \mathbf{y}), 2)$ // split scores and actual values in two sub-samples
11 **for** $(\mathbf{s}_j, \mathbf{y}_j) \in J$ **do** // for each sub-sample
12 $\quad \theta_{l_j}, \theta_{u_j} \leftarrow EstimateThetasAUC(\mathbf{s}_j, \mathbf{y}_j)$
13 $\theta_{u^*} \leftarrow \frac{1}{\sqrt{2}}\theta_u + (1 - \frac{1}{\sqrt{2}}) \sum_{i=1}^{2} \frac{\theta_{u_i}}{2}$ // estimate of θ_u
14 $\theta_{l^*} \leftarrow \frac{1}{\sqrt{2}}\theta_l + (1 - \frac{1}{\sqrt{2}}) \sum_{i=1}^{2} \frac{\theta_{l_i}}{2}$ // estimate of θ_l
15 $mid \leftarrow \lfloor n \cdot (\theta_{u^*} + \theta_{l^*})/2 \rfloor$ // compute mid point position
16 $\delta \leftarrow n \cdot (1 - c)/2$ // half-width of rejection area
17 $u' \leftarrow \min\{\lfloor mid + \delta \rfloor, n\}$ // upper bound position
18 $l' \leftarrow \max\{1, \lfloor mid - \delta \rfloor\}$ // lower bound position
19 $\mathbf{ss} \leftarrow sort(\mathbf{s})$ // sort scores ascending
20 $\theta_{l'}, \theta_{u'} \leftarrow \mathbf{ss}[l'], \mathbf{ss}[u']$ // bounds of g
21 $h \leftarrow H.fit(\mathbf{X}, \mathbf{y})$ // train classifier
22 $g \leftarrow$ **lambda** $\mathbf{x} : 1 - \mathbb{1}(\theta_{l'} \leq h.score(\mathbf{x}) \leq \theta_{u'})$ // selection function
23 **return** (h, g)

AUCross iterates over a stratified K-fold partitioning of the training set (lines 3-6 of Alg. 2), where, for each of the folds k, we use $K - 1$ folds to train a classifier h_k (using H) and to predict scores $h_k(\mathbf{X}_k)$ over the fold k (lines 4-5). We store all these predictions in $\mathbf{s}$ (line 7), and we use them to compute quantiles according to Prop. 3 (line 9 and Alg. 1). Since θ_l and θ_u are quantiles estimates, we can improve their second-order behaviour by randomly splitting the obtained scores and the target variable $(\mathbf{s}, \mathbf{y})$ into two parts (line 10) and repeat the estimation of quantiles over subsamples (lines 11-12). We combine the

Table 1: Experimental datasets.

Dataset	# Features	Training Size	Test Size	Positive Rate
Adult	55	30,162	15,060	.246
LendingClub	65	1,364,697	445,912	.225
GiveMe	12	112,500	37,500	.067
UCICredit	23	22,500	7,500	.221
CSDS1 (1)	155	230,409	76,939	.144
CSDS2 (1)	35	37,100	12,533	.018
CSDS3 (1)	144	71,177	23,288	.253
CatsVsDogs	64x64	20,000	5,000	.500
CIFAR-10-cat	32x32	50,000	10,000	.100

estimates according to Thm. 4 to obtain the final quantiles θ_{l^*} and θ_{u^*} (lines 13-14).

The quantile estimation methods considered so far, and in particular θ_l and θ_u in Prop. 3, are intended to maximize the AUC independently of the desired coverage c. To address the minimum coverage constraint c, we centre the rejection area at the midpoint between the instances with score θ_{l^*} and θ_{u^*} (line 15). Such a mid point mid is the median of the distribution of scores from θ_{l^*} up to θ_{u^*}. We consider a rejection area $[l', u']$ centered in mid and with width $n \cdot (1 - c)$, also checking not to exceed the range $[1, n]$ (lines 16-18). The final selective classifier (h, g) is then obtained by: *(i)* fitting H on the whole sample to get h (line 21); *(ii)* setting the bounds in the selection function g to the scores $\theta_{l'}$ and $\theta_{u'}$ at the boundaries l' and u' of the rejection area (lines 20 and 22).

We point out that the bounds computed by AUCross may be sup-optimal, either because they do not achieve the minimum target coverage c or do not maximize the AUC. Even when restricting to the class of score-bounded selection functions, the reached AUC may be sub-optimal: for instance, we used (8) as a sufficient but not necessary condition during the proof of Prop. 1. We investigate the experimental performance of AUCross in the next section, also concerning such theoretical caveats.

6 Experiments

We run experiments on nine real-world datasets (seven tabular datasets and two image datasets). We split available data instances into 75% for training selective classifiers and 25% for performance testing. When possible, the split was based on timestamps to allow for out-of-time validation. Otherwise, a stratified random split is performed. A summary of the experimental datasets is reported in Table 1, including the number of features, size for training and test, and the rate of positives in the test set. For the sake of space, we detail the pre-processing procedures in the Appendix. We only mention here that, since we are interested in a binary classification task, we labelled all the cat images of *CIFAR-10-cat* as the positive class, while images of other classes are considered as negatives. Experiments were run on a machine with 96 cores equipped with Intel(R) Xeon(R) Gold 6342 CPU @ 2.80GHz and two NVIDIA RTX A6000, OS Ubuntu 20.04.4, programming language Python 3.8.12.

AUCross *vs* Oracle The first experiment evaluates how well the AUCross algorithm approximates the optimal solution to the problem of maximizing

Table 2: Absolute deviation in empirical AUC of AUCross w.r.t. an Oracle.

c	Adult	Lending	GiveMe	UCICredit	CSDS1	CSDS2	CSDS3	CatsVsDogs	CIFAR-10-cat
.99	.0003	.0000	.0004	.0008	.0003	.0057	.0001	.0000	.0001
.95	.0011	.0000	.0009	.0008	.0012	.0050	.0005	.0000	.0002
.90	.0022	.0000	.0004	.0007	.0024	.0080	.0011	.0000	.0006
.85	.0033	.0000	.0003	.0013	.0033	.0027	.0017	.0004	.0016
.80	.0033	.0000	.0006	.0007	.0039	.0006	.0020	.0004	.0049
.75	.0037	.0001	.0005	.0001	.0053	.0027	.0023	.0000	.0193

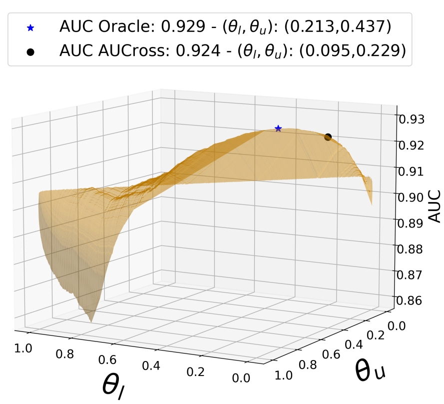

Fig. 4: AUC by varying bounds: Adult and $c = .90$.

$\widehat{AUC}(h, g|S_n)$ s.t. $\hat{\phi}(g|S_n) \geq c$ over the test set S_n, considering the family of score-bounded selection functions. For tabular data, h is fixed to a LightGBM classifier (28) with default hyperparameters. For image data, we used a VGG architecture as a base classifier (38), trained with cross-entropy loss. We enumerate all the score-bounded selection functions $g(\mathbf{x}) = 0$ iff $\theta_l \leq h(\mathbf{x}) \leq \theta_u$ by varying the bounds $\theta_l, \theta_u \in \{h(\mathbf{x}_i) \mid (\mathbf{x}_i, y) \in S_n\}$ such that $\hat{\phi}(g|S_n) \geq c$. The ORACLE approach chooses the function g which maximizes $\widehat{AUC}(h, g|S_n)$. It requires knowing the true class of instances in S_n; hence it is not feasible in practice. Table 2 shows the difference between the empirical AUC ($\widehat{AUC}$) achieved by the ORACLE and the one of AUCROSS on the test sets of the experimental datasets (the value of the empirical AUC for AUCROSS is reported in Table 3 and discussed later on). We notice that AUCROSS reaches the ORACLE performance for the *LendingClub* and *CatsVsDogs* datasets. For the datasets *Adult*, *CSDS1*, *CSDS3* and *CIFAR-10-cat*, there are larger violations for smaller c's, while we do not see such a trend for *GiveMe*, *UCICredit* and *CSDS2*. This suggests that the strategy of centring the rejection area at a midpoint (line 15 of Alg. 2) might be sub-optimal. Figure 4 shows the performance over the space of bounds in a specific case, highlighting the bounds of ORACLE and AUCROSS.

AUCROSS *vs baselines* We compare the performance of AUCROSS to a few baselines. The first one is a bounded-improvement version of the plug-in rule (24) (PLUGIN). PLUGIN uses softmax as confidence function, i.e., in the binary case, $g(x) = \mathbb{1}(\max\{h(x), 1 - h(x)\} > \theta)$, and it estimates the θ parameter as the $(1 - c)$ quantile on a validation set. Such an approach requires splitting the training set into a dataset for building the classifier and a validation set for estimating θ (10% of the original dataset in our experiments). In contrast, AUCROSS allows for building the classifier using the whole training set. Second, we consider two hybrid versions of AUCROSS and PLUGIN: (1) a plug-in rule specialized for AUC (PLUGINAUC), i.e., that uses a score-bounded selection function by estimating bounds as in Alg. 1 on a validation set; (2) a cross-fitting version of PLUGIN (called SCROSS by (35)), where the selection function is softmax and the estimation of the bounds exploits the second-order improvement of of Thm. 4. These two variants allow us to evaluate the relative contributions of the two key elements of our approach: the estimation of AUC-specific bounds (Prop. 3), and the cross fitting approach paired with the second-order quantile estimation strategy (Thm. 4). We also consider two state-of-the-art methods for selective classification, namely the SelectiveNet (18) (SELNET), and the Self Adaptive Training (26) (SAT) methods. The approach from (37) is not included in the baselines since it rejects the comparison of *pairs* of instances, while we focus on the rejection of predictions on instances in isolation.

For tabular data, SELNET and SAT are built using a ResNet structure (20), while AUCROSS, PLUGIN, PLUGINAUC and SCROSS are based on a LightGBM classifier. We report in the Appendix results for other base classifiers. For *CatsVsDogs*, a VGG architecture (38) is used for all the methods. For all the DNN approaches, we set 300 epochs in training, Stochastic Gradient Descent as an optimizer, a learning rate of .1 decreased by a factor .5 every 25 epochs, as in

Table 3: Performance metrics (1,000 bootstrap runs over the test set, results as mean ± stdev). V (for violation) is the absolute difference between the empirical coverage and the target coverage c.

		Empirical Coverage						Selective AUC					
	c	AUCross	PlugIn	PlugInAUC	SCross	SelNet	SAT	AUCross	PlugIn	PlugInAUC	SCross	SelNet	SAT
Adult	.99	**.989 ± .001**	.992 ± .001	.991 ± .001	.993 ± .001	.985 ± .001	.998 ± .001	**.929 ± .003**	.928 ± .003	.928 ± .003	.928 ± .003	.901 ± .003	.902 ± .003
	.95	**.950 ± .002**	.947 ± .002	.950 ± .002	**.950 ± .002**	.954 ± .002	.984 ± .002	**.935 ± .003**	**.935 ± .003**	.934 ± .003	.935 ± .003	.907 ± .003	.903 ± .003
	.90	.896 ± .003	**.900 ± .003**	.898 ± .003	.904 ± .003	.897 ± .003	.966 ± .002	**.943 ± .002**	.941 ± .003	.942 ± .003	.941 ± .003	.917 ± .003	.904 ± .003
	.85	.849 ± .003	.846 ± .003	.850 ± .003	**.850 ± .003**	.772 ± .004	.952 ± .002	**.950 ± .002**	.947 ± .003	.949 ± .002	.948 ± .003	.848 ± .005	.904 ± .003
	.80	.797 ± .004	.794 ± .004	.793 ± .004	**.802 ± .004**	.793 ± .004	.934 ± .003	**.958 ± .002**	.952 ± .003	.957 ± .002	.953 ± .003	.924 ± .004	.904 ± .003
	.75	**.750 ± .004**	.743 ± .004	.744 ± .004	.746 ± .004	.736 ± .004	.919 ± .003	**.963 ± .002**	.958 ± .003	.962 ± .002	.958 ± .003	.917 ± .004	.904 ± .003
Lending	.99	.996 ± .001	.995 ± .001	.997 ± .001	**.994 ± .001**	.997 ± .001	.073 ± .001	.983 ± .001	**.984 ± .001**	.984 ± .001	.983 ± .001	.973 ± .001	.664 ± .004
	.95	.980 ± .001	.970 ± .001	.980 ± .001	**.969 ± .001**	.985 ± .001	.073 ± .001	**.985 ± .001**	**.985 ± .001**	**.985 ± .001**	.984 ± .001	.973 ± .001	.664 ± .004
	.90	.959 ± .001	**.939 ± .001**	.958 ± .001	**.938 ± .001**	.928 ± .001	.073 ± .001	**.987 ± .001**	.986 ± .001	**.987 ± .001**	.986 ± .001	.981 ± .001	.664 ± .004
	.85	.936 ± .001	.909 ± .001	.935 ± .001	**.908 ± .001**	.866 ± .001	.073 ± .001	**.989 ± .001**	.988 ± .001	**.989 ± .001**	.987 ± .001	.977 ± .001	.664 ± .004
	.80	.912 ± .001	.880 ± .001	.911 ± .001	.879 ± .001	**.850 ± .001**	.073 ± .001	.990 ± .001	.989 ± .001	**.991 ± .001**	.989 ± .001	.991 ± .001	.664 ± .004
	.75	.886 ± .001	.850 ± .001	.884 ± .001	**.849 ± .001**	.880 ± .001	.073 ± .001	**.992 ± .001**	.990 ± .001	**.992 ± .001**	.990 ± .001	.976 ± .001	.664 ± .004
GiveMe	.99	**.990 ± .001**	**.990 ± .001**	**.990 ± .001**	**.990 ± .001**	**.944 ± .002**	.746 ± .003	.868 ± .004	.861 ± .004	**.869 ± .004**	.859 ± .004	.747 ± .007	.730 ± .006
	.95	**.950 ± .002**	.949 ± .002	.947 ± .002	**.950 ± .002**	.944 ± .002	.727 ± .003	.874 ± .004	.824 ± .006	**.875 ± .004**	.826 ± .006	.748 ± .007	.729 ± .006
	.90	.898 ± .002	.903 ± .002	.897 ± .002	.902 ± .002	**.900 ± .002**	.705 ± .003	**.883 ± .004**	.784 ± .007	**.883 ± .004**	.784 ± .007	.708 ± .008	.728 ± .007
	.85	.848 ± .002	.856 ± .002	.849 ± .002	.852 ± .002	**.848 ± .002**	.699 ± .003	**.890 ± .004**	.759 ± .008	**.890 ± .004**	.757 ± .008	.658 ± .009	.728 ± .007
	.80	.795 ± .003	.807 ± .003	.801 ± .003	**.800 ± .003**	.802 ± .003	.694 ± .003	**.898 ± .004**	.744 ± .009	.897 ± .004	.743 ± .009	.677 ± .009	.727 ± .007
	.75	.748 ± .003	.756 ± .003	.750 ± .003	.748 ± .003	**.754 ± .003**	.689 ± .003	**.904 ± .004**	.738 ± .010	**.904 ± .004**	.734 ± .010	.676 ± .010	.727 ± .007
UCICredit	.99	.988 ± .002	.993 ± .002	.993 ± .001	.993 ± .001	**.989 ± .002**	1.000± .000	.772 ± .008	.772 ± .007	**.774 ± .007**	.770 ± .008	.765 ± .007	.768 ± .007
	.95	.944 ± .003	.948 ± .003	.946 ± .003	**.951 ± .003**	.909 ± .004	1.000± .000	.778 ± .008	.769 ± .008	**.780 ± .007**	.765 ± .008	.718 ± .009	.768 ± .007
	.90	.896 ± .004	**.897 ± .004**	.900 ± .004	.906 ± .004	.904 ± .004	.994 ± .001	.785 ± .008	.758 ± .008	**.787 ± .008**	.754 ± .009	.707 ± .009	.769 ± .007
	.85	.840 ± .005	**.851 ± .005**	.846 ± .005	.848 ± .005	.855 ± .005	.994 ± .001	.792 ± .008	.746 ± .009	**.795 ± .008**	.734 ± .009	.668 ± .009	.769 ± .007
	.80	.789 ± .005	**.803 ± .005**	**.797 ± .005**	**.797 ± .005**	.810 ± .005	.994 ± .001	.800 ± .008	.722 ± .010	**.802 ± .008**	.710 ± .010	.644 ± .010	.769 ± .007
	.75	.737 ± .006	.751 ± .006	.739 ± .006	**.750 ± .005**	.759 ± .005	.994 ± .001	.809 ± .008	.695 ± .011	**.810 ± .008**	.679 ± .011	.620 ± .011	.769 ± .007
CSDS1	.99	**.990 ± .001**	.980 ± .001	.991 ± .001	.980 ± .001	.985 ± .001	.984 ± .001	**.686 ± .003**	.675 ± .003	**.686 ± .003**	.675 ± .003	.671 ± .003	.669 ± .003
	.95	**.950 ± .001**	.918 ± .001	.949 ± .001	.917 ± .001	.924 ± .001	.937 ± .001	**.689 ± .003**	.654 ± .004	**.689 ± .003**	.655 ± .004	.652 ± .004	.654 ± .004
	.90	**.898 ± .002**	.853 ± .002	.898 ± .002	.849 ± .002	.863 ± .002	.861 ± .002	**.693 ± .003**	.640 ± .004	**.694 ± .003**	.641 ± .004	.638 ± .004	.635 ± .004
	.85	**.849 ± .002**	.792 ± .002	.845 ± .002	.790 ± .002	.800 ± .002	.803 ± .002	.697 ± .003	.631 ± .004	**.698 ± .003**	.631 ± .004	.630 ± .004	.625 ± .004
	.80	**.799 ± .002**	.737 ± .002	.795 ± .002	.736 ± .002	.740 ± .002	.739 ± .002	.702 ± .004	.624 ± .004	**.703 ± .003**	.620 ± .004	.618 ± .004	.616 ± .004
	.75	**.748 ± .002**	.681 ± .002	.747 ± .002	.682 ± .002	.686 ± .002	.687 ± .002	.706 ± .004	.613 ± .005	**.707 ± .004**	.616 ± .005	.609 ± .005	.608 ± .005
CSDS2	.99	**.990 ± .001**	.984 ± .002	.991 ± .001	.983 ± .002	.980 ± .002	1.000± .000	.616 ± .020	.585 ± .020	.587 ± .020	.616 ± .020	.607 ± .020	**.623 ± .019**
	.95	**.952 ± .002**	.927 ± .003	.954 ± .002	.916 ± .003	.894 ± .003	1.000± .000	**.619 ± .020**	.572 ± .021	.589 ± .020	.581 ± .022	.587 ± .022	.623 ± .019
	.90	**.904 ± .003**	.851 ± .004	.907 ± .003	.834 ± .004	.805 ± .004	.880 ± .003	**.620 ± .020**	.566 ± .023	.590 ± .021	.570 ± .023	.560 ± .024	.588 ± .022
	.85	**.860 ± .004**	.773 ± .004	.861 ± .004	.761 ± .004	.712 ± .005	.817 ± .004	**.631 ± .021**	.552 ± .024	.590 ± .021	.572 ± .025	.562 ± .026	.589 ± .023
	.80	**.811 ± .004**	.715 ± .005	.818 ± .004	.687 ± .005	.663 ± .005	.756 ± .004	**.631 ± .021**	.532 ± .025	.592 ± .021	.555 ± .026	.547 ± .027	.582 ± .024
	.75	**.760 ± .004**	.651 ± .005	.776 ± .004	.620 ± .005	.564 ± .005	.697 ± .005	**.635 ± .021**	.536 ± .027	.592 ± .021	.541 ± .028	.579 ± .029	.575 ± .026
CSDS3	.99	**.991 ± .001**	.992 ± .001	.992 ± .001	**.991 ± .001**	.988 ± .001	.992 ± .001	**.851 ± .003**	.850 ± .003	.850 ± .003	.850 ± .003	.844 ± .003	.844 ± .003
	.95	**.948 ± .002**	.957 ± .002	.960 ± .002	.955 ± .002	.942 ± .002	.951 ± .002	**.858 ± .003**	.852 ± .003	.855 ± .003	.853 ± .003	.851 ± .003	.849 ± .003
	.90	**.901 ± .002**	.909 ± .002	.909 ± .002	.911 ± .002	.913 ± .002	.902 ± .002	**.865 ± .003**	.855 ± .003	.863 ± .003	.856 ± .004	.850 ± .004	.855 ± .004
	.85	**.849 ± .003**	.859 ± .003	.856 ± .003	.865 ± .003	.867 ± .003	.856 ± .003	**.873 ± .003**	.857 ± .004	.871 ± .003	.858 ± .004	.851 ± .004	.858 ± .004
	.80	**.805 ± .003**	.808 ± .003	.811 ± .003	.818 ± .003	.796 ± .003	.819 ± .003	**.880 ± .003**	.859 ± .004	.878 ± .003	.859 ± .004	.859 ± .004	.858 ± .004
	.75	.759 ± .003	**.758 ± .003**	.765 ± .003	.769 ± .003	.765 ± .003	.776 ± .003	**.887 ± .003**	.858 ± .004	.884 ± .003	.859 ± .004	.843 ± .004	.853 ± .004
CatsVsDogs	.99	.992 ± .002	**.990 ± .002**	.991 ± .002	.992 ± .002	.991 ± .002	.993 ± .002	.984 ± .003	**.985 ± .002**	**.985 ± .002**	.985 ± .002	.979 ± .003	**.985 ± .002**
	.95	.962 ± .003	.965 ± .003	**.954 ± .004**	.959 ± .003	**.954 ± .003**	.959 ± .003	.986 ± .003	.985 ± .002	**.987 ± .002**	.987 ± .003	**.987 ± .002**	.987 ± .002
	.90	.920 ± .004	.910 ± .004	**.907 ± .005**	.917 ± .004	.891 ± .005	.910 ± .005	.987 ± .003	.984 ± .002	.988 ± .002	.988 ± .003	**.990 ± .002**	.989 ± .002
	.85	.883 ± .005	.854 ± .005	.858 ± .005	.871 ± .005	**.852 ± .006**	.862 ± .005	.987 ± .003	.984 ± .002	.989 ± .002	.988 ± .003	**.990 ± .002**	.990 ± .002
	.80	.841 ± .006	**.797 ± .006**	.805 ± .006	.807 ± .006	.809 ± .006	.818 ± .006	.987 ± .003	.985 ± .002	.990 ± .002	.989 ± .003	**.991 ± .002**	.990 ± .002
	.75	.781 ± .006	.740 ± .007	**.749 ± .007**	.744 ± .007	.765 ± .007	.768 ± .007	.987 ± .003	.983 ± .002	.990 ± .002	.988 ± .003	**.992 ± .002**	.990 ± .002
CIFAR10-CAT	.99	1.000± .001	.991 ± .001	.989 ± .002	.992 ± .001	.992 ± .001	**.990 ± .002**	.890 ± .009	.915 ± .007	.909 ± .008	**.922 ± .007**	.896 ± .008	.893 ± .008
	.95	.998 ± .001	**.950 ± .003**	.948 ± .003	.953 ± .003	.956 ± .003	.952 ± .003	.889 ± .009	.895 ± .009	**.911 ± .007**	.906 ± .009	.902 ± .008	.863 ± .011
	.90	.994 ± .001	**.899 ± .004**	.898 ± .003	.903 ± .003	.904 ± .003	.913 ± .003	.889 ± .009	.896 ± .009	**.913 ± .007**	.851 ± .013	.899 ± .010	.760 ± .017
	.85	.987 ± .002	.849 ± .004	**.851 ± .004**	.886 ± .004	.863 ± .004	.854 ± .004	.888 ± .009	.898 ± .009	**.916 ± .007**	.815 ± .016	.676 ± .022	.574 ± .024
	.80	.969 ± .002	.796 ± .005	**.799 ± .004**	.882 ± .004	.806 ± .005	.803 ± .005	.885 ± .009	.906 ± .008	**.919 ± .007**	.804 ± .017	.359 ± .026	.572 ± .024
	.75	.945 ± .003	**.749 ± .005**	**.749 ± .005**	.879 ± .004	.763 ± .005	.759 ± .005	.870 ± .011	.905 ± .008	**.921 ± .007**	.797 ± .018	.408 ± .028	.575 ± .025
	#	21/54	12/54	8/54	15/54	8/54	1/54	28/54	4/54	29/54	6/54	5/54	2/54
	V	.026 ± .046	.021 ± .029	.013 ± .026	.028 ± .036	.029 ± .041	.171 ± .251						

the original papers. All the parameters of base classifiers are left as the default ones. See the Appendix for details.

Table 3 shows performance metrics (mean ± stdev) evaluated on 1,000 bootstrap runs of the test set: the empirical coverage $\hat{\phi}(g|S_n)$ and the empirical AUC $\widehat{AUC}(h, g|S_n)$. Resorting to bootstrap allows for calculating standard errors of the metrics[2] (36). Moreover, it allows for quantifying the generalizability of the methods to perturbations of the test set. Additional results on the selective

[2] Other approaches include confidence intervals for AUC (8; 7; 19) or direct estimation of its variance (4).

accuracy and the positive rate[3] metrics are reported in the Appendix. Results in Table 3 show that AUCross achieves an empirical coverage close to the target c in most of the cases for tabular data. The most significant violation occurs for *LendingClub*, where also all the other methods fail to reach the target coverage. In particular, SAT performs poorly on such a dataset. The other largest violation occurs for *CIFAR-10-cat*. Here, also the other cross-fitting-based algorithm (SCross) fails to reach the target one. Interestingly, the runner-up method w.r.t. coverage is PlugInAUC, which has the smallest average violation. A striking case is the *CSDS2* dataset, which is the most imbalanced one with a positive rate of 1.8%. Here, both AUCross and PlugInAUC have small violations, while all other approaches have larger ones. We argue that this is due to the bounds determined by Prop. 1 and Prop. 2, which are specific for positives and negatives, respectively, while the other methods calibrate the selection function independently of the class labels. Regarding the empirical AUC, AUCross and PlugInAUC outperform the other methods in most cases. For unbalanced datasets such as for *GiveMe*, *CSDS1* and *CSDS2*, the empirical AUC drops for smaller coverages, contrarily to what is expected. Such behaviour does not occur for those methods that estimate bounds specifically for the AUC, i.e. AUCross and PlugInAUC. We show in the Appendix that a trade-off exists in terms of selective accuracy, as both AUCross and PlugInAUC are outperformed concerning the accuracy over the accepted region. This is not surprising since they are optimizing the AUC metric, and it is well-known that accuracy and AUC cannot be jointly optimized (6). The results in the Appendix also show that optimizing accuracy compromises the positive rate, while our approach is fairer in this sense.

Finally, consider the running time performances (see the Appendix for details). AUCross and SCross require K executions of the base classifier (loop at lines 3-6 of Alg. 2), while all other baselines train a single (selective) classifier. On tabular data, using efficient base classifiers results in a reasonably low total running time. Those two strategies are instead computationally expensive for image data, where DNN base classifiers are adopted. However, we observe that PlugInAUC exhibits performances comparable to SelNet and SAT for *CatsVsDogs*, and it outperforms them in the case of *CIFAR-10-cat*. In summary, we recommend using AUCross for tabular datasets and PlugInAUC for image datasets.

7 Conclusions

Selective classification can help prevent poor or even harmful decisions by abstaining from making an output, possibly demanding the decision to a human. We have extended the selective classification framework to a widely used classifier evaluation metric, the Area Under the ROC Curve. Through an analytic characterization, we devised methods for computing an estimator of the best score-bounded selection function. These methods are effective and outperform existing approaches designed for optimizing accuracy.

[3] Positive rate in the accepted region, compared to the positive rate in the overall test set, measures fairness of the selection function w.r.t. the class labels.

Limitations that require future work include the following. First, the approach should be extended to multi-class classification, for which the notion of AUC has been considered, e.g., the Volume Under the Surface (14), the average of pairwise binary AUCs (the M metric) (21), or the AUC-μ (30). Second, since selective classification might amplify unfair decisions (27), we intend to study how to account for fairness metrics in the context of AUC-based selective classification. Third, as suggested by our experimental results, we could better determine the bounds for a target coverage c, reconsidering the choice of centring the rejection area in the midpoint of the bounds of Prop. 3. As shown in Table 2, this is especially relevant for large c's. Lastly, AUC-based selective classification can be extended to the metric of weighted AUC (13; 29), where instances are weighted by importance.

Reproducibility Data and source code can be downloaded from https://-github.com/andrepugni/AUCbasedSelectiveClassification.

Acknowledgements Work supported by the XAI project, funded by the European Union's Horizon 2020 Excellent Science European Research Council (ERC) programme under g.a. No. 834756. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the EU. Neither the EU nor the granting authority can be held responsible for them.

References

[1] Barddal, J.P., Loezer, L., Enembreck, F., Lanzuolo, R.: Lessons learned from data stream classification applied to credit scoring. Expert Syst. Appl. **162**, 113899 (2020). https://doi.org/10.1016/j.eswa.2020.113899, https://doi.org/10.1016/j.eswa.2020.113899

[2] Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., Robins, J.: Double/debiased machine learning for treatment and structural parameters. The Econometrics Journal **21**(1), C1–C68 (01 2018). https://doi.org/10.1111/ectj.12097, https://doi.org/10.1111/ectj.12097

[3] Chow, C.K.: On optimum recognition error and reject tradeoff. IEEE Trans. Inf. Theory **16**(1), 41–46 (1970)

[4] Cleves, M.A.: Comparative assessment of three common algorithms for estimating the variance of the area under the nonparametric receiver operating characteristic curve. The Stata Journal **2**, 280–289 (2002)

[5] Cortes, C., DeSalvo, G., Mohri, M.: Boosting with abstention. In: NIPS. pp. 1660–1668 (2016)

[6] Cortes, C., Mohri, M.: AUC optimization vs. error rate minimization. In: NIPS. pp. 313–320. MIT Press (2003)

[7] Cortes, C., Mohri, M.: Confidence intervals for the area under the ROC curve. In: NIPS. pp. 305–312 (2004)

[8] DeLong, E.R., DeLong, D.M., Clarke-Pearson, D.L.: Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. Biometrics **44**(3), 837–845 (1988)

[9] Dua, D., Graff, C.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml

[10] El-Yaniv, R., Wiener, Y.: On the foundations of noise-free selective classification. J. Mach. Learn. Res. **11**, 1605–1641 (2010)

[11] Engelmann, B.: Measures of a rating's discriminative power: Applications and limitations. In: Engelmann, B., Rauhmeier, R. (eds.) The Basel II Risk Parameters, pp. 269–291. Springer, 2 edn. (2011)

[12] Fawcett, T.: An introduction to ROC analysis. Pattern Recognit. Lett. **27**(8), 861–874 (2006)

[13] Fawcett, T.: ROC graphs with instance-varying costs. Pattern Recognit. Lett. **27**(8), 882–891 (2006)

[14] Ferri, C., Hernández-Orallo, J., Salido, M.A.: Volume under the ROC surface for multi-class problems. In: ECML. Lecture Notes in Computer Science, vol. 2837, pp. 108–120. Springer (2003)

[15] Franc, V., Průša, D.: On discriminative learning of prediction uncertainty. In: ICML. Proceedings of Machine Learning Research, vol. 97, pp. 1963–1971. PMLR (2019)

[16] Fumera, G., Roli, F.: Support vector machines with embedded reject option. In: SVM. Lecture Notes in Computer Science, vol. 2388, pp. 68–82. Springer (2002)

[17] Geifman, Y., El-Yaniv, R.: Selective classification for deep neural networks. In: NIPS. pp. 4878–4887 (2017)

[18] Geifman, Y., El-Yaniv, R.: Selectivenet: A deep neural network with an integrated reject option. In: ICML. Proceedings of Machine Learning Research, vol. 97, pp. 2151–2159. PMLR (2019)

[19] Goncalves, L., Subtil, A., Oliveira, M.R., de Zea Bermudez, P.: ROC curve estimation: An overview. Revstat - Statistical Journal **12**, 1–20 (2014)

[20] Gorishniy, Y., Rubachev, I., Khrulkov, V., Babenko, A.: Revisiting deep learning models for tabular data. In: NeurIPS. pp. 18932–18943 (2021)

[21] Hand, D.J., Till, R.J.: A simple generalisation of the area under the ROC curve for multiple class classification problems. Mach. Learn. **45**(2), 171–186 (2001)

[22] He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009)

[23] Hendrickx, K., Perini, L., der Plas, D.V., Meert, W., Davis, J.: Machine learning with a reject option: A survey. CoRR **abs/2107.11277** (2021)

[24] Herbei, R., Wegkamp, M.H.: Classification with reject option. Can. J. Stat. **34**(4), 709–721 (2006)

[25] Huang, J., Ling, C.X.: Using AUC and accuracy in evaluating learning algorithms. IEEE Trans. Knowl. Data Eng. **17**(3), 299–310 (2005)

[26] Huang, L., Zhang, C., Zhang, H.: Self-adaptive training: beyond empirical risk minimization. In: NeurIPS (2020)

[27] Jones, E., Sagawa, S., Koh, P.W., Kumar, A., Liang, P.: Selective classification can magnify disparities across groups. In: ICLR. OpenReview.net (2021)

[28] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T.: LightGBM: A highly efficient gradient boosting decision tree. In: NIPS. pp. 3146–3154 (2017)

[29] Keilwagen, J., Grosse, I., Grau, J.: Area under precision-recall curves for weighted and unweighted data. PLoS ONE **9**(3), e92209 (2014)
[30] Kleiman, R., Page, D.: AUCμ: A performance metric for multi-class machine learning models. In: ICML. Proceedings of Machine Learning Research, vol. 97, pp. 3439–3447. PMLR (2019)
[31] Knight, K., Bassett, G.W.: Second order improvements of sample quantiles using subsamples (2003), *Unpublished manuscript*, http://www.utstat.toronto.edu/keith/papers/subsample.pdf
[32] Krizhevsky, A.: Learning multiple layers of features from tiny images. Tech. rep., University of Toronto (2009)
[33] Liu, Z., Wang, Z., Liang, P.P., Salakhutdinov, R., Morency, L., Ueda, M.: Deep gamblers: Learning to abstain with portfolio theory. In: NeurIPS. pp. 10622–10632 (2019)
[34] Pietraszek, T.: Optimizing abstaining classifiers using ROC analysis. In: ICML. ACM International Conference Proceeding Series, vol. 119, pp. 665–672. ACM (2005)
[35] Pugnana, A., Ruggieri, S.: A model-agnostic heuristics for selective classification. In: AAAI. AAAI Press (2023)
[36] Rajkomar, A., Oren, E., Chen, K., Dai, A.M., Hajaj, N., Hardt, M., Liu, P.J., Liu, X., Marcus, J., Sun, M., et al.: Scalable and accurate deep learning with electronic health records. NPJ Digital Medicine **1**(1), 1–10 (2018)
[37] Shen, S., Yang, B., Gao, W.: AUC optimization with a reject option. In: AAAI. pp. 5684–5691. AAAI Press (2020)
[38] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
[39] Tortorella, F.: A ROC-based reject rule for dichotomizers. Pattern Recognit. Lett. **26**(2), 167–180 (2005)
[40] Vovk, V.: Conditional validity of inductive conformal predictors. Mach. Learn. **92**(2-3), 349–376 (2013)
[41] Wu, Y.C., Lee, W.C.: Alternative performance measures for prediction models. PLoS ONE **9**(3), e91249 (2014)
[42] Yang, T., Ying, Y.: AUC maximization in the era of big data and AI: A survey. ACM Comput. Surv. **55**(8), 172:1–172:37 (2023)
[43] Yang, T., Ying, Y.: AUC maximization in the era of big data and AI: A survey. ACM Comput. Surv. **55**(8), 172:1–172:37 (2023)
[44] Yeh, I., Lien, C.: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Syst. Appl. **36**(2), 2473–2480 (2009)

Supplementary Materials for AUC-based selective classification

A APPENDIX

A.1 Proofs

Proposition 1.
Proof. The missing part of the proof consists of the following equivalence:

$$
\begin{aligned}
\bar{G} > G \text{ iff } & \bar{A}(n-1) > An \\
\text{iff } & \{ \text{ by } (7) \ \} \\
\text{iff } & ((A + \frac{1}{2} - \left[\frac{(r'(\mathbf{x}_i) - 1) + (t(\mathbf{x}_i) - 1) + 1/2}{n^+ n}\right]) \frac{n^+ n}{(n^+ - 1)(n-1)} - \frac{1}{2})(n-1) > An \\
\text{iff } & (A + \frac{1}{2} - \frac{(r'(\mathbf{x}_i) + t(\mathbf{x}_i) - 1.5)}{n^+ n}) \frac{n^+ n}{(n^+ - 1)} - \frac{(n-1)}{2} > An \\
\text{iff } & A \frac{n^+ n}{(n^+ - 1)} + \frac{n^+ n}{2(n^+ - 1)} - \frac{(r'(\mathbf{x}_i) + t(\mathbf{x}_i) - 1.5)}{(n^+ - 1)} - \frac{(n-1)}{2} > An \\
\text{iff } & A \frac{n}{(n^+ - 1)} + \frac{n^+ n}{2(n^+ - 1)} - \frac{(r'(\mathbf{x}_i) + t(\mathbf{x}_i) - 1.5)}{(n^+ - 1)} - \frac{(n-1)}{2} > 0 \\
\text{iff } & A \frac{n}{(n^+ - 1)} + \frac{n^+ n - (n-1)(n^+ - 1)}{2(n^+ - 1)} - \frac{(r'(\mathbf{x}_i) + t(\mathbf{x}_i) - 1.5)}{(n^+ - 1)} > 0 \\
\text{iff } & An + \frac{n + n^+ - 1}{2} - (r'(\mathbf{x}_i) + t(\mathbf{x}_i) - 1.5) > 0 \\
\text{iff } & An + \frac{n + n^+}{2} - t(\mathbf{x}_i) > r'(\mathbf{x}_i) - 1 \\
\text{iff } & An + \frac{n + n^+}{2} - t(\mathbf{x}_i) \geq r'(\mathbf{x}_i) \\
\text{iff } & An + \frac{n - n^+}{2} + (n^+ - t(\mathbf{x}_i)) \geq r'(\mathbf{x}_i) \\
\text{iff } & \frac{r'(\mathbf{x}_i)}{n} \leq A + \frac{p^-}{2} + \frac{(n^+ - t(\mathbf{x}_i))}{n}
\end{aligned}
$$

Proposition 2.
Proof. We can equivalently show the result for the Gini coefficient. Let $G = A/(A+B)$, and let $\bar{G} = \bar{A}/(\bar{A}+\bar{B})$ be the Gini coefficient after removing (a.k.a., abstaining on) one negative instance $\mathbf{x}_i$. We have $\bar{A}+\bar{B} = (n^- - 1)/(2(n-1))$. Then $\bar{G} > G$ iff

$$\bar{A}(n-1)n^- > An(n^- - 1) \tag{11}$$

As in Proposition 1, we can link the area $\bar{A}$ after the removal of the negative instance $\mathbf{x}_i$ to the original area A as follows:

$$\bar{A} + \frac{1}{2} = (A + \frac{1}{2} - \frac{t(\mathbf{x}_i)}{n^+ n})\frac{n}{n-1}$$

As for the positive case, this can be intuitively understood by looking at Figure 5, where the grey area highlights the loss from the original CAP plot after we remove the negative instance. Then we rescale to account for the new number of instances, as in Figure 6. Then:

$$\begin{aligned}
\bar{G} > G \text{ iff } & ((A + \frac{1}{2} - \frac{t(\mathbf{x}_i)}{n^+ n})\frac{n}{n-1} - \frac{1}{2})(n-1)n^- > An(n^- - 1) \\
\text{iff } & (A + \frac{1}{2} - \frac{t(\mathbf{x}_i)}{n^+ n})nn^- - \frac{(n-1)n^-}{2} > An(n^- - 1) \\
\text{iff } & Ann^- + \frac{nn^-}{2} - \frac{t(\mathbf{x}_i)}{n^+}n^- - \frac{(n-1)n^-}{2} > Ann^- - An \\
\text{iff } & An + \frac{nn^-}{2} - \frac{(n-1)n^-}{2} > \frac{t(\mathbf{x}_i)}{n^+}n^- \\
\text{iff } & A\frac{n}{n^-} + \frac{1}{2} > \frac{t(\mathbf{x}_i)}{n^+} \\
\text{iff } & A\frac{n}{n^-} + \frac{1}{2} - \frac{1}{n^+} \geq \frac{t(\mathbf{x}_i)}{n^+}
\end{aligned}$$

Since $\widehat{AUC}(h, g'|S_n) = (G+1)/2 = (2A/p^- + 1)/2 = A^n/_{n^-} + {}^1/_2$, and

$$\frac{t(\mathbf{x}_i)}{n^+} \leq \widehat{AUC}(h, g|S_n) - \frac{1}{n^+} \tag{12}$$

is assumed to hold, we have the $\bar{G} > G$ (a.k.a., the conclusion of Proposition 2) after removing *one* negative instance. Moreover, if (12) holds for a second negative instance $\mathbf{x}_i$, then:

$$\frac{t(\mathbf{x}_i)}{n^+} \leq \widehat{AUC}(h, g|S_n) - \frac{1}{n^+} \leq \widehat{AUC}(h, g'|S_n) - \frac{1}{n^+}$$

where g' abstains on the first negative instance. In fact, $\bar{G} > G$ implies that $\widehat{AUC}(h, g|S_n) < \widehat{AUC}(h, g'|S_n)$. Therefore, we can iterate the conclusion that the Gini coefficient (or, equivalently, empirical AUC) increases by abstaining on any number of negative instances that satisfy the assumption of Proposition 2.

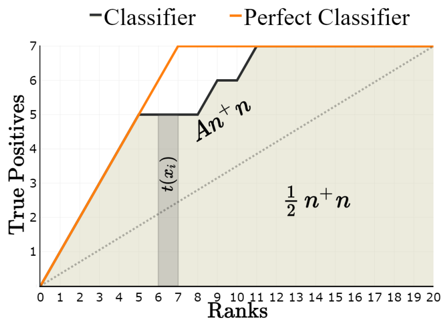

Fig. 5: CAP plot before removing a negative instance.

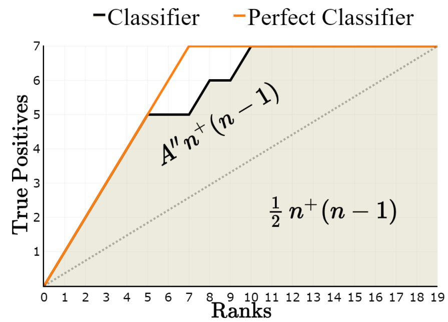

Fig. 6: CAP plot after removing a negative instance.

A.2 Datasets description

Adult is an extract from the 1994 US Census, with class the binarization of income into $\leq 50K$ and $> 50K$. The final training set contains 30,162 instances and 55 features after one-hot encoding. The test set size is 15,060.

LendingClub regards repaying a loan obtained by an online platform. We used a temporal split to build the final training set (1,364,697 instances) and the test set (445,912 instances). The dataset has 65 features.

The *GiveMe* dataset aims at predicting the financial distress of a borrower within two years. The training and the test set were obtained by stratified random sampling, and they contain 12 features, and 112,500 and 37,500 instances respectively.

UCICredit regards credit card defaults in Taiwan. This dataset from (9) concerns whether or not a credit card holder will default in the next six months (44). Training and test sets were obtained by stratified random sampling. The training set includes 22,500 instances (7,500 for the test set) and 23 features.

CSDS1, *CSDS2* and *CSDS3* - from (1) - regard predicting defaults in repaying a loan: within six months for *CSDS1* (data span over 15 months), within 2 months for *CSDS2* (data span over 25 months), and within three months for *CSDS3* (data span over 16 months). Training set and test set were divided through a timestamp variable. The training set of *CSDS1* consists of 230,409 instances and 155 features (test set size is 76,939). For *CSDS2*, the training set contains 37,100 instances and 35 features (test set size is 12,533). For *CSDS3*, the training set contains 71,177 instances and 144 features (test set size 23,288).

The *CatsVsDogs* dataset is a collection of cats and dogs images. The task here is to distinguish between the two species. The training and test sets were obtained as described in (33). The training set contains 20,000 images, each one of 64x64 pixels. The test set consists of 5,000 images.

Finally we considered the image dataset *CIFAR-10-cat* from (32). For each of the 10 class labels, we have 5,000 32x32 images in the training set and 1,000 in the test set. We transformed it into a binary classification task by using the *cat* label as the positive class.

A.3 Models

AUCross *and* PlugInAUC. In the main paper we used as base classifier a LightGBM classifier with default parameters. In Tables 6-9 we provide also results for a Logistic Regression and a Random Forest Classifier from *sklearn* package with default parameters, a ResNet implementation with default parameters from *rtdl* (20) and a XGBoost from *xgboost* package with default parameters.

PlugIn *and* SCross. As for AUCross, we considered a LightGBM classifier with default parameters.

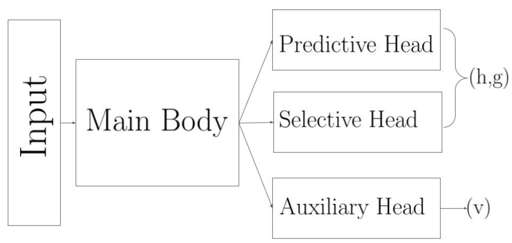

Fig. 7: Scheme for Selective Net architecture.

SelNet. Selective Net is a selective model (h, g) that optimizes at the same time both $h(\mathbf{x})$ and $g(\mathbf{x})$. Its schema is summarized in Figure 7. The architecture is based on four distinct parts: the main body, the predictive head, the selective head and the auxiliary head. The input is initially processed by the main body: it consists of deep layers that are shared by all the three heads. Any type of architecture can be used in this part (e.g., convolutional layers, linear layers, recurrent layers ecc.). The predictive head provides the final prediction $h(\mathbf{x})$; the selective head outputs the selective function $g(\mathbf{x})$; the auxiliary head is used to ensure that the main body part is exposed to all training instances, i.e., it is used to avoid that SelNet overfits on the accepted instances. We used *pytorch* to model SelNet. For tabular datasets, we built the main body part using ResNet with default parameters provided by *rtdl* (20), as authors claim that ResNet is a valid baseline on tabular data. For images, we used as the main body the VGG16 architecture (38) as done in the original paper (18). We then added the classification head, the selection head and the auxiliary head following SelNet paper. For CSDS1, Lending and GiveMe the prediction head and the auxiliary head were made by a first linear layer with 512 nodes followed by a batch normalization layer and ReLu activation; a second layer with 256 nodes, batch normalization and ReLu activation; a final dense layer with 128 nodes ending with two nodes and softmax activation. For the other datasets the classification and auxiliary heads were made by a single linear layer with 128 nodes and a final softmax activation. We built selective heads using a linear layer with 128 nodes, batch normalization, relu activation and another 64-node linear layer ending with a single node and Sigmoid activation. All the models are available in the code here. We point out that the lack of a clear design methodology of the SelectiveNet structure for a given dataset is a major drawback of SelNet compared to the flexibility our model-agnostic method. Models were trained, for an expected coverage c, using the same loss function as in (18):

$$\mathcal{L} = \alpha \frac{\frac{1}{n}\sum_{i=1}^{n} l(h(\mathbf{x}_i), y_i) g(\mathbf{x}_i)}{\hat{\phi}(g|S_n)} + \lambda(\max(0, c - \hat{\phi}(g|S_n)))^2 + (1-\alpha)\frac{1}{n}\sum_{i=1}^{n} l(v(\mathbf{x}_i), y_i), \tag{13}$$

where $h(\mathbf{x}_i)$ is the classification head prediction, $l(h(\mathbf{x}_i), y_i)$ is the cross entropy loss, $g(\mathbf{x}_i)$ is the selection head output over instance i, $v(\mathbf{x}_i)$ is the auxiliary head prediction and $\hat{\phi}(g|S_n) = \sum_{i=1}^{n} g(\mathbf{x}_i)/n$ is the empirical coverage. Both parameters λ and α are set as in (18) to $\lambda = 32$ and $\alpha = .5$. The batch size was 512 for Lending; 128 for GiveMe, CSDS1, CSDS2, CSDS3 and image data; 32 for Adult and UCICredit. The learning procedure was run for 300 epochs with 13 as loss and it used as an optimizer Stochastic Gradient Descent setting *learning-rate=.1, momentum=.9,Nesterov=True* and a decay of .5 every 25 epochs as in the original paper. The training was performed over 90% of training set instances while we used the remaining 10% to calibrate the selective head.

SAT. Let us consider the standard binary classification problem where the classifier h can produce a score for instance i belonging to class 0 or 1[4], i.e. in our main paper $s(\mathbf{x}_i) = s_1(\mathbf{x}_i)$. SAT (26) introduces an extra class v (representing abstention) during training and replace the confidence function with the score for the additional class v. This allows for training a selective classifier in an end-to-end fashion. Given a batch of data pairs $\{(\mathbf{x}_i, y_i)\}$ of size M, the model score $s_a(\mathbf{x}_i)$ for class a, and its exponential moving average t_i for each sample, we optimize the classifier h by minimizing:

$$\mathcal{L}(h_\theta) = -\frac{1}{M}\sum_{i=1}^{M}[t_{i,y_i}\log(s_{y_i}(\mathbf{x}_i)) + (1 - t_{i,y_i})\log s_v(\mathbf{x}_i)] \tag{14}$$

where $s_{y_i}(\mathbf{x}_i)$ denotes the score attributed by the classifier to the true class of instance i. This loss is a composition of two terms: the first one measures the standard cross-entropy loss between prediction and original label y_i; the second term acts as the selection function and identifies uncertain samples in the dataset. The value t_{i,y_i} trades-off these two terms: if t_{i,y_i} is very small, the sample is treated as uncertain and the second term enforces the selective classifier to learn to abstain from this sample; if t_{i,y_i} is close to 1, the loss recovers the standard cross entropy minimization and enforces the selective classifier to make perfect predictions. The

[4] The problem is symmetrical in the binary setting.

code was based on the *pytorch* implementation available here. We employed the same batch sizes as for SelNet and we set up training details as in the original paper of SAT: we used the loss in 14 for 300 epochs and as an optimizer Stochastic Gradient Descent, setting *learning-rate=.1, momentum=.9,Nesterov=True* and a decay of .5 every 25 epochs. The training was performed over 90% of training set instances while we used the remaining 10% to calibrate the selection function.

A.4 Other metrics considered

We report results for selective accuracy and positive rate in Table 4. As discussed in the main paper, AUCross and PlugInAUC do not guarantee improvements in terms of accuracy whenever the target coverage decreases. Interestingly, they are able to maintain the positive rate more stable than the compared approaches. Finally, we report training times for all the methods in Table 5. PlugInAUC and PlugIn are clear winners over tabular datases as they can exploit fast classifiers. Both AUCross and SCross pay a factor proportional to the number of folds K used in the cross-fitting part of the algorithm. This extra cost can be potentially mitigated on tabular datasets by parallelizing the cross-fitting procedure. Finally, notice that SelNet is the only approach which require a separate run for each target coverage c.

A.5 Results for different classifiers

We report in Tables 6-9 the results for AUCross and PlugInAUC using different classifiers over tabular datasets. Regarding coverage, we see the harshest violations for both AUCross-RandForest and PlugInAUC-RandForest over *CSDS2* and for AUCross-ResNet over *GiveMe*. Coverage violations occur also for *Lending* dataset, independently of the classifier. Regarding the AUC, both AUCross and PlugInAUC succeed in increasing it while target coverage drops, regardless of the considered base classifier. At the same time, we notice that for all the base classifiers lowering coverage does not guarantee selective accuracy to increase, highlighting once more the trade-off between these two metrics. Finally, we see similar results across all the classifiers for positive rate.

A.6 Results for different K's

We report empirical coverage and selective AUC for different choices of the parameter K in Table 10. The default value $K = 5$ shows a slightly better trade-off between empirical coverage and selectice AUC.

Table 4: Performance metrics (1,000 bootstrap runs over the test set, results as mean ± stdev).

		Selective Accuracy						Positive Rate					
	c	AUCross	PlugIn	PlugInAUC	SCross	SelNet	SAT	AUCross	PlugIn	PlugInAUC	SCross	SelNet	SAT
Adult	.99	.870 ± .003	**.872 ± .003**	.870 ± .003	.871 ± .003	.848 ± .003	.845 ± .003	**.246 ± .004**	.245 ± .004	**.246 ± .004**	.245 ± .004	.242 ± .004	**.246 ± .004**
	.95	.875 ± .003	**.888 ± .003**	.874 ± .003	**.888 ± .003**	.858 ± .003	.845 ± .003	**.246 ± .004**	.234 ± .004	.247 ± .004	.234 ± .004	.235 ± .004	.247 ± .004
	.90	.882 ± .003	**.903 ± .003**	.880 ± .003	.902 ± .003	.872 ± .003	.845 ± .003	**.246 ± .004**	.221 ± .004	.248 ± .004	.220 ± .004	.225 ± .004	.249 ± .004
	.85	.888 ± .003	**.920 ± .003**	.887 ± .003	.919 ± .003	.886 ± .003	.846 ± .003	**.247 ± .004**	.203 ± .004	.249 ± .004	.206 ± .004	.115 ± .003	.250 ± .004
	.80	.899 ± .003	**.936 ± .003**	.897 ± .003	.934 ± .003	.901 ± .003	.846 ± .003	**.245 ± .004**	.185 ± .004	.249 ± .004	.190 ± .004	.180 ± .004	.251 ± .004
	.75	.907 ± .003	**.950 ± .003**	.905 ± .003	**.950 ± .003**	.908 ± .003	.845 ± .004	**.245 ± .005**	.169 ± .004	.250 ± .005	.170 ± .004	.149 ± .004	.252 ± .004
Lending	.99	**.899 ± .001**	**.899 ± .001**	**.899 ± .001**	**.899 ± .001**	.870 ± .001	.754 ± .003	**.224 ± .001**	.222 ± .001	.224 ± .001	**.222 ± .001**	.223 ± .001	.249 ± .003
	.95	.908 ± .001	.909 ± .001	.909 ± .001	**.910 ± .001**	.873 ± .001	.754 ± .003	**.217 ± .001**	.206 ± .001	**.217 ± .001**	.205 ± .001	.215 ± .001	.249 ± .003
	.90	.921 ± .001	.923 ± .001	.921 ± .001	**.924 ± .001**	.912 ± .001	.754 ± .003	**.207 ± .001**	.184 ± .001	**.207 ± .001**	.183 ± .001	.191 ± .001	.249 ± .003
	.85	.934 ± .001	.937 ± .001	.934 ± .001	**.938 ± .001**	.885 ± .001	.754 ± .003	.197 ± .001	.162 ± .001	.197 ± .001	.161 ± .001	.134 ± .001	**.249 ± .003**
	.80	.948 ± .001	**.951 ± .001**	.949 ± .001	**.951 ± .001**	.926 ± .001	.754 ± .003	.186 ± .001	.140 ± .001	.186 ± .001	.139 ± .001	.157 ± .001	**.249 ± .003**
	.75	.964 ± .001	.963 ± .001	**.965 ± .001**	.963 ± .001	.924 ± .001	.754 ± .003	.174 ± .001	.118 ± .001	.174 ± .001	.116 ± .001	.143 ± .001	**.249 ± .003**
GiveMe	.99	.938 ± .002	.942 ± .002	.938 ± .002	.942 ± .002	**.954 ± .002**	.916 ± .002	.068 ± .002	.063 ± .002	**.067 ± .002**	.063 ± .002	.047 ± .002	.085 ± .002
	.95	.937 ± .002	**.956 ± .002**	.937 ± .002	**.956 ± .002**	.954 ± .002	.915 ± .002	**.068 ± .002**	.046 ± .002	**.068 ± .002**	.047 ± .002	.047 ± .002	.086 ± .002
	.90	.938 ± .002	**.967 ± .001**	.937 ± .002	**.967 ± .001**	.961 ± .001	.913 ± .002	**.067 ± .002**	.034 ± .001	**.068 ± .002**	.034 ± .001	.040 ± .001	.088 ± .002
	.85	.939 ± .002	**.973 ± .001**	.938 ± .002	**.973 ± .001**	.967 ± .001	.913 ± .002	**.067 ± .002**	.028 ± .001	**.068 ± .002**	.028 ± .001	.034 ± .001	.088 ± .002
	.80	.939 ± .002	.977 ± .001	.938 ± .002	**.978 ± .001**	.969 ± .001	.912 ± .002	**.067 ± .002**	.024 ± .001	**.067 ± .002**	.023 ± .001	.032 ± .001	.089 ± .002
	.75	.939 ± .002	.980 ± .001	.939 ± .002	**.981 ± .001**	.971 ± .001	.912 ± .002	**.067 ± .002**	.021 ± .001	**.068 ± .002**	.020 ± .001	.030 ± .001	.089 ± .002
UCICredit	.99	.811 ± .005	.814 ± .005	.812 ± .005	.814 ± .005	**.815 ± .005**	.812 ± .005	**.222 ± .005**	.220 ± .005	**.222 ± .005**	.220 ± .005	.220 ± .005	**.222 ± .005**
	.95	.810 ± .005	.826 ± .005	.811 ± .005	**.827 ± .005**	.824 ± .005	.812 ± .005	.225 ± .006	.210 ± .005	.224 ± .005	.209 ± .005	.180 ± .005	**.222 ± .005**
	.90	.809 ± .005	.838 ± .005	.811 ± .005	**.839 ± .005**	.830 ± .005	.813 ± .005	.228 ± .006	.194 ± .005	.226 ± .006	.195 ± .005	.176 ± .005	**.221 ± .005**
	.85	.806 ± .005	.849 ± .005	.811 ± .005	**.855 ± .005**	.845 ± .005	.813 ± .005	.233 ± .006	.179 ± .005	.229 ± .006	.176 ± .005	.156 ± .005	**.221 ± .005**
	.80	.806 ± .006	.863 ± .005	.810 ± .006	**.867 ± .005**	.858 ± .005	.813 ± .005	.236 ± .006	.161 ± .005	.232 ± .006	.158 ± .005	.143 ± .005	**.221 ± .005**
	.75	.806 ± .006	.872 ± .005	.808 ± .006	**.875 ± .005**	.871 ± .005	.813 ± .005	.239 ± .006	.143 ± .005	.237 ± .006	.141 ± .005	.130 ± .005	**.221 ± .005**
CSDS1	.99	.857 ± .002	**.863 ± .002**	.857 ± .002	**.863 ± .002**	.861 ± .002	.862 ± .002	.145 ± .002	.139 ± .002	**.144 ± .002**	.138 ± .002	.140 ± .002	.139 ± .002
	.95	.856 ± .002	**.875 ± .002**	.856 ± .002	**.875 ± .002**	.873 ± .002	.871 ± .002	**.146 ± .002**	.126 ± .002	**.146 ± .002**	.126 ± .002	.128 ± .002	.130 ± .002
	.90	.854 ± .002	**.885 ± .002**	.855 ± .002	**.885 ± .002**	.882 ± .002	.883 ± .002	**.147 ± .002**	.116 ± .002	**.147 ± .002**	.116 ± .002	.119 ± .002	.118 ± .002
	.85	.853 ± .002	**.892 ± .002**	.853 ± .002	**.892 ± .002**	.890 ± .002	.890 ± .002	**.149 ± .002**	.109 ± .002	**.149 ± .002**	.109 ± .002	.111 ± .002	.111 ± .002
	.80	.852 ± .002	.898 ± .002	.852 ± .002	**.899 ± .002**	.897 ± .002	.897 ± .002	**.150 ± .002**	.103 ± .002	**.150 ± .002**	.102 ± .002	.104 ± .002	.104 ± .002
	.75	.850 ± .002	**.904 ± .002**	.850 ± .002	**.904 ± .002**	.902 ± .002	.902 ± .002	**.152 ± .002**	.097 ± .002	**.152 ± .002**	.097 ± .002	.099 ± .002	.099 ± .002
CSDS2	.99	.982 ± .002	.982 ± .002	.982 ± .002	.982 ± .002	**.983 ± .002**	.982 ± .002	**.019 ± .002**	**.019 ± .002**	**.019 ± .002**	**.019 ± .002**	.018 ± .002	**.019 ± .002**
	.95	.982 ± .002	.983 ± .002	.982 ± .002	.985 ± .002	**.984 ± .002**	.982 ± .002	.019 ± .002	**.018 ± .002**	.019 ± .002	.016 ± .002	.017 ± .002	.019 ± .002
	.90	.982 ± .002	.984 ± .002	.982 ± .002	.985 ± .002	**.986 ± .002**	.985 ± .002	**.019 ± .002**	.017 ± .002	**.019 ± .002**	.016 ± .002	.015 ± .002	.016 ± .002
	.85	.982 ± .002	.985 ± .002	.981 ± .002	.986 ± .002	**.986 ± .002**	.985 ± .002	**.019 ± .002**	.016 ± .002	**.019 ± .002**	.015 ± .002	.015 ± .002	.016 ± .002
	.80	.982 ± .002	.986 ± .002	.981 ± .002	.987 ± .002	**.987 ± .002**	.986 ± .002	**.019 ± .002**	.015 ± .002	**.019 ± .002**	.014 ± .002	.014 ± .002	.015 ± .002
	.75	.981 ± .002	.986 ± .002	.981 ± .002	.987 ± .002	**.988 ± .002**	.987 ± .002	**.019 ± .002**	.015 ± .002	.020 ± .002	.014 ± .002	.013 ± .002	.014 ± .002
CSDS3	.99	.814 ± .003	**.816 ± .003**	.813 ± .003	**.816 ± .003**	.810 ± .003	.809 ± .003	**.253 ± .003**	.252 ± .003	.254 ± .003	.251 ± .003	.251 ± .003	.252 ± .003
	.95	.817 ± .003	.826 ± .003	.816 ± .003	**.827 ± .003**	.822 ± .003	.820 ± .003	**.254 ± .003**	.242 ± .003	**.254 ± .003**	.242 ± .003	.244 ± .003	.243 ± .003
	.90	.820 ± .003	**.841 ± .003**	.819 ± .003	**.841 ± .003**	.830 ± .003	.835 ± .003	**.254 ± .003**	.228 ± .003	.255 ± .003	.229 ± .003	.233 ± .003	.232 ± .003
	.85	.825 ± .003	**.856 ± .003**	.823 ± .003	.855 ± .003	.844 ± .003	.850 ± .003	**.254 ± .003**	.214 ± .003	.255 ± .003	.215 ± .003	.219 ± .003	.219 ± .003
	.80	.829 ± .003	**.871 ± .003**	.826 ± .003	.869 ± .003	.860 ± .003	.861 ± .003	**.254 ± .004**	.199 ± .003	.256 ± .004	.200 ± .003	.206 ± .003	.207 ± .003
	.75	.835 ± .003	**.884 ± .003**	.830 ± .003	.883 ± .003	.871 ± .003	.872 ± .003	**.254 ± .004**	.182 ± .003	.257 ± .004	.185 ± .003	.178 ± .003	.188 ± .003
CatsVsDogs	.99	**.954 ± .004**	.950 ± .004	.944 ± .004	.956 ± .003	.941 ± .004	.950 ± .004	**.500 ± .008**	**.505 ± .008**	.503 ± .008	**.500 ± .008**	**.500 ± .008**	**.500 ± .008**
	.95	**.968 ± .003**	.949 ± .004	.953 ± .004	.969 ± .003	.964 ± .003	.964 ± .003	**.498 ± .008**	.514 ± .008	.512 ± .008	.495 ± .008	.494 ± .008	.497 ± .008
	.90	.977 ± .003	.947 ± .004	.971 ± .003	.979 ± .003	**.978 ± .003**	.977 ± .003	.494 ± .008	.531 ± .008	.518 ± .008	.490 ± .008	.482 ± .008	**.496 ± .008**
	.85	**.984 ± .002**	.944 ± .004	.982 ± .003	.985 ± .002	.983 ± .002	**.984 ± .002**	.488 ± .008	.546 ± .008	.524 ± .008	.488 ± .008	.485 ± .008	**.494 ± .008**
	.80	.987 ± .002	.942 ± .004	.986 ± .002	.989 ± .002	**.990 ± .002**	.985 ± .002	.473 ± .008	.556 ± .008	.528 ± .008	**.493 ± .008**	.476 ± .008	.482 ± .008
	.75	.988 ± .002	.938 ± .004	.988 ± .002	.990 ± .002	**.990 ± .002**	.987 ± .002	.445 ± .008	.561 ± .009	.529 ± .009	**.500 ± .009**	.472 ± .008	.464 ± .008
CIFAR10-CAT	.99	.952 ± .003	.952 ± .003	.953 ± .003	.956 ± .002	.952 ± .003	**.958 ± .002**	**.101 ± .004**	.092 ± .003	.102 ± .004	.097 ± .004	.098 ± .004	.096 ± .004
	.95	.952 ± .003	.952 ± .003	.951 ± .003	.969 ± .002	.967 ± .002	**.969 ± .002**	**.100 ± .004**	.072 ± .003	.106 ± .004	.078 ± .003	.084 ± .003	.075 ± .003
	.90	.953 ± .003	.950 ± .003	.950 ± .003	**.977 ± .002**	.977 ± .002	.973 ± .002	**.100 ± .004**	.074 ± .003	.111 ± .004	.049 ± .003	.064 ± .003	.044 ± .003
	.85	.954 ± .003	.947 ± .003	.948 ± .003	**.979 ± .002**	.977 ± .002	.976 ± .002	**.099 ± .004**	.077 ± .003	.115 ± .004	.040 ± .003	.024 ± .002	.025 ± .002
	.80	.959 ± .003	.945 ± .003	.946 ± .003	**.980 ± .002**	.984 ± .002	.975 ± .002	**.095 ± .004**	.080 ± .004	.122 ± .004	.037 ± .003	.017 ± .002	.026 ± .002
	.75	.971 ± .002	.942 ± .003	.943 ± .003	.980 ± .002	**.986 ± .002**	.975 ± .002	**.084 ± .003**	**.084 ± .004**	.129 ± .004	.036 ± .003	.015 ± .002	.026 ± .002
	#	4/54	21/54	2/54	29/54	12/54	3/54	39/54	4/54	20/54	5/54	1/54	14/54

Table 5: Time required for running different methods (in seconds).

Training Time (seconds)

	c	AUCross	Plug-In	Plug-In-AUC	SCross	SelNet	SAT
Adult	.99	1.43	**0.22**	**0.22**	1.15	1787.93	2319.66
	.95	1.43	**0.22**	**0.22**	1.15	1779.74	2319.66
	.90	1.43	**0.22**	**0.22**	1.15	1786.99	2319.66
	.85	1.43	**0.22**	**0.22**	1.15	1781.64	2319.66
	.80	1.43	**0.22**	**0.22**	1.15	1789.44	2319.66
	.75	1.43	**0.22**	**0.22**	1.15	1788.53	2319.66
Lending	.99	13.16	**2.55**	**2.55**	12.10	9168.31	14182.26
	.95	13.16	**2.55**	**2.55**	12.10	9198.68	14182.26
	.90	13.16	**2.55**	**2.55**	12.10	9187.33	14182.26
	.85	13.16	**2.55**	**2.55**	12.10	9204.72	14182.26
	.80	13.16	**2.55**	**2.55**	12.10	9165.67	14182.26
	.75	13.16	**2.55**	**2.55**	12.10	9199.28	14182.26
GiveMe	.99	1.60	**0.27**	**0.27**	1.39	2341.00	3104.83
	.95	1.60	**0.27**	**0.27**	1.39	2323.07	3104.83
	.90	1.60	**0.27**	**0.27**	1.39	2332.69	3104.83
	.85	1.60	**0.27**	**0.27**	1.39	2332.97	3104.83
	.80	1.60	**0.27**	**0.27**	1.39	2326.58	3104.83
	.75	1.60	**0.27**	**0.27**	1.39	2333.06	3104.83
UCICredit	.99	1.26	**0.20**	**0.20**	1.11	1547.44	2070.54
	.95	1.26	**0.20**	**0.20**	1.11	1549.75	2070.54
	.90	1.26	**0.20**	**0.20**	1.11	1527.84	2070.54
	.85	1.26	**0.20**	**0.20**	1.11	1547.15	2070.54
	.80	1.26	**0.20**	**0.20**	1.11	1548.82	2070.54
	.75	1.26	**0.20**	**0.20**	1.11	1541.74	2070.54
CSDS1	.99	5.60	**1.02**	**1.02**	5.31	4815.42	5524.00
	.95	5.60	**1.02**	**1.02**	5.31	4812.96	5524.00
	.90	5.60	**1.02**	**1.02**	5.31	4813.66	5524.00
	.85	5.60	**1.02**	**1.02**	5.31	4817.26	5524.00
	.80	5.60	**1.02**	**1.02**	5.31	4818.25	5524.00
	.75	5.60	**1.02**	**1.02**	5.31	4812.89	5524.00
CSDS2	.99	1.23	**0.21**	**0.21**	1.41	609.99	357.92
	.95	1.23	**0.21**	**0.21**	1.41	609.32	357.92
	.90	1.23	**0.21**	**0.21**	1.41	606.45	357.92
	.85	1.23	**0.21**	**0.21**	1.41	606.47	357.92
	.80	1.23	**0.21**	**0.21**	1.41	610.34	357.92
	.75	1.23	**0.21**	**0.21**	1.41	605.20	357.92
CSDS3	.99	5.07	**0.80**	**0.80**	4.60	1188.31	748.88
	.95	5.07	**0.80**	**0.80**	4.60	1193.35	748.88
	.90	5.07	**0.80**	**0.80**	4.60	1192.50	748.88
	.85	5.07	**0.80**	**0.80**	4.60	1190.79	748.88
	.80	5.07	**0.80**	**0.80**	4.60	1191.03	748.88
	.75	5.07	**0.80**	**0.80**	4.60	1191.56	748.88
CatsVsDogs	.99	14668.55	**2624.68**	**2624.68**	14475.64	2700.46	2735.28
	.95	14668.55	**2624.68**	**2624.68**	14475.64	2699.13	2735.28
	.90	14668.55	**2624.68**	**2624.68**	14475.64	2733.35	2735.28
	.85	14668.55	**2624.68**	**2624.68**	14475.64	2769.44	2735.28
	.80	14668.55	**2624.68**	**2624.68**	14475.64	2766.80	2735.28
	.75	14668.55	**2624.68**	**2624.68**	14475.64	2762.34	2735.28
CIFAR10-CAT	.99	17401.61	**3026.98**	**3026.98**	16924.30	3317.14	3239.50
	.95	17401.61	**3026.98**	**3026.98**	16924.30	3296.70	3239.50
	.90	17401.61	**3026.98**	**3026.98**	16924.30	3337.28	3239.50
	.85	17401.61	**3026.98**	**3026.98**	16924.30	3358.18	3239.50
	.80	17401.61	**3026.98**	**3026.98**	16924.30	3262.57	3239.50
	.75	17401.61	**3026.98**	**3026.98**	16924.30	3294.08	3239.50
	#	0/54	54/54	54/54	0/54	0/54	0/54

Table 6: Empirical coverage for AUCross and PlugInAUC using different classifiers (1,000 bootstrap runs over the test set, results as mean ± stdev).

		Empirical Coverage							
		AUCross				PlugInAUC			
	c	Logistic	RandForest	ResNet	XGBoost	Logistic	RandForest	ResNet	XGBoost
Adult	.99	.991 ± (.001)	.988 ± (.001)	.994 ± (.001)	**.990 ± (.001)**	.988 ± (.001)	**.989 ± (.001)**	.988 ± (.001)	.992 ± (.001)
	.95	**.950 ± (.002)**	.945 ± (.002)	.961 ± (.002)	.951 ± (.002)	.947 ± (.002)	.949 ± (.002)	.946 ± (.002)	**.951 ± (.002)**
	.90	**.901 ± (.003)**	.893 ± (.003)	.912 ± (.003)	.891 ± (.003)	.896 ± (.003)	.894 ± (.003)	.903 ± (.003)	**.899 ± (.003)**
	.85	**.850 ± (.003)**	.853 ± (.003)	.840 ± (.004)	.846 ± (.003)	.848 ± (.003)	.836 ± (.004)	.849 ± (.003)	**.850 ± (.003)**
	.80	**.800 ± (.004)**	.804 ± (.004)	.792 ± (.004)	.793 ± (.004)	.799 ± (.004)	.783 ± (.004)	.801 ± (.004)	**.800 ± (.004)**
	.75	.747 ± (.004)	**.752 ± (.004)**	.735 ± (.004)	.744 ± (.004)	.744 ± (.004)	.743 ± (.004)	**.749 ± (.004)**	.748 ± (.004)
Lending	.99	.996 ± (.001)	**.992 ± (.001)**	.995 ± (.001)	.996 ± (.001)	.996 ± (.001)	**.992 ± (.001)**	.995 ± (.001)	.996 ± (.001)
	.95	.976 ± (.001)	**.971 ± (.001)**	.976 ± (.001)	.980 ± (.001)	.976 ± (.001)	**.971 ± (.001)**	.975 ± (.001)	.980 ± (.001)
	.90	.951 ± (.001)	**.945 ± (.001)**	.951 ± (.001)	.958 ± (.001)	.951 ± (.001)	**.945 ± (.001)**	.949 ± (.001)	.958 ± (.001)
	.85	.926 ± (.001)	**.918 ± (.001)**	.925 ± (.001)	.936 ± (.001)	.926 ± (.001)	**.917 ± (.001)**	.922 ± (.001)	.936 ± (.001)
	.80	.900 ± (.001)	**.890 ± (.001)**	.898 ± (.001)	.912 ± (.001)	.899 ± (.001)	**.884 ± (.001)**	.894 ± (.001)	.912 ± (.001)
	.75	.871 ± (.001)	**.856 ± (.001)**	.868 ± (.001)	.886 ± (.001)	.871 ± (.001)	**.855 ± (.001)**	.865 ± (.001)	.886 ± (.001)
GiveMe	.99	**.990 ± (.001)**	.942 ± (.002)	.944 ± (.002)	**.990 ± (.001)**	**.991 ± (.001)**	.944 ± (.002)	.977 ± (.001)	**.991 ± (.001)**
	.95	.953 ± (.002)	.942 ± (.002)	.755 ± (.003)	**.951 ± (.002)**	**.950 ± (.002)**	.944 ± (.002)	.930 ± (.002)	**.948 ± (.002)**
	.90	.904 ± (.002)	.858 ± (.002)	.640 ± (.003)	**.898 ± (.002)**	**.898 ± (.002)**	.863 ± (.002)	.884 ± (.002)	**.898 ± (.002)**
	.85	**.855 ± (.002)**	.788 ± (.003)	.570 ± (.003)	.849 ± (.002)	.844 ± (.002)	.794 ± (.003)	.841 ± (.002)	**.849 ± (.002)**
	.80	.805 ± (.003)	.762 ± (.003)	.499 ± (.003)	**.799 ± (.003)**	.796 ± (.003)	.780 ± (.003)	**.797 ± (.003)**	**.797 ± (.003)**
	.75	.758 ± (.003)	.741 ± (.003)	.431 ± (.003)	**.750 ± (.003)**	.746 ± (.003)	.667 ± (.003)	.748 ± (.003)	**.750 ± (.003)**
UCICredit	.99	.991 ± (.002)	.949 ± (.003)	.987 ± (.002)	**.990 ± (.002)**	**.990 ± (.002)**	.954 ± (.003)	**.991 ± (.002)**	**.990 ± (.002)**
	.95	**.953 ± (.003)**	.949 ± (.003)	.939 ± (.003)	.947 ± (.003)	.962 ± (.003)	.904 ± (.004)	.946 ± (.003)	**.948 ± (.003)**
	.90	**.903 ± (.004)**	.878 ± (.004)	.873 ± (.004)	.894 ± (.004)	.913 ± (.004)	.857 ± (.005)	**.901 ± (.004)**	.894 ± (.004)
	.85	**.853 ± (.005)**	.827 ± (.005)	.814 ± (.005)	.840 ± (.005)	.864 ± (.004)	.838 ± (.005)	**.849 ± (.005)**	.848 ± (.005)
	.80	**.808 ± (.005)**	.777 ± (.005)	.759 ± (.005)	.784 ± (.005)	.816 ± (.005)	.775 ± (.005)	**.796 ± (.005)**	.806 ± (.005)
	.75	**.762 ± (.005)**	.712 ± (.006)	.706 ± (.006)	.729 ± (.006)	.770 ± (.005)	.733 ± (.006)	.744 ± (.006)	**.752 ± (.005)**
CSDS1	.99	**.991 ± (.001)**	.913 ± (.002)	**.991 ± (.001)**	**.991 ± (.001)**	.991 ± (.001)	.961 ± (.001)	**.990 ± (.001)**	.991 ± (.001)
	.95	**.951 ± (.001)**	.913 ± (.002)	.953 ± (.001)	**.951 ± (.001)**	**.950 ± (.001)**	.914 ± (.001)	.949 ± (.001)	**.950 ± (.001)**
	.90	.902 ± (.002)	.870 ± (.002)	.905 ± (.002)	**.901 ± (.002)**	**.900 ± (.002)**	.872 ± (.002)	.899 ± (.002)	.899 ± (.002)
	.85	.852 ± (.002)	.822 ± (.002)	.854 ± (.002)	**.851 ± (.002)**	.851 ± (.002)	.820 ± (.002)	.851 ± (.002)	**.850 ± (.002)**
	.80	.803 ± (.002)	.784 ± (.002)	.797 ± (.002)	**.799 ± (.002)**	**.800 ± (.002)**	.786 ± (.002)	.798 ± (.002)	**.800 ± (.002)**
	.75	.752 ± (.002)	.729 ± (.002)	.744 ± (.002)	**.749 ± (.002)**	**.749 ± (.002)**	.698 ± (.002)	.746 ± (.002)	.752 ± (.002)
CSDS2	.99	.993 ± (.001)	.653 ± (.005)	**.990 ± (.001)**	**.990 ± (.001)**	**.990 ± (.001)**	.829 ± (.004)	.989 ± (.001)	.991 ± (.001)
	.95	.953 ± (.002)	.653 ± (.005)	.942 ± (.003)	**.949 ± (.002)**	**.950 ± (.003)**	.829 ± (.004)	.947 ± (.003)	.954 ± (.002)
	.90	**.903 ± (.003)**	.653 ± (.005)	.881 ± (.003)	.890 ± (.003)	.894 ± (.003)	.829 ± (.004)	**.902 ± (.003)**	.907 ± (.003)
	.85	**.854 ± (.004)**	.653 ± (.005)	.824 ± (.004)	.834 ± (.004)	.841 ± (.004)	.650 ± (.005)	.856 ± (.004)	**.851 ± (.004)**
	.80	**.803 ± (.004)**	.653 ± (.005)	.761 ± (.004)	.781 ± (.004)	**.798 ± (.004)**	.650 ± (.005)	.806 ± (.004)	.806 ± (.004)
	.75	**.756 ± (.004)**	.653 ± (.005)	.703 ± (.005)	.732 ± (.004)	**.749 ± (.004)**	.516 ± (.005)	.756 ± (.004)	.754 ± (.004)
CSDS3	.99	**.990 ± (.001)**	.980 ± (.001)	.992 ± (.001)	.989 ± (.001)	.992 ± (.001)	.989 ± (.001)	**.991 ± (.001)**	**.991 ± (.001)**
	.95	.948 ± (.002)	.933 ± (.002)	.954 ± (.002)	**.949 ± (.002)**	.955 ± (.002)	.941 ± (.002)	**.952 ± (.002)**	.955 ± (.002)
	.90	**.899 ± (.002)**	.887 ± (.003)	.907 ± (.002)	.903 ± (.002)	.905 ± (.002)	.896 ± (.003)	**.901 ± (.003)**	.906 ± (.002)
	.85	**.851 ± (.003)**	.831 ± (.003)	.860 ± (.003)	.854 ± (.003)	.857 ± (.003)	.838 ± (.003)	**.850 ± (.003)**	.860 ± (.003)
	.80	**.799 ± (.003)**	.795 ± (.003)	.789 ± (.003)	.814 ± (.003)	.799 ± (.003)	**.803 ± (.003)**	**.803 ± (.003)**	.814 ± (.003)
	.75	**.744 ± (.003)**	.738 ± (.003)	**.744 ± (.003)**	.760 ± (.003)	.746 ± (.003)	**.748 ± (.003)**	.753 ± (.003)	.762 ± (.003)
	#	22/42	7/42	3/42	16/42	12/42	9/42	13/42	18/42
	V	.015 ± .034	.064 ± .087	.064 ± .108	.019 ± .038	.017 ± .034	.056 ± .069	.016 ± .032	.017 ± .038

Table 7: Selective AUC for AUCross and PlugInAUC using different classifiers (1,000 bootstrap runs over the test set, results as mean ± stdev).

		Selective AUC							
		AUCross				PlugInAUC			
	c	Logistic	RandForest	ResNet	XGBoost	Logistic	RandForest	ResNet	XGBoost
Adult	.99	.903 ± .003	.888 ± .003	.901 ± .003	**.928 ± .003**	.904 ± .003	.886 ± .003	.906 ± .003	**.927 ± .003**
	.95	.910 ± .003	.895 ± .003	.907 ± .003	**.934 ± .003**	.911 ± .003	.892 ± .003	.912 ± .003	**.934 ± .003**
	.90	.918 ± .003	.903 ± .003	.914 ± .003	**.944 ± .002**	.918 ± .003	.900 ± .003	.919 ± .003	**.942 ± .003**
	.85	.926 ± .003	.910 ± .003	.925 ± .003	**.950 ± .002**	.925 ± .003	.908 ± .003	.927 ± .003	**.949 ± .002**
	.80	.934 ± .003	.918 ± .003	.933 ± .003	**.959 ± .002**	.932 ± .003	.916 ± .003	.934 ± .003	**.956 ± .002**
	.75	.941 ± .003	.925 ± .003	.942 ± .003	**.965 ± .002**	.939 ± .003	.922 ± .003	.942 ± .003	**.962 ± .002**
Lending	.99	.956 ± .001	.977 ± .001	.981 ± .001	**.987 ± .001**	.956 ± .001	.976 ± .001	.975 ± .001	**.987 ± .001**
	.95	.958 ± .001	.979 ± .001	.983 ± .001	**.988 ± .001**	.958 ± .001	.979 ± .001	.978 ± .001	**.989 ± .001**
	.90	.962 ± .001	.982 ± .001	.986 ± .001	**.990 ± .001**	.962 ± .001	.982 ± .001	.981 ± .001	**.990 ± .001**
	.85	.965 ± .001	.984 ± .001	.988 ± .001	**.992 ± .001**	.965 ± .001	.984 ± .001	.985 ± .001	**.992 ± .001**
	.80	.968 ± .001	.987 ± .001	.991 ± .001	**.993 ± .001**	.968 ± .001	.987 ± .001	.988 ± .001	**.993 ± .001**
	.75	.971 ± .001	.989 ± .001	.993 ± .001	**.994 ± .001**	.971 ± .001	.989 ± .001	.991 ± .001	**.994 ± .001**
GiveMe	.99	.699 ± .006	.836 ± .005	.670 ± .006	**.865 ± .004**	.698 ± .006	.840 ± .005	.834 ± .005	**.862 ± .004**
	.95	.703 ± .006	.836 ± .005	.688 ± .007	**.871 ± .004**	.702 ± .006	.840 ± .005	.840 ± .005	**.869 ± .004**
	.90	.708 ± .006	.846 ± .005	.705 ± .008	**.878 ± .004**	.708 ± .006	.850 ± .005	.847 ± .005	**.876 ± .004**
	.85	.713 ± .006	.854 ± .005	.720 ± .008	**.885 ± .004**	.714 ± .006	.857 ± .005	.853 ± .005	**.883 ± .004**
	.80	.720 ± .006	.858 ± .005	.732 ± .008	**.892 ± .004**	.720 ± .006	.859 ± .005	.859 ± .005	**.891 ± .004**
	.75	.726 ± .006	.861 ± .005	.749 ± .009	**.899 ± .004**	.727 ± .007	.870 ± .005	.866 ± .005	**.897 ± .004**
UCICredit	.99	.701 ± .009	.760 ± .008	**.768 ± .007**	.767 ± .008	.702 ± .009	.757 ± .008	**.768 ± .007**	.760 ± .008
	.95	.704 ± .009	.760 ± .008	**.774 ± .007**	.774 ± .008	.704 ± .009	.764 ± .008	**.774 ± .007**	.765 ± .008
	.90	.709 ± .009	.769 ± .008	**.783 ± .007**	.780 ± .008	.708 ± .009	.770 ± .008	**.779 ± .007**	.772 ± .008
	.85	.714 ± .009	.776 ± .008	**.792 ± .007**	.787 ± .008	.713 ± .009	.772 ± .008	**.785 ± .007**	.778 ± .008
	.80	.719 ± .009	.784 ± .008	**.799 ± .007**	.794 ± .008	.718 ± .009	.780 ± .008	**.793 ± .008**	.783 ± .008
	.75	.724 ± .009	.794 ± .008	**.807 ± .007**	.801 ± .008	.723 ± .009	.786 ± .008	**.802 ± .008**	.790 ± .008
CSDS1	.99	.676 ± .003	.636 ± .004	.679 ± .003	**.680 ± .003**	.676 ± .003	.634 ± .003	.679 ± .003	**.680 ± .003**
	.95	.679 ± .003	.636 ± .004	.682 ± .003	**.683 ± .003**	.679 ± .003	.636 ± .004	**.683 ± .003**	**.683 ± .003**
	.90	.682 ± .003	.639 ± .004	.685 ± .003	**.686 ± .003**	.682 ± .003	.639 ± .004	.686 ± .003	**.687 ± .003**
	.85	.685 ± .003	.640 ± .004	**.690 ± .004**	**.690 ± .003**	.685 ± .004	.641 ± .004	.690 ± .004	**.691 ± .003**
	.80	.689 ± .004	.643 ± .004	**.694 ± .004**	**.694 ± .004**	.689 ± .004	.644 ± .004	.693 ± .004	**.695 ± .003**
	.75	.693 ± .004	.644 ± .004	**.698 ± .004**	.697 ± .004	.693 ± .004	.647 ± .004	.697 ± .004	**.698 ± .004**
CSDS2	.99	**.615 ± .019**	.604 ± .023	.615 ± .019	.575 ± .021	.616 ± .019	.587 ± .021	**.624 ± .020**	.577 ± .019
	.95	.615 ± .019	.604 ± .023	**.622 ± .020**	.574 ± .021	.621 ± .020	.587 ± .021	**.626 ± .020**	.578 ± .019
	.90	.619 ± .020	.604 ± .023	**.623 ± .020**	.574 ± .021	.627 ± .020	.587 ± .021	**.628 ± .020**	.580 ± .019
	.85	.622 ± .020	.604 ± .023	**.628 ± .020**	.576 ± .022	.631 ± .021	.580 ± .023	**.633 ± .021**	.583 ± .020
	.80	**.626 ± .021**	.604 ± .023	.619 ± .020	.571 ± .022	.636 ± .021	.580 ± .023	**.639 ± .022**	.585 ± .020
	.75	**.633 ± .022**	.604 ± .023	.616 ± .020	.568 ± .022	**.644 ± .022**	.591 ± .025	.639 ± .022	.588 ± .021
CSDS3	.99	.840 ± .003	.838 ± .003	.840 ± .003	**.845 ± .003**	.840 ± .003	.835 ± .003	**.845 ± .003**	.843 ± .003
	.95	.846 ± .003	.845 ± .003	.845 ± .003	**.851 ± .003**	.844 ± .003	.841 ± .003	**.850 ± .003**	.848 ± .003
	.90	.852 ± .003	.851 ± .003	.851 ± .003	**.858 ± .003**	.851 ± .003	.847 ± .003	**.857 ± .003**	.856 ± .003
	.85	.859 ± .003	.859 ± .003	.858 ± .003	**.865 ± .003**	.856 ± .003	.855 ± .004	**.864 ± .003**	.863 ± .003
	.80	.866 ± .003	.865 ± .004	.869 ± .003	**.871 ± .003**	.865 ± .003	.861 ± .004	**.870 ± .003**	.869 ± .003
	.75	.873 ± .003	.873 ± .004	.875 ± .003	**.879 ± .003**	.872 ± .003	.868 ± .004	**.878 ± .003**	.877 ± .003
	#	3/42	0/42	12/42	29/42	1/42	0/42	18/42	24/42

Table 8: Selective accuracy for AUCross and PlugInAUC using different classifiers (1,000 bootstrap runs over the test set, results as mean ± stdev).

		Selective Accuracy							
		AUCross				PlugInAUC			
	c	Logistic	RandForest	ResNet	XGBoost	Logistic	RandForest	ResNet	XGBoost
Adult	.99	.846 ± .003	.838 ± .004	.844 ± .003	**.870 ± .003**	.847 ± .003	.839 ± .004	.844 ± .003	**.869 ± .003**
	.95	.851 ± .003	.842 ± .004	.847 ± .004	**.874 ± .003**	.851 ± .003	.842 ± .004	.847 ± .003	**.874 ± .003**
	.90	.856 ± .003	.847 ± .004	.853 ± .004	**.883 ± .003**	.855 ± .003	.847 ± .004	.853 ± .003	**.880 ± .003**
	.85	.863 ± .004	.853 ± .004	.861 ± .004	**.890 ± .003**	.861 ± .004	.851 ± .004	.860 ± .004	**.888 ± .003**
	.80	.870 ± .004	.860 ± .004	.870 ± .004	**.901 ± .003**	.868 ± .004	.858 ± .004	.866 ± .004	**.896 ± .003**
	.75	.879 ± .004	.868 ± .004	.883 ± .003	**.911 ± .003**	.875 ± .004	.864 ± .004	.874 ± .004	**.905 ± .003**
Lending	.99	.848 ± .001	.878 ± .001	.883 ± .001	**.903 ± .001**	.848 ± .001	.877 ± .001	.874 ± .001	**.904 ± .001**
	.95	.857 ± .001	.890 ± .001	.894 ± .001	**.913 ± .001**	.857 ± .001	.888 ± .001	.884 ± .001	**.914 ± .001**
	.90	.869 ± .001	.903 ± .001	.909 ± .001	**.927 ± .001**	.869 ± .001	.902 ± .001	.896 ± .001	**.928 ± .001**
	.85	.882 ± .001	.918 ± .001	.924 ± .001	**.941 ± .001**	.882 ± .001	.917 ± .001	.910 ± .001	**.942 ± .001**
	.80	.896 ± .001	.933 ± .001	.940 ± .001	**.957 ± .001**	.895 ± .001	.936 ± .001	.925 ± .001	**.957 ± .001**
	.75	.911 ± .001	.952 ± .001	.957 ± .001	**.975 ± .001**	.911 ± .001	.951 ± .001	.941 ± .001	**.974 ± .001**
GiveMe	.99	.934 ± .002	.935 ± .002	.933 ± .002	**.937 ± .002**	.934 ± .002	.935 ± .002	.933 ± .002	**.936 ± .002**
	.95	.934 ± .002	.935 ± .002	.929 ± .002	**.937 ± .002**	.934 ± .002	.935 ± .002	.932 ± .002	**.936 ± .002**
	.90	.934 ± .002	.934 ± .002	.928 ± .002	**.937 ± .002**	.934 ± .002	.934 ± .002	.931 ± .002	**.936 ± .002**
	.85	.933 ± .002	.932 ± .002	.928 ± .002	**.937 ± .002**	.934 ± .002	.932 ± .002	.930 ± .002	**.936 ± .002**
	.80	.934 ± .002	.933 ± .002	.926 ± .002	**.938 ± .002**	.935 ± .002	.932 ± .002	.929 ± .002	**.937 ± .002**
	.75	.934 ± .002	.933 ± .002	.923 ± .003	**.939 ± .002**	.935 ± .002	.929 ± .002	.929 ± .002	**.937 ± .002**
UCICredit	.99	.784 ± .005	.806 ± .005	**.814 ± .005**	.805 ± .005	.785 ± .005	.805 ± .005	**.814 ± .005**	.805 ± .005
	.95	.780 ± .005	.806 ± .005	**.812 ± .005**	.805 ± .005	.783 ± .005	.804 ± .005	**.812 ± .005**	.804 ± .005
	.90	.777 ± .006	.803 ± .005	**.809 ± .005**	.803 ± .005	.777 ± .006	.801 ± .006	**.810 ± .005**	.803 ± .005
	.85	.773 ± .006	.801 ± .006	**.808 ± .006**	.800 ± .006	.774 ± .006	.799 ± .006	**.806 ± .006**	.802 ± .005
	.80	.769 ± .006	.800 ± .006	**.806 ± .006**	.796 ± .006	.770 ± .006	.795 ± .006	**.804 ± .006**	.800 ± .006
	.75	.765 ± .006	.799 ± .006	**.802 ± .006**	.794 ± .006	.765 ± .006	.793 ± .006	**.806 ± .006**	.798 ± .006
CSDS1	.99	**.857 ± .002**	.846 ± .002	**.857 ± .002**	.856 ± .002	**.857 ± .002**	.847 ± .002	**.857 ± .002**	.856 ± .002
	.95	**.856 ± .002**	.846 ± .002	**.856 ± .002**	.855 ± .002	**.856 ± .002**	.845 ± .002	**.856 ± .002**	.854 ± .002
	.90	**.854 ± .002**	.844 ± .002	**.854 ± .002**	.854 ± .002	**.855 ± .002**	.844 ± .002	**.854 ± .002**	.853 ± .002
	.85	**.852 ± .002**	.842 ± .002	**.853 ± .002**	.852 ± .002	**.852 ± .002**	.841 ± .002	**.853 ± .002**	.851 ± .002
	.80	**.851 ± .002**	.840 ± .002	**.851 ± .002**	.850 ± .002	**.851 ± .002**	.840 ± .002	**.851 ± .002**	.850 ± .002
	.75	**.850 ± .002**	.836 ± .002	.849 ± .002	.848 ± .002	**.850 ± .002**	.834 ± .002	.849 ± .002	.848 ± .002
CSDS2	.99	**.982 ± .002**	.980 ± .002	**.982 ± .002**	**.982 ± .002**	**.982 ± .002**	.981 ± .002	**.982 ± .002**	**.982 ± .002**
	.95	**.982 ± .002**	.980 ± .002	**.982 ± .002**	**.982 ± .002**	**.982 ± .002**	.981 ± .002	**.982 ± .002**	**.982 ± .002**
	.90	**.982 ± .002**	.980 ± .002	**.982 ± .002**	.981 ± .002	**.982 ± .002**	.981 ± .002	**.982 ± .002**	**.982 ± .002**
	.85	**.982 ± .002**	.980 ± .002	.981 ± .002	.981 ± .002	**.982 ± .002**	.980 ± .002	**.982 ± .002**	**.982 ± .002**
	.80	**.982 ± .002**	.980 ± .002	.980 ± .002	.980 ± .002	**.982 ± .002**	.980 ± .002	**.982 ± .002**	**.982 ± .002**
	.75	**.982 ± .002**	.980 ± .002	.980 ± .002	.980 ± .002	**.982 ± .002**	.979 ± .002	**.982 ± .002**	**.982 ± .002**
CSDS3	.99	.808 ± .003	.808 ± .003	.806 ± .003	**.811 ± .003**	**.809 ± .003**	.804 ± .003	.808 ± .003	.808 ± .003
	.95	.810 ± .003	.810 ± .003	.807 ± .003	**.813 ± .003**	.809 ± .003	.805 ± .003	.809 ± .003	**.810 ± .003**
	.90	.811 ± .003	.813 ± .003	.808 ± .003	**.815 ± .003**	.809 ± .003	.807 ± .003	.811 ± .003	**.814 ± .003**
	.85	.812 ± .003	.817 ± .003	.810 ± .003	**.818 ± .003**	.810 ± .003	.811 ± .003	.813 ± .003	**.817 ± .003**
	.80	.814 ± .003	.820 ± .003	.817 ± .003	**.822 ± .003**	.812 ± .003	.815 ± .003	.816 ± .003	**.821 ± .003**
	.75	.817 ± .003	.826 ± .003	.820 ± .003	**.828 ± .003**	.815 ± .003	.819 ± .003	.821 ± .003	**.825 ± .003**
	#	12/42	0/42	14/42	26/42	13/42	0/42	17/42	29/42

Table 9: Positive rate for AUCross and PlugInAUC using different classifiers (1,000 bootstrap runs over the test set, results as mean ± stdev).

		Positive Rate							
		AUCross				PlugInAUC			
	c	Logistic	RandForest	ResNet	XGBoost	Logistic	RandForest	ResNet	XGBoost
Adult	.99	**.246 ± .004**	**.246 ± .004**	**.246 ± .004**	**.246 ± .004**	**.246 ± .004**	**.246 ± .004**	**.246 ± .004**	**.246 ± .004**
	.95	.245 ± .004	**.246 ± .004**	**.246 ± .004**	**.246 ± .004**	.246 ± .004	**.247 ± .004**	**.247 ± .004**	.246 ± .004
	.90	.245 ± .004	**.246 ± .004**	.245 ± .004	**.246 ± .004**	.247 ± .004	**.248 ± .004**	.246 ± .004	.248 ± .004
	.85	.245 ± .004	.245 ± .004	.246 ± .004	**.246 ± .004**	.248 ± .004	**.250 ± .004**	.246 ± .004	.246 ± .004
	.80	**.244 ± .004**	**.244 ± .004**	.242 ± .004	**.244 ± .004**	.247 ± .004	**.250 ± .004**	.246 ± .004	.247 ± .004
	.75	**.243 ± .004**	**.243 ± .005**	.238 ± .005	**.243 ± .005**	.248 ± .004	**.250 ± .005**	.245 ± .005	.248 ± .005
Lending	.99	**.224 ± .001**	.222 ± .001	.223 ± .001	**.224 ± .001**	**.224 ± .001**	.222 ± .001	**.224 ± .001**	**.224 ± .001**
	.95	**.216 ± .001**	.213 ± .001	.214 ± .001	**.216 ± .001**	**.217 ± .001**	.213 ± .001	.216 ± .001	.216 ± .001
	.90	**.206 ± .001**	.202 ± .001	.202 ± .001	.205 ± .001	**.207 ± .001**	.202 ± .001	**.207 ± .001**	.205 ± .001
	.85	**.196 ± .001**	.190 ± .001	.190 ± .001	.193 ± .001	**.196 ± .001**	.190 ± .001	**.196 ± .001**	.195 ± .001
	.80	**.184 ± .001**	.179 ± .001	.177 ± .001	.181 ± .001	**.185 ± .001**	.175 ± .001	.184 ± .001	.182 ± .001
	.75	**.171 ± .001**	.165 ± .001	.164 ± .001	.168 ± .001	**.172 ± .001**	.163 ± .001	.171 ± .001	.170 ± .001
GiveMe	.99	.067 ± .002	**.068 ± .002**	**.068 ± .002**	.067 ± .002	**.068 ± .002**	**.068 ± .002**	**.068 ± .002**	**.068 ± .002**
	.95	.067 ± .002	.068 ± .002	**.072 ± .002**	.068 ± .002	.067 ± .002	.068 ± .002	**.069 ± .002**	.068 ± .002
	.90	.068 ± .002	.070 ± .002	**.072 ± .002**	.068 ± .002	.067 ± .002	.069 ± .002	**.070 ± .002**	.068 ± .002
	.85	.068 ± .002	.072 ± .002	**.073 ± .002**	.068 ± .002	.067 ± .002	**.071 ± .002**	**.071 ± .002**	.068 ± .002
	.80	.067 ± .002	.071 ± .002	**.075 ± .002**	.068 ± .002	.067 ± .002	.071 ± .002	**.072 ± .002**	.068 ± .002
	.75	.067 ± .002	.070 ± .002	**.077 ± .003**	.067 ± .002	.066 ± .002	**.075 ± .002**	.073 ± .002	.067 ± .002
UCICredit	.99	.222 ± .005	**.226 ± .006**	.223 ± .005	.222 ± .005	.222 ± .005	**.224 ± .005**	.222 ± .005	.222 ± .005
	.95	**.227 ± .006**	.226 ± .006	.226 ± .006	.223 ± .005	.225 ± .005	**.227 ± .006**	.226 ± .006	.224 ± .005
	.90	.231 ± .006	**.232 ± .006**	**.232 ± .006**	.227 ± .006	.230 ± .006	**.232 ± .006**	.229 ± .006	.227 ± .006
	.85	.235 ± .006	.235 ± .006	**.237 ± .006**	.232 ± .006	.234 ± .006	.234 ± .006	**.236 ± .006**	.230 ± .006
	.80	.239 ± .006	.239 ± .006	**.242 ± .006**	.238 ± .006	.238 ± .006	**.241 ± .006**	.240 ± .006	.233 ± .006
	.75	.243 ± .006	.243 ± .007	**.249 ± .007**	.242 ± .006	.244 ± .006	**.245 ± .006**	.241 ± .006	.238 ± .006
CSDS1	.99	.144 ± .002	**.147 ± .002**	.145 ± .002	.145 ± .002	.144 ± .002	**.146 ± .002**	.144 ± .002	.144 ± .002
	.95	.145 ± .002	**.147 ± .002**	.146 ± .002	.146 ± .002	.145 ± .002	**.147 ± .002**	.145 ± .002	.146 ± .002
	.90	.147 ± .002	**.148 ± .002**	.147 ± .002	.147 ± .002	.147 ± .002	**.148 ± .002**	.147 ± .002	.147 ± .002
	.85	.149 ± .002	**.150 ± .002**	.148 ± .002	.149 ± .002	.149 ± .002	**.150 ± .002**	.148 ± .002	.149 ± .002
	.80	.150 ± .002	**.151 ± .002**	.150 ± .002	**.151 ± .002**	.150 ± .002	**.151 ± .002**	.150 ± .002	.150 ± .002
	.75	.152 ± .002	**.154 ± .002**	.152 ± .002	.153 ± .002	.151 ± .002	**.155 ± .002**	.152 ± .002	.153 ± .002
CSDS2	.99	.019 ± .002	**.021 ± .002**	.019 ± .002	.019 ± .002	.019 ± .002	**.020 ± .002**	.019 ± .002	.019 ± .002
	.95	.019 ± .002	**.021 ± .002**	.019 ± .002	.019 ± .002	.019 ± .002	**.020 ± .002**	.019 ± .002	.019 ± .002
	.90	.019 ± .002	**.021 ± .002**	.019 ± .002	.019 ± .002	.019 ± .002	**.020 ± .002**	.019 ± .002	.019 ± .002
	.85	.019 ± .002	**.021 ± .002**	.020 ± .002	.019 ± .002	.019 ± .002	**.021 ± .002**	.019 ± .002	.018 ± .002
	.80	.019 ± .002	**.021 ± .002**	**.021 ± .002**	.020 ± .002	.019 ± .002	**.021 ± .002**	.019 ± .002	.019 ± .002
	.75	.019 ± .002	**.021 ± .002**	**.021 ± .002**	**.021 ± .002**	.019 ± .002	**.022 ± .002**	.019 ± .002	.019 ± .002
CSDS3	.99	**.254 ± .003**	**.254 ± .003**	**.254 ± .003**	**.254 ± .003**	**.254 ± .003**	**.254 ± .003**	**.254 ± .003**	**.254 ± .003**
	.95	**.255 ± .003**	**.255 ± .003**	**.255 ± .003**	.254 ± .003	**.256 ± .003**	.255 ± .003	**.256 ± .003**	.255 ± .003
	.90	**.258 ± .003**	.256 ± .003	.257 ± .003	.255 ± .003	**.259 ± .003**	.256 ± .003	.258 ± .003	.254 ± .003
	.85	**.261 ± .003**	.257 ± .004	.259 ± .003	.256 ± .003	**.263 ± .003**	.257 ± .004	.260 ± .003	.255 ± .004
	.80	**.263 ± .004**	.257 ± .004	.258 ± .004	.256 ± .004	**.265 ± .004**	.256 ± .004	.261 ± .004	.255 ± .004
	.75	**.266 ± .004**	.257 ± .004	.259 ± .004	.255 ± .004	**.268 ± .004**	.257 ± .004	.261 ± .004	.256 ± .004
	#	16/42	22/42	16/42	11/42	14/42	27/42	13/42	4/42

Table 10: Performance metrics for AUCross using different number of folds K over tabular datasets (1,000 bootstrap runs over the test set, results as mean ± stdev).

		Empirical Coverage					**Selective AUC**				
	c	$K=2$	$K=3$	$K=5$	$K=7$	$K=10$	$K=2$	$K=3$	$K=5$	$K=7$	$K=10$
Adult	.99	**.990 ± .001**	.991 ± .001	.989 ± .001	.991 ± .001	.991 ± .001	**.929 ± .003**	.928 ± .003	**.929 ± .003**	.928 ± .003	.928 ± .003
	.95	.952 ± .002	.951 ± .002	**.950 ± .002**	.949 ± .002	.947 ± .002	.935 ± .003	.935 ± .003	.935 ± .003	.935 ± .003	**.936 ± .003**
	.90	.896 ± .003	.897 ± .003	.896 ± .003	.896 ± .003	**.900 ± .003**	**.943 ± .002**	**.943 ± .002**	**.943 ± .002**	**.943 ± .002**	**.943 ± .002**
	.85	.848 ± .003	**.850 ± .003**	.849 ± .003	.848 ± .003	.849 ± .003	**.950 ± .002**	**.950 ± .002**	**.950 ± .002**	**.950 ± .002**	**.950 ± .002**
	.80	.796 ± .004	.797 ± .004	.797 ± .004	.799 ± .004	**.800 ± .004**	**.958 ± .002**	**.958 ± .002**	**.958 ± .002**	**.957 ± .002**	**.957 ± .002**
	.75	**.750 ± .004**	.752 ± .004	**.750 ± .004**	.753 ± .004	.751 ± .004	.963 ± .002	.963 ± .002	.963 ± .002	**.964 ± .002**	**.964 ± .002**
Lending	.99	**.996 ± .001**	**.996 ± .001**	**.996 ± .001**	**.996 ± .001**	**.996 ± .001**	.983 ± .001	.983 ± .001	.983 ± .001	.983 ± .001	.983 ± .001
	.95	**.980 ± .001**	**.980 ± .001**	**.980 ± .001**	**.980 ± .001**	**.980 ± .001**	.985 ± .001	.985 ± .001	.985 ± .001	.985 ± .001	.985 ± .001
	.90	.958 ± .001	**.959 ± .001**	**.959 ± .001**	**.959 ± .001**	**.959 ± .001**	.987 ± .001	.987 ± .001	.987 ± .001	.987 ± .001	.987 ± .001
	.85	.935 ± .001	**.936 ± .001**	**.936 ± .001**	**.936 ± .001**	**.936 ± .001**	.989 ± .001	.989 ± .001	.989 ± .001	.989 ± .001	.989 ± .001
	.80	**.912 ± .001**	**.912 ± .001**	**.912 ± .001**	**.912 ± .001**	**.912 ± .001**	.990 ± .001	.990 ± .001	.990 ± .001	.990 ± .001	.990 ± .001
	.75	**.886 ± .001**	**.886 ± .001**	**.886 ± .001**	**.886 ± .001**	**.886 ± .001**	.992 ± .001	.992 ± .001	.992 ± .001	.992 ± .001	.992 ± .001
GiveMe	.99	**.990 ± .001**	**.990 ± .001**	**.990 ± .001**	**.990 ± .001**	.991 ± .001	**.868 ± .004**	**.868 ± .004**	**.868 ± .004**	**.868 ± .004**	.867 ± .004
	.95	.951 ± .002	**.950 ± .002**	**.950 ± .002**	**.950 ± .002**	**.950 ± .002**	**.874 ± .004**	**.874 ± .004**	**.874 ± .004**	**.874 ± .004**	**.874 ± .004**
	.90	**.900 ± .002**	.899 ± .002	.898 ± .002	.899 ± .002	.898 ± .002	.882 ± .004	.882 ± .004	**.883 ± .004**	**.883 ± .004**	**.883 ± .004**
	.85	**.850 ± .002**	.849 ± .002	.848 ± .002	.849 ± .002	.849 ± .002	**.890 ± .004**	**.890 ± .004**	**.890 ± .004**	**.890 ± .004**	**.890 ± .004**
	.80	.794 ± .003	.795 ± .003	.795 ± .003	**.796 ± .003**	**.796 ± .003**	**.898 ± .004**	.897 ± .004	**.898 ± .004**	**.898 ± .004**	.897 ± .004
	.75	.747 ± .003	.749 ± .003	.748 ± .003	.749 ± .003	**.750 ± .003**	**.904 ± .004**	.903 ± .004	**.904 ± .004**	**.904 ± .004**	.903 ± .004
UCICredit	.99	.989 ± .002	.989 ± .002	.988 ± .002	.988 ± .002	**.990 ± .002**	.772 ± .008	.772 ± .007	.772 ± .008	.772 ± .008	.772 ± .008
	.95	.944 ± .003	**.946 ± .003**	.944 ± .003	.945 ± .003	**.946 ± .003**	.778 ± .008	.778 ± .008	.778 ± .008	.778 ± .008	.778 ± .008
	.90	.886 ± .004	.894 ± .004	.896 ± .004	.895 ± .004	**.900 ± .004**	**.786 ± .008**	.785 ± .008	.785 ± .008	.784 ± .008	.784 ± .008
	.85	.830 ± .005	.838 ± .005	.840 ± .005	.840 ± .005	**.843 ± .005**	**.794 ± .008**	.793 ± .007	.792 ± .008	.792 ± .008	.792 ± .008
	.80	.775 ± .005	.789 ± .005	.789 ± .005	.788 ± .005	**.795 ± .005**	**.802 ± .008**	.800 ± .008	.800 ± .008	.800 ± .008	.800 ± .008
	.75	.720 ± .006	.737 ± .006	.737 ± .006	.735 ± .006	**.745 ± .006**	**.811 ± .008**	.808 ± .008	.809 ± .008	.809 ± .008	.807 ± .008
CSDS1	.99	.991 ± .001	**.990 ± .001**	**.990 ± .001**	**.990 ± .001**	**.990 ± .001**	.686 ± .003	.686 ± .003	.686 ± .003	.686 ± .003	.686 ± .003
	.95	**.950 ± .001**	**.950 ± .001**	**.950 ± .001**	.949 ± .001	**.950 ± .001**	.689 ± .003	.689 ± .003	.689 ± .003	.689 ± .003	.689 ± .003
	.90	.898 ± .002	**.899 ± .002**	.898 ± .002	.898 ± .002	.898 ± .002	.693 ± .003	.693 ± .003	.693 ± .003	.693 ± .003	.693 ± .003
	.85	.848 ± .002	**.849 ± .002**	**.849 ± .002**	.848 ± .002	.848 ± .002	.697 ± .003	.697 ± .003	.697 ± .003	.697 ± .003	.697 ± .003
	.80	.796 ± .002	**.799 ± .002**	**.799 ± .002**	.797 ± .002	.797 ± .002	.702 ± .004	.702 ± .004	.702 ± .004	.702 ± .004	.702 ± .004
	.75	.746 ± .002	**.748 ± .002**	**.748 ± .002**	.747 ± .002	.747 ± .002	.706 ± .004	.706 ± .004	.706 ± .004	.706 ± .004	.706 ± .004
CSDS2	.99	**.990 ± .001**	.992 ± .001	**.990 ± .001**	**.990 ± .001**	.992 ± .001	.615 ± .020	**.618 ± .020**	.616 ± .020	**.618 ± .020**	.617 ± .020
	.95	.952 ± .002	**.951 ± .002**	.952 ± .002	.956 ± .002	.956 ± .002	**.622 ± .020**	.619 ± .020	.619 ± .020	.619 ± .020	.621 ± .020
	.90	**.904 ± .003**	**.904 ± .003**	**.904 ± .003**	.909 ± .003	.911 ± .003	.620 ± .020	**.626 ± .020**	.620 ± .020	.625 ± .020	.624 ± .020
	.85	**.854 ± .004**	**.854 ± .004**	.860 ± .004	.861 ± .003	.866 ± .003	.622 ± .020	.629 ± .021	**.631 ± .021**	.626 ± .021	.626 ± .021
	.80	.808 ± .004	**.804 ± .004**	.811 ± .004	.814 ± .004	.819 ± .004	.628 ± .021	**.636 ± .021**	.631 ± .021	.632 ± .021	.630 ± .021
	.75	.760 ± .004	**.754 ± .004**	.760 ± .004	.765 ± .004	.770 ± .004	.628 ± .021	**.637 ± .021**	.635 ± .021	.634 ± .021	.630 ± .021
CSDS3	.99	**.991 ± .001**	.992 ± .001	**.991 ± .001**	**.991 ± .001**	**.991 ± .001**	.851 ± .003	.851 ± .003	.851 ± .003	.851 ± .003	.851 ± .003
	.95	.948 ± .002	**.950 ± .002**	.948 ± .002	**.950 ± .002**	.951 ± .002	**.858 ± .003**	.857 ± .003	**.858 ± .003**	.857 ± .003	.857 ± .003
	.90	.902 ± .002	**.901 ± .002**	**.901 ± .002**	.902 ± .002	**.901 ± .002**	.865 ± .003	.865 ± .003	.865 ± .003	.865 ± .003	.865 ± .003
	.85	**.850 ± .003**	.853 ± .003	.849 ± .003	**.850 ± .003**	**.850 ± .003**	.873 ± .003	.873 ± .003	.873 ± .003	.873 ± .003	.873 ± .003
	.80	.809 ± .003	.810 ± .003	**.805 ± .003**	**.805 ± .003**	**.805 ± .003**	.879 ± .003	.879 ± .003	**.880 ± .003**	**.880 ± .003**	**.880 ± .003**
	.75	.760 ± .003	.759 ± .003	.759 ± .003	**.758 ± .003**	**.758 ± .003**	.886 ± .003	**.887 ± .003**	**.887 ± .003**	**.887 ± .003**	**.887 ± .003**
	#	15/42	23/42	20/42	16/42	24/42	29/42	28/42	31/42	30/42	27/42
	V	.019 ± .038	.017 ± .038	.018 ± .038	.018 ± .038	.018 ± .038					