MSL: Multi-class Scoring Lists for Interpretable Incremental Decision-Making

Stefan Heid[1](✉), Jaroslaw Kornowicz[2](✉), Jonas Hanselle[1,3], Kirsten Thommes[2], and Eyke Hüllermeier[1,3,4]

[1] LMU Munich, Geschwister-Scholl-Platz 1, 80539 Munich, Germany
{stefan.heid,jonas.hanselle,eyke}@lmu.de
[2] Paderborn University, Warburger Street 100, 33098 Paderborn, Germany
{jaroslaw.koronowicz,kirsten.thommes}@upb.de
[3] Munich Center for Machine Learning (MCML), Munich, Germany
[4] German Research Center for Artificial Intelligence (DFKI, DSA), Kaiserslautern, Germany

Abstract. A scoring list is a sequence of simple decision models, where features are incrementally evaluated and scores of satisfied features are summed to be used for threshold-based decisions or for calculating class probabilities. In this paper, we introduce a new multi-class variant and compare it against previously introduced binary classification variants for incremental decisions, as well as multi-class variants for classical decision-making using all features. Furthermore, we introduce a new multi-class dataset to assess collaborative human-machine decision-making, which is suitable for user studies with non-expert participants. We demonstrate the usefulness of our approach by evaluating predictive performance and compared to the performance of participants without AI help.

Keywords: machine learning · decision support · scoring systems · user study

1 Introduction

Machine Learning (ML) methods have achieved remarkable accomplishments in various application domains. While complex and powerful methods like deep neural networks offer state-of-the-art predictive accuracy, they lack transparency and inherent explainability, which are key requirements for high-stakes decision-making [3]. In general, there are two competing approaches navigating the accuracy-explainability trade-off in ML [2,4]: On the one hand, complex models may be accompanied by simplistic *post-hoc* explanation methods like LIME [39] and SHAP [28]. These can be applied to any predictive and complex model and help mitigate some lack of transparency by explaining individual predictions. Yet, they fail to provide full transparency.

R. Guidotti et al. (Eds.): xAI 2025, CCIS 2578, pp. 117–139, 2026.
https://doi.org/10.1007/978-3-032-08327-2_6

An alternative approach is the use of less complex models that are genuinely interpretable, also known as *ante-hoc* explanation. Corresponding models have a restricted, simple structure that humans can inspect, offering a global understanding of how different features influence the model predictions without the need for additional explanation. This inherent property of explainability makes them an appropriate choice for decision support in high-stakes domains [42] when human understanding and accountability are required.

One of the most prominent model classes of this kind are scoring systems with a long-standing tradition in clinical decision-making [38]. Simply put, they assign an integer-valued score to each (binary) feature, and a decision is made by comparing the sum of all scores for present features to a threshold. Recently, the need for situation-adapted decision models of such kind has been addressed with Probabilistic Scoring Lists (PSL) [15], for which a prediction can be made with any prefix of features in an ordered list. This allows for adjusting the decision process by stopping the feature acquisition once a prediction can be made with sufficient confidence for the decision context at hand. A PSL is a simple model that can be handled by lay persons [19].

While these methods have shown promising performance for the binary case, they have not yet been adapted to polychotomous decision situations in which three or more options are considered. However, many real-world applications are multi-class problems, at least if there is more than one option available (in addition to "do nothing"). For instance, in many medical situations, there is more than one treatment available in addition to "do nothing", which makes this scenario already a multi-class problem.

In this paper, we introduce *Multi-class Scoring Lists* (MSL), an extension of PSL to accommodate multi-class predictions. We evaluate the MSL's predictive performance against various baselines on benchmark datasets, and we observe a favorable compromise between accuracy and interpretability. Additionally, we introduce a new dataset rooted in the sports domain that is particularly well-suited for studies on human-AI interaction. To this end, we have conducted a first study to compare participants' predictive performance on the dataset with the introduced model class.

2 Related Work

Scoring systems are widely utilized in medical applications, including the assessment of atrial fibrillation [27], pancreatitis [32], pneumonia [21], strokes [12], and infants [52]. While their simplistic architecture may result in reduced accuracy, their transparency and ease of use allow for application without computational support. Additionally, such transparency and interpretability can lead to higher acceptance. However, the potential increase in cognitive load compared to so-called "black-box" decision support systems should be considered to avoid causing the opposite effect [29,36].

Traditionally, scoring systems have been manually designed based on domain expertise. However, recent advancements have introduced data-driven methods, such as Supersparse Linear Integer Models (SLIM) and RiskSLIM, which employ mixed-integer programming (MIP) [46,47], as well as Interval Coded Scoring (ICS) [5,7,8].

Although these models have demonstrated effectiveness in binary decision-making, there remains a need for scoring systems capable of handling multi-class classification. Established scoring systems are either considering the pure binary setting, like the PERC rule [22], or scenarios in which multiple classes exhibit an ordinal structure, most notably risk classes in the clinical setting, e.g., the SAPS or APACHE scores [23,30]. In the first case, the total score is compared to a threshold to make the decision, while in the latter case, the risk classes correspond to predefined intervals, and membership is determined by checking in which interval the total score falls. However, little attention has been paid to the multi-class setting with nominal categories, and only a few proposed methods exist. Rouzot et al. propose a one-versus-rest decomposition on top of SLIM for solving multi-class classification problems [41]. While this is a natural approach to transforming a binary into a multi-class classifier, the resulting decomposition has one classifier per class. The more recent approach, MISS, uses a multinomial approach instead [13]. Many existing multi-class approaches leverage mixed-integer nonlinear programming for model learning [13,41].

Despite their potential, both binary and multi-class scoring models face a critical limitation: they become inapplicable when essential feature data is unavailable. This challenge arises in scenarios where data acquisition is costly or when decision-makers operate under time constraints, limiting the available information [6,45].

To address these constraints, adaptive decision support frameworks are required. One approach involves decision lists, which apply predefined rules for prediction. If no applicable rule is found, the decision-making process is deferred to the next rule in the sequence [40]. Heid et al. [18] propose a framework of complexity-ordered catalogues of models, where each successive model incorporates an additional rule compared to its predecessor, along with a methodology for learning these models. Expanding on this concept, probabilistic scoring lists have been introduced [15]. These systems, structured as sequentially dependent scoring models, function similarly to decision lists but provide probabilistic rather than deterministic predictions, akin to RiskSLIM.

3 Multi-class Scoring List

We consider a decision-making scenario in which decisions have to be made for varying contexts that are specified in terms of binary features $\mathcal{F} = \{f_1, \dots f_K\}$. Moreover, decisions are incremental in the sense that the concrete values $x_i \in \{0, 1\}$ of these features are acquired in a stagewise fashion, one after another, in a prespecified order. At each of these stages $1, \dots, K$, the decision-maker (DM) has the option to make a decision immediately or gather additional evidence in terms of further feature values, until all features are exhausted. When learning an arbitrary set of classifiers, e.g., logistic regression models, those models do not share any parameters, which makes it impossible to carry over partial results from previous stages. Decision lists on the other hand are a joint model and can also be interpreted as a sequence of models with coherence constraints.

The multi-class scoring list (MSL) is a decision support model tailored to this scenario and is formally defined as follows:

Definition 1. *A* ***multi-class scoring list*** *(MSL) over candidate features $\mathcal{F}$ and score set $\mathcal{S} \subset \mathbb{Z}$ is a triple $h = \langle F, \boldsymbol{S}, \boldsymbol{b} \rangle$, where $F = (f_1, \dots, f_K)$ is a list of (distinct) features from $\mathcal{F}$, $\boldsymbol{S} \in \mathcal{S}^{C \times K}$ is a score matrix and $\boldsymbol{b} \in \mathcal{S}^C$ is a bias term, where $\mathcal{Y}$ is the set of classes and $C = |\mathcal{Y}|$ is the number of elements therein.*

At prediction time, stagewise decisions are formed in the following manner.

- Let $\boldsymbol{s}^{(k)} = (s_1^{(k)}, \dots, s_C^{(k)})$ denote the cumulative score vector at stage k, with $s_c^{(k)}$ the score of class c. At stage $k = 0$, where no features have been evaluated yet, the scores are formed by the bias term
$$\boldsymbol{s}^{(0)} = \boldsymbol{b}$$
that can be interpreted as a general tendency towards a certain decision when no information is available.
- For subsequent stages $k > 0$, the cumulative scores are given by
$$s_c^{(k)} = s_c^{(k-1)} + S_{c,k} \cdot x_k, \quad \forall\, c \in \mathcal{Y}$$
where $S_{c,k}$ is the score associated with class c at stage k (feature f_k) in the score matrix $\boldsymbol{S}$.
- After computing the cumulative class scores, the prediction for stage k can be conducted by computing the argmax set of these scores
$$\hat{y} = \underset{c \in \mathcal{Y}}{\arg\max}\, s_c^{(k)} \tag{1}$$
Note that the argmax of the cumulative scores may indeed be ambiguous due to the discrete nature of the scores. Hence, the prediction $\hat{y}$ can be set-valued, if several classes are scored maximally likely. This is a natural way for the predictor to express its uncertainty about a predictive outcome [31].
- Another practical interpretation of the cumulative class scores $s_c^{(k)}$ is to use them as logits for the softmax function. This way, we can obtain probabilistic predictions
$$\widehat{p}_c = \frac{\exp\left(s_c^{(k)}\right)}{\sum_{c' \in \mathcal{Y}} \exp\left(s_{c'}^{(k)}\right)}, \quad \forall\, c \in \mathcal{Y} \tag{2}$$
where $\widehat{p}_c$ denotes the estimated probability for class c. Therefore, multiple maximal scores in the discrete decision scenario will be converted into equal predictive probabilities in the probabilistic setting.

– At every stage k, the decision maker can either exit with decision (1) or continue the process and acquire the next feature f_{k+1}. This question will mainly be answered on the basis of the probability estimates (2), which provides information about the confidence in the decision (1).

Table 1. Example of a multi-class scoring list for football player classification. The numerical features have been binarized through thresholding (The binarization thresholds are as follows: *Many shots* > 0.55; *Long Playing Time* > 78.8; *High Pass Success Rate* > 74.5; *Many Aerial Duels Won/Match* > 0.65; *Tall Player* > 183.5.). The model was trained using a score set $\{0, \pm1, \pm2, \pm3\}$ and L_2 regularization of 10^{-6}.

Feature	Forward	Midfielder	Defender	Goalkeeper
⟨Bias⟩	0	1	1	0
Many Shots	2	2	0	–3
Long Playing Time	–3	–1	2	3
High Pass Success Rate	–1	1	1	–1
Many Aerial Duels Won/Match	2	0	1	-3
Tall Player	–1	–1	–1	2

Table 1 shows an exemplary MSL for classifying positions of football players. The first row corresponds to the bias term: Here, the class *forward* and the class *goalkeeper* have a score of 0, while *midfielder* and *defender* have a score of 1. These bias scores, which are available before acquiring any feature values, hint at the marginal distribution of classes. Overall, there are more midfielders and defenders in a team than there are goalkeepers and forwards. The first feature acquired is the average number of shots per match (second row). This feature carries positive evidence for the classes *forward* and *midfielder*, no evidence for *defender*, and strong negative evidence for *goalkeeper*. Again, this is intuitively reasonable, as most shots are performed by players in offensive positions and definitely not by goalkeepers. This can be continued until all features have been consumed, and the final prediction is formed.

The MSLs score set $\mathcal{S}$ is specified in advance according to the DMs preferences and typically comprises a set of small integers reflecting different levels of "evidence" in favor or against a decision. For example, the score set $\mathcal{S} = \{0, \pm1, \pm2, \pm3\}$ distinguishes three levels of evidence: weak, medium, and strong. Assigning a score of $+1$ to a feature then means that the presence of that feature provides weak evidence in favor of a decision, whereas a score of -3 means strong evidence against that decision. Restricting the magnitude and number of admissible scores ensures that the resulting model is cognitively tractable for a human expert. The influence of an individual feature can be immediately understood and communicated, and in principle, predictions could even be made without the help of computing devices.

3.1 Connections to Other Interpretable Probabilistic Classifiers

Given the simple and inherently interpretable structure of MSL, one may wonder how it distinguishes itself from other simple probabilistic classifiers. Most notably, MSL resembles a multinomial logistic regression (MLR) with two major differences: First, MLR has unbounded real-valued coefficients which are harder to understand than integer-valued scores that stem from a small, predefined score-set. Secondly, MLR does not provide stagewise predictions but uses the full feature set for all predictions.

Another natural connection can be drawn to the Naïve Bayes (NB) classifier, which models the posterior probability of class c given a feature vector $\boldsymbol{x}$ as

$$P(c \mid \boldsymbol{x}) = \frac{P(c)\prod_{k=1}^{K} P(x_k \mid c)}{P(\boldsymbol{x})}.$$

Taking the logarithm on both sides yields

$$\log P(c \mid \boldsymbol{x}) = \log P(c) + \sum_{k=1}^{K} \log P(x_k \mid c) - \log P(\boldsymbol{x}) \tag{3}$$

which shows the relation to MSL. The $\log P(c)$ correspond to the bias term $\boldsymbol{b}$ and the log-likelihoods $\log P(x_k \mid c)$ in the sum correspond to the stagewise class scores $s_c^{(k)}$. As $P(\boldsymbol{x})$ is constant across all classes, it only serves to normalize the values to form a valid probability distribution over class labels and can be neglected.

Unlike MSL and MLR, NB can make predictions with any subset of features, even without adhering to a predefined order, making it an interesting choice for situated decision support. However, there are again two major disadvantages compared to MSL: The log-likelihoods in NB are not restricted to a predefined score set, yielding the same disadvantages regarding score complexity as MLR. Additionally, the probability estimates in NB are built upon the naïve assumption of conditional independence and are formed by normalization. MSL can implicitly model feature dependencies by selecting scores that reflect the combined influence of multiple correlated features on the predicted probability.

3.2 Learning Multi-Class Scoring Lists

Consider a standard supervised learning setting in which the data generating process is characterized by a joint probability distribution $P(\boldsymbol{x}, y)$ over $\mathcal{X} \times \mathcal{Y}$. Given a loss function $\ell(\hat{y}, y)$ that quantifies how different a prediction $\hat{y}$ is from the true outcome y, the risk of a classifier $h \colon \mathcal{X} \longrightarrow \mathcal{Y}$ is defined as

$$R(h) = \mathbb{E}\left[\ell\left(h(\boldsymbol{x}), y\right)\right] = \int \ell\left(h(x), y\right)\, dP(\boldsymbol{x}, y). \tag{4}$$

As the distribution $P(\boldsymbol{x}, y)$ is unknown, the true risk is substituted with the empirical risk on observed training data $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}, y)\}_{n=1}^{N}$:

$$R_{\text{emp}}(h) = \frac{1}{N}\sum_{n=1}^{N} \ell\big(h(\boldsymbol{x}_{\boldsymbol{n}}), y_n\big)$$

In our listwise scenario, we are not considering a single model, but rather a sequence of models $h = (h_1, \dots, h_K) \in \mathcal{H}^K$, where $\mathcal{H}$ is an underlying hypothesis space (in our case the set of scoring systems). The learning objective is to find such a sequence of models, or decision list, that has minimal global risk throughout the stages, i.e.,

$$h^* \in \operatorname*{arg\,min}_{h \in \mathcal{H}} \; R(h_1) \oplus R(h_2) \oplus \cdots \oplus R(h_K)\,, \tag{5}$$

where $\oplus$ is a suitable aggregation operator[1] (e.g., the sum).

It is important to note that an optimal decision list (5) does not necessarily consist of elements h_k that have minimal stagewise risk, as the stage-optimal models may not constitute a valid MSL due to conflicting feature selections and score assignments. Hence, the problem is not decomposable in the sense that we could simply identify optimal models for the individual stages and combine them into a decision list.

In the following, we propose a learning algorithm for inferring MSLs from training data $\mathcal{D}_{\text{train}}$. The learning algorithm has to identify three components, that is, the order of features F, the score matrix $\boldsymbol{S}$, and the bias term $\boldsymbol{b}$. Note that this search space is rather large, precisely, the number of candidate MSLs for a score set $\mathcal{S}$, K features, and C classes is

$$K! \cdot |\mathcal{S}|^{(C \cdot (K+1))}\,,$$

as it consists of all possible feature permutations and score assignments. Needless to say, an exhaustive search in such a huge space is not feasible. Thus, a heuristic approach has to be employed, that does not consider all candidate solutions. A natural strategy is to build the model bottom-up and stage by stage, starting with an empty list, first identifying the bias term, and then adding locally optimal features and score assignments for each stage consecutively.

An illustration of such a greedy forward selection procedure is given in Algorithm 1. The function EVALUATE is used to compute a loss value for candidate solutions, fully specified through F, $\boldsymbol{S}$ and $\boldsymbol{b}$, given the training data $\mathcal{D}_{\text{train}}$. The core of the greedy forward selection is the loop starting in line 2, that continues until all available features have been added to the MSL. In the first iteration, the bias term $\boldsymbol{b}$ is identified by considering all possible $\boldsymbol{b} \in \mathcal{S}^C$.

Afterwards, the subsequent stages are constructed: In each iteration, the locally optimal extension of the current MSL is identified by selecting the feature $f \in \bar{F}$ and corresponding score vector $\boldsymbol{s} \in \mathcal{S}^C$ that minimizes the loss achieved on the training data in line 6. As there are $|\mathcal{S}|^C$ many possible score vectors and $|\bar{F}|$ many remaining features, this step takes $|\mathcal{S}|^C \cdot |\bar{F}|$ many calls of EVALUATE. In the beginning, we start with the full feature set and have $|\bar{F}| = K$, which is

[1] The learning algorithm we propose below is of heuristic (greedy) nature and does not directly optimize a specific global risk. Therefore, the concrete form of $\oplus$ is not that important. The essential property assumed by the algorithm is the monotonicity of $\oplus$, which is naturally fulfilled by all meaningful candidates.

reduced by 1 in each iteration, as features are being added to the MSL. This results in

$$|\mathcal{S}|^C + \frac{K(K+1)}{2} \cdot |\mathcal{S}|^C \in \mathcal{O}(K^2 \cdot |\mathcal{S}|^C)$$

overall calls of EVALUATE for identifying the entire MSL including the bias term.

The loss function ℓ can be instantiated with any meaningful loss that compares a class label y with probability estimates. We choose the well-established cross-entropy loss

$$\ell(\widehat{\boldsymbol{p}}, y) = -\log \widehat{p}_y \,, \tag{6}$$

where $\widehat{p}_y$ is the predicted probability for the true class label y.

To further trade-off interpretability and performance, an L_2-loss of all scores can be added to the cross-entropy loss as a regularizer. This yields models with even smaller scores with often little to no expense in performance.

4 Football Player Dataset

Along with the MSL, we introduce a dataset containing the career statistics of football players and their position (*goalkeeper*, *defender*, *midfielder*, and *forward*) as the classification label. Although a classification of players to their positions may not look like a very important problem, it provides distinct advantages in experimental human-(X)AI interaction research.

In human-(X)AI interaction experiments, participants are often assigned classification tasks drawn from various datasets and task types, such as quiz question answering [10], and playing chess moves [9]. While several well-known tabular datasets exist for binary classification and regression in human-(X)AI experiments (e.g., income [24], recidivism [49], or house pricing [43]), there is a lack of comparable datasets for multi-class classification [26]. This gap is partly due to the common practice of recruiting lay participants—often via crowdsourcing platforms like Prolific—which necessitates easy understandable tasks to ensure valid results.

In our view, popular multi-class datasets in the machine learning literature, such as *iris* [11], *wine* [1], and *heart* [20], do not fully meet this criterion and therefore cannot be as readily adapted for human-(X)AI research as the aforementioned binary and regression datasets. Football, as the world's most popular sport, offers a clear advantage in this context. Its universal appeal ensures that a diverse participant pool is already familiar with the game, enhancing both task engagement and the reliability of study outcomes.

Our raw dataset comprises 5,449 active professional football players from eight professional leagues[2]. It includes all players who were on their teams' rosters at the time of data collection. The dataset contains performance statistics spanning each player's entire career up to the time the dataset was compiled

[2] England and Germany (1st and 2nd divisions), and the top-tier (1st division) leagues in Spain, France, Italy, Portugal, the Netherlands, Russia, Turkey, and the USA.

Algorithm 1: Greedy MSL

```
input  : dataset $\mathcal{D}_{\text{train}}$ , number of classes $C$,
         set of all features $\mathcal{F}$ including the $\emptyset$ for the bias term
         the available scores $\mathcal{S}$, loss function $\ell$ to evaluate a hypothesis
output : MSL model $h$
$F, \boldsymbol{S} \leftarrow (), []$
# While not all features have been used. For set difference and
  inequality operators we treat $F$ as a set for ease of notation.
while $F \neq \mathcal{F}$ do
    # In the first iteration compute the bias term
    if $\emptyset \notin F$ then
        $\boldsymbol{b} \leftarrow \arg\min_{\boldsymbol{b} \in \mathcal{S}^C} \left\{\text{EVALUATE}\left((), [], \boldsymbol{b}\right)\right\}$
    # Select all remaining features
    $\bar{F} \leftarrow \mathcal{F} \setminus F$
    # Evaluate remaining features with all comb. of scores per class
    $f, \boldsymbol{s} \leftarrow \arg\min_{f \in \bar{F}, \boldsymbol{s} \in \mathcal{S}^C} \left\{\text{EVALUATE}\left(F \parallel (f), \begin{bmatrix} \boldsymbol{S} \\ \boldsymbol{s} \end{bmatrix}, \boldsymbol{b}\right)\right\}$
    $F \leftarrow F \parallel (f)$
    $\boldsymbol{S} \leftarrow \begin{bmatrix} \boldsymbol{S} \\ \boldsymbol{s} \end{bmatrix}$
return $h = \langle F, \boldsymbol{S}, \boldsymbol{b} \rangle$

Function EVALUATE($F, \boldsymbol{S}, \boldsymbol{b}$):
    $L \leftarrow 0$
    for $(\boldsymbol{x}, y) \in \mathcal{D}_{train}$ do
        $\boldsymbol{x}_F \leftarrow \boldsymbol{x}[F]$ /* Select features $F$ of instance $x$        */
        # Matrix product of scores and selected features and bias
        $\boldsymbol{s} \leftarrow \boldsymbol{S}\boldsymbol{x}_F + \boldsymbol{b}$
        $\widehat{\boldsymbol{p}} \leftarrow \text{SOFTMAX}(\boldsymbol{s})$ /* Softmax probabilities acc. to Eq. 2    */
        $L \leftarrow L + \ell(\widehat{\boldsymbol{p}}, y)$
    return $L$
```

(11th November 2024). In addition to basic information such as *name*, *nationality*, *age*, *height*, and *current team*, the dataset provides a variety of performance metrics, including the *number of matches played*, *total minutes played*, *goals*, *assists*, *yellow* and *red cards*, *shots*, *pass success percentage*, *aerial duels won percentage*, and each player's primary *playing position*. To our knowledge, no comparable dataset exists. Other publicly available football datasets typically include information from only a single season or provide fewer performance indicators.

Because some players occupy multiple positions (e.g., *central defender* or *defensive midfielder*), various approaches to handling such cases are possible. For our evaluation, we chose four broad categories—*goalkeeper*, *defender*, *midfielder*, and *forward*. Players who could be assigned to more than one of these four

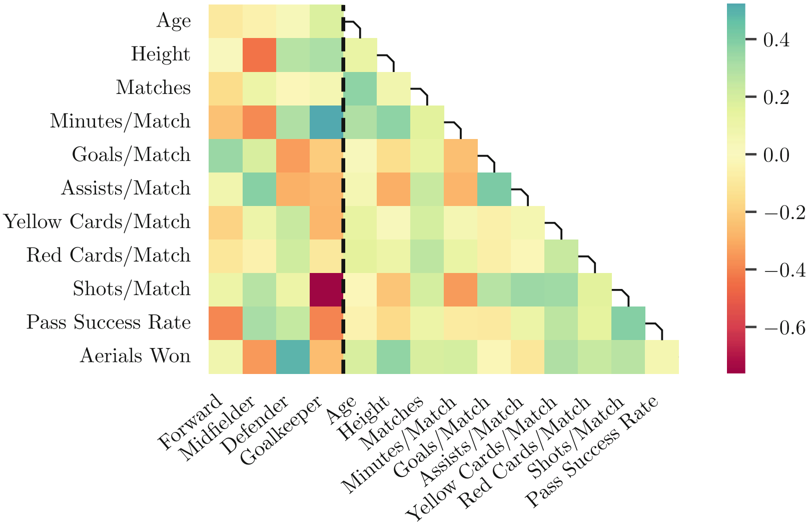

Fig. 1. Correlation of classes and features in football player dataset.

categories were removed, leading to the exclusion of 1,582 players (29%). An additional 256 players (6.6%) were removed due to missing data, resulting in 3,611 players in the cleaned dataset. Figure 1 presents the cross-correlation of the classes and features on the dataset.

5 User Study

We conducted an online user study as a benchmark for the MSL. Our main objectives were to assess how accurately participants perform the classification task and gauge the dataset's comprehensibility. Moreover, to determine the potential for automated decision support, we wanted to compare the performance of human decision-makers with the performance of a data-driven approach, namely a machine learning model.

We recruited 31 participants through the Prolific platform. Each participant was asked to predict the playing positions of football players drawn from our dataset. The study included detailed instructions, which were verified through comprehension checks. Before making their predictions, participants completed four Likert-scale questions assessing their familiarity with football. They were then asked to describe their decision-making process during the classification tasks, after which they received feedback on their responses.

The study included incentives: participants received a fixed payment of € 2 and an additional € 0.40 for each correct prediction. Only UK residents with English as their native language were eligible to participate. Moreover, participants were required to have a Prolific acceptance rate of at least 95% and to have successfully completed more than 10 prior studies on the platform.

The dataset was adapted specifically for this user study: we included only players who had participated in more than 50 games, as those with fewer games were particularly difficult to classify during a pretest. This criterion removed 54% of the 3,611 players, but the remaining total of 1,957 players was still sufficient. Additionally, we included a variable called *Man of the Match*, which cannot be published for legal reasons.

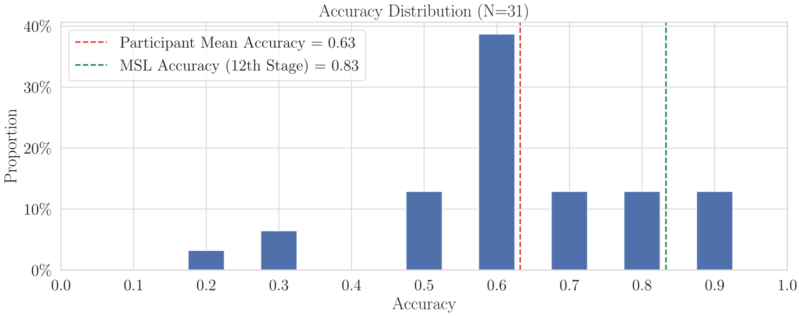

Fig. 2. Distribution of participants' accuracy on the football dataset. The red line shows the mean (63%), and the green line shows the accuracy of the 12th stage of an MSL model (83%).

The remaining dataset contained 1,957 players, which was then split into training and test sets. Only players from the test set were presented to participants to allow a fair comparison. Each participant was randomly assigned 10 players, ensuring the selected positions mirrored the overall class distribution. The participant were not made aware of this stratification.

Figure 2 shows the distribution of participants' classification accuracies compared to the MSL. For a fair comparison, the MSL is trained on the training data and evaluated on the same test samples as the participants. Their average accuracy of 63% fell below that of the MSL model, which achieved up to 83%. These results demonstrate that meaningful classifications are possible from humans (crowd-sourced workers), but also that performance can be improved through data-driven methods based on machine learning. Pearson correlation between accuracy and self-reported football knowledge ($r = 0.246$, $p = 0.165$) suggests that greater familiarity with soccer did not necessarily lead to better performance, although this may be due to self-selection effects in the study or insufficient sample size.

Figure 3 further analyses the classification errors with the help of a confusion matrix. In general, the participants make less precise decisions, however, many of the participants can better judge whether a player plays in the *forward* position. Albeit, this is not due to misclassifications, but is caused by many ties during prediction. Since the MSL implementation is configured to resolve ties at random, this yields to sub-par performance for ambiguous decisions. Yet, this is not an

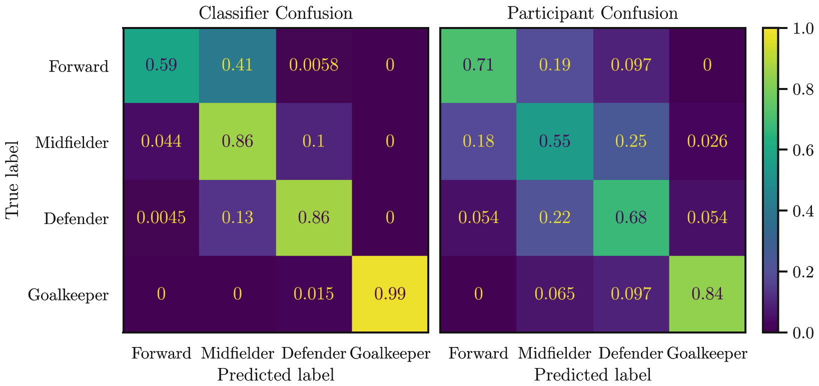

Fig. 3. Confusion matrices for classifier predictions (left) and participant classifications (right). The matrices show the distribution of predicted labels for each true label, with row summing to 1.

issue in a decision-support setting, as the classifier will yield both potential classes, allowing the decision maker to disambiguate.

6 Evaluation

In this section, we provide an evaluation of our newly introduced classifier on various datasets including the football player dataset presented in Sect. 4. The detailed experimental setup and implementation is publicly available[3] as is the implementation of the learning algorithm[4].

6.1 Datasets

To evaluate our classifier, we use well-known binary and multi-class datasets from the UCI repository in addition to our newly introduced dataset.

Table 2 provides an overview of all used datasets. For all datasets we report the entropy with respect to the base of the class count. A uniform class balance will, therefore, yield and entropy of 1. A dataset with 1:2 class-imbalance will yield an entropy of 0.92. The three binary datasets stem from the medical domain. Note, that the `ilp` is therefore relatively unbalanced, with significantly more positive samples (416) than negative samples (167). The multi-class datasets include the previously introduced football player dataset as well as one harder dataset: the customer segmentation dataset, also used in [13].

Since the MSL classifier can only work with binary features, all numerical features have been binarized by calculating a threshold to minimize the expected entropy over the two subsets, similar to splits of a decision stump. Note, that

[3] https://github.com/TRR318/pub-msl.

[4] https://github.com/TRR318/scikit-psl.

Table 2. Overview of the datasets used in the evaluation. Entropy is calculated to the base of the number of classes of the dataset.

Name	Classes	Instances	Features	Entropy	Task	OpenML	Ref.
breast	2	116	9	0.99	Breast cancer	42900	[35]
ilp	2	583	10	0.86	Liver disease	41945	[37]
diabetes	2	768	8	0.93	Diabetes	37	[44]
wine	3	178	13	0.99	Wine origin	187	[1]
player	4	3611	11	0.90	Football player position	46764	ours
segmentation	4	6665	9	1.00	Customer category		[48]

binarization will be problematic if features do not exhibit a monotonic relationship with the target classes. The (close-to) optimal split is selected by employing a hierarchical search heuristic introduced in [15]. The categorical features in the segmentation dataset were one-hot-encoded. The detailed dataset preparation can be found in the experimental repository.

6.2 Setup and Baselines

To evaluate the out-of-sample performance of the classifiers, all experiments have been conducted using Monte Carlo cross-validation (MCCV) with 20 splits where $\frac{2}{3}$ of the data was used for training and the remainder held back for evaluation. The resulting performances have been aggregated and are reported by mean performance and its 95% confidence interval. All experiments have been executed on a single core of a Intel i7-9750H and parallelized over the folds. The total training time of all experiments was more than 40h when parallelized over 12 cores and mostly dominated by the evaluation of MISS, one of our baselines. All MSL instances were learned without regularization and configured with a score set of $\{0, \pm 1, \pm 2, \pm 3\}$. Some metrics, like accuracy, precision, or informedness, do not rely on probability predictions but on discrete classifications. However, the discrete nature of MSLs small score set will often yields ties, especially in earlier stages of the classifier. For example, if only the bias term is evaluated (ref. Table 1), there might be multiple classes with the same maximal total score. In the case of such a set-valued prediction, we select one of the highest-scoring classes uniformly at random.

In each evaluation, we train the PSL and MSL models on all features of the training dataset. Both classifiers create a decision list, i.e., a sequences of decision models for on a nested sequence of features. We call these models "stages". All other baseline models only create single decision models for a specific set of features. Table 3 provides an overview of the training and evaluation method for each stage and the baseline models. In the following paragraphs, we explain in detail how those baseline models can be adapted to those stages.

In Sect. 3.1, we have shown the connection to NB. Using only the likelihoods $P(x_k \mid c)$ of the features available at stage k, NB can naturally be extended to

Table 3. Overview of all models used in the evaluation. k is the number of features used in the kth stage. $model_k$ is the model at the stage k

Model	Training	Training features	Evaluation per stage	Consistent
PSL	global	all	global	✓
MSL	global	all	global	✓
NB	global	all	features of MSL_k	✓
MISS	local	k	features of MISS_k	✗
LR, RF, XGB	local	features of MSL_k	features of MSL_k	✗

the setting of scoring lists. Similarly to the MSL, the NB classifier is trained on all features of the training dataset. At prediction time, we only use the same features that the MSL has selected on that stage.

Grzeszczyk et al. [13] introduced learning algorithm for multinomial scoring systems. Apart from the fact that miss cannot natively produce decision lists, we consider this model closely related to our work. The MISS model at each stage was trained with all features but parametrized to use exactly as many features as the MSL did on this stage. Note, that this will not create a consistent list of models, as selected features and assigned scores can be completely different between each model. We have executed MISS with two different timeouts throughout the experiments. MISS_{90} and MISS_{1800} refers to a training timeout of 90 s, and 30 min vice-versa.

Finally, we have selected three additional models as the baseline that have been trained and evaluated on the same subset of features that the MSL selected on the stage: Logistic Regression (LR), Random Forest (RF), and XGBoost (XGB). Overall, we can see that MISS has the largest amount of freedom of all models with respect to feature selection, as only the *number* of features is dictated by the MSLs stage.

To evaluate our model, we rely on two metrics: accuracy (classification rate) and *expected calibration error* (ECE). While the classification rate (fraction of correct predictions) is a standard measure of the correctness of the learner's final (deterministic) decisions, calibration aims to assess the model's probability estimates. Here, we adopt a standard notion of classifier calibration called confidence-calibration: A probabilistic classifier producing predictions $\hat{\boldsymbol{p}}(x) = (\hat{p}_1(x), \ldots, \hat{p}_C(x))$ is (confidence-)calibrated, if

$$P\big(y = \arg\max_i \hat{p}_i(x) \mid \max_i \hat{p}_i(x) = \alpha\big) = \alpha$$

for all $\alpha \in [0, 1]$. In words, if the model reports α-confidence in its decision, i.e., the probability predicted for the (presumably) most probable class is α, then this decision is indeed correct with probability α. For example, among all decisions for which the model reports a confidence of 80%, indeed 80% of the cases are correct. While this notion of calibration can be criticized (e.g., because it does not condition on the instance x itself), it does appear useful from the point of view of explainability and informed decision-making. In particular, it provides reasonable support for the stopping condition: A calibrated confidence at stage k of the decision process provides the decision maker with a clear idea of how safe or risky it might be to stop and make a final decision at that stage.

Practically, as ground-truth probabilities cannot be observed in the data, the calibration of a model is measured in terms of the *expected calibration error* (ECE), which is based on the partitioning of the unit interval into a set of bins (intervals) $B_1, \ldots, B_m$. Formally, ECE is then defined as follows [14]:

$$ECE = \sum_{j=1}^{m} \frac{|B_j|}{N} \left| \mathrm{acc}(B_j) - \mathrm{conf}(B_j) \right|, \tag{7}$$

where N is the number of data points, $|B_j|$ is the number of points falling in bin B_j, $\mathrm{acc}(B_j)$ is the fraction of points in bin B_j for which the model predicted correctly (i.e., the accuracy in that bin), and $\mathrm{conf}(B_j)$ the average confidence reported by the model for points in B_j. We rely on the implementation of Kumar et al. for an unbiased estimate of the ECE [25].

6.3 Classification Accuracy

Binary classification problems can be interpreted in two ways: Either as the presence of absence of the positive label or as a genuine two class problem. This allows comparing the PSL model, which can only make predictions towards the positive class and the MSL which collects evidence towards all alternative classes. Recall, that Naïve Bayes and MSL operate on the same features at prediction time.

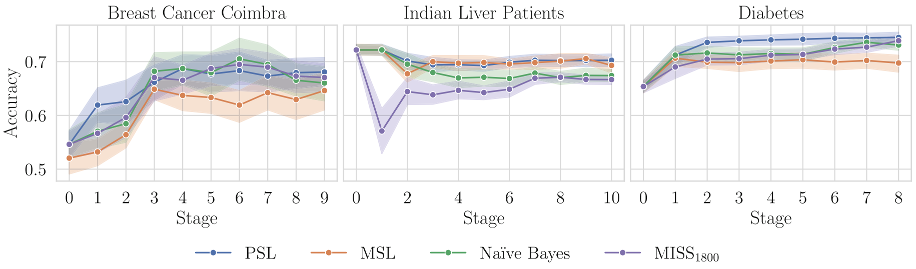

Fig. 4. Classifier accuracy across different stages for all binary datasets. The shaded regions represent confidence intervals of the mean.

Overall, the predictive performance of the compared classifiers yield mixed results on the binary datasets as seen in Fig. 4. While MISS performes good on the `breast` and `diabetes` dataset it exhibits poor accuracy on the unbalanced `ilp` dataset. MSL performes generally sligtly worse than the PSL which is particularly tuned for binary classification problems. On the particularly small `breast` dataset 20 MCCV splits appear to have insufficient statistical power to clearly distinguish classifiers performance.

In the multi class setting, we cannot compare to the PSL. Hence, we add multinomial logistic regression and two less interpretable decision models (RF, XGB).

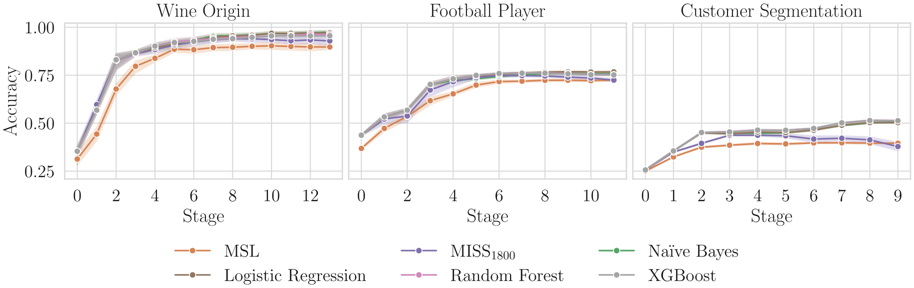

Fig. 5. Classifier accuracy across different stages for all multi-class datasets. The shaded regions represent confidence intervals of the mean.

Figure 5 shows the accuracy of the classifiers across the datasets sorted by sample size. While the MSL performance is worse in general, it must be noted that the MSL and NB construct one list of models that are consistent to each other: Feature subsets form a nested sequence, and the score assigned to a feature remains constant across stages. This is arguably important from an interpretability point of view [18]. The remaining classifiers can create different models for each stage, thereby compromising interpretability. While LR, RF, and XGB at least use the same features that the MSL uses, MISS will only use the same number of features. The parametrizations across those models are not consistent. Still, the MSL performs similarly well to the other classifiers. The performance of MISS declines on the largest dataset (`segmentation`) as more and more features become available. This can only be explained by the 30 min timeout, meaning the models still have a large optimality gap.

6.4 Probability Calibration of the Classifier

In this section we analyze the classifiers probability calibration against the same baselines used in the previous chapter.

Figure 6 shows that all classifiers provide fairly calibrated probability estimates, except for the Naïve Bayes classifier, which is known to be a good classifier but a sub-par probability estimator [51].

On the multi-class datasets (ref. Figure 7) MISS performs slightly worse when only little features are available. In absolute terms, most models exhibit low calibration errors across all stages. The strikingly bad performance of MISS on the `segmentation` can again be explained by the premature terminated training

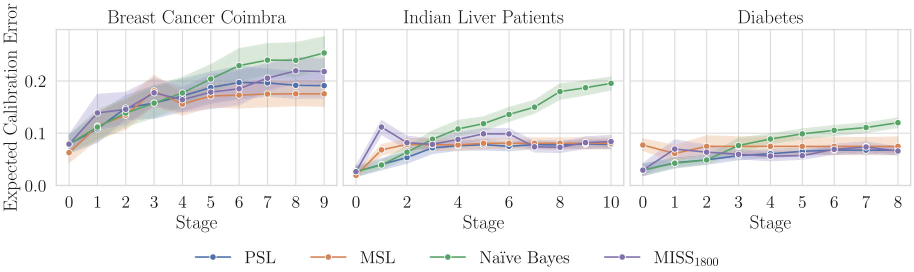

Fig. 6. Expected Calibration Error across different stages for three binary datasets. The plots compare the calibration performance of four models: PSL, MSL, Naïve Bayes, and MISS_{1800}. The shaded regions represent confidence intervals.

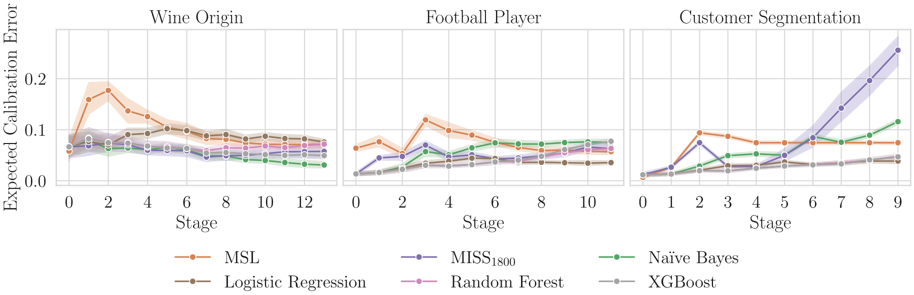

Fig. 7. Expected Calibration Error across different stages for all multi-class datasets. The plots compare the calibration performance of four models: PSL, MSL, Naïve Bayes, and MISS_{1800}. The shaded regions represent confidence intervals.

due to timeouts. This can also be seen on the `player` dataset, which is also stopped due to timeouts for stages 8 and following. Fortunately, on this dataset, only a relatively small optimality gap is retained after exhausting the 30 min training budget.

6.5 Runtime Analysis

In the previous sections, we have seen mostly competitive performance of the MISS classifier. However, particularly on the `segmentation` dataset, the performance was often suboptimal, even though the MISS baseline, was the one with the most flexibility as it was only constrained regarding the number of features used.

The MISS classifier is learned by solving a mixed integer program with the help of the `cplex` solver. This can yield provably optimal solutions with respect to the loss function and the training data. However, this training method is also very costly in terms of training time. This is exacerbated in the scenario of decision lists, because many decision models have to be learned independently.

Figure 8 shows the performance of the MSL classifier and two parametrizations of the MISS classifier: one with 90 s and one with 30 min. With only 90 s per stage, the performance of MISS already stagnates after 3 features and hardly exceeds the performance of the MSL, even though the MSL will additionally enforce coherence of the whole decision list. Even with 30 min, stages 6 and following time out, however, with significantly higher performance, which can even be seen in the slight performance decrease after stage 8. For the `segmentation` dataset, not even 30 min per stage are sufficient and large optimality gaps remain.

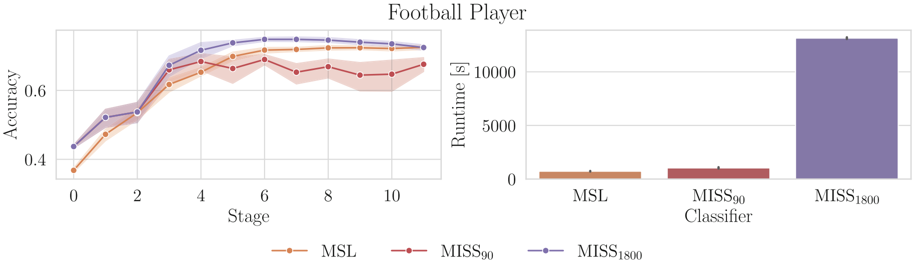

Fig. 8. Accuracy and runtime analysis for the Football Player dataset. The left plot shows accuracy across different stages for MSL and the two 90s and 30min timeout configurations of MISS. The right plot shows the total training time for all stages.

7 Conclusion

In the search for explainable AI, two approaches are currently pursued: post-hoc explaination of complex models and inference of inherently (ante-hoc) explainable models. Although the former approach has been fostered by advances in generative AI, very recent research has shown that explaining complex or even black-box models in easy terms can result in undesirable outcomes, including overreliance on AI if predictions are accompanied by explanations that appear to be comprehensive [17].

In this paper, we therefore pursue a different path to improve the performance of AI in (human) decision-making tasks. We propose a method for learning scoring systems that are commonly used and widely accepted for decision support in real-world applications. In contrast to existing approaches, our method is able to handle problems with more than two choice alternatives. Moreover, by constructing a coherent decision list instead of a single model, MSL supports a stagewise decision-making process, where a decision can be made as soon as enough evidence has been accumulated.

Not less importantly, MSL is inherently explainable due to its restriction to integer scores, its simple additive structure, and the coherence of the models that form a decision list (feature subsets are nested and scores remain unchanged). Admittedly, compared to black-box models or models being less restricted (e.g., additive models with real-valued instead of integer scores, such as logistic regression), MSL may exhibit slightly weaker predictive performance. However, the loss in performance is in general not very high and appears to be acceptable in view of the gain in explainability. Future work should empirically investigate MSL with regard to interpretability and explainability, particularly examining how the stages are used in different decision-making scenarios and how this affects decision quality.

We evaluated human performance on a specific dataset that is especially suited for analyzing AI-human collaborative decision-making, and show that humans perform significantly worse than our approach. Despite this, we believe that a hybrid approach—where a human expert supports a machine learning algorithm in constructing an MSL, or more broadly, engages in an AI-human co-construction of decision models—is a promising direction that we plan to explore in future work, especially given that prior research has shown human-in-the-loop approaches can enhance model performance [34,50], improve decision-making [19], and increase model acceptance [33], even though such methods may be limited when experts are biased [16]. Broadly speaking, the idea is to let the human support or correct decisions about the order of features, the scores assigned to features, etc. This might be beneficial for the learning algorithm, in particular to counteract the heuristic nature of its greedy search strategy. At the same time, a hybrid approach could be appealing for the human expert and increase the acceptance and adoption of automatic decision support — a model that a human expert co-constructed herself will likely increase acceptance, trust, and understanding compared to a model that was constructed in a purely data-driven way and impose on the expert from outside.

Acknowledgements. We gratefully acknowledge funding by the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG): TRR 318/1 2021 – 438445824.

References

1. Aeberhard, S., Forina, M.: Wine. UCI Machine Learning Repository (1992). https://doi.org/10.24432/C5PC7J
2. Ahmed, M.U., Barua, S., Begum, S., Islam, M.R., Weber, R.O.: When a CBR in hand is better than twins in the bush (2023)
3. Barredo Arrieta, A., et al.: Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible AI. Heuristic Optimization **58**, 82–115 (2020). https://doi.org/10.1016/j.inffus.2019.12.012
4. Bell, A., Solano-Kamaiko, I., Nov, O., Stoyanovich, J.: It's just not that simple: an empirical study of the accuracy-explainability trade-off in machine learning for public policy. In: ACM Conference on Fairness, Accountability, and Transparency. pp. 248–266 (2022). https://doi.org/10.1145/3531146.3533090
5. Belle, V.M.C.A.V., et al.: A mathematical model for interpretable clinical decision support with applications in gynecology. PLOS ONE **7**(3), e34312 (2012). https://doi.org/10.1371/journal.pone.0034312

6. Bianchi, F., Piroddi, L., Bemporad, A., Halasz, G., Villani, M., Piga, D.: Active preference-based optimization for human-in-the-loop feature selection. Eur. J. Control. **66**, 100647 (2022). https://doi.org/10.1016/j.ejcon.2022.100647
7. Billiet, L., Huffel, S.V., Belle, V.V.: Interval coded scoring index with interaction effects - A sensitivity study. In: International Conference on Pattern Recognition Applications and Methods (ICPRAM), vol. 2, pp. 33–40 (2016). https://doi.org/10.5220/0005646500330040
8. Billiet, L., Van Huffel, S., Van Belle, V.: Interval coded scoring extensions for larger problems. In: IEEE Symposium on Computers and Communications (ISCC), pp. 198–203 (2017). https://doi.org/10.1109/ISCC.2017.8024529
9. Das, D., Chernova, S.: Leveraging rationales to improve human task performance. In: International Conference on Intelligent User Interfaces (IUI), pp. 510–518 (2020). https://doi.org/10.1145/3377325.3377512
10. Feng, S., Boyd-Graber, J.: What can AI do for me? evaluating machine learning interpretations in cooperative play. In: International Conference on Intelligent User Interfaces (IUI), pp. 229–239 (2019). https://doi.org/10.1145/3301275.3302265
11. Fisher, R.A.: Iris. UCI Machine Learning Repository (1936). https://doi.org/10.24432/C56C76
12. Gage, B.F., Waterman, A.D., Shannon, W., Boechler, M., Rich, M.W., Radford, M.J.: Validation of clinical classification schemes for predicting stroke results from the national registry of atrial fibrillation. JAMA **285**(22), 2864–2870 (2001). https://doi.org/10.1001/jama.285.22.2864
13. Grzeszczyk, M.K., Trzciński, T., Sitek, A.: MISS: multiclass interpretable scoring systems. In: SIAM International Conference on Data Mining (SDM), pp. 55–63 (2024). https://doi.org/10.1137/1.9781611978032.7
14. Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q.: On calibration of modern neural networks. In: International Conference on Machine Learning (ICML), pp. 1321–1330 (2017)
15. Hanselle, J., Heid, S., Fürnkranz, J., Hüllermeier, E.: Probabilistic scoring lists for interpretable machine learning. Mach. Learn. **114**(3), 55 (2025). https://doi.org/10.1007/s10994-024-06705-w
16. Hanselle, J., Kornowicz, J., Heid, S., Thommes, K., Hüllermeier, E.: Comparing humans and algorithms in feature ranking: a case-study in the medical domain. In: Leyer, M., Wichmann, J. (eds.) Lernen, Wissen, Daten, Analysen (LWDA), vol. 3630, pp. 430–441 (2023)
17. He, G., Aishwarya, N., Gadiraju, U.: Is conversational XAI all you need? Human-AI decision-making with a conversational XAI assistant (2025)
18. Heid, S., Hanselle, J., Fürnkranz, J., Hüllermeier, E.: Learning decision catalogues for situated decision making: The case of scoring systems. Inter. J. Approximate Reas. (IJAR) **171**, 109190 (2024). https://doi.org/10.1016/J.IJAR.2024.109190
19. Heid, S., Kornowicz, J., Hanselle, J., Hüllermeier, E., Thommes, K.: Human-AI co-construction of interpretable predictive models: the case of scoring systems. In: Workshop on Computational Intelligence (CI), pp. 233–252 (2024)
20. Janosi, A., Steinbrunn, W., Pfisterer, M., Detrano, R.: Heart disease. UCI Machine Learning Repository (1989). https://doi.org/10.24432/C52P4X
21. Jeong, B.H., et al.: Performances of prognostic scoring systems in patients with healthcare-associated pneumonia. Clin. Infect. Dis. **56**(5), 625–632 (2013). https://doi.org/10.1093/cid/cis970
22. Kline, J.A., Mitchell, A.M., Kabrhel, C., Richman, P., Courtney, D.M.: Clinical criteria to prevent unnecessary diagnostic testing in emergency department patients with suspected pulmonary embolism. J. Thromb. Haemost. **2**(8), 1247–1255 (2004)

23. Knaus, W.A., et al.: The APACHE III prognostic system: risk prediction of hospital mortality for critically III hospitalized adults. Chest **100**(6), 1619–1636 (1991)
24. Kohavi, R.: Census income. UCI Machine Learning Repository (1996). https://doi.org/10.24432/C5GP7S
25. Kumar, A., Liang, P., Ma, T.: Verified uncertainty calibration. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 3787–3798 (2019)
26. Lai, V., Chen, C., Smith-Renner, A., Liao, Q.V., Tan, C.: Towards a science of human-AI decision making: an overview of design space in empirical human-subject studies. In: Conference on Fairness, Accountability, and Transparency, pp. 1369–1385 (2023). https://doi.org/10.1145/3593013.3594087
27. Lip, G.Y., Nieuwlaat, R., Pisters, R., Lane, D.A., Crijns, H.J.: Refining clinical risk stratification for predicting stroke and thromboembolism in atrial fibrillation using a novel risk factor-based approach: The euro heart survey on atrial fibrillation. Chest **137**(2), 263–272 (2010)
28. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 4765–4774 (2017)
29. Mahmud, H., Islam, A.K.M.N., Ahmed, S.I., Smolander, K.: What influences algorithmic decision-making? a systematic literature review on algorithm aversion. Technol. Forecast. Soc. Chang. **175**, 121390 (2022). https://doi.org/10.1016/j.techfore.2021.121390
30. Moreno, R.P., et al.: On behalf of the SAPS 3 Investigators: SAPS 3From evaluation of the patient to evaluation of the intensive care unit. Part 2: Development of a prognostic model for hospital mortality at icu admission. Intensive Care Med. **31**(10), 1345–1355 (2005). https://doi.org/10.1007/s00134-005-2763-5
31. Mortier, T., Hüllermeier, E., Dembczynski, K., Waegeman, W.: Set-valued prediction in hierarchical classification with constrained representation complexity. In: Cussens, J., Zhang, K. (eds.) Uncertainty in Artificial Intelligence (UAI), vol. 180, pp. 1392–1401 (2022)
32. Mounzer, R., et al.: Comparison of existing clinical scoring systems to predict persistent organ failure in patients with acute pancreatitis. Gastroenterology **142**, 1476–1482 (2012). https://doi.org/10.1053/j.gastro.2012.03.005
33. Muijlwijk, H., Willemsen, M.C., Smyth, B., IJsselsteijn, W.A.: Benefits of human-AI interaction for expert users interacting with prediction models: a study on marathon running. In: International Conference on Intelligent User Interfaces (IUI), pp. 245–258 (2024). https://doi.org/10.1145/3640543.3645205
34. Nahar, J., Imam, T., Tickle, K.S., Chen, Y.P.P.: Computational intelligence for heart disease diagnosis: a medical knowledge driven approach. Expert Syst. Appl. **40**(1), 96–104 (2013). https://doi.org/10.1016/j.eswa.2012.07.032
35. Patrcio, M., Pereira, J., Crisstomo, J., Matafome, P., Seia, R., Caramelo, F.: Breast cancer coimbra. UCI Machine Learning Repository (2018). https://doi.org/10.24432/C52P59
36. Poursabzi-Sangdeh, F., Goldstein, D.G., Hofman, J.M., Wortman Vaughan, J.W., Wallach, H.: Manipulating and measuring model interpretability. In: Conference on Human Factors in Computing Systems (CHI), pp. 1–52 (2021). https://doi.org/10.1145/3411764.3445315
37. Ramana, B., Venkateswarlu, N.: Indian liver patient. UCI Machine Learning Repository (2022). https://doi.org/10.24432/C5D02C
38. Rapsang, A.G., Shyam, D.C.: Scoring systems in the intensive care unit: a compendium. Indian J. Critical Care Med. **18**(4), 220–228 (2014). https://doi.org/10.4103/0972-5229.130573

39. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?": explaining the predictions of any classifier. In: International Conference on Knowledge Discovery and Data Mining (SIGKDD), pp. 1135–1144 (2016). https://doi.org/10.1145/2939672.2939778
40. Rivest, R.L.: Learning decision lists. Mach. Learn. **2**(3), 229–246 (1987). https://doi.org/10.1007/BF00058680
41. Rouzot, J., Ferry, J., Huguet, M.J.: Learning optimal fair scoring systems for multi-class classification. In: International Conference on Tools with Artificial Intelligence (ICTAI), pp. 197–204 (2022). https://doi.org/10.1109/ICTAI56018.2022.00036
42. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. **1**(5), 206–215 (2019). https://doi.org/10.1038/S42256-019-0048-X
43. Sanyal, S., Biswas, S.K., Das, D., Chakraborty, M., Purkayastha, B.: Boston house price prediction using regression models. In: International Conference on Intelligent Technologies, pp. 1–6 (2022)
44. Smith, J.W., Everhart, J.E., Dickson, W.C., Knowler, W.C., Johannes, R.S.: Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. In: Computational Applications Medical Care, pp. 261–265 (1988)
45. Taheri, E., Wang, C., Zahmat Doost, E.: Emergency decision-making under an uncertain time limit. Inter. J. Disaster Risk Reduct. **95**, 103832 (2023). https://doi.org/10.1016/j.ijdrr.2023.103832
46. Ustun, B., Rudin, C.: Supersparse linear integer models for optimized medical scoring systems. Mach. Learn. **102**(3), 349–391 (2016). https://doi.org/10.1007/S10994-015-5528-6
47. Ustun, B., Rudin, C.: Learning optimized risk scores. J. Mach. Learn. Res. **20**, 150:1–150:75 (2019)
48. Vidhya, A.: Customer segmentation. Kaggle (2020)
49. Wang, C., Han, B., Patel, B., Rudin, C.: In pursuit of interpretable, fair and accurate machine learning for criminal recidivism prediction. J. Quant. Criminol. **39**(2), 519–581 (2023)
50. Yang, Y., Kandogan, E., Li, Y., Sen, P., Lasecki, W.S.: A study on interaction in human-in-the-loop machine learning for text analytics. In: IUI Workshops (2019)
51. Zadrozny, B., Elkan, C.P.: Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In: International Conference on Machine Learning (ICML) (2001)
52. Zeng, Z., Shi, Z., Li, X.: Comparing different scoring systems for predicting mortality risk in preterm infants: a systematic review and network meta-analysis. Front. Pediatr. **11**, 1287774 (2023). https://doi.org/10.3389/fped.2023.1287774