Formal Explanations of Black-Box Ranking Functions

Francesco Chiariello[1][0000−0001−7855−7480] and Joao Marques-Silva[2][0000−0002−6632−3086]

[1] RWTH Aachen University, Aachen, Germany
francesco.chiariello@ml.rwth-aachen.de
[2] ICREA, University of Lleida, Lleida, Spain
jpms@icrea.cat

Abstract. Ranking functions play a crucial role in supporting decision-making processes across various critical domains. Given their widespread use, coupled with the fact that these functions are often directly learned from data, it is becoming more and more important to provide explanations that make the underlying models more transparent. In this paper, we propose the first formal approach to explain ranking functions. Our approach is model-agnostic, requiring only black-box access to the ranking function. We study the formal properties of this new approach, including an analysis of the complexity of computing an explanation. To demonstrate its feasibility, we implement our approach and conduct an experimental evaluation using as a case study a neural network model for predicting breast cancer recurrence.

Keywords: Ranking · Preferences · Utility Functions · Machine Learning · Explainable Artificial Intelligence

1 Introduction

Ranking is a fundamental task in many decision-making processes, such as job recruitment, college admissions, and loan approval [57]. A critical domain is healthcare scheduling, where risk scores help prioritise hospital operations according to the urgency and severity of patient conditions. Given the significant impact that rankings have on our lives, it is therefore essential to provide clear explanations that ensure transparency, understanding, and trust. This necessity is even more pressing if we consider that such rankings are increasingly often determined by machine learning algorithms. Despite this, the problem of explaining ranking functions has often been overlooked in eXplainable AI (XAI) research, which has mainly focused on classification and regression tasks [20]. Driven by fairness concerns in machine learning, some studies have started to tackle this gap [55,18,45]. Nevertheless, these works only rely on heuristic methods, such as approximate Shapley values [50,36], which can sometimes produce misleading results for human decision-makers [23].

Formal XAI (FXAI) [38,17] offers a promising alternative to heuristic methods by grounding explanations in logical definitions that enhance their interpretability. However, this research line has so far only considered classification and regression [39,6]. A naive approach to applying FXAI to ranking functions involves reducing the ranking task to binary classification. Specifically, one can construct a binary classifier that outputs 1 if and only if the ranking holds. Explaining a ranking then amounts to explaining why the classifier outputs 1. This construction was outlined by Labreuche [33], who, however, regarded it as impractical. This is because the classifier takes as input the concatenation of the vectors in the ranking, which substantially increases the dimensionality of the feature space. This higher dimensionality, in turn, negatively affects both the time needed to compute the explanations and their overall quality. Furthermore, this approach overlooks the fact that the new feature space is essentially composed of duplicates of the same features, once for each vector in the ranking.

We address the issues of previous work by introducing the first formal definitions for explanations of ranking functions. These definitions allow a feature of a vector to be part of an explanation if and only if the corresponding features in all the other vectors are also included. This results in a more natural definition, as explanations are directly defined within the original feature space. Furthermore, the reduced number of features involved makes the computation of these explanations practically feasible. We then investigate the formal properties of the proposed explanations and establish several results that support our framework, including the monotonicity of (weak) explanations. This key property allows us to cast the problem of computing an explanation of a ranking as an instance of the Minimal Set over a Monotone Predicate problem [41], which we solve using a deletion-based algorithm, akin to computing Minimal Unsatisfiable Subsets in Boolean Satisfiability [37]. Yet, our approach is *model-agnostic*, enabling its application to black-box models, whether large-scale or proprietary. We discuss the complexity of the algorithm, highlighting its greater efficiency compared to the naive approach. Finally, we provide an implementation of the approach and conduct an experimental evaluation based on the following case study, which serves as a proof of concept.

Case Study. We assume a neural network model f that estimates the probability $f(\mathbf{x}) \in \mathbb{R}$ of a patient $\mathbf{x}$ experiencing breast cancer recurrence within five years after surgery. These probabilities induce a priority ranking over patients, which can be used, for instance, to schedule medical appointments. Each patient profile is represented as a vector $\mathbf{x}$ of categorical features. While more features are considered in the experiments, we focus here on three illustrative ones: *age* (the patient's age group), *tumor-size* (the size of the tumour), and *deg-malig* (the degree of malignancy), with domain sizes of six, eleven, and three, respectively. For example, a patient profile $\mathbf{v} = (3, 8, 2)$ corresponds to an individual aged between 50–59 years (category 3), with a tumour size of 45–49 mm (category 8), and a malignancy degree of 2. Our goal is then to explain the ranking produced by the neural network over a given group of patients by isolating a set of features that alone account for the ranking.

The remainder of the paper is structured as follows. Section 2 provides the necessary background on formal explainability for classifiers and order theory, including an introduction to ranking functions. Section 3 defines (weak) abductive explanations for rankings and analyses their properties. Section 4 details an algorithm for computing these explanations and examines its computational complexity. Section 5 describes our case study and the experiments conducted. Section 6 reviews relevant literature. Finally, Section 7 concludes the paper with possible directions for future research.

2 Background

This section reviews key definitions from Formal Explainability [38] and Order Theory [48]. For a comprehensive treatment, please refer to the cited sources.

2.1 Formal Explanations of Classifiers

Let $\mathcal{F} = \{1, \ldots, m\}$ denote a *feature set*, with each *feature* $i \in \mathcal{F}$ having an associated domain $\mathbb{D}_i$. These domains collectively define the *feature space* $\mathbb{F} = \mathbb{D}_1 \times \cdots \times \mathbb{D}_m$. The points $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{F}$ in the feature space are also referred to as *feature vectors*, or simply vectors.

We say that two vectors $\mathbf{x}, \mathbf{v} \in \mathbb{F}$, *agree* on features $\mathcal{S} \subseteq \mathcal{F}$, denoted $\mathbf{x} \sim_{\mathcal{S}} \mathbf{v}$, if $\forall i \in \mathcal{S}, x_i = v_i$. That is, when projected into the subspace defined by $\mathcal{S}$, the two vectors are indistinguishable. Note that the agreement relation $\sim_{\mathcal{S}}$ is trivially an equivalence relation on $\mathbb{F}$. We also define

$$[\mathbf{v}]_{\mathcal{S}} := [\mathbf{v}]_{\sim_{\mathcal{S}}} = \{\mathbf{x} \in \mathbb{F} \mid \mathbf{x} \sim_{\mathcal{S}} \mathbf{v}\}$$

as the equivalence class of $\mathbf{v}$ under $\sim_{\mathcal{S}}$, consisting of all points $\mathbf{x}$ that agree with $\mathbf{v}$ on $\mathcal{S}$.

Let $\mathcal{K} = \{c_1, \ldots, c_K\}$ be a set of classes, and let $\kappa : \mathbb{F} \to \mathcal{K}$ be a classifier.

Definition 1 (Weak Abductive Explanation (WeakAXp)).
A set $\mathcal{S} \subseteq \mathcal{F}$ of features is a weak abductive explanation *(or WeakAXp for short) for the explanation problem $(\kappa; \mathbf{v})$, if it holds that:*

$$\forall \mathbf{x} \in [\mathbf{v}]_{\mathcal{S}}, \kappa(\mathbf{x}) = \kappa(\mathbf{v}) \tag{1}$$

In other words, a WeakAXp is a set of features such that any vector $\mathbf{x}$ that agrees with $\mathbf{v}$ on those features is mapped to the same class of $\mathbf{v}$. We also write $\mathsf{WeakAXp}(\mathcal{S})$ to denote that $\mathcal{S}$ is a WeakAXp, omitting the dependence on the explanation problem to simplify the notation.

Definition 2 (Abductive Explanation (AXp)). *A set $\mathcal{S} \subseteq \mathcal{F}$ is an* abductive explanation *(or AXp) for the explanation problem $(\kappa; \mathbf{v})$ if $\mathcal{S}$ is a subset-minimal WeakAXp, that is,*

$$\mathsf{WeakAXp}(\mathcal{S}) \ \wedge \ \forall \mathcal{S}' \subsetneq \mathcal{S}, \neg\mathsf{WeakAXp}(\mathcal{S}') \tag{2}$$

AXps are also known as PI-explanations [51], as they correspond to the prime implicants of the classification $\kappa(\mathbf{v})$, or more precisely, the prime implicants of formula (1) when interpreted as a Boolean function over the features $\mathcal{F}$.

2.2 Order Theory

In what follows, let S be a finite set. A *preorder* $\preceq$ on S is a binary relation on S that is both reflexive and transitive. A preorder is said to be *total* if it is also strongly connected; that is, if any two elements are comparable. A total preorder is also referred to as a *ranking*. Rankings are commonly known as *preferences* in microeconomic theory [9], where they serve as models for consumer behaviour. If a ranking is also antisymmetric—thus forming a proper order—it is called a *linear order*. A preorder $\preceq_i$ on S induces an equivalence relation $\sim_i$ on S, defined by $a \sim_i b \iff a \preceq_i b \wedge b \preceq_i a$. Given two rankings $\preceq_1, \preceq_2$, the ranking $\preceq_1$ is *finer* than $\preceq_2$ if $\preceq_1 \subseteq \preceq_2$ or, equivalently, $\sim_1 \subseteq \sim_2$. Rankings are more general than linear orders in that they allow for ties, with elements being tied if they belong to the same equivalence class.

A *ranking function*, or *ranker*, (also known as *utility function* in microeconomic theory) on S is a function $f : S \to \mathbb{R}$. A ranking function f on S induces a ranking $\preceq_f$ on S, by defining $a \preceq_f b \iff f(a) \leq f(b)$. Conversely, given a ranking $\preceq$ on S, there exists a ranking function $f : S \to \mathbb{R}$ such that $\preceq$ coincides with the ranking induced by f, i.e., $\preceq = \preceq_f$. Consequently, the terms ranking and ranking function can be used interchangeably. We also blur the distinction between a ranking $\preceq$ and the rankings $\preceq_{\restriction_{S'}}$ obtained by restricting $\preceq$ to $S' \subseteq S$. Without loss of generality, one can assume the range of f to be an initial segment $\{1, \ldots, K\}$ of $\mathbb{N}$.

It is worth noting that if $(\mathcal{K}, \preceq_{\mathcal{K}})$ is linearly ordered, a classifier $\kappa : \mathbb{F} \to \mathcal{K}$ acts as a ranker on $\mathbb{F}$, inducing the ranking $\preceq_\kappa$ defined by

$$\mathbf{x} \preceq_\kappa \mathbf{x}' \iff \kappa(\mathbf{x}) \preceq_{\mathcal{K}} \kappa(\mathbf{x}') \tag{3}$$

3 Formal Explanations of Rankers

Let $\mathbb{F} = \mathbb{D}_1 \times \ldots \times \mathbb{D}_m$ be a feature space, and let $f : \mathbb{F} \to \mathbb{R}$ be a ranking function on $\mathbb{F}$. Given two feature vectors $\mathbf{v}, \mathbf{v}' \in \mathbb{F}$, such that $\mathbf{v} \preceq_f \mathbf{v}'$, i.e., $f(\mathbf{v}) \leq f(\mathbf{v}')$, we address the question: *why is* $\mathbf{v}'$ *ranked at least as highly as* $\mathbf{v}$? We indicate with $(f; \mathbf{v}, \mathbf{v}')$ this explanation problem.

Definition 3 (Weak Abductive Explanation (WeakAXp)). *Let $\mathcal{S} \subseteq \mathcal{F}$ be a set of features. We say that $\mathcal{S}$ is a* Weak Abductive Explanation *(or WeakAXp for short) for the explanation problem $(f; \mathbf{v}, \mathbf{v}')$ if*

$$\forall\, (\mathbf{x}, \mathbf{x}') \in [\mathbf{v}]_{\mathcal{S}} \times [\mathbf{v}']_{\mathcal{S}},\ \mathbf{x} \preceq_f \mathbf{x}' \tag{4}$$

In other words, a WeakAXp is a set $\mathcal{S}$ of features such that, for each pair $\mathbf{x}, \mathbf{x}' \in \mathbb{F}$ of vectors such that $\mathbf{x} \sim_{\mathcal{S}} \mathbf{v}$ and $\mathbf{x}' \sim_{\mathcal{S}} \mathbf{v}'$, the ranking $\mathbf{x} \preceq_f \mathbf{x}'$ is preserved. We also write $\mathsf{WeakAXp}(\mathcal{S})$ to denote that $\mathcal{S}$ is a WeakAXp.

The following theorem establishes the monotonicity of WeakAXps for rankers and is analogous to the corresponding result for classifiers [38].

Theorem 1 (Monotonicity of WeakAXps). *If $\mathcal{S}$ is a WeakAXp and $\mathcal{S} \subseteq \mathcal{S}''$, then $\mathcal{S}''$ is also a WeakAXp.*

In practice, one can consider different rankings that vary in how finely they distinguish between alternatives. Such differences often arise from introducing tie-breaking criteria to resolve some or all ties, resulting in finer rankings derived from coarser ones. The following theorem establishes a relationship between the WeakAXps of rankers with different levels of granularity.

Theorem 2. *Let $\preceq_1$ and $\preceq_2$ be rankings on $\mathbb{F}$ such that $\preceq_1$ is finer than $\preceq_2$, i.e., $\preceq_1 \subseteq \preceq_2$. Then every WeakAXp of $\preceq_1$ is also a WeakAXp of $\preceq_2$.*

We now proceed to define abductive explanations.

Definition 4 (Abductive Explanation (AXp)). *A set $\mathcal{S} \subseteq \mathcal{F}$ is an* abductive explanation *(AXp) for the explanation problem $(f; \mathbf{v}, \mathbf{v}')$ if $\mathcal{S}$ is a subset-minimal WeakAXp, that is,*

$$\mathsf{WeakAXp}(\mathcal{S}) \ \land \ \forall \mathcal{S}' \subsetneq \mathcal{S}, \neg\mathsf{WeakAXp}(\mathcal{S}') \tag{5}$$

Observe that formula (5) is formally analogous to formula (2) for classifiers, i.e., they are syntactically equivalent. However, it is important to note that we have redefined the semantics of predicate $\mathsf{WeakAXp}$ in the context of rankings. Note also that, by Theorem 1, determining whether $\mathcal{S}$ is an AXp requires checking only the $|\mathcal{S}|$ maximal proper subsets $\mathcal{S}' = \mathcal{S} \setminus \{i\}$ with $i \in \mathcal{S}$, rather than all $2^{|\mathcal{S}|} - 1$ proper subsets.

The following theorem relates WeakAXps of classifiers and rankers.

Theorem 3. *Let $k : \mathbb{F} \to \mathcal{K}$ be a classifier, with $\mathcal{K}$ linearly ordered, and let $\mathbf{v}, \mathbf{v}' \in \mathbb{F}$ be two point such that $\mathbf{v} \preceq_\kappa \mathbf{v}'$. Then, if $\mathcal{S}$ is a WeakAXp for both the explanations problems $(\kappa; \mathbf{v})$ and $(\kappa; \mathbf{v}')$, then $\mathcal{S}$ is a WeakAXp also for $(\kappa; \mathbf{v}, \mathbf{v}')$.*

This follows directly from the fact that the classes of $\mathbf{x} \in [\mathbf{v}]_\mathcal{S}$ and $\mathbf{x}' \in [\mathbf{v}']_\mathcal{S}$ are fixed. The converse does not necessarily hold. In fact, formula (4) allows their classes to vary freely, as long as every $\mathbf{x}'$ is ranked at least as highly as every $\mathbf{x}$.

We began our analysis by addressing the question of why $\mathbf{v} \preceq_f \mathbf{v}'$. In fact, formula (4) can be easily adapted to the case $\mathbf{v}^{(1)} \preceq_f \ldots \preceq_f \mathbf{v}^{(n)}$. Finally, although our focus has been on AXps, contrastive explanations [26] could be adapted to the ranking setting in a similar manner.

3.1 On the Need for FXAI for Rankers

One could be tempted to explain a ranking function f by reducing such a problem to classification. One can, in fact, consider the binary classifier $\kappa_f : \mathbb{F}^2 \to \{0, 1\}$ defined by

$$\kappa_f(\mathbf{x}, \mathbf{x}') = \begin{cases} 1, & \text{if } \mathbf{x} \preceq_f \mathbf{x}', \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Explaining why $\mathbf{v} \preceq_f \mathbf{v}'$, then reduces to explain why κ_f evaluate to 1 on the vector $(\mathbf{v}, \mathbf{v}')$ of the feature space $\mathbb{F}^2$. This approach, which we term the *naive approach* has some shortcomings. Indeed, it differs from ours in that it treats the

i-th component of $\mathbf{v}$ independently from the i-th component of $\mathbf{v}'$, resulting (i) in explanations defined over the new feature set $\mathcal{F} \cup \mathcal{F}'$ obtained by adding a primed copy for each feature, and (ii) a consequently higher complexity for computing such explanations. Note also that the requirement $x_i = v_i \land x'_i = v'_i$ for each $i \in \mathcal{S}$, cannot be reproduced using domain constraints [19]. This is because computing an AXp involves considering a different set $\mathcal{S}$ at each step, which results in varying constraints on $\mathbf{x}$ and $\mathbf{x}'$. On the contrary, domain constraints are fixed over the feature space.

It is also useful to compare our work with approaches for explaining regression models $f : \mathbb{F} \to \mathbb{R}$, such as neural networks [54,28] and regression trees [6,8]. These methods explain a prediction $f(\mathbf{v})$ by treating all points within a circular neighbourhood of $f(\mathbf{v})$—with radius specified by a user parameter—as if they were equal to that value. In contrast, by considering the ranking induced by f rather than the exact values, our approach can handle real-valued functions without any approximation or the need for additional parameters.

3.2 On the Concept of Best Explanations

Explanation problems can admit multiple AXps, raising the question of which AXp to prefer. A natural choice is to favor smaller AXps, ideally those that are cardinality-minimal. However, computing cardinality-minimal explanations can be computationally intensive [27]. Moreover, even cardinality-minimal explanations may not be unique.

To address this, we define a score function $score : 2^{\mathcal{F}} \to \mathbb{R}$ by posing

$$score(\mathcal{S}) = \min_{(\mathbf{x},\mathbf{x}') \in [\mathbf{v}]_{\mathcal{S}} \times [\mathbf{v}']_{\mathcal{S}}} (f(\mathbf{x}') - f(\mathbf{x})). \tag{7}$$

A set $\mathcal{S}$ is then a WeakAXp if and only if $score(\mathcal{S}) \geq 0$. We define a preference relation $\preceq$ on $2^{\mathcal{F}}$ by posing

$$\mathcal{S}_1 \preceq \mathcal{S}_2 \quad \text{iff} \quad score(\mathcal{S}_1) \leq score(\mathcal{S}_2).$$

This preference relation can then serve as a tie-breaking criterion among explanations of the same size. The concept of score is particularly meaningful when f carries significance beyond mere ranking, a common situation as demonstrated in our case study, where the ranking is based on the probabilities of breast cancer recurrence. Note also that our preference relation is defined directly by the explanation problem and therefore forms a more objective basis for identifying the best explanations, as opposed to approaches requiring the incorporation of formal models of the explainee [5].

4 Algorithms

In this section, we start by introducing a model-agnostic algorithm for verifying whether a set $\mathcal{S} \subseteq \mathcal{F}$ is a WeakAXp. We then combine it with a deletion-based

Algorithm 1: Verify WeakAXp.

Input: Ranker f, points $\mathbf{v}_1$, $\mathbf{v}_2$, candidate set $\mathcal{S}$, feature space $\mathbb{F}$, cache memo
Output: WeakAXp($\mathcal{S}$)

```
1  Function GenerateVectors(v, S, F):
2      foreach i ∈ S do
3          D_i ← {v_i};
4      return ∏_{i=1}^{m} D_i // as an iterator
5  Function CachedPredict(x, memo):
6      if x ∉ memo then
7          memo[x] ← f(x);
8      return memo[x]
9  foreach x_1 ∈ GenerateVectors(v_1, S, F) do
10     p_1 ← CachedPredict(x_1, memo);
11     foreach x_2 ∈ GenerateVectors(v_2, S, F) do
12         if p_1 > CachedPredict(x_2, memo) then
13             return false;
14 return true;
```

search strategy in order to compute an AXp. Our implementation is optimised to minimise redundant calls to both the ranker and the verification procedure. While the definitions in the previous section require no assumptions about the domains $\mathbb{D}_i$, in this section we assume them to be finite.

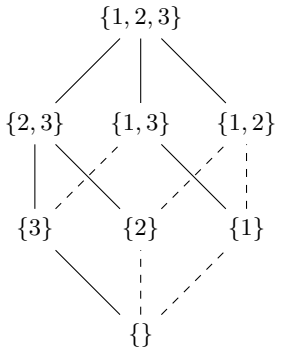
4.1 Verification of a WeakAXp

We now describe Algorithm 1 for verifying whether a set $\mathcal{S} \subseteq \mathcal{F}$ is a WeakAXp for the explanation problem $(f, \mathbf{v}, \mathbf{v}')$. Such algorithm works by searching for a counterexample $(\mathbf{x}, \mathbf{x}') \in [\mathbf{v}]_{\mathcal{S}} \times [\mathbf{v}']_{\mathcal{S}}$ to the claim of $\mathsf{WeakAXp}(\mathcal{S})$ being true (lines 9–13). If no such pair is encountered, then $\mathcal{S}$ is certified as a WeakAXp (line 14). The vectors $\mathbf{x}$ are generated (lines 1-4) by fixing $x_i = v_i$ for each $i \in \mathcal{S}$ (lines 2-3), while letting x_j varying freely within the domain $\mathbb{D}_j$ for $j \in \mathcal{F} \setminus \mathcal{S}$ (analogously for $\mathbf{x}'$). This implies that the number of vectors grows exponentially with $|\mathcal{F} \setminus \mathcal{S}|$, rather than $|\mathcal{F}|$, making verification particularly efficient for large $\mathcal{S}$. Additionally, vectors are generated on-the-fly via an iterator (line 4), eliminating the need to construct the equivalence class $[\mathbf{v}]_{\mathcal{S}}$. Finally, we apply memoisation (lines 5-8) to cache predictions and avoid redundant computation, including across repeated invocations of the algorithm.

Theorem 4. *Let $d = \max_{i \in \mathcal{F}} |\mathbb{D}_i|$. Algorithm 1 for verifying a WeakAXp $\mathcal{S} \subseteq \mathcal{F}$ containing k of the m features, requires at most $2d^{m-k}$ calls to the function f.*

It is worth mentioning that, while Algorithm 1 systematically explores the entire feature space to guarantee correctness, its model-agnostic nature makes

Fig. 1. Hasse diagram of the search space for $m = 3$ features. Dashed lines indicate child nodes skipped during traversal.



Algorithm 2: Deletion-based Computation of an AXp.

Input: $\mathcal{S} \subseteq \mathcal{F}$
Output: AXp $\mathcal{S}' \subseteq \mathcal{S}$ or `None`

```
if not WeakAXp(S) then
    return None
S' ← S
for i ∈ S do
    if WeakAXp(S' \ {i}) then
        S' ← S' \ {i}
return S'
```

it naturally compatible with heuristic strategies such as random sampling to approximate the verification process—following an approach similar in spirit to that used in many popular heuristic XAI techniques [47,46,36].

4.2 Computing an AXp

The monotonicity of WeakAXps, as established by Theorem 1, enables us to reformulate the problem of computing an AXp as an instance of the Minimal Set over a Monotone Predicate (MSMP) problem [41], which can be efficiently solved using the deletion-based approach presented in Algorithm 2. The basic idea is as follows. Given a WeakAXp $\mathcal{S}$, the algorithm iteratively attempts to refine it by deleting one element at a time in order to find a proper subset that is still a WeakAXp. If such a deletion is possible, the algorithm proceeds with the resulting subset. This process repeats until no single element can be removed without losing the WeakAXp property, at which point the algorithm concludes that the subset is an AXp and returns it. Note that an AXp $\mathcal{S}' \subseteq \mathcal{S}$ exists if and only if $\mathcal{S}$ is a WeakAXp. Therefore, the algorithm begins by checking this condition, returning `None` otherwise. Note also that the feature set $\mathcal{F}$ is a WeakAXp, since we assume $\mathbf{v} \preceq_f \mathbf{v}'$, and can thus be used as the starting point of the search. This search can be viewed as a traversal of the lattice of subsets of $\mathcal{F}$, ordered by set inclusion. Figure 1 shows the Hasse diagram of this lattice for $m = 3$ features. Note how the deletion-based algorithm leverages monotonicity by testing only the maximal proper subsets. In addition, it avoids testing all such subsets by excluding the subsets $\mathcal{S}' \setminus \{i\}$, for those $i \in \mathcal{S}$ that have already been looped over. In fact, these subsets are contained in previously visited sets that were not WeakAXps and, again due to monotonicity, cannot be WeakAXps themselves. For example—referring to Figure 1 and assuming, for illustrative purposes, the lattice is traversed by selecting features i in increasing order—if the algorithm reaches $\{1, 3\}$, it can infer that $\{3\}$ is not a WeakAXp

and can therefore be skipped. This is illustrated in the figure by a dashed line. Our implementation of Algorithm 2 selects features uniformly at random to avoid biasing the returned AXps toward certain features. The algorithm returns the first AXp encountered during the traversal, not necessarily a cardinality-minimal one. However, it is worth noting that the AXps returned tend to be small in practice, as there are more paths leading to a smaller AXp than to a larger one. Also, one could run the algorithm multiple times to try to find smaller AXps. Alternatively, one could modify the algorithm to continue beyond the first AXp found.

Finally, to contextualise the efficiency of our algorithm, consider again a ranking $\mathbf{v}^{(1)} \preceq_f \ldots \preceq_f \mathbf{v}^{(n)}$ between n vectors. While Algorithm 2 has a constant query complexity with respect to n, the naive approach scales linearly. This clearly demonstrates the computational advantage of our approach compared to the naive one.

5 Case Study

The experiments were carried out using Google Colab with no hardware acceleration. The code was written in Python using TensorFlow/Keras as the deep learning framework and is available at https://github.com/fracchiariello/jelia2025.

5.1 Experimental Setup

Dataset. We use the Breast Cancer dataset from the UCI Machine Learning Repository [59], which contains real-world data about breast cancer recurrence within five years after surgery. The dataset consists of 286 instances, each with 9 categorical features and assigned to one of 2 classes. Of these instances, 201 exhibit no cancer recurrence, whereas 85 indicate its occurrence, corresponding to a recurrence rate of around 30%. Table 1 lists the feature names along with the sizes of their corresponding domains.

Problem formulation. Predicting the recurrence of breast cancer is a binary classification problem. We address it via regression, by training a neural network to predict the probability of recurrence. These probabilities then define the desired ranking over the patients. This corresponds to a pointwise approach in Learning to Rank terminology [35]. Note that one could have chosen to predict the class and then consider the classifier-induced ranking, as defined in (3). However, we prefer to work with probabilities since they are more informative, as formalised by Theorem 2, enabling us to distinguish between patients within the same class.

Dataset preparation. We denote cancer recurrence with 1 and its absence with 0. To enable the neural network to handle categorical variables, we one-hot encode them. This results in a 43-dimensional feature space, representing 299376 distinct possible patients. We then split the dataset, allocating 80% for training and 20% for testing.

Table 1. Features and domains for the Breast Cancer dataset.

Feature	Name	$\lvert\mathcal{D}_i\rvert$
0	*age*	6
1	*menopause*	3
2	*tumor-size*	11
3	*inv-nodes*	7
4	*node-caps*	3
5	*deg-malig*	3
6	*breast*	2
7	*breast-quad*	6
8	*irradiat*	2

Table 2. Summary of the Keras model.

Layer type	Shape	Param #
Dense (ReLU)	(43, 64)	2816
Dense (ReLU)	(64, 32)	2080
Dense (sigmoid)	(32, 1)	33
Trainable params		4929
Optimizer params		9860
Total params		**14789**

Model Architecture. We consider a feedforward neural network with 3 dense layers, as shown in Table 2. The ReLU activation function is applied to the first two layers, while the output layer uses the sigmoid function, ensuring the output stays within the (0,1) range, therefore representing a probability.

Training. We train the model using the Adam optimiser and binary cross-entropy as the loss function. The trained model achieves an accuracy of 72% and an F_1 score of 53%. In comparison, a baseline model that always predicts 0 (the a priori most probable class) achieves an accuracy of 64% but an F_1 score of 0. It is important to note that the dataset is incomplete, meaning the available features do not allow for effective discrimination between the classes. Our results align with the performances reported in previous studies [42,14]. Moreover, our primary objective is to explain the ranking induced by the model rather than optimising its performance on the machine learning task.

5.2 Experiments

For the first experiment, we randomly sample the feature space to select 1000 pairs of feature vectors $\mathbf{v}, \mathbf{v}'$, ordering each pair such that $\mathbf{v} \preceq_f \mathbf{v}'$ to ensure the existence of AXps, with f corresponding to our neural network model. Note that the search for AXps is performed in the original feature space, as shown in Table 1, rather than in the 43-dimensional feature space of the neural network, with vectors one-hot encoded before being input into the network for processing. Table 3 reports the time required to compute the AXps. We observe that the average time increases as the size of the returned explanation decreases, consistently with Theorem 4. It is also worth noting how the support (i.e., the number of explanations for each size) varies. Specifically, the largest number of explanations corresponds to size 7, and this number gradually decreases as we move away from it. The column about the standard deviation shows that the execution time tends to vary considerably, even for explanations of the same size. This variability is expected, as execution time depends on several factors, including the specific features in the AXps, the sizes of their corresponding domains, the

Table 3. Execution Time for computing AXps, ordered by their size.

Exp. Size	Avg Time (s)	Std Dev (s)	Support
9	2.38	0.47	49
8	5.75	3.87	236
7	14.51	12.45	393
6	37.03	36.02	259
5	95.64	70.05	62
4	314.75	0.00	1
Overall	23.01	35.88	1000

Table 4. Feature Occurrences in the AXps.

Feat.	0	1	2	3	4	5	6	7	8
Occur.	881	715	957	764	847	727	572	913	572

size of intermediate sets encountered during lattice traversal, and the time required to determine whether these sets constitute a WeakAXp. We also report in Table 4 the number of occurrences of each feature across the 1000 AXps. In this regard, it is important to recall that our implementation of Algorithm 2 selects features uniformly at random, thereby avoiding skew in the count toward any particular feature. Interestingly, the size of the tumour (*tumor-size*, feature 2) emerges as the most common feature, appearing in 957 of the 1000 AXps.

So far, we have examined how AXps change when varying the pairs $(\mathbf{v}, \mathbf{v}')$, gaining deeper insights into the overall structure of the feature space. We now shift our focus to analyzing the different AXps generated for a specific pair of patients, selected from the test set and reported in Table 6. For patient $\mathbf{v}'$, cancer recurrence was observed, whereas for $\mathbf{v}$, it was not. The neural network f correctly classifies these cases, assigning prediction scores $f(\mathbf{v}) = 0.08$ and $f(\mathbf{v}') = 0.97$. To generate explanations, we ran Algorithm 2 ten times: six times producing explanations of size 7 and four times of size 6, with the execution times reported in Table 5. Table 6 lists the solutions of the smaller size (with $\mathcal{S}_1$ returned twice). Given the relatively small size of the feature space, it is feasible to exhaustively verify that these solutions are indeed cardinality-minimal. Notably, all explanations agree on features 0, 2, 4, 5, and 8, differing only in the final selected feature. This brings us to the question of which explanation to prefer. To address this, we refer to the definition of the score as introduced in Equation (7). The computed scores are: $\mathcal{S}_1 = 0.305$, $\mathcal{S}_2 = 0.002$, and $\mathcal{S}_3 = 0.292$, so that $\mathcal{S}_2 \prec \mathcal{S}_3 \prec \mathcal{S}_1$. These scores can be calculated concurrently with the verification of WeakAXps by modifying Algorithm 1 to track the minimum value of $f(\mathbf{x}') - f(\mathbf{x})$ encountered. These findings not only help prioritise explanations but also suggest an implicit ranking of features based on both their occurrences across AXps and the relative scores of the explanations in which they appear: $1 \prec_f 6 \prec_f 7 \prec_f 3 \prec_f 0 \sim_f 2 \sim_f 4 \sim_f 5 \sim_f 8$.

Table 5. Execution time for computing different AXps for the same explanation problem.

Exp. Size	Avg Time (s)	Support
7	24.03	6
6	65.24	4
Overall	40.51	10

Table 6. Smallest AXps found.

$\mathcal{F}$	0	1	2	3	4	5	6	7	8
$\mathbf{v}$	2	2	3	0	1	1	1	3	0
$\mathbf{v}'$	4	0	3	3	2	2	0	2	1
$\mathcal{S}_1$	1	0	1	1	1	1	0	0	1
$\mathcal{S}_2$	1	0	1	0	1	1	1	0	1
$\mathcal{S}_3$	1	0	1	0	1	1	0	1	1

6 Related Work

FXAI has received significant attention in recent years [38,17], with approaches ranging from knowledge compilation techniques [51,52] to the use of reasoning engines (e.g., SAT, SMT or ILP solvers) [27,43]. Research has explored various dimensions, including enumeration of explanations [26], and explainability queries [7,4,22]. In addition, different types of explanations have been proposed, such as probabilistic explanations [53,3,31] and feature importance scores [34]. Techniques to efficiently navigate the feature space under constraints have also been examined [19,56,16]. Various studies focus on models with specific properties, such as monotonic classifiers [40,24]. Other research targets particular classes, such as decision trees [29,30,22] and decision lists [25], to develop practically efficient algorithms. The main challenge of formal explainability remains its ability to scale to more complex models. However, recent advancements have led to significant improvements in this area [54,28].

The problem of explaining preferences over a combinatorial structure was first studied in the domain of multi-criteria decision-making [32]. [33], for example, considers explanations for weights-based decision models. Ranking functions were then studied in [55], where the authors quantify the importance of each feature in a score-based ranker and consider other measures about stability and diversity of the ranking, useful for fairness considerations [57]. [18] proposes participation metrics to explain monotonic ranking functions, quantifying feature importance based on the analysis of the functions themself. [1] propose a framework to explain competitive rankings based on the analysis of the local impact of each feature to quantify its importance. [21] proposed an approach to explain pairwise comparisons based on Shapley values. Later [45] proposed to use Shapley values to explain score-based rankers, rather than learned pairwise comparisons. [49] uses a greedy algorithm based on Shapley values to compute counterfactual explanations for an item to reach a desired rank position, while keeping all other items fixed. Note how this is a simpler problem than the one we consider here, where we allow the features of all the items to vary.

Related research areas that consider rankings include information retrieval and recommender systems [58,2,13]. However, these studies differ considerably from the aforementioned works and the present paper, as they focus on ranking w.r.t. a given query usually expressed in natural language.

Explaining preferences has received increasing attention in Computational Social Choice to justify election outcomes [10,11,44]. However, these works typically represent candidates as atomic objects, with explanations referring to voter preferences or voting rules, rather than on specific features of the candidates. Interestingly, FXAI was recently applied in this context [15].

7 Conclusion

In this paper, we introduced the first formal definitions of explanations for ranking functions. Although applying FXAI to rankings has been considered straightforward through a reduction to classification [33], such a reduction is computationally prohibitive, limiting both research and practical use of formal explainability for rankings and preferences. By introducing definitions tailored specifically to ranking functions—resembling those for classifiers but not reducible to them—we presented the first practically feasible approach to formal explainability in the ranking setting, provided the number of features remains manageable. Our formal approach is model-agnostic, relying on querying the system, and scales linearly with the inference time of the model used to solve the specific ranking problem. We established several key properties of these definitions, including the monotonicity of WeakAXps, and demonstrated how a deletion-based algorithm can efficiently compute AXps by leveraging this property. Recognising that an explanation problem can admit multiple (cardinality-minimal) solutions, we introduce a score-based criterion to serve as a tie-breaker among such solutions. To the best of our knowledge, this is the first proposal in FXAI to introduce a preference criterion over the explanations, beyond their size, that does not rely on incorporating the preferences of the explainee. We implemented our approach and tested it on a real-world use case: a neural network model trained to predict breast cancer recurrence. Our experiments demonstrated the feasibility of the approach and highlighted the connection to the theoretical results.

On a terminological note, while we prefer the generic terms *ranking* and *ranking functions*, which also emphasise their connection to machine learning tasks, these correspond respectively to *preferences* and *utility functions* over combinatorial domains [12]. As such, the relevance of our work for AI extends beyond machine learning to encompass autonomous agents and multi-agent systems.

This paper aimed at laying down the theoretical foundations for applying FXAI to ranking functions. As a result, the challenge of efficiently computing these explanations and scaling the approach to larger problems remains an open question. Our model-agnostic approach enables the verification of explanations for black-box ranking functions. However, it is worth exploring alternative formal approaches that leverage logic-based representations of these functions, which could yield improved performance for specific classes of models. Additionally, investigating probabilistic approaches may help address the challenge posed by large numbers of features. These avenues for future research could further enhance the applicability and efficiency of FXAI in explaining ranking functions.

Acknowledgments

This work has been partially supported by the European Research Council (ERC), Grant agreement No. 885107; by the Excellence Strategy of the Federal Government and the NRW Länder, Germany; and by the Spanish Government under grant PID 2023-152814OB-I00.

References

1. Anahideh, H., Mohabbati-Kalejahi, N.: Local explanations of global rankings: Insights for competitive rankings. IEEE Access **10**, 30676–30693 (2022)
2. Anand, A., Lyu, L., Idahl, M., Wang, Y., Wallat, J., Zhang, Z.: Explainable information retrieval: A survey. CoRR **abs/2211.02405** (2022)
3. Arenas, M., Barceló, P., Orth, M.A.R., Subercaseaux, B.: On computing probabilistic explanations for decision trees. In: NeurIPS (2022)
4. Audemard, G., Bellart, S., Bounia, L., Koriche, F., Lagniez, J., Marquis, P.: On the computational intelligibility of boolean classifiers. In: KR. pp. 74–86 (2021)
5. Audemard, G., Bellart, S., Bounia, L., Koriche, F., Lagniez, J., Marquis, P.: On preferred abductive explanations for decision trees and random forests. In: IJCAI. pp. 643–650. ijcai.org (2022)
6. Audemard, G., Bellart, S., Lagniez, J., Marquis, P.: Computing abductive explanations for boosted regression trees. In: IJCAI. pp. 3432–3441. ijcai.org (2023)
7. Audemard, G., Koriche, F., Marquis, P.: On tractable XAI queries based on compiled representations. In: KR. pp. 838–849 (2020)
8. Audemard, G., Lagniez, J., Marquis, P.: On the computation of contrastive explanations for boosted regression trees. In: ECAI. Frontiers in Artificial Intelligence and Applications, vol. 392, pp. 1083–1091. IOS Press (2024)
9. Barten, A.P., Böhm, V.: Consumer theory. Handbook of mathematical economics **2**, 381–429 (1982)
10. Boixel, A., Endriss, U.: Automated justification of collective decisions via constraint solving. In: AAMAS. pp. 168–176. International Foundation for Autonomous Agents and Multiagent Systems (2020)
11. Cailloux, O., Endriss, U.: Arguing about voting rules. In: AAMAS. pp. 287–295. ACM (2016)
12. Chevaleyre, Y., Endriss, U., Lang, J., Maudet, N.: Preference handling in combinatorial domains: From AI to social choice. AI Mag. **29**(4), 37–46 (2008)
13. Chowdhury, T., Rahimi, R., Allan, J.: Rank-lime: Local model-agnostic feature attribution for learning to rank. In: ICTIR. pp. 33–37. ACM (2023)
14. Clark, P., Niblett, T.: Induction in noisy domains. In: EWSL. pp. 11–30. Sigma Press, Wilmslow (1987)
15. Contet, C., Grandi, U., Mengin, J.: Abductive and contrastive explanations for scoring rules in voting. In: ECAI. Frontiers in Artificial Intelligence and Applications, vol. 392, pp. 3565–3572. IOS Press (2024)
16. Cooper, M., Amgoud, L.: Abductive explanations of classifiers under constraints: Complexity and properties. In: 26th European Conference on Artificial Intelligence (ECAI 2023). pp. à–paraître. IOS Press (2023)
17. Darwiche, A.: Logic for explainable AI. In: LICS. pp. 1–11 (2023)
18. Gale, A., Marian, A.: Explaining monotonic ranking functions. Proceedings of the VLDB Endowment **14**(4), 640–652 (2020)

19. Gorji, N., Rubin, S.: Sufficient reasons for classifier decisions in the presence of domain constraints. In: AAAI. pp. 5660–5667. AAAI Press (2022)
20. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. **51**(5), 93:1–93:42 (2019). https://doi.org/10.1145/3236009, https://doi.org/10.1145/3236009
21. Hu, R., Chau, S.L., Huertas, J.F., Sejdinovic, D.: Explaining preferences with shapley values. In: NeurIPS (2022)
22. Huang, X., Izza, Y., Ignatiev, A., Marques-Silva, J.: On efficiently explaining graph-based classifiers. In: KR. pp. 356–367 (2021)
23. Huang, X., Marques-Silva, J.: On the failings of shapley values for explainability. Int. J. Approx. Reason. **171**, 109112 (2024)
24. Hurault, A., Marques-Silva, J.: Certified logic-based explainable AI - the case of monotonic classifiers. In: TAP. Lecture Notes in Computer Science, vol. 14066, pp. 51–67. Springer (2023)
25. Ignatiev, A., Marques-Silva, J.: Sat-based rigorous explanations for decision lists. In: SAT. Lecture Notes in Computer Science, vol. 12831, pp. 251–269. Springer (2021)
26. Ignatiev, A., Narodytska, N., Asher, N., Marques-Silva, J.: From contrastive to abductive explanations and back again. In: AI*IA. Lecture Notes in Computer Science, vol. 12414, pp. 335–355. Springer (2020)
27. Ignatiev, A., Narodytska, N., Marques-Silva, J.: Abduction-based explanations for machine learning models. In: AAAI. pp. 1511–1519. AAAI Press (2019)
28. Izza, Y., Huang, X., Morgado, A., Planes, J., Ignatiev, A., Marques-Silva, J.: Distance-restricted explanations: Theoretical underpinnings & efficient implementation. In: KR (2024)
29. Izza, Y., Ignatiev, A., Marques-Silva, J.: On explaining decision trees. CoRR **abs/2010.11034** (2020)
30. Izza, Y., Ignatiev, A., Marques-Silva, J.: On tackling explanation redundancy in decision trees. J. Artif. Intell. Res. **75**, 261–321 (2022)
31. Izza, Y., Ignatiev, A., Narodytska, N., Cooper, M.C., Marques-Silva, J.: Provably precise, succinct and efficient explanations for decision trees. CoRR **abs/2205.09569** (2022)
32. Keeney, R.L., Raiffa, H.: Decisions with multiple objectives: preferences and value trade-offs. Cambridge university press (1993)
33. Labreuche, C.: A general framework for explaining the results of a multi-attribute preference model. Artif. Intell. **175**(7-8), 1410–1448 (2011)
34. Letoffe, O., Huang, X., Asher, N., Marques-Silva, J.: From SHAP scores to feature importance scores. CoRR **abs/2405.11766** (2024)
35. Liu, T.: Learning to rank for information retrieval. Found. Trends Inf. Retr. **3**(3), 225–331 (2009)
36. Lundberg, S.M., Lee, S.: A unified approach to interpreting model predictions. In: NIPS. pp. 4765–4774 (2017)
37. Marques-Silva, J.: Minimal unsatisfiability: Models, algorithms and applications (invited paper). In: ISMVL. pp. 9–14. IEEE Computer Society (2010)
38. Marques-Silva, J.: Logic-based explainability in machine learning. In: Reasoning Web. pp. 24–104 (2022)
39. Marques-Silva, J.: Logic-based explainability: Past, present and future. In: ISoLA (4). Lecture Notes in Computer Science, vol. 15222, pp. 181–204. Springer (2024)

40. Marques-Silva, J., Gerspacher, T., Cooper, M.C., Ignatiev, A., Narodytska, N.: Explanations for monotonic classifiers. In: ICML. Proceedings of Machine Learning Research, vol. 139, pp. 7469–7479. PMLR (2021)
41. Marques-Silva, J., Janota, M., Mencía, C.: Minimal sets on propositional formulae. problems and reductions. Artif. Intell. **252**, 22–50 (2017)
42. Michalski, R.S., Mozetic, I., Hong, J., Lavrac, N.: The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In: AAAI. pp. 1041–1047. Morgan Kaufmann (1986)
43. Narodytska, N., Shrotri, A.A., Meel, K.S., Ignatiev, A., Marques-Silva, J.: Assessing heuristic machine learning explanations with model counting. In: SAT. Lecture Notes in Computer Science, vol. 11628, pp. 267–278. Springer (2019)
44. Peters, D., Procaccia, A.D., Psomas, A., Zhou, Z.: Explainable voting. In: NeurIPS (2020)
45. Pliatsika, V., Fonseca, J., Wang, T., Stoyanovich, J.: Sharp: Explaining rankings with shapley values. CoRR **abs/2401.16744** (2024)
46. Ribeiro, M.T., Singh, S., Guestrin, C.: "why should I trust you?": Explaining the predictions of any classifier. In: KDD. pp. 1135–1144. ACM (2016)
47. Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: High-precision model-agnostic explanations. In: AAAI. pp. 1527–1535. AAAI Press (2018)
48. Rudeanu, S.: Sets and ordered structures. Bentham Science Publishers (2012)
49. Salimiparsa, M.: Counterfactual explanations for rankings. In: Canadian AI. Canadian Artificial Intelligence Association (2023)
50. Shapley, L.S., et al.: A value for n-person games (1953)
51. Shih, A., Choi, A., Darwiche, A.: A symbolic approach to explaining bayesian network classifiers. In: IJCAI. pp. 5103–5111. ijcai.org (2018)
52. Shih, A., Choi, A., Darwiche, A.: Compiling bayesian network classifiers into decision graphs. In: AAAI. pp. 7966–7974. AAAI Press (2019)
53. Wäldchen, S., MacDonald, J., Hauch, S., Kutyniok, G.: The computational complexity of understanding binary classifier decisions. J. Artif. Intell. Res. **70**, 351–387 (2021)
54. Wu, M., Wu, H., Barrett, C.W.: Verix: Towards verified explainability of deep neural networks. In: NeurIPS (2023)
55. Yang, K., Stoyanovich, J., Asudeh, A., Howe, B., Jagadish, H.V., Miklau, G.: A nutritional label for rankings. In: SIGMOD Conference. pp. 1773–1776. ACM (2018)
56. Yu, J., Ignatiev, A., Stuckey, P.J., Narodytska, N., Marques-Silva, J.: Eliminating the impossible, whatever remains must be true: On extracting and applying background knowledge in the context of formal explanations. In: AAAI. pp. 4123–4131. AAAI Press (2023)
57. Zehlike, M., Yang, K., Stoyanovich, J.: Fairness in ranking, part I: score-based ranking. ACM Comput. Surv. **55**(6), 118:1–118:36 (2023)
58. Zhang, Y., Chen, X.: Explainable recommendation: A survey and new perspectives. Found. Trends Inf. Retr. **14**(1), 1–101 (2020)
59. Zwitter, M., Soklic, M.: Breast Cancer. UCI Machine Learning Repository (1988), DOI: https://doi.org/10.24432/C51P4M