Proxy Attribute Discovery in Machine Learning Datasets via Inductive Logic Programming

Rafael Gonçalves(✉), Filipe Gouveia, Inês Lynce, and José Fragoso Santos

INESC-ID & Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal
{rafael.s.goncalves,filipe.gouveia,ines.lynce,
jose.fragoso}@tecnico.ulisboa.pt

Abstract. The issue of fairness is a well-known challenge in Machine Learning (ML) that has gained increased importance with the emergence of Large Language Models (LLMs) and generative AI. Algorithmic bias can manifest during the training of ML models due to the presence of sensitive attributes, such as gender or racial identity. One approach to mitigate bias is to avoid making decisions based on these protected attributes. However, indirect discrimination can still occur if sensitive information is inferred from proxy attributes. To prevent this, there is a growing interest in detecting potential proxy attributes before training ML models. In this case study, we report on the use of Inductive Logic Programming (ILP) to discover proxy attributes in training datasets, with a focus on the ML classification problem. While ILP has established applications in program synthesis and data curation, we demonstrate that it can also advance the state of the art in proxy attribute discovery by removing the need for prior domain knowledge. Our evaluation shows that this approach is effective at detecting potential sources of indirect discrimination, having successfully identified proxy attributes in several well-known datasets used in fairness-awareness studies.

Keywords: Fairness, Machine Learning, Proxy Attributes, Inductive Logic Programming.

1 Introduction

Recent years have seen a dramatic rise in the popularity of Artificial Intelligence (AI) and Machine Learning (ML). ML algorithms are now deeply integrated into our daily lives, playing critical roles in tasks ranging from loan approval [44] to recidivism prediction [47]. While this shift towards machine-guided decision-making has streamlined processes that were once labor-intensive, it has also raised significant concerns about the potential for discriminatory behavior in AI. For example, some ML models have exhibited biases based on gender [15] or racial identity [3], underscoring the need for fair ML.

A common strategy to ensure fairness in ML models is to prevent them from relying on sensitive information during the learning process [39]. This approach distinguishes between two types of attributes in training datasets: *protected* and *non-protected* [7,16,24,25]. Protected attributes are potential sources

A. Gurfinkel and M. Heule (Eds.): TACAS 2025, LNCS 15697, pp. 343–363, 2025.
https://doi.org/10.1007/978-3-031-90653-4_17

ID	Marital	Country	Sex
1	husband	usa	male
2	wife	usa	female
3	husband	uk	male
4	single	uk	female
5	single	usa	male

Table 1. Example dataset with non-protected attributes **Marital** and **Country**, and protected attribute **Sex**. The non-protected attribute **Marital** is a proxy of **Sex** (when **Marital** = *husband* or **Marital** = *wife*).

of direct discrimination (*e.g.*, gender and race), and should not be used as decision points [31]. However, indirect discrimination may still occur if some of the non-protected attributes leak information about the protected ones. For example, evidence suggests that an individual's race can, in some cases, be inferred from their ZIP code [17]. As a result, a model using the non-protected attribute **ZIP Code** as a decision point might still exhibit racial bias, even if it does not directly use the protected attribute **Race**. Hence, we say that **ZIP Code** is a *proxy attribute* [51] of **Race**, since the latter can be inferred from the former.

Proxy Attributes. A proxy attribute is an attribute that has a causal effect on some other attribute [40,51]. In other words, proxy attributes allow us to at least partially derive the values of another attribute, as seen with **ZIP Code** and **Race**. While these attributes can be either protected or non-protected, we generally use the term *proxy attribute* to refer specifically to a non-protected attribute that infers information about a protected one.

Consider the example shown in Table 1. The dataset has two non-protected attributes, **Marital** and **Country**, and one protected attribute, **Sex**. For clarity, the dataset also includes a column **ID** that uniquely identifies each row. Note that this column is not actually part of the dataset and is neither a protected nor a non-protected attribute.

In the example, **Marital** acts as a proxy attribute of **Sex** when it has value *husband* or *wife*. Specifically, when **Marital** = *husband*, **Sex** has value *male*, and when **Marital** = *wife*, **Sex** has value *female*. In other words, the attribute **Marital** determines **Sex**, although the opposite does not necessarily hold (*e.g.*, **Sex** = *male* does not determine the value of **Marital**). Furthermore, we cannot infer the value of **Sex** when **Marital** = *single*, *i.e.*, the relation between proxy and protected attributes is only partially defined. This is a common occurrence in real-world datasets, as we shall discuss later.

Proxy Attribute Discovery. This case study reports on the use of Inductive Logic Programming (ILP) [14,19,37,48] to identify proxy attributes in ML train-

Approach	No Domain Knowledge	Partiality	Noise	Arithmetic	Expressivity
ILP	✓	✓	✓	✓	✓
Causal Graphs	×	✓	✓	×†	×
Program Synthesis	✓	×	✓	✓	✓

Table 2. Comparison between different approaches for proxy attribute discovery: ILP, causal graphs and program synthesis. While some causal approaches may support arithmetic†, they require substantial user effort to do so, as opposed to the other two.

ing datasets. The quality of an algorithm for proxy attribute discovery can be evaluated based on the following five criteria:

1. **Dependence on domain knowledge:** the degree to which the algorithm relies on user-provided domain knowledge to detect proxy attributes.
2. **Support for partial proxy attributes:** the ability to support proxy relations that are only partially defined (*e.g.*, the **Marital** proxy in Table 1).
3. **Support for noisy data:** the ability to handle conflicts, such as rows with similar proxy attributes having different protected attributes.
4. **Support for arithmetic relations:** the ability to infer arithmetic relations beyond simple equalities, such as order relations (*e.g.*, less-than, greater-than relations) or arithmetic operations (*e.g.*, addition, multiplication).
5. **Expressivity of the output:** the extent to which the algorithm's output provides insight into the proxy attributes, *i.e.*, whether it simply identifies them or explicitly describes the proxy relation.

Table 2 shows a comparison between three different approaches for proxy attribute discovery: ILP, causal graphs and program synthesis. Causal graphs are directed acyclic graphs where nodes represent attributes and edges capture the relations between them. While causal approaches are the most popular in prior work [31,35,39], they are limited by their dependence on the availability of domain knowledge, such as the need for user-provided causal graphs [31,35] or the categorization of numeric attributes [39]. Furthermore, their output is not expressive, and while some causal approaches support arithmetic [39], they require extensive manual preprocessing to do so.

In contrast, program synthesis techniques [41,46,52], although not yet applied in this domain, offer a promising alternative for proxy attribute discovery. Following such an approach, a synthesis tool would try to derive a function that computes protected attributes from proxy attributes. However, these techniques offer no native support for partial proxy attributes, as existing table-based synthesis methods mostly focus on fully defined operations, such as SQL queries or table transformations [41]. ILP, on the other hand, follows a rule-based synthesis approach, which offers native support for partiality in the inferred relations.

Contributions. The main contribution of this case study is a new methodology for proxy attribute discovery based on ILP (§4), which we have implemented as

a Python tool, PADTAI, available online [22]. We have applied this tool to 10 real-world datasets, and detected up to 83 proxy relations, corresponding to 49 potential proxy attributes, in 8 of those (§5). Our evaluation shows that ILP is an effective technique for proxy attribute discovery, and compares favorably to more traditional causal approaches.

2 Related Work

Fairness. AI fairness has emerged as an increasingly critical research area in recent years [1,2,5,10,20,29,53]. Despite this, there is still no general consensus on how to define and evaluate fairness in AI systems [51]. In the context of the Machine Learning classification problem, which this case study focuses on, Verma and Rubin [51] identify three main categories of fairness definitions: **(i)** *statistical measures* [4,9,11,18,26,32]; **(ii)** *similarity-based measures* [18,21,35], which are grounded in the principle that similar individuals should have similar classifications; and **(iii)** *causal reasoning* [31,35,39,45], which bases its definition of fairness on the causal relations between attributes.

Most fairness definitions in ML algorithms differentiate between protected and non-protected attributes. Protected attributes are sensitive features for which non-discriminatory behaviour should be established (*e.g.*, race, religion, or gender). Much of the research on algorithmic fairness actively attempts to find biases against these attributes with respect to various fairness metrics [18,35,50]. Other approaches employ more complex methods; for instance, Ignatiev et al. [30] exploit formal reasoning techniques to rigorously analyze and build fair ML models. However, to the best of our knowledge, we are the first to apply ILP to identify potential sources of discrimination.

Proxy Attributes. Indirect discrimination arising from proxy attributes has been studied using causal approaches [31,35]. These approaches assume that the classification problem and training dataset are accompanied by a causal graph that models the domain; for instance, Kusner et al. [35] use a causal graph to model the relations between grades (*e.g.*, *LSAT*, *GPA* and *FYA*) and race and gender in the *Lawschool* dataset [54]. However, causal graphs of a specific domain are often hard to find, especially in the social contexts where fairness is most relevant, making causal methods largely inaccessible to non-experts.

A notable exception is the work of Le Quy et al. [39], who circumvent the need for predefined causal graphs. Instead, the authors learn Bayesian networks [28] directly from datasets by approaching the task as an optimization problem. While this method reduces the need for domain knowledge compared to previous approaches [31,35], it still requires user input to categorize numeric attributes and handle arithmetic relations. In contrast, our ILP-based approach supports both categorical and numeric attributes natively and requires minimal user input.

Day	Temperature	Humid	Rain
sunday	cold	yes	yes
monday	hot	yes	no
tuesday	cold	no	no
wednesday	hot	no	no

Table 3. Example dataset for an ILP task.

3 Background: Inductive Logic Programming

Inductive Logic Programming (ILP) [14,19,37,48] is a form of Machine Learning that relies on *logic programs* to learn models. As with other forms of ML, ILP tries to induce a hypothesis from a set of training examples. However, unlike most forms of ML, where data is given in tabular format, ILP is *rule-based*, *i.e.*, it encodes both the data and the learned model as a set of logical rules. As a result, ILP learns *relations* between attributes, whereas most other methods learn functions. Consider, for instance, a classification problem whose goal is to predict the correct label of a given input. In this case, a typical ML model learns a complex function that transforms a vector of attributes into a prediction. In contrast, ILP learns a logical program that relates attributes to labels.

Suppose we want to predict if it is going to rain. We are given a dataset with meteorological data about the first four days of the week, and whether it rained or not (Table 3). This is a classification problem: given a set of examples, our goal is to induce a hypothesis that generalizes and correctly predicts the target label **Rain**. However, ILP does not rely on table-based learning; instead, the input table must be encoded as a set of logical rules. In general, an ILP system takes three sets as input: B, E^+ and E^-. The set B is the *background knowledge*, which serves a similar purpose to features in traditional ML approaches. The sets E^+ and E^- are the positive and negative examples respectively. In our case, we build these sets as follows:

$$B = \left\{ \begin{array}{l} \text{cold}(sunday). \\ \text{cold}(tuesday). \\ \text{hot}(monday). \\ \text{hot}(wednesday). \\ \text{humid}(sunday). \\ \text{humid}(monday). \end{array} \right\} \qquad E^+ = \{ \text{ rain}(sunday). \ \}$$

$$E^- = \left\{ \begin{array}{l} \text{rain}(monday). \\ \text{rain}(tuesday). \\ \text{rain}(wednesday). \end{array} \right\}$$

Note that these sets essentially translate Table 3 into a logical program: the background knowledge represents the features, while the examples represent the target labels. Given this input, an ILP system will try to find a hypothesis that covers as many positive examples and as few negative examples as possible. In

```
pm(m). pf(f).
ph(h). pw(w). ps(s).                    } DP
pusa(usa). puk(uk).
pmarital(1,h). pcountry(1,usa).
pmarital(2,w). pcountry(2,usa).
pmarital(3,h). pcountry(3,uk).          } CR
pmarital(4,s). pcountry(4,uk).
pmarital(5,s). pcountry(5,usa).
```

(a)

```
pos(psex(1,m)).
pos(psex(2,f)).
pos(psex(3,m)).
pos(psex(4,f)).
pos(psex(5,m)).
```

(b)

Fig. 1. Encoding the dataset from Table 1 in Popper: (a) background knowledge: decision points (**DP**) and column relations (**CR**), and (b) examples.

our case, for instance, the system might induce the hypothesis that it rains if it is cold and humid. Put formally:

$$H = \{\ \text{rain}(A)\ \text{:-}\ \text{cold}(A),\ \text{humid}(A).\ \}$$

where rain(A) is called the *head* of the rule and cold(A), humid(A) the *body*.

While ILP tools may differ in their specific features, the underlying principles remain largely the same. For this reason, much of our procedure for proxy attribute discovery easily translates to various systems. To ground our discussion, we will follow the syntax of Popper [14], a state-of-the-art ILP system. We elaborate on our rationale for choosing Popper in §5. Popper expects three files as input: **(i)** a *bias* file with syntactic and semantic information to restrict the search space; **(ii)** a *background knowledge* file containing predicates that the system may use in rule bodies; and **(iii)** an *examples* file with positive and/or negative examples for testing the derived rules. The bias includes the types of each column, which we infer automatically from the dataset, as well as the specifications (name and arity) of the predicates that represent them.

4 Proxy Attribute Discovery with ILP

In this section, we report on the use of ILP to discover proxy attributes in training datasets for ML models. Here, our main point of concern is how to model the detection of proxy attributes as an ILP problem. We discuss the basic dataset encoding strategy (§4.1), preprocessing techniques (§4.2), the handling of numeric attributes (§4.3), and the use of thresholds to filter the output (§4.4).

4.1 Basic Encoding

Our first priority is to establish a systematic procedure for encoding a simple ML dataset as valid input for our target ILP system. As an example, we revisit the dataset from Table 1. Recall that this dataset has two non-protected attributes,

Marital and **Country**, a protected attribute **Sex**, and that **Marital** is a proxy attribute of **Sex** when **Marital** = *husband* or **Marital** = *wife*. Informally, our goal is to encode each attribute as a logical relation, allowing the ILP system to derive rules that associate the protected attribute with the proxy attribute. Fig. 1 gives an abridged view of the encoding. Note that the background knowledge encodes the non-protected columns and the values appearing in the dataset (called *decision points*), while the examples encode the protected column.

Let us then formalize the encoding process. In a nutshell, we aim to define an encoding function $\mathcal{E}(T)$ that takes as input a dataset T and encodes it as an ILP program. We can formally define a dataset T as:

$$T = \sum_{i=1}^{n} (\rho_i, t_i)$$

Here, a dataset is described as a sum of n rows (ρ_i, t_i), where ρ_i is a tuple of non-protected attributes and t_i the protected attribute for some row i. For example, the first row of Table 1 is defined as $((husband, usa), male)$. Note that the column **ID** is not encoded, as it is neither a protected nor a non-protected attribute. In general, a dataset may have any number of protected attributes; however, for modeling purposes, we assume that each dataset has only one. We can easily transform a dataset with k protected attributes into k datasets with a single protected attribute by simply isolating each protected attribute at a time.

Given a dataset T, we define its encoding, $\mathcal{E}(T)$, as:

$$\mathcal{E}(T) = \left(\bigcup_{c \in \text{Cols} \cup \{t\}} \bigcup_{v \in \text{Vals}(c)} p_v(v) \right) \quad \textbf{(Decision Points)}$$

$$\cup \left(\bigcup_{i=1}^{n} \bigcup_{c \in \text{Cols}} p_c(i, \rho_i(c)) \right) \quad \textbf{(Column Relations)}$$

$$\cup \left(\bigcup_{i=1}^{n} \mathsf{pos}(p_t(i, t_i)) \right) \quad \textbf{(Examples)}$$

where p_x is constructed by appending the value of x to the character p (*e.g.*, if $x = m$, then $p_x = pm$). The intuition behind this encoding is more involved. We explain each component of the sum individually:

- **Decision Points:** We encode a value v as a *decision point* $p_v(v)$. Values encompass both the non-protected and protected attributes present in T. For example, the protected attribute *male* (shortened to m for the sake of simplicity) is encoded as $pm(m)$ (*cf.* Fig. 1a, line 1).
- **Column Relations:** We encode a non-protected entry $\rho_i(c)$, representing the value of column c in some row i, as a *column relation* $p_c(i, \rho_i(c))$. Intuitively, $\rho_i(c)$ isolates the value of the non-protected column c in the tuple of non-protected attributes ρ_i. For example, the non-protected entry *usa* in row 1 is encoded as $pcountry(1, usa)$ (*cf.* Fig. 1a, line 4).

ID	Marital	Race
1	husband	black
2	wife	white
3	husband	white
4	single	hispanic
5	single	white
6	husband	black

(a)

ID	Marital	Race
1	husband	black
2	wife	white

(b)

Table 4. Example dataset with non-protected attribute **Marital**, and protected attribute **Race**: (a) before, and (b) after removing conflicts.

– **Examples:** We encode a protected entry t_i, representing the value of the target label t in some row i, as an *example* $\mathsf{pos}(p_t(i, t_i))$. The meta-predicate pos is a Popper builtin and simply states that the example is *positive* (*i.e.*, that the relation $p_t(i, t_i)$ is true). For example, the protected entry *male* (shortened to m) in row 1 is encoded as $\mathsf{pos}(psex(1, m))$ (*cf.* Fig. 1b, line 1).

In the current example, given the encoding shown in Fig. 1, an ILP system would learn the rules $\text{sex}(X, Y)$:- $\text{pmarital}(X, Z)$, $\text{ph}(Z)$, $\text{pm}(Y)$, corresponding to the case **Marital** $= husband$, and $\text{sex}(X, Y)$:- $\text{pmarital}(X, Z)$, $\text{pw}(Z)$, $\text{pf}(Y)$, corresponding to the case **Marital** $= wife$.

4.2 Preprocessing Techniques

Although the basic encoding strategy described in §4.1 is effective for small synthetic datasets (*e.g.*, Table 1), it often falls short when applied in the wild. To address this, we discuss the application of preprocessing techniques, focusing on two key issues: *conflicts between rows*, which we solve via majority voting, and *size limitations*, which we mitigate through sampling.

Removing Conflicts. A *conflict* between two or more rows occurs when all their non-protected attributes are identical but the protected attribute differs. Put formally, two rows i, j are conflicting if $\rho_i = \rho_j$ but $t_i \neq t_j$. This is a common issue in real-world datasets; an example of how it might happen is given in Table 4a. The dataset has two sets of conflicts: rows 1, 3 and 6, and rows 4 and 5. While a human can easily recognize that this example is not incorrect, many ILP systems interpret such conflicts as errors and struggle to produce meaningful output. Popper, for instance, often fails to discover relevant relations in datasets with conflicts, as it cannot find a set of rules that accurately predicts the protected attribute across all conflicting rows.

ID	Score	Race
1	80	white
2	74	hispanic
3	83	white
4	72	white
5	75	black

Table 5. Example dataset with non-protected attribute **Score**, and protected attribute **Race**. The non-protected attribute **Score** is a proxy of **Race** (when **Score** > 75).

To remove conflicts, our tool implements a straightforward majority voting system. Given a set of conflicting rows, the system proceeds as follows: if any protected attribute holds a relative majority, it keeps a single row with that attribute; otherwise, it discards all rows. Applying these principles to Table 4a, the tool produces the dataset shown in Table 4b. In the example, the protected attribute **Race** $=$ *black* holds a relative majority in rows 1, 3 and 6; hence, the preprocessed dataset keeps row 1. In contrast, no protected attribute holds a relative majority in rows 4 and 5, which are discarded. Since this approach may occasionally introduce false positives (*e.g.*, **Marital** acting as a proxy of **Race** in Table 4b), rules inferred from the preprocessed dataset must be validated against the original dataset (*cf.* §4.4).

Sampling. Real-world datasets often have tens or even hundreds of thousands of rows. Applying our methodology directly to such datasets would result in extremely large ILP tasks. For instance, a dataset containing 10 non-protected attributes and 20 000 rows would produce $10 \times 20\,000 = 200\,000$ column relations. This poses a significant challenge to our approach, as even state-of-the-art ILP systems cannot efficiently handle inputs of that size.

To address this, our tool samples a random subset of rows from the dataset and applies the ILP procedure exclusively to this sample. The default sample size varies with the size of the dataset, and should balance performance and representativeness. Once the procedure terminates, the tool validates the rules inferred from the sampled rows against the entire dataset, accepting only those that perform well across all rows (*cf.* §4.4).

4.3 Numeric Attributes

Consider the example shown in Table 5. The dataset has one non-protected attribute, **Score**, and one protected attribute, **Race**. Here, **Score** acts as a proxy attribute of **Race**; specifically, when **Score** > 75, **Race** has value *white*. This relation cannot be discovered with our basic encoding methodology, which only supports equality-based reasoning for numeric attributes. However, as in

```
pw(w). ph(h). pb(b).
p72(72). p74(74). p75(75).
p80(80). p83(83).                                   } DP
pscore(1,80). pscore(2,74).
pscore(3,83). pscore(4,72).                         } CR
pscore(5,75).
lt(72,74). lt(72,75). lt(72,80).
lt(72,83). lt(74,75). lt(74,80).
lt(74,83). lt(75,80).                               } PR
lt(75,83). lt(80,83).
```

(a)

```
pos(prace(1,w)).
pos(prace(2,h)).
pos(prace(3,w)).
pos(prace(4,w)).
pos(prace(5,b)).
```

(b)

Fig. 2. Encoding the dataset from Table 5 in Popper: (a) background knowledge: decision points (**DP**), column relations (**CR**) and pair relations (**PR**), and (b) examples.

this example, some datasets (*e.g.*, the *Ricci* dataset [49]) contain proxy relations that can only be expressed through arithmetic relations beyond simple equalities.

In the context of ILP, one possible technique for reasoning about these relations is to explicitly encode them on the numeric values that appear in the sampled rows. For example, in Table 5, we can encode the underlying order relation using the less-than operator in tandem with the equality. Fig. 2 gives an abridged view of the application of this technique. The general structure is similar to that of Fig. 1, with the key difference being the inclusion of the less-than relations between integer values in the background knowledge. Note that the encoding of these relations is *grounded*, *i.e.*, it is explicitly defined for all relevant integer pairs rather than being expressed as a mathematical rule. While this has an impact on the size of the background knowledge file, it improves the performance of the ILP procedure by pruning the search space [13].

As before, we can formalize the encoding process for some dataset T. To account for arithmetic relations, our implementation is parametric on a set of relations to be grounded. For simplicity, however, we focus the formalism solely on the less-than operator. Hence, given a dataset T, we define its parameterized encoding, $\mathcal{E}_{lt}(T)$, as:

$$\mathcal{E}_{lt}(T) = \mathcal{E}(T) \qquad \textbf{(Basic Encoding)}$$

$$\cup \left(\bigcup_{c \in \text{Cols} \cup \{t\}} \; \bigcup_{\substack{i<j \\ i,j \in \text{Ints}(c)}} lt(i,j) \right) \qquad \textbf{(Pair Relations)}$$

Here, the less-than operator is encoded as a set of *pair relations* $lt(i, j)$ representing its grounding for all integers appearing in the sampled rows. Note that to parameterize $\mathcal{E}(T)$ on other relations, one needs only modify the extended encoding. For instance, addition can be modeled by changing the inner union to $\bigcup_{i,j \in \text{Ints}(c)} sum(i, j, i + j)$. By default, we encode only the less-than relation.

ID	Marital	Sex	
1	husband	male	(**TP**)
2	wife	female	(**TN**)
3	husband	female	(**FP**)
4	wife	male	(**FN**)

Table 6. Measuring **TP**/**TN**/**FP**/**FN** for the rule $\text{sex}(X, \textit{male})$:- $\text{marital}(X, \textit{husband})$.

In the current example, given the encoding shown in Fig. 2, an ILP system would learn the rule $\text{race}(X, Y)$:- $\text{pscore}(X, Z),\ \text{lt}(75, Z),\ \text{pw}(Y)$.

4.4 Thresholds

Identifying proxy attributes often requires a degree of flexibility in determining what qualifies as a proxy. For example, in Table 1, **Marital** acts as a proxy of **Sex** only when **Marital** = *husband* or **Marital** = *wife*, which will be reflected in the rules found by the ILP procedure. In general, it is acceptable for the inferred rules not to cover certain rows. Conversely, some rows may be misclassified due to noise or because the protected attribute is only strongly correlated with the proxy rather than directly implied by it. Therefore, rules must be evaluated against a set of criteria that allows for some leniency while still ensuring that overly specific or highly inaccurate rules are filtered out.

Metrics. For a given rule, we measure the number of rows where: **(i)** the proxy and protected attributes match the body and head of the rule, respectively (**TP**); **(ii)** neither the proxy nor the protected attributes match the body and head of the rule (**TN**); **(iii)** the proxy attributes match the body of the rule but the protected attribute does not match the head (**FP**); and **(iv)** the proxy attributes do not match the body of the rule but the protected attribute matches the head (**FN**). As an example, consider Table 6, which illustrates these concepts for the rule $\text{sex}(X, \textit{male})$:- $\text{marital}(X, \textit{husband})$, written in simplified notation for clarity. We base our filtering criteria on three metrics: recall (R), precision (P) and coverage (C), defined in the standard way:

$$R = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \qquad P = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}} \qquad C = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}}$$

Informally, recall ensures that rules are broad enough to filter out overly specific relations, precision that rules maintain a high level of accuracy, and coverage that rules apply to a statistically significant portion of the dataset. An attribute qualifies as a proxy only if there exists a rule pertaining to it that meets specific thresholds: by default, rules should be comprehensive (above 15% recall and 10% coverage) and highly accurate (above 85% precision). The choice of thresholds is critical in determining which proxies are found; we discuss them further in §5.3.

System	Open Source	Maintained	Documented	Noise	Arithmetic
Popper [14]	✓	✓	✓	✓	✓
Aleph [48]	✓	×	✓	✓	✓
Metagol [43]	✓	×	✓	×	✓
ILASP [38]	×	✓	✓	✓	✓

Table 7. Comparison between different ILP tools: Popper, Aleph, Metagol and ILASP.

5 Evaluation

To evaluate our approach, we developed a Python tool based on the methodology outlined in §4, with Popper [14] as its underlying ILP system. Our tool, PADTAI, is open-source and available online [22]. We have applied PADTAI to 10 real-world datasets, and detected up to 83 proxy relations, corresponding to 49 potential proxy attributes, in 8 of those. Here, we give an overview of our choice of ILP system and datasets (§5.1), report on the results (§5.2) and discuss the strengths and limitations of our approach (§5.3).

5.1 Experimental Setup

System Selection. To determine the most suitable ILP system for our tool's backend, we define five criteria: **(i)** open-source availability; **(ii)** active maintenance; **(iii)** quality and comprehensiveness of documentation; **(iv)** support for noise; and **(v)** support for arithmetic relations. Despite the wide array of available ILP systems, we narrow our focus to four that best fit our task: Popper [14], which we discussed previously; Aleph [48], a longstanding and widely-used system; and two modern ILP tools, Metagol [43] and ILASP [38].

Table 7 gives an abridged comparison of the evaluated systems. With the exception of Metagol, all tools offer support for both noise and arithmetic relations, and are well-documented. However, both Aleph and Metagol are no longer maintained, while ILASP, despite being actively maintained, is not open-source. Popper, on the other hand, meets all our criteria and is easy to use, making it the ideal choice for our experiments.

Datasets. We evaluate our approach against 10 real-world datasets with well-known sources of indirect discrimination [39]:

- The *Adult* dataset [33], which consists of US census data, and whose goal is to predict whether a person's annual income exceeds US$50 000. The dataset has three protected attributes: *race*, *sex* and *age*. We expect to find two proxy relations: between *relationship* and *sex*, and between *education* and *race*.
- The *KDD* dataset [8], which contains data from US population surveys, and whose goal is also to predict whether a person's annual income exceeds US$50 000. The dataset has two protected attributes: *race* and *sex*.

We expect to find two proxy relations between *detailed-household-summary-in-household*/*tax-filer-stat* and *sex*/*race*.

- The *German Credit* dataset [27], which contains samples of bank account holders and is used for credit risk assessment prediction. The dataset has two protected attributes: *age* and *foreign-worker*. We expect to find a proxy relation between *housing* and *age*.
- The *Bank Marketing* dataset [42], which contains data from marketing campaigns of a Portuguese banking institution, and whose goal is to predict whether a client will make a deposit subscription. The dataset has two protected attributes: *age* and *marital*. We expect to find a proxy relation between *job* and *marital*.
- The *Credit Card* dataset [55], which contains data of customers' default payments in Taiwan and has been used for predicting future default situations. The dataset has three protected attributes: *education*, *marriage* and *sex*. We expect to find a proxy relation between *age* and *marriage*.
- The *COMPAS* dataset [3], which contains criminological data and has been used for recidivism risk prediction. The dataset has two protected attributes: *race* and *sex*. We expect to find two proxy relations: between *score-text* and *race*, and between *v-score-text* and *sex*.
- The *Ricci* dataset [49], which contains data on promotion decisions within a fire department, and whose goal is to predict whether an individual receives a promotion based on their exam results. The dataset has one protected attribute: *race*. We expect to find a proxy relation between *oral* and *race*.
- The *Students* dataset [12], which contains student data from two Portuguese high schools, and whose goal is to predict students' final year grades. The dataset has two protected attributes: *sex* and *age*. We expect to find a proxy relation between *walc* and *sex*.
- The *OULAD* dataset [36], which contains information about students and their activities in virtual learning environments, and whose goal is to predict the success of a student. The dataset has one protected attribute: *gender*. We expect to find a proxy relation between *code-module* and *gender*.
- The *Lawschool* dataset [54], which contains US law school admission records from 1991, and whose goal is to predict if a candidate will pass the bar exam. The dataset has two protected attributes: *gender* and *race*. We expect to find two proxy relations between *lsat*/*ugpa* and *gender*/*race*.

A summary of the tested datasets and experimental results is given in [23, Appendices A and C]. For a more comprehensive discussion, see [39].

Establishing the Baseline. In the literature, there is no ground-truth list of proxy relations for the evaluated datasets. To establish our baseline of expected proxy relations, we began by analyzing the Bayesian networks given in [39], enumerating all non-protected attributes that were directly connected to protected attributes by an edge. This analysis signaled 41 attributes as proxy candidates, listed in [23, Appendix B].

Of the 41 candidates, five belonged to the *Lawschool* dataset, which was previously studied in [35]. We cross-checked these five candidates against the causal

relations identified in [35], leaving us with two validated proxy relations in this dataset. For the remaining 36 candidates, we employed a random forest classifier [6,34] to check whether the values of the corresponding protected attributes could be inferred with a Gini impurity level of 0.01. This process yielded another 12 expected proxy relations in the nine analyzed datasets.

The resulting list comprises the 14 expected proxy relations given above, and serves as a benchmark for comparison with our approach. While the list is not exhaustive, and, in fact, our methodology uncovered proxy relations beyond this baseline (*cf.* §5.2), it provides a point of reference for our experiments.

Experimental Setup. All tests were executed on a Ubuntu machine (22.04.4 LTS) with an Intel Core i5-8250U CPU and 8GB of RAM, using Popper 4.2.0[1] and SWI-Prolog 9.2.5. The datasets were preprocessed to remove the target labels and isolate each protected attribute individually. Each experiment was performed three times, with a maximum timeout of 20 minutes (1200 seconds) per run. The results were obtained by taking the union of all discovered rules from the three runs.

5.2 Results

We tested each dataset for potential proxy attributes with varying thresholds for recall, precision, and coverage. Table 8 shows the number of detected proxy relations under different threshold settings, ranging from strict (20/90/15) to permissive (5/80/2.5). The number of expected proxy relations is indicated next to each dataset. Additionally, we provide the average recall, precision, and coverage values for the detected relations under each threshold setting, along with the total number of detected relations. For example, in the *Adult* dataset, we identified 3 rules that meet the default thresholds (15/85/10), with an average recall, precision, and coverage of 32/95/23.

We detected the expected proxy relations in nearly all cases, except for the *Bank Marketing* and *COMPAS* datasets. In both instances, the relevant proxy relations were identified, but did not meet our minimum thresholds and were therefore discarded. The remaining proxy relations were detected at varying threshold settings depending on the dataset. For example, in the *Adult* dataset, we detected the *relationship* proxy at the highest thresholds (20/90/15), while in the *Credit Card* dataset, we only detected the *age* proxy at the lowest (5/80/2.5).

We also identified several proxy relations that have not been previously discussed in the literature. For instance, in the *Ricci* dataset, we detected a relation between *combine* and *race*: $\text{race}(X, \textit{white})$:- $\text{combine}(X, Y), \text{lt}(79, Y)$, written in simplified notation for clarity, with recall, precision, and coverage of 25/94/14. Informally, this rule states that an individual is white if their *combine* score is greater than 79. Notably, Le Quy et al. [39] overlook this relation because they categorize *combine* into two groups, $\{< 70, \geq 70\}$, failing to identify the correct splitting point for this attribute.

[1] Available at https://github.com/logic-and-learning-lab/Popper/releases/v4.2.0.

Dataset	Recall / Precision / Coverage (%)			
	20 / 90 / 15	**15 / 85 / 10**	10 / 80 / 5	5 / 80 / 2.5
Adult (2)	1 (61/100/41)	**3 (32/95/23)**	6 (22/93/16)	27 (10/91/8)
KDD (2)	–	**1 (38/87/18)**	2 (25/83/12)	14 (9/86/6)
German Credit (1)	6 (32/98/31)	**10 (26/98/25)**	13 (23/98/21)	17 (19/97/18)
Bank Marketing (1)	–	–	–	–
Credit Card (1)	–	–	–	1 (8/83/4)
COMPAS (2)	–	–	–	–
Ricci (1)	–	**1 (25/94/14)**	2 (27/97/11)	5 (17/95/6)
Students (1)	–	–	3 (11/82/6)	13 (7/87/4)
OULAD (1)	–	–	–	2 (5/90/3)
Lawschool (2)	–	–	–	4 (7/87/6)
Total (14)	7 (36/98/32)	**15 (28/96/23)**	26 (22/94/17)	83 (11/91/9)

Table 8. Number of detected proxy relations per dataset with varying thresholds, along with the average recall, precision, and coverage for the identified relations. The number beside each dataset indicates the number of proxy relations we expected to find. Results for the default approach are highlighted in **bold**.

In addition, we identified proxy relations involving multiple proxy attributes. In the *Adult* dataset, we found a relation between *native-country*/*education* and *race*: race$(X, white)$:- native-country(X, usa), education(X, bsc), with recall, precision, and coverage of 16/92/14. Informally, this rule states that an individual is white if they have a bachelor's degree and are from the US. While prior work [39] found causal relations between *native-country*, *education*, and *race*, their combination in a single proxy relation was previously unknown. In contrast, our approach explicitly identifies the proxy relation.

Finally, Table 9 shows the maximum recall, precision, and coverage values found in the proxy relations detected for each dataset. The corresponding values for the other metrics are provided in parentheses. For example, in the *Credit Card* dataset, the maximum detected recall was 7.65%, with the corresponding rule having 83% precision and 4% coverage. Note that these values may fall below the minimum thresholds, resulting in their exclusion from Table 8 (*e.g.*, in the *Bank Marketing* and *COMPAS* datasets). Furthermore, notice that while the maximum precision is consistently high, it often comes from rules with low recall/coverage. On the other hand, the maximum recall and coverage vary across datasets, and in some cases even fail to meet the thresholds, despite being typically associated with high precision rules.

5.3 Discussion

The results demonstrate that ILP is an effective technique for proxy attribute discovery, having successfully identified most of the proxy attributes discussed

Dataset	Max. Rec. (%)	Max. Prec. (%)	Max. Cov. (%)
Adult	61.15 (100**P**/41**C**)	99.99 (61**R**/41**C**)	41.27 (61**R**/100**P**)
KDD	37.58 (87**P**/18**C**)	99.65 (1**R**/1**C**)	18.00 (38**R**/87**P**)
German Credit	48.39 (98**P**/47**C**)	100.00 (9**R**/9**C**)	46.60 (48**R**/98**P**)
Bank Marketing	6.86 (94**P**/2**C**)	100.00 (0**R**/0**C**)	1.94 (7**R**/94**P**)
Credit Card	7.65 (83**P**/4**C**)	100.00 (0**R**/0**C**)	4.07 (8**R**/83**P**)
COMPAS	4.64 (90**P**/4**C**)	100.00 (0**R**/0**C**)	3.75 (5**R**/90**P**)
Ricci	29.63 (100**P**/7**C**)	100.00 (30**R**/7**C**)	14.41 (25**R**/94**P**)
Students	11.54 (73**P**/6**C**)	100.00 (6**R**/3**C**)	6.16 (10**R**/83**P**)
OULAD	5.37 (82**P**/2**C**)	91.28 (4**R**/2**C**)	2.88 (5**R**/91**P**)
Lawschool	9.12 (86**P**/8**C**)	100.00 (0**R**/0**C**)	7.67 (9**R**/86**P**)

Table 9. Maximum recall (**R**), precision (**P**), and coverage (**C**) values per dataset. The corresponding values of the other metrics are shown in parentheses.

in the literature, as well as some that were previously unknown. This discussion highlights the strengths and limitations of our approach, and how it compares to more traditional causal methods.

Thresholds. The modeling choice that most affects our approach is threshold selection. Stricter thresholds lead to lower detection rates, but have a higher likelihood of detecting true proxy attributes. In turn, more permissive thresholds boost detection rates, but at the risk of introducing false positives. There are no universally optimal thresholds; the appropriate choice depends on the specific dataset under analysis. For instance, in the *Credit Card* dataset, we only identify the *age* proxy at the lowest threshold settings, while in the *German Credit* dataset, nearly every attribute is flagged as a proxy under similar conditions.

In general, we have found that the limiting factors in our approach are typically recall and coverage, rather than precision. This is in line with the results from Tables 8 and 9: while the average recall/coverage tends to drop significantly when we lower the thresholds, the average precision tends to remain constant. Our sensitivity to these metrics can be attributed to two main factors. First, as we have already discussed, the choice of thresholds impacts which rules are discarded and which are not. Second, ILP systems prioritize inferring general rules over highly specific ones. This means that if a proxy relation has a very low coverage (*e.g.*, under 1%), the dataset sample fed into the ILP system may contain too few instances where the rule applies. In such cases, the system will discard the rule, as it is not general enough to induce [13].

Sampling. The use of sampling in the ILP procedure introduces a degree of non-determinism into our methodology. While high recall/coverage rules are consis-

tently detected across different runs, low recall/coverage rules may be overlooked if not enough relevant rows are sampled. We have observed this non-determinism in practice, which is why we ran each experiment multiple times.

Comparison with Causal Methods. The impact of the different metrics varies significantly depending on the discovery method used. Causal approaches tend to be more sensitive to lower precision. For example, Le Quy et al. [39] fail to discover the relation between *marital-status* and *race* in the *Adult* dataset, while we detect it with a precision of approximately 87%. In contrast, our approach is more sensitive to lower coverage, as evidenced by its failure to detect proxies in the *Bank Marketing* and *COMPAS* datasets.

Another difference between the two methods lies in the handling of numeric attributes. As mentioned earlier, Le Quy et al. [39] are able to detect proxy relations involving arithmetic operations (*e.g.*, in the *Ricci* dataset). However, their approach relies on the categorization of numeric attributes, which requires domain knowledge and user effort. As a result, if the user fails to choose the appropriate categories, the system may miss relevant causal relations (*e.g.*, the *combine* proxy in the *Ricci* dataset). In contrast, our approach supports arithmetic relations natively, and does not require any user-guided preprocessing.

Finally, the proxy relations we detect are directly interpretable and can be easily validated. This stands in contrast to causal approaches, as shown by the relation between *native-country*/*education* and *race* discussed earlier. While causal methods may detect such proxy attributes [39], their output provides limited insight into the underlying proxy relations. In contrast, our approach explicitly identifies these relations, making them straightforward to interpret and validate.

6 Conclusions

Ensuring fairness in Machine Learning has become increasingly critical. While efforts to reduce bias often focus on preventing direct discrimination arising from sensitive attributes such as gender or race, ML models may still exhibit indirect discrimination due to the presence of proxy attributes in training datasets.

In this case study, we reported on a methodology for proxy attribute discovery based on Inductive Logic Programming, which we implemented as a Python tool, PADTAI, available online [22]. We have applied this tool to 10 real-world datasets, and detected up to 83 proxy relations, corresponding to 49 potential proxy attributes, in 8 of those. Our evaluation shows that ILP is not only an effective approach for proxy attribute discovery, but can also advance the state of the art by removing the need for prior domain knowledge.

Acknowledgments. We thank the anonymous reviewers for their comments and insightful feedback. This work was supported by Fundação para a Ciência e Tecnologia (FCT) and IAPMEI via a CMU Portugal Dual Degree PhD fellowship (ref. 2024.12581.PRT), INESC-ID multiannual funding (ref. UIDB/50021/2020), and projects RIGA (ref. 2022.03537.PTDC), WebCAP (ref. 2024.07393.IACDC), and SmartRetail (ref. C6632206063-00466847).

References

1. Adebayo, J.A.: FairML: ToolBox for diagnosing bias in predictive modeling. Master's thesis, Massachusetts Institute of Technology (2016)
2. Aïvodji, U., Ferry, J., Gambs, S., Huguet, M.J., Siala, M.: Faircorels, an open-source library for learning fair rule lists. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. p. 4665–4669. CIKM '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3459637.3481965
3. Angwin, J., Larson, J., Kirchner, L., Mattu, S.: Machine bias (May 2016), https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
4. Berk, R., Heidari, H., Jabbari, S., Kearns, M., Roth, A.: Fairness in criminal justice risk assessments: The state of the art. Sociological Methods & Research **50**(1), 3–44 (2021). https://doi.org/10.1177/0049124118782533
5. Bird, S., Kenthapadi, K., Kiciman, E., Mitchell, M.: Fairness-aware machine learning: Practical challenges and lessons learned. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. p. 834–835. WSDM '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3289600.3291383
6. Breiman, L.: Random forests. Machine Learning **45**(1), 5–32 (Oct 2001). https://doi.org/10.1023/A:1010933404324
7. Calmon, F.P., Wei, D., Vinzamuri, B., Ramamurthy, K.N., Varshney, K.R.: Optimized pre-processing for discrimination prevention. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 3995–4004. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
8. Census-Income (KDD). UCI Machine Learning Repository (2000). https://doi.org/10.24432/C5N30T
9. Chouldechova, A.: Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. Big Data **5**(2), 153–163 (2017). https://doi.org/10.1089/big.2016.0047
10. Chouldechova, A., Roth, A.: A snapshot of the frontiers of fairness in machine learning. Commun. ACM **63**(5), 82–89 (Apr 2020). https://doi.org/10.1145/3376898
11. Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., Huq, A.: Algorithmic decision making and the cost of fairness. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 797–806. KDD '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3097983.3098095
12. Cortez, P., Silva, A.: Using data mining to predict secondary school student performance. In: Proceedings of 5th Annual Future Business Technology Conference (Apr 2008)
13. Cropper, A., Dumančić, S.: Inductive logic programming at 30: A new introduction. Journal of Artificial Intelligence Research **74**, 765–850 (Jun 2022). https://doi.org/10.1613/jair.1.13507
14. Cropper, A., Morel, R.: Learning programs by learning from failures. Machine Learning **110**(4), 801–856 (Apr 2021). https://doi.org/10.1007/s10994-020-05934-z
15. Datta, A., Tschantz, M.C., Datta, A.: Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. Proceedings on Privacy Enhancing Technologies **2015**(1), 92–112 (Apr 2015). https://doi.org/10.1515/popets-2015-0007

16. Datta, A., Fredrikson, M., Ko, G., Mardziel, P., Sen, S.: Proxy non-discrimination in data-driven systems (Jul 2017). https://doi.org/10.48550/arXiv.1707.08120
17. Datta, A., Fredrikson, M., Ko, G., Mardziel, P., Sen, S.: Use privacy in data-driven systems: Theory and experiments with machine learnt programs. In: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. p. 1193–1210. CCS '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3133956.3134097
18. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference. p. 214–226. ITCS '12, Association for Computing Machinery, New York, NY, USA (2012). https://doi.org/10.1145/2090236.2090255
19. Evans, R., Grefenstette, E.: Learning explanatory rules from noisy data (Nov 2017). https://doi.org/10.48550/arXiv.1711.04574
20. Friedler, S.A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E.P., Roth, D.: A comparative study of fairness-enhancing interventions in machine learning. In: Proceedings of the Conference on Fairness, Accountability, and Transparency. p. 329–338. FAT* '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3287560.3287589
21. Galhotra, S., Brun, Y., Meliou, A.: Fairness testing: testing software for discrimination. In: Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering. p. 498–510. ESEC/FSE 2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3106237.3106277
22. Gonçalves, R., Gouveia, F., Lynce, I., Fragoso Santos, J.: Proxy Attribute Discovery in Machine Learning Datasets via Inductive Logic Programming (Artifact) (2025). https://doi.org/10.5281/zenodo.14618504
23. Gonçalves, R., Gouveia, F., Lynce, I., Fragoso Santos, J.: Proxy Attribute Discovery in Machine Learning Datasets via Inductive Logic Programming (Extended Version) (2025). https://doi.org/10.5281/zenodo.14757542
24. Hajian, S., Domingo-Ferrer, J.: Direct and Indirect Discrimination Prevention Methods, pp. 241–254. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). https://doi.org/10.1007/978-3-642-30487-3_13
25. Hajian, S., Domingo-Ferrer, J.: A methodology for direct and indirect discrimination prevention in data mining. IEEE Transactions on Knowledge and Data Engineering **25**(7), 1445–1459 (2013). https://doi.org/10.1109/TKDE.2012.72
26. Hardt, M., Price, E., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 29, pp. 3315–3323. Curran Associates, Inc. (2016)
27. Hofmann, H.: Statlog (German Credit Data). UCI Machine Learning Repository (1994). https://doi.org/10.24432/C5NC77
28. Holmes, D.E., Jain, L.C. (eds.): Innovations in Bayesian networks, Studies in Computational Intelligence, vol. 156. Springer, Berlin, Germany (Sep 2008)
29. Holstein, K., Wortman Vaughan, J., Daumé, H., Dudik, M., Wallach, H.: Improving fairness in machine learning systems: What do industry practitioners need? In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems. p. 1–16. CHI '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3290605.3300830
30. Ignatiev, A., Cooper, M.C., Siala, M., Hebrard, E., Marques-Silva, J.: Towards formal fairness in machine learning. In: Simonis, H. (ed.) Principles and Practice of Constraint Programming. pp. 846–867. Springer International Publishing, Cham (2020)

31. Kilbertus, N., Rojas-Carulla, M., Parascandolo, G., Hardt, M., Janzing, D., Schölkopf, B.: Avoiding discrimination through causal reasoning. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 656–666. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
32. Kleinberg, J.: Inherent trade-offs in algorithmic fairness. In: Abstracts of the 2018 ACM International Conference on Measurement and Modeling of Computer Systems. p. 40. SIGMETRICS '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3219617.3219634
33. Kohavi, R.: Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. p. 202–207. KDD'96, AAAI Press (1996)
34. Krzywinski, M., Altman, N.: Classification and regression trees. Nature Methods **14**(8), 757–758 (Aug 2017). https://doi.org/10.1038/nmeth.4370
35. Kusner, M., Loftus, J., Russell, C., Silva, R.: Counterfactual fairness. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. p. 4069–4079. NIPS'17, Curran Associates Inc., Red Hook, NY, USA (2017)
36. Kuzilek, J., Hlosta, M., Zdrahal, Z.: Open university learning analytics dataset. Scientific Data **4**(1), 170171 (Nov 2017). https://doi.org/10.1038/sdata.2017.171
37. Law, M., Russo, A., Broda, K.: Inductive learning of answer set programs. In: Logics in Artificial Intelligence. pp. 11–325. Springer International Publishing, Cham (2014)
38. Law, M., Russo, A., Broda, K.: The ILASP system for learning answer set programs (2015), www.ilasp.com
39. Le Quy, T., Roy, A., Iosifidis, V., Zhang, W., Ntoutsi, E.: A survey on datasets for fairness-aware machine learning. WIREs Data Mining and Knowledge Discovery **12**(3), e1452 (Mar 2022). https://doi.org/10.1002/widm.1452
40. Madras, D., Creager, E., Pitassi, T., Zemel, R.: Fairness through causal awareness: Learning causal latent-variable models for biased data. In: Proceedings of the Conference on Fairness, Accountability, and Transparency. p. 349–358. FAT* '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3287560.3287564
41. Martins, R., Chen, J., Chen, Y., Feng, Y., Dillig, I.: Trinity: an extensible synthesis framework for data science. Proc. VLDB Endow. **12**(12), 1914–1917 (Aug 2019). https://doi.org/10.14778/3352063.3352098
42. Moro, S., Cortez, P., Rita, P.: A data-driven approach to predict the success of bank telemarketing. Decision Support Systems **62**, 22–31 (2014). https://doi.org/10.1016/j.dss.2014.03.001
43. Muggleton, S.H., Lin, D., Tamaddoni-Nezhad, A.: Meta-interpretive learning of higher-order dyadic datalog: Predicate invention revisited. Machine Learning **100**(1), 49–73 (Jul 2015). https://doi.org/10.1007/s10994-014-5471-y
44. Mukerjee, A., Biswas, R., Deb, K., Mathur, A.P.: Multi–objective evolutionary algorithms for the risk–return trade–off in bank loan management. International Transactions in Operational Research **9**(5), 583–597 (2002). https://doi.org/10.1111/1475-3995.00375
45. Nabi, R., Shpitser, I.: Fair inference on outcomes. Proceedings of the AAAI Conference on Artificial Intelligence **32**(1) (Apr 2018). https://doi.org/10.1609/aaai.v32i1.11553
46. Orvalho, P., Terra-Neves, M., Ventura, M., Martins, R., Manquinho, V.: Squares : A sql synthesizer using query reverse engineering. Proc. VLDB Endow. **13**(12), 2853–2856 (2020). https://doi.org/10.14778/3352063.3352098

47. Perry, W.L., McInnis, B., Price, C.C., Smith, S.C., Hollywood, J.S.: Predictive Policing: The Role of Crime Forecasting in Law Enforcement Operations. RAND Corporation (2013)
48. Srinivasan, A.: The Aleph manual (2001), https://www.cs.ox.ac.uk/activities/programinduction/Aleph/aleph.html
49. Supreme Court of the United States: Ricci v. DeStafano. 557 U.S. 557 (2009)
50. Tramèr, F., Atlidakis, V., Geambasu, R., Hsu, D., Hubaux, J.P., Humbert, M., Juels, A., Lin, H.: Fairtest: Discovering unwarranted associations in data-driven applications. In: 2017 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 401–416 (2017). https://doi.org/10.1109/EuroSP.2017.29
51. Verma, S., Rubin, J.: Fairness definitions explained. In: Proceedings of the International Workshop on Software Fairness. p. 1–7. FairWare '18, Association for Computing Machinery, New York, NY, USA (2018). https://doi.org/10.1145/3194770.3194776
52. Wang, C., Cheung, A., Bodik, R.: Synthesizing highly expressive sql queries from input-output examples. In: Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation. p. 452–466. PLDI 2017, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3062341.3062365
53. Warner, R., Sloan, R.H.: Making artificial intelligence transparent: Fairness and the problem of proxy variables. Criminal Justice Ethics **40**(1), 23–39 (2021). https://doi.org/10.1080/0731129X.2021.1893932
54. Wightman, L.F.: LSAC national longitudinal bar passage study. In: LSAC Research Report Series (1998), https://eric.ed.gov/?id=ED469370
55. Yeh, I.C., hui Lien, C.: The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients. Expert Systems with Applications **36**(2, Part 1), 2473–2480 (2009). https://doi.org/j.eswa.2007.12.020