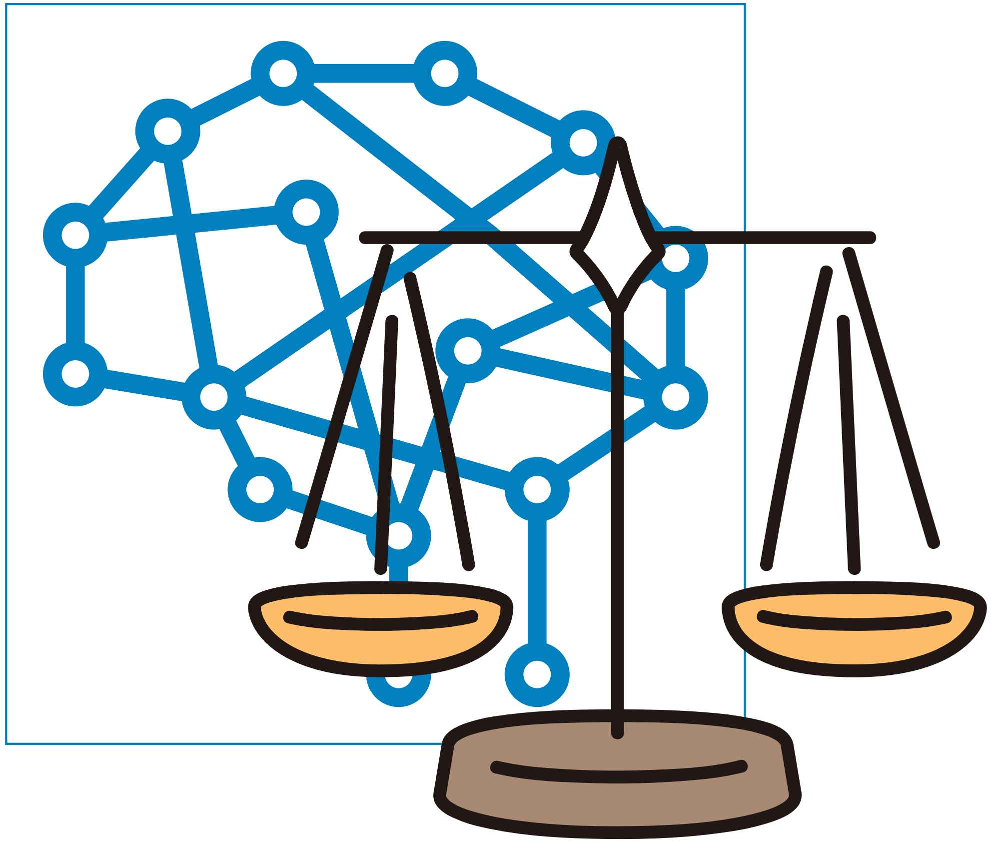
Explainable
Global Fairness Verification of Tree-Based Classifiers

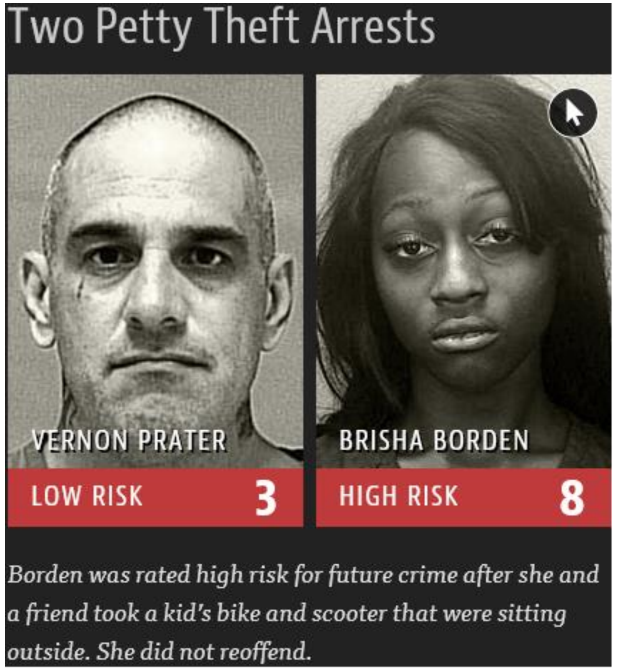


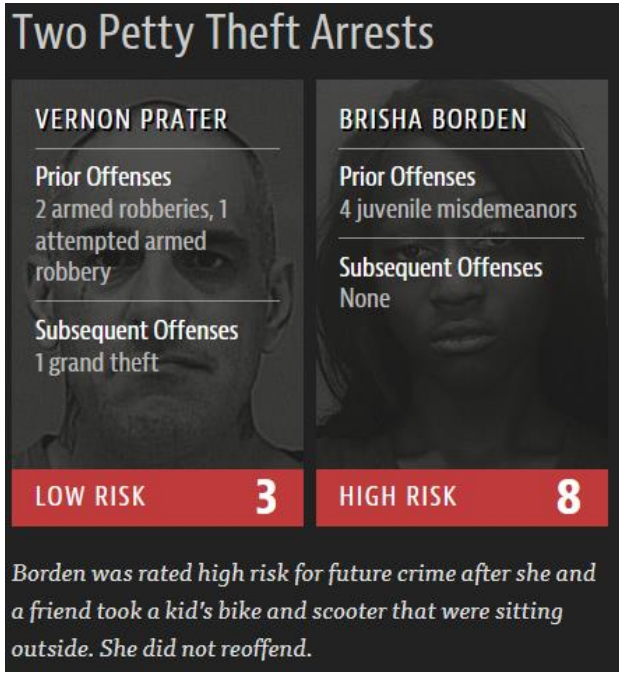
Stefano Calzavara, **Lorenzo Cazzaro**, Claudio Lucchese, Federico Marcuzzi

Ca' Foscari University of Venice

Is Machine Learning Unfair?

Example: Machine Learning (ML) used to predict recidivity in USA*

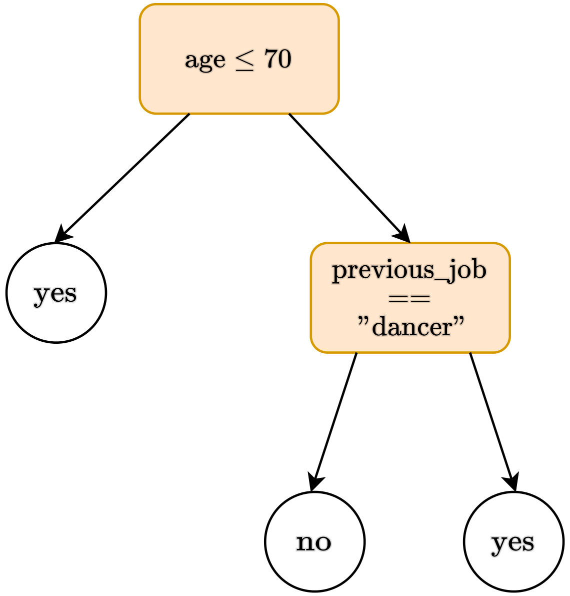




Non-recidivist black people were twice as likely to be labelled high risk than non-recidivist white people.

*https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

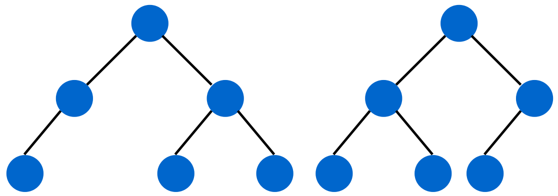
ML classifier for
loan requests



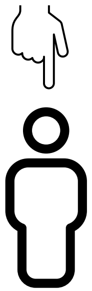
Can We Provide Fairness Guarantees about the behaviour of a ML Classifier?

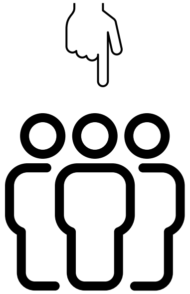
Fairness Guarantees

Local properties: predicate over an **instance** or a **specific test set** of instances.

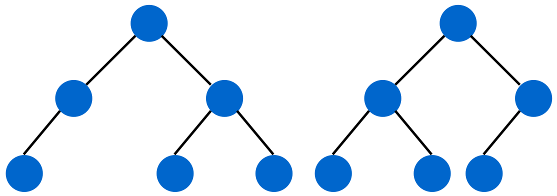


It's fair on





Global properties: they predicate over **a (continuous and unbounded) subset of instances**.



It's fair on people described by

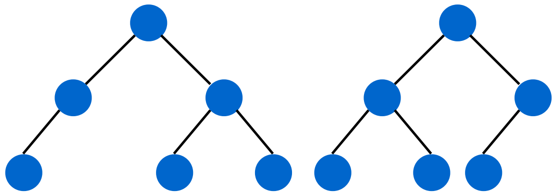
age > 70 and job = «prof»

Research Problem

There are not proposals in the literature to verify **global fairness** for tree-based classifiers…

Problem:

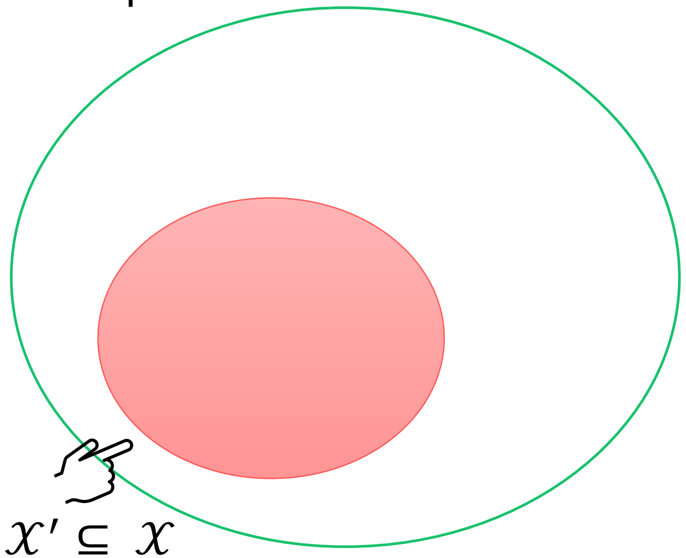
Tree based classifier T



Set of sensitive features S

{sex, race}

Feature space $\mathcal{X}$



In this subset $\mathcal{X}' \subseteq \mathcal{X}$
the ML model is fair

Our Contribution

We present a new approach to the global fairness verification of tree-based classifiers.

Our analysis synthesizes a set of sufficient conditions for fairness:

Conditions as **logical formulas**

Global conditions: predicate over the entire feature space

{age > 70 and job = «prof»,
credit_account < 4000 and age < 35 and housing = «rent»}

Explainable formulas: readily understandable

Our verification approach is **proved:**

- **Sound**: fairness is certified for any instance satisfying some formulas.
- **Complete**: the formulas can characterize all the instances where the classifier is fair.

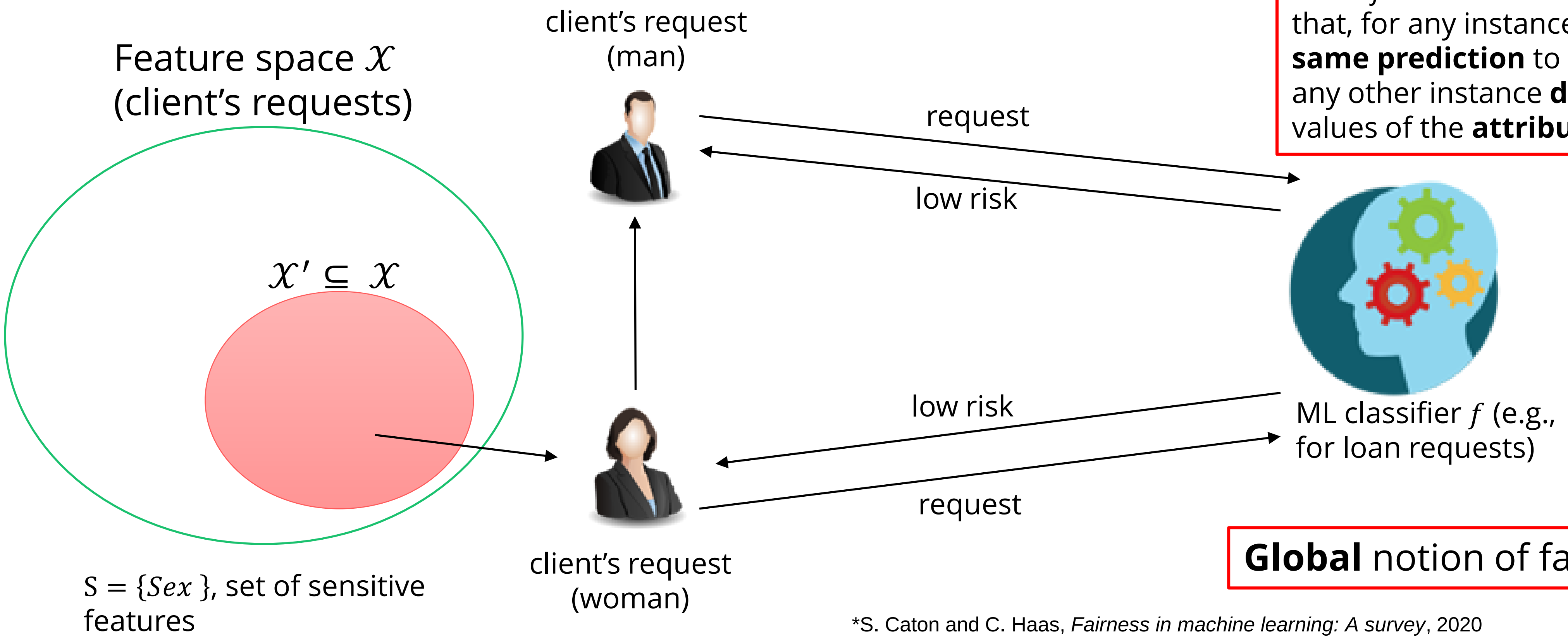
Considered Fairness Property

Causal Discrimination

We focus on **individual fairness***: give similar predictions to similar individuals.

In particular, we focus on **lack of causal discrimination****.

Lack of causal discrimination requires that, for any instance in $\mathcal{X}'$, f gives the **same prediction** to **the instance** and any other instance **differing** only for the values of the **attributes in $\mathbf{S}$**!



Global notion of fairness on $\mathcal{X}'$!

*S. Caton and C. Haas, *Fairness in machine learning: A survey*, 2020

**S. Galhotra, Y. Brun, and A. Meliou, *Fairness testing: testing software for discrimination*, ESEC/FSE 2017

Lack of causal discrimination and Stability

Lack of causal discrimination is connected to the **stability*** property:

- Suppose to have an instance $\vec{x} \subseteq \mathcal{X}$ and a set of possible adversarial manipulations $\mathrm{A}(\vec{x})$;
- f is *stable* on $\vec{x}$ if and only if $\forall \vec{z} \in A(\vec{x})$: $f(\vec{z}) = f(\vec{x})$. It's a **local** property.
- Lack of causal discrimination: changes to the sensitive features in S must not affect the predictions of the classifier.

*F. Ranzato and M. Zanella, *Abstract interpretation of decision tree ensemble classifiers,* AAAI 2020

The Synthesis Algorithm

Data-independent Stability Analysis

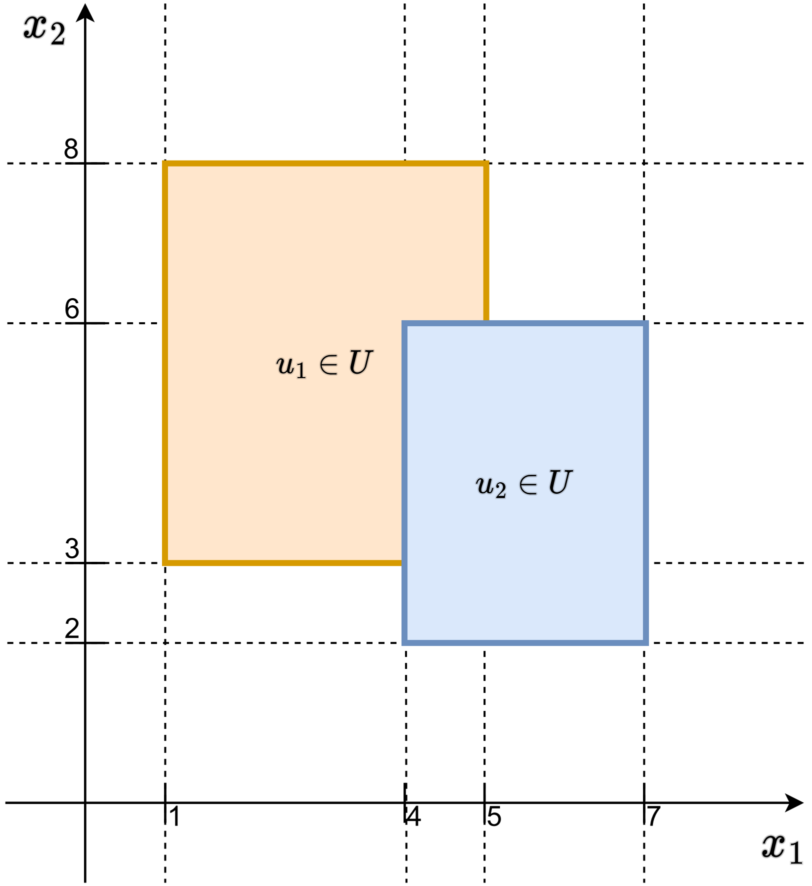
For tree-based models, exploit a

Data-Independent Stability Analysis algorithm (DISA)*:

- **Input**: tree-based model T and the definition of an attacker $A(\vec{x})$ (e.g., she manipulates the sensitive features of $\vec{x}$).
- **Output**: **set of hyper-rectangles $\boldsymbol{U}$** that **over-approximates** the subsets of the feature space on which $\boldsymbol{T}$ **is unstable.**

⬇

$\boldsymbol{T}$ **might perform causal discrimination** on these subsets of the feature space!



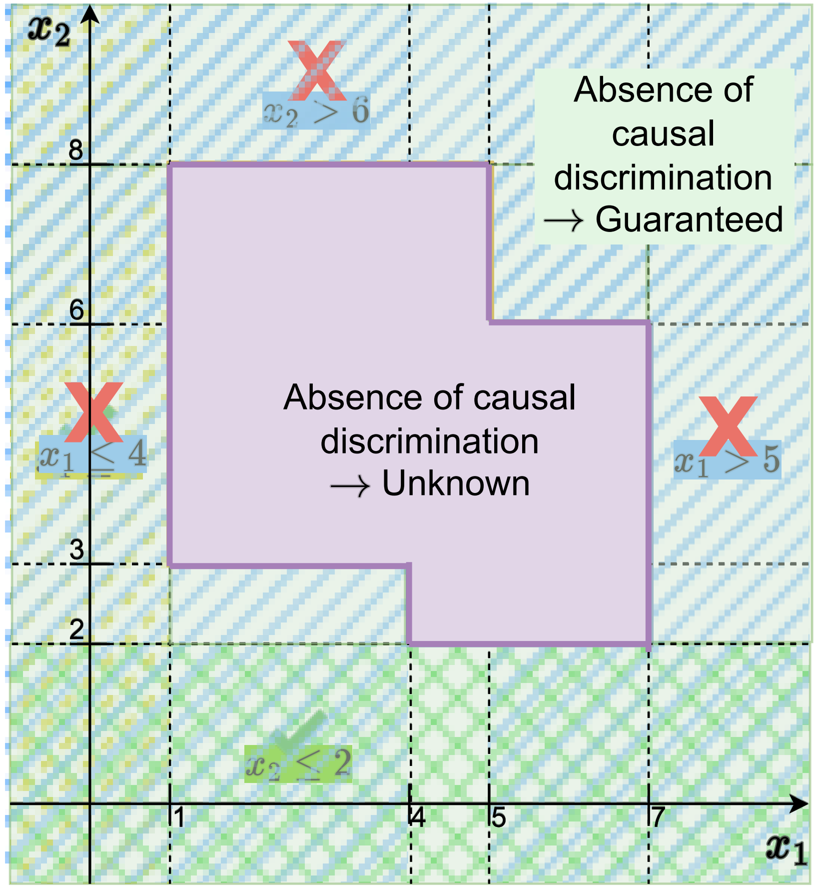
*S. Calzavara, L. Cazzaro, C. Lucchese, F. Marcuzzi, S. Orlando, *Beyond Robustness: Resilience Verification of Tree-Based Classifiers*, Computers&Security (2022)

Synthesis algorithm – Generate conditions

The synthesizer generates formulas **predicating on instances outside** the hyper-rectangles, i.e., where the ML classifier presents lack of causal discrimination!

The synthesis algorithm takes in input the set of hyper-rectangles U from the DISA:

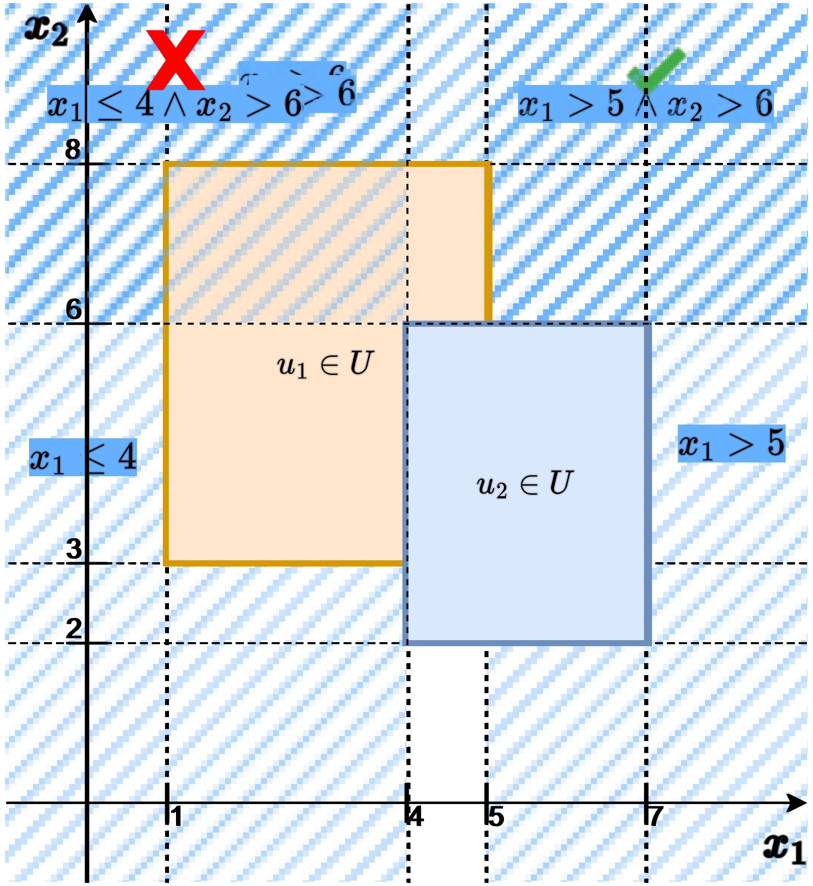
- It starts generating formulas with a predicate on **one single feature**.
- Check if some formulas of complexity 1 predicate only over instances outside the hyper-rectangles. Example: $x_1 \leq 1$.
- Some formulas may identify subsets of the feature space **that intersect some hyper-rectangles**.



Synthesis algorithm – Generate longer conditions

After the initial generation:

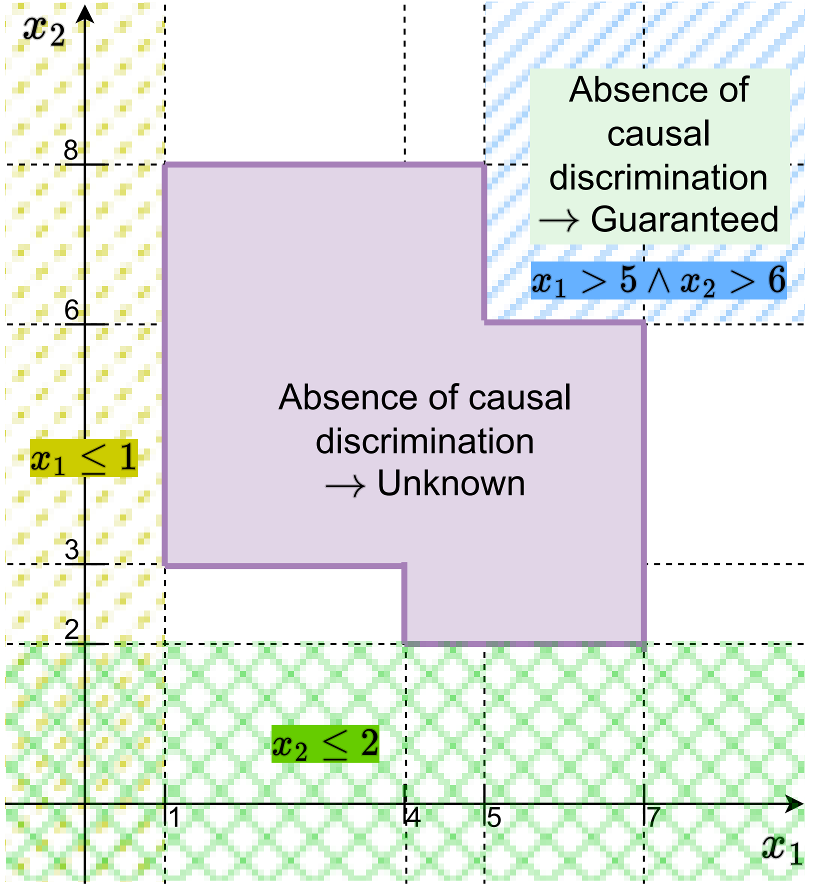
- Formulas that intersect hyper-rectangles are combined togheter to generate longer conditions. Example: $x_1 > 5 \ \wedge x_2 > 6$.
- **Check the new conditions** against the hyper-rectangles.
- **Continue performing the combination-check steps** until a stopping criteria is met (e.g., number of iterations).
- At the iteration k, formulas of complexity k are generated.



Synthesis algorithm - Summary

The synthesizer is an **iterative algorithm** that:

- Generates **increasingly complex sufficient conditions** ensuring lack of causal discrimination.
- The conditions predicate on instances outside the hyper-rectangles, i.e., where the ML classifier shows lack of causal discrimination.
- First iterations → formulas **easy to understand** (**explainable**).
- The more computational resources are available, the more complex conditions may be generated.
- **Sound and Complete**



Experimental Evaluation

Experimental evaluation

Setting:

- ML classifier: Random Forest.
- Adult dataset (+ other two datasets)
- D_{test} $\rightarrow$ test set.
- D_{rand} $\rightarrow$ set of 100000 random instances → larger view of the feature space.
- The set of sensitive attributes is $S = \{sex\}$.

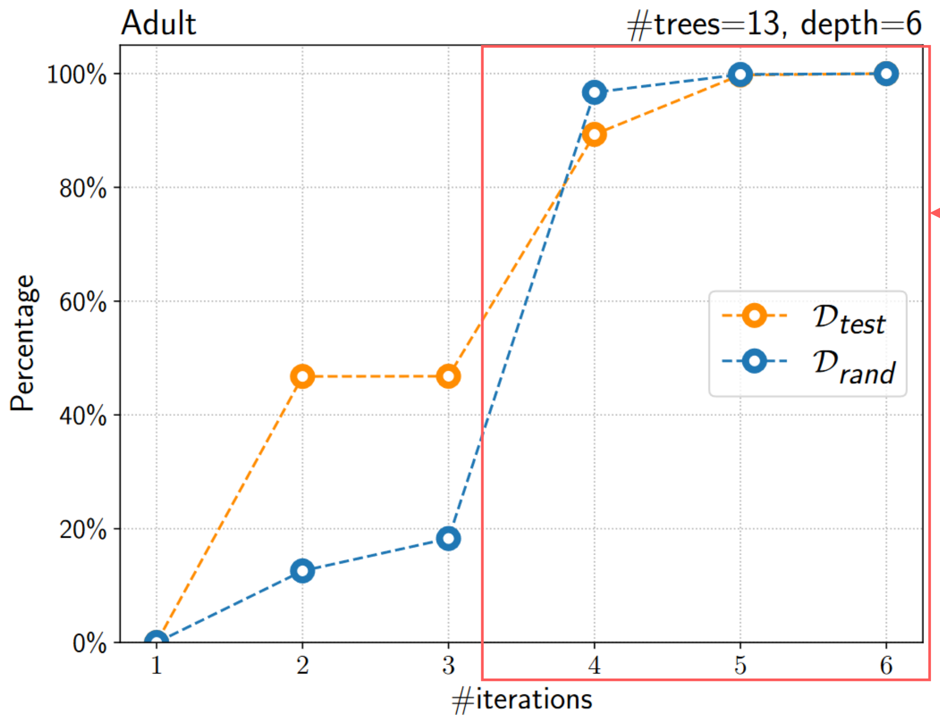
Evaluation along three different axes:

- Precision of the analysis (see the full paper for details).
- **Explainability of the generated conditions**.
- Performance evaluation (see the full paper for details).

Experimental evaluation - Coverage

Question: how much is the subset of the feature space outside the hyper-rectangles (i.e., where the ML model is fair) covered by the conditions?

Method: we compute the **percentage of instances** covered by the fairness conditions.

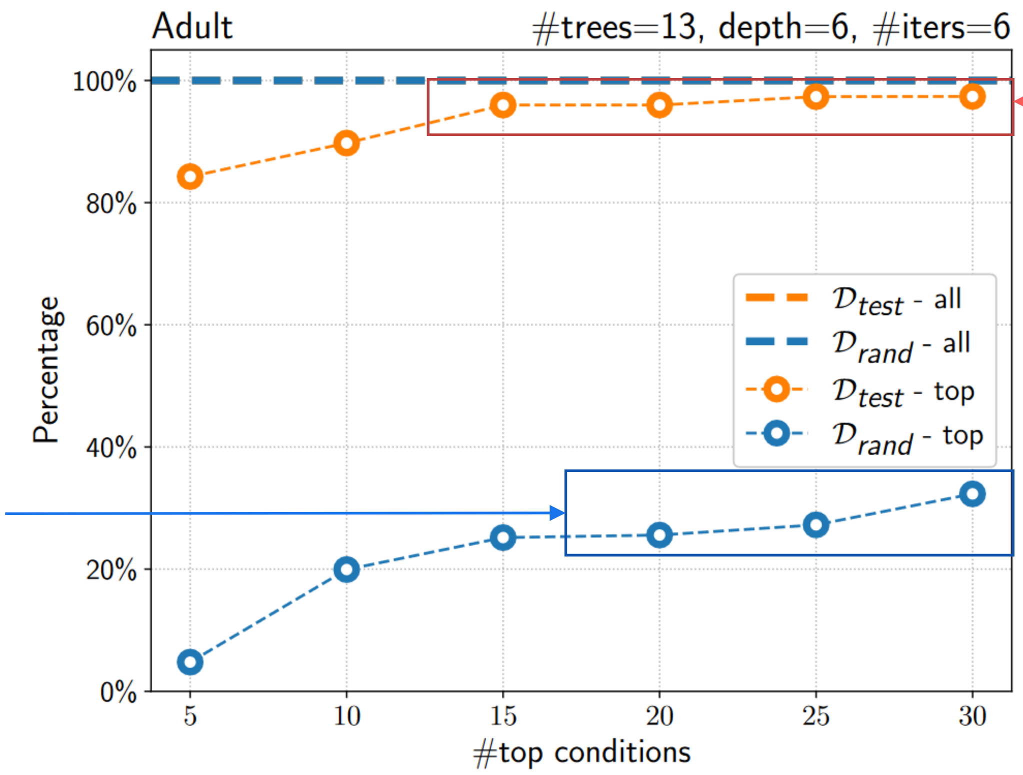


Answer: short logical formulas are expressive enough to establish useful fairness proofs!

Problem: *the number of generated formulas may increase significantly*, e.g., more than 300 formulas after 5 iterations.

Experimental evaluation – Top k formulas

Question: is a subset of the generated formulas sufficient to cover a «large» part of the subset of the feature space on which the ML model is fair?
Method: we select the set of the top k most important formulas using a greedy strategy.



A small number of formulas is sufficient to characterize the fairness guarantees on D_test.

More formulas are needed to cover synthetic instances in D_rand.

Answer: the number of important formulas **is relatively small** in practice!

Conclusion

Is ML unfair? Maybe, but **we are able to produce guarantees ensuring lack of causal discrimination for the ML classifier!**

Our analysis synthesizes a set of sufficient conditions for fairness:

Conditions as **logical formulas**

Global conditions: predicate over the entire feature space

{age > 70 and job = «prof»,
credit_account < 4000 and age < 35 and housing = «rent»}

Explainable conditions: readily understandable

Our analysis is precise, explainable and reasonably efficient (details in the full paper)!

Lorenzo Cazzaro
Second-year Ph.D. student in Computer Science

@LorenzoCazz

lorenzo.cazzaro@unive.it

Ca' Foscari University of Venice

Lorenzo Cazzaro

LorenzoCazzaro

Thank you! Questions?