

# LOGIC-BASED EXPLAINABLE ARTIFICIAL INTELLIGENCE

---

Joao Marques-Silva

ICREA & Univ. Lleida, Catalunya, Spain

ESSLLI, Bochum, Germany, July 2025

# Brief bio sketch...

- Research:
  - Boolean satisfiability (SAT) – clause learning used in CDCL
  - Optimization & inconsistency
  - Quantification, model checking, uses of CEGAR
  - Logic encodings, description logics
  - Model-based diagnosis
  - ...
  - Rigorous explainability – new topic since 2019
- Career:
  - Research Professor – ICREA & Univ. Lleida, Catalunya, [Spain](#)
  - Directeur de Recherche – CNRS & IRIT, Toulouse, [France](#)
  - SFI Stokes Prof. of CS – Univ. College Dublin, [Ireland](#)
  - Full Professor of CS – Univ. Southampton, [United Kingdom](#)
  - Full Professor of CS – Univ. Lisbon – FCUL, [Portugal](#)
  - Full Professor of CS – Univ. Lisbon – IST, [Portugal](#)
  - ...
  - PhD at Univ. Michigan, Ann Arbor, MI, [USA](#) (1995)

# Lecture 01

# Remarkable ML successes – e.g. LLM ChatGPT for knowledge

Explain Sudoku puzzles. Give me an example.  
Thanks!



## What is Sudoku?

Sudoku is a logic-based number placement puzzle. It consists of a 9x9 grid divided into nine 3x3 subgrids, also called "regions." The goal is to fill the grid with numbers from 1 to 9 such that:

1. Each row contains the numbers 1 through 9, without repetition.
2. Each column contains the numbers 1 through 9, without repetition.
3. Each 3x3 subgrid contains the numbers 1 through 9, without repetition.

The puzzle starts with some numbers pre-filled, and your task is to fill in the blanks following the rules above. The difficulty of the puzzle depends on how many numbers are pre-filled and their arrangement.

## Example Puzzle

Here's an example of a Sudoku puzzle:

5	3		7					
6			1	9	5			
	9	8				6		
8			6			3		
4			8	3		1		
7			2			6		
6			2	8				
			4	1	9	5		
			8			7	9	

Copy

## How to Solve It

1. **Start with easy numbers:** Look for rows, columns, or subgrids where most numbers are already filled in. Use the process of elimination to determine the missing numbers.
2. **Use pencil marks:** If you're unsure, pencil in all possible candidates for a blank cell and narrow them down as you progress.
3. **Keep the rules in mind:** Continuously check the rows, columns, and subgrids to ensure you don't duplicate any numbers.
4. **Look for patterns:** Advanced strategies like "naked pairs" and "hidden singles" may help solve more challenging puzzles.

Would you like help solving this example or generating a new one?



# Remarkable ML successes – e.g. LLM ChatGPT for programming

code for selecting columns of pandas  
dataframe matching some value

To select columns in a pandas DataFrame that match a specific value, you can use a combination of boolean indexing and `.any()` or `.all()` methods. Here's a simple example:

## Example Code

Let's assume you want to select all columns containing a specific value (say, 42).

```
python Copy  
  
import pandas as pd  
  
# Example DataFrame  
data = {  
    'A': [10, 42, 10],  
    'B': [5, 5, 42],  
    'C': [42, 1, 42],  
    'D': [2, 2, 2]  
}  
df = pd.DataFrame(data)  
  
# Specify the value to match  
value_to_match = 42  
  
# Selecting columns that contain the specified value  
matching_columns = df.columns[(df == value_to_match).  
  
# Filter the DataFrame to include only matching column  
df_matching = df[matching_columns]  
  
print(df_matching)
```

## Explanation

1. `(df == value_to_match)` creates a DataFrame of booleans where `True` indicates cells that match the specified value.
2. `.any()` is used on columns (default `axis=0`) to determine if any cell in each column matches the value.
3. `df.columns[...]` is used to filter the columns to include only those with matching values.
4. `df[matching_columns]` creates a new DataFrame with only the matching columns.

## Output

For the example given above, if `value_to_match` is 42, the output will be:

Copy

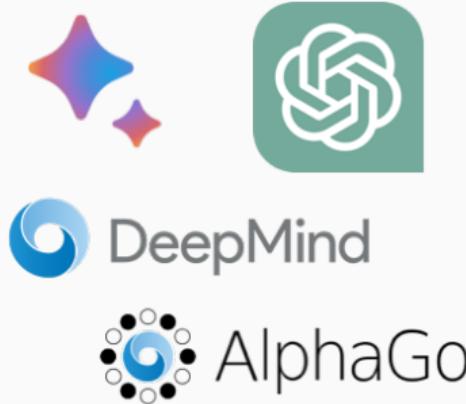
	A	B	C
0	10	5	42
1	42	5	1
2	10	42	42

Only columns A, B, and C are returned, as they contain the value 42.

# Recent & ongoing ML successes

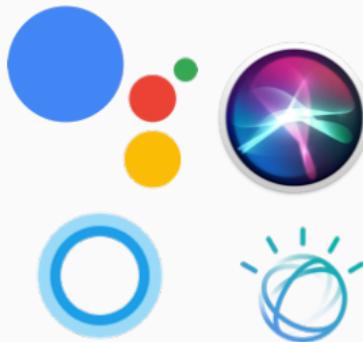
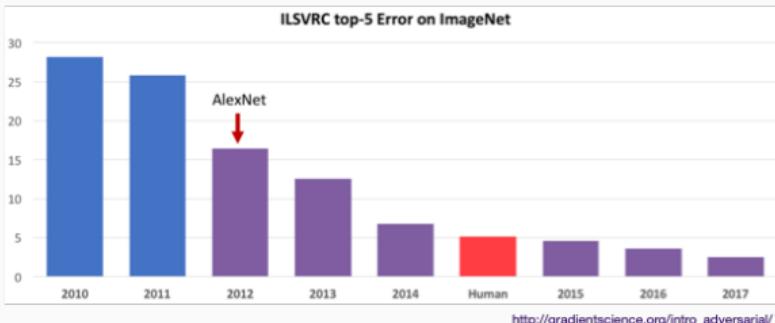


<https://en.wikipedia.org/wiki/Waymo>



AlphaGo Zero & Alpha Zero

## Image & Speech Recognition



[https://fr.wikipedia.org/wiki/Pepper\\_\(robot\)](https://fr.wikipedia.org/wiki/Pepper_(robot))

## Troubling ML issues – e.g. LLM ChatGPT for puzzles (Jan'25)

Count the number of occurrences of the letter 'L' in the word 'LOLLAPALOOZA'

To count the occurrences of the letter 'L' in the word **LOLLAPALOOZA**, we can visually inspect it:

The word is: **LOLLAPALOOZA**.

Now, let's count the 'L's:

- There are **3 L's** in the word.

**Answer:**

The number of occurrences of the letter 'L' is **3**.

<https://arxiv.org/html/2405.19616v2>

# Troubling ML issues – e.g. LLM ChatGPT for puzzles (Jan'25)

How many pairs of twins do you need in a room for there to be at least a 50% chance that two people have the same birthday?

This question is a variation of the famous **Birthday Problem**, where the goal is to determine how many people are needed in a room for at least a 50% chance of two sharing the same birthday.

Here, you're dealing with **pairs of twins**. A pair of twins share the same birthday, so the effective number of unique birthdays to consider is reduced compared to regular individuals.

**Step-by-Step Solution:**

- 1. Revisiting the Regular Birthday Problem:** For a 50% chance of two people sharing a birthday in a room of unrelated individuals, you need about **23 people**. This assumes 365 equally likely days for a birthday.
- 2. Impact of Twins:**
  - Each pair of twins reduces the diversity of birthdays because both members of the pair share a single birthday.
  - Therefore, each pair of twins counts as **1 effective individual** in terms of the Birthday Problem.
- 3. Calculation for Pairs of Twins:**
  - For a 50% chance of a shared birthday, you need **23 effective individuals**.
  - Since each pair of twins is equivalent to **1 effective individual**, you need **23 pairs of twins** to achieve the same probability.

**Final Answer:**

You need **23 pairs of twins** in the room for at least a 50% chance that two people have the same birthday.

↓

<https://arxiv.org/html/2405.19616v2>

# Can we trust ML models?

- Accuracy in training/test data
- Complex ML models are **brittle**
  - Extensive work on finding adversarial examples
  - Extensive work on learning robust ML models
- More recently, complex ML models **hallucinate**
  - LLMs also fail at basic reasoning tasks
- Need to validate operation of ML model, with rigor
  - Explanations; robustness; verification

## ML models are brittle – adversarial examples



## ML models are brittle – adversarial examples



## ML models are brittle – adversarial examples



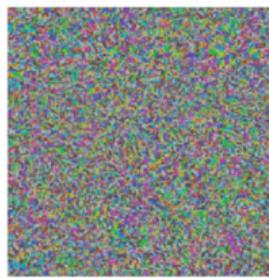
# Adversarial examples can be very problematic

Original image



+ 0.04 ×

Adversarial noise



Adversarial example



Dermatoscopic image of a benign melanocytic nevus, along with the diagnostic probability computed by a deep neural network.



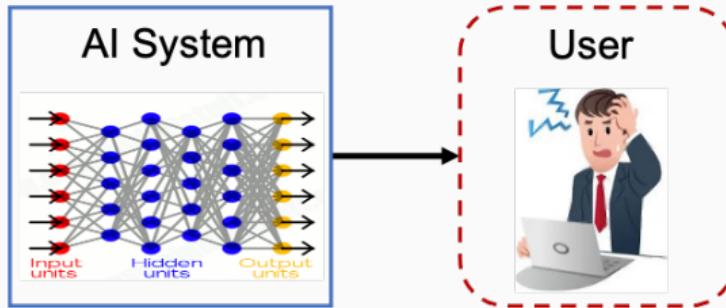
Perturbation computed by a common adversarial attack technique.

Combined image of nevus and attack perturbation and the diagnostic probabilities from the same deep neural network.



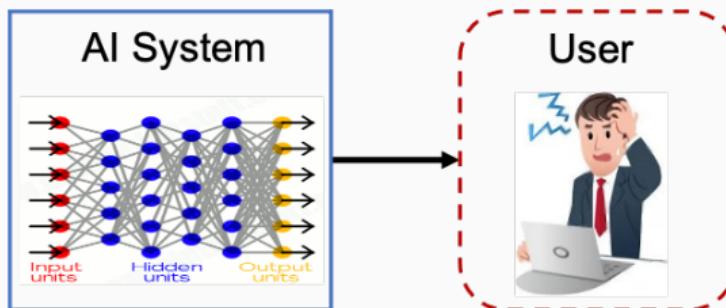
Finlayson et al., Nature 2019

# The road to trust – eXplainable AI (XAI)



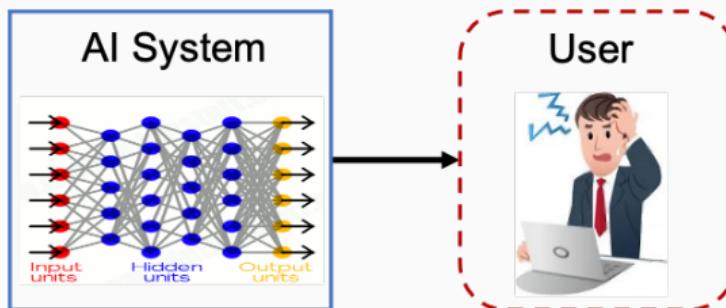
- Complex ML models are **opaque**
- Goal of **XAI**: **to help humans understand ML models**

# The road to trust – eXplainable AI (XAI)



- Complex ML models are **opaque**
- Goal of XAI: **to help humans understand ML models**
- How to XAI?
  - Answer to **Why (the prediction)?** questions
  - Answer to **Why Not (some other prediction)?** questions

# The road to trust – eXplainable AI (XAI)



- Complex ML models are **opaque**
- Goal of XAI: **to help humans understand ML models**
- How to XAI?
  - Answer to **Why (the prediction)?** questions
  - Answer to **Why Not (some other prediction)?** questions
- What about **rigor**?
  - Sub-symbolic (i.e. non-symbolic) methods?
  - Symbolic (i.e. logic-based) methods?

# Importance of XAI

REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 27 April 2016

on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

(Text with EEA relevance)

European Union regulations on algorithmic decision-making  
and a “right to explanation”

Bryce Goodman,<sup>1\*</sup> Seth Flaxman,<sup>2</sup>

We summarize the potential impact that the European Union's new General Data Protection Regulation will have on the routine use of machine-learning algorithms. Slated to take effect as law across the European Union in 2018, it will place restrictions on automated individual decision making (that is, algorithms that make decisions based on user-level predictors) that "significantly affect" users. When put into practice, the law may also effectively create a right to explanation, whereby a user can ask for an explanation of an algorithmic decision that significantly affects them. We argue that while this law may pose large challenges for industry, it highlights opportunities for computer scientists to take the lead in designing algorithms and evaluation frameworks that avoid discrimination and enable explanation.

Proposal for a

REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE  
(ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION  
LEGISLATIVE ACTS

## Explainable Artificial Intelligence (XAI)



David Gunning  
DARPA/I2O  
Program Update November 2017



© DARPA

European Commission > Strategy > Digital Single Market > Reports and studies >

Digital Single Market

REPORT / STUDY | 8 April 2019

Ethics guidelines for trustworthy AI

# Importance of XAI

European Union regulation  
and a “right to explanation”

Bryce Goodman

■ We summarize the potential impact that the European Union's new General Data Protection Regulation will have on the routine use of machine-learning algorithms. Slated to take effect as law across the European Union in 2018, it will place restrictions on automated individual decision making (that is, algorithms that make decisions based on user-level predictors) that "significantly affect" users. When put into practice, the law may also effectively create a right to explanation, whereby a user can ask for an explanation of an algorithmic decision that significantly affects them. We argue that while this law may pose large challenges for industry, it highlights opportunities for computer scientists to take the lead in designing algorithms and evaluation frameworks that avoid discrimination and enable explanation.

REGULATION (EU) 2016/679

In order to trust deployed AI systems, on the part of the user and the provider, we must not only improve their robustness,<sup>5</sup> but also develop ways to make their reasoning intelligible. Intelligibility will help us spot AI that makes mistakes due to distributional drift or incomplete representations of goals and features. Intelligibility will also facilitate control by humans in increasingly common collaborative human/AI teams. Furthermore, intelligibility will help humans learn from AI. Finally, there are legal reasons to want intelligible AI, including the European GDPR and a growing need to assign liability when AI errs.

Weld & Bansal, CACM, Jun'19  
Due November 2017

THE COUNCIL

(Data protection and on the free movement of data Regulation)

Proposal for a

EAN PARLIAMENT AND OF THE COUNCIL

RULES ON ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS

European Commission > Strategy > Digital Single Market > Reports and studies >

Digital Single Market

REPORT / STUDY | 8 April 2019

Ethics guidelines for trustworthy AI



© DARPA



Search

European Commission > Strategy > Digital Single Market > Reports and studies >

Digital Single Market

REPORT / STUDY | 8 April 2019

## Ethics guidelines for trustworthy AI

Following the publication of the draft ethics guidelines in December 2018 to which more than 500 comments were received, the independent expert group presents today their ethics guidelines for trustworthy artificial intelligence.

About Artificial intelligence

Blog posts

News

# XAI & the principle of explicability



European Commission > Strategy > Digital Single Market > Reports and

Digital Single Market

REPORT / ST

The principle of explicability

- Explicability is crucial for building and maintaining users' trust in AI systems. This means that processes need to be transparent, the capabilities and purpose of AI systems openly communicated, and decisions – to the extent possible – explainable to those directly and indirectly affected. Without such information, a decision cannot be duly contested. An explanation as to why a model has generated a particular output or decision (and what combination of input factors contributed to that) is not always possible. These cases are referred to as 'black box' algorithms and require special attention. In those circumstances, other explicability measures (e.g. traceability, auditability and transparent communication on system capabilities) may be required, provided that the system as a whole respects fundamental rights. The degree to which explicability is needed is highly dependent on the context and the severity of the consequences if that output is erroneous or otherwise inaccurate.<sup>33</sup>

About Artificial  
intelligence

Blog posts

News

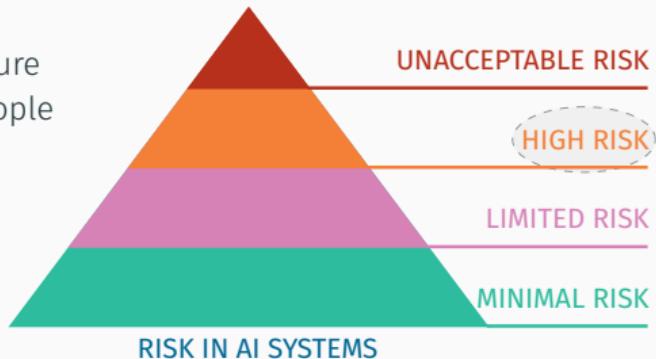
& thousands of recent papers!

# XAI for high-risk & safety-critical applications

- **High-risk** (EU regulations):

- Law enforcement
- Management and operation of critical infrastructure
- Biometric identification and categorization of people
- ...

[EU21b, EU21a]



# XAI for high-risk & safety-critical applications

- **High-risk** (EU regulations):

- Law enforcement
- Management and operation of critical infrastructure
- Biometric identification and processing
- ...

otherwise incorrect or unjust manner. Furthermore, the exercise of important procedural fundamental rights, such as the right to an effective remedy and to a fair trial as well as the right of defence and the presumption of innocence, could be hampered, in particular, where such AI systems are not sufficiently transparent, explainable and documented.

[21b, EU21a]

**EU AI Act, 2021, page 27**

**HIGH RISK**

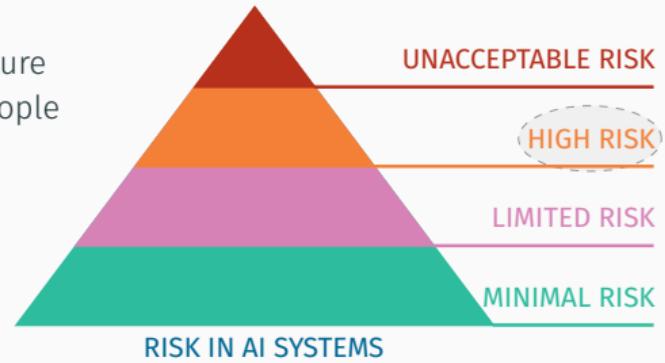


# XAI for high-risk & safety-critical applications

- **High-risk** (EU regulations):

- Law enforcement
- Management and operation of critical infrastructure
- Biometric identification and categorization of people
- ...

[EU21b, EU21a]



- And **safety-critical**:

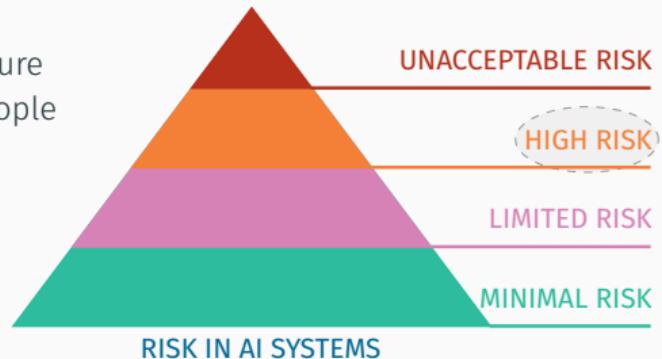
- Self-driving cars
- Autonomous vehicles
- Autonomous aerial devices
- ...

# XAI for high-risk & safety-critical applications

- **High-risk** (EU regulations):

- Law enforcement
- Management and operation of critical infrastructure
- Biometric identification and categorization of people
- ...

[EU21b, EU21a]



- And **safety-critical**:

- Self-driving cars
- Autonomous vehicles
- Autonomous aerial devices
- ...



**Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead**

Cynthia Rudin

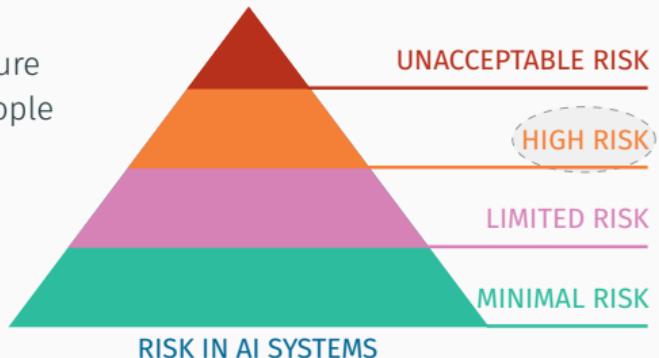
May 2019

# XAI for high-risk & safety-critical applications

- **High-risk** (EU regulations):

- Law enforcement
- Management and operation of critical infrastructure
- Biometric identification and categorization of people
- ...

[EU21b, EU21a]



- And **safety-critical**:

- Self-driving cars
- Autonomous vehicles
- Autonomous aerial devices
- ...

- **Correctness of explanations is paramount!**

- To build trust
- To help debug AI systems
- To prevent (catastrophic) accidents
- ...

PERSPECTIVE

<https://doi.org/10.1038/s42256-019-0048-x>

nature  
machine intelligence

**Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead**

Cynthia Rudin

May 2019

# XAI for high-risk & safety-critical applications

- **High-risk** (EU regulations):

- Law enforcement
- Management and operation of critical infrastructure
- Biometric identification and categorization of people
- ...

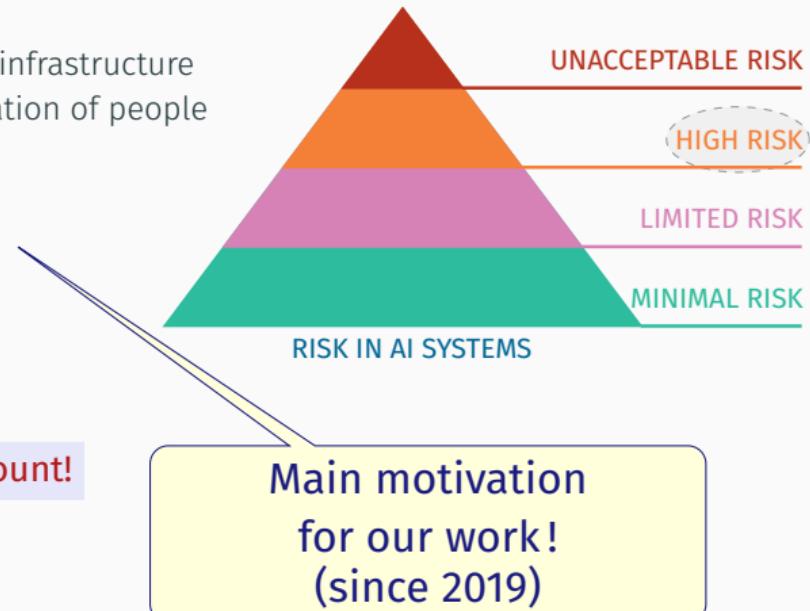
[EU21b, EU21a]

- And **safety-critical**:

- Self-driving cars
- Autonomous vehicles
- Autonomous aerial devices
- ...

- **Correctness of explanations is paramount!**

- To build trust
- To help debug AI systems
- To prevent (catastrophic) accidents
- ...



# Can we trust (non-symbolic) XAI? – some questions

- Many proposed **solutions** for XAI
  - Most, and the better-known, are heuristic
  - I.e. no guarantees of rigor
- Many proposed **uses** of XAI
- Regular complaints about issues with existing (heuristic) methods of XAI

# Can we trust (non-symbolic) XAI? – some questions

- Many proposed **solutions** for XAI
  - Most, and the better-known, are heuristic
  - I.e. no guarantees of rigor
- Many proposed **uses** of XAI
- Regular complaints about issues with existing (heuristic) methods of XAI
- **Q:** Can heuristic XAI be trusted in high-risk and/or safety-critical domains?
- **Q:** Can we validate results of heuristic XAI?

# What have we been up to? 1. Created the field of symbolic (formal) XAI – I

[MI22, Mar22, MS23, Mar24]

- Rigorous, logic-based, definitions of explanations
  - Relationship with abduction – abductive explanations (AXps)
  - Contrastive explanations (CXps) [Mil19]
- Duality between AXps & CXps
  - AXps are MHSes of CXps and vice-versa

# What have we been up to? 1. Created the field of symbolic (formal) XAI – I

[MI22, Mar22, MS23, Mar24]

- Rigorous, logic-based, definitions of explanations
  - Relationship with abduction – abductive explanations (AXps)
  - Contrastive explanations (CXps) [Mil19]
- Duality between AXps & CXps
  - AXps are MHSes of CXps and vice-versa
- Tractability results
  - Devised efficient poly-time algorithms

# What have we been up to? 1. Created the field of symbolic (formal) XAI – I

[MI22, Mar22, MS23, Mar24]

- Rigorous, logic-based, definitions of explanations
  - Relationship with abduction – abductive explanations (AXps)
  - Contrastive explanations (CXps) [Mil19]
- Duality between AXps & CXps
  - AXps are MHSes of CXps and vice-versa
- Tractability results
  - Devised efficient poly-time algorithms
- Intractability results
  - Devised efficient methods
  - Links with automated reasoners

# What have we been up to? 1. Created the field of symbolic (formal) XAI – I

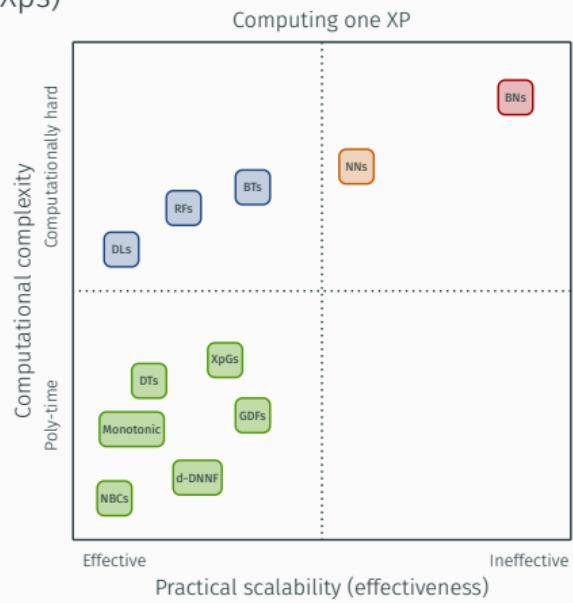
[MI22, Mar22, MS23, Mar24]

- Rigorous, logic-based, definitions of explanations
  - Relationship with abduction – abductive explanations (AXps)
  - Contrastive explanations (CXps) [Mil19]
- Duality between AXps & CXps
  - AXps are MHSes of CXps and vice-versa
- Tractability results
  - Devised efficient poly-time algorithms
- Intractability results
  - Devised efficient methods
  - Links with automated reasoners
- Wealth of computational problems related with AXps/CXps

# What have we been up to? 1. Created the field of symbolic (formal) XAI – I

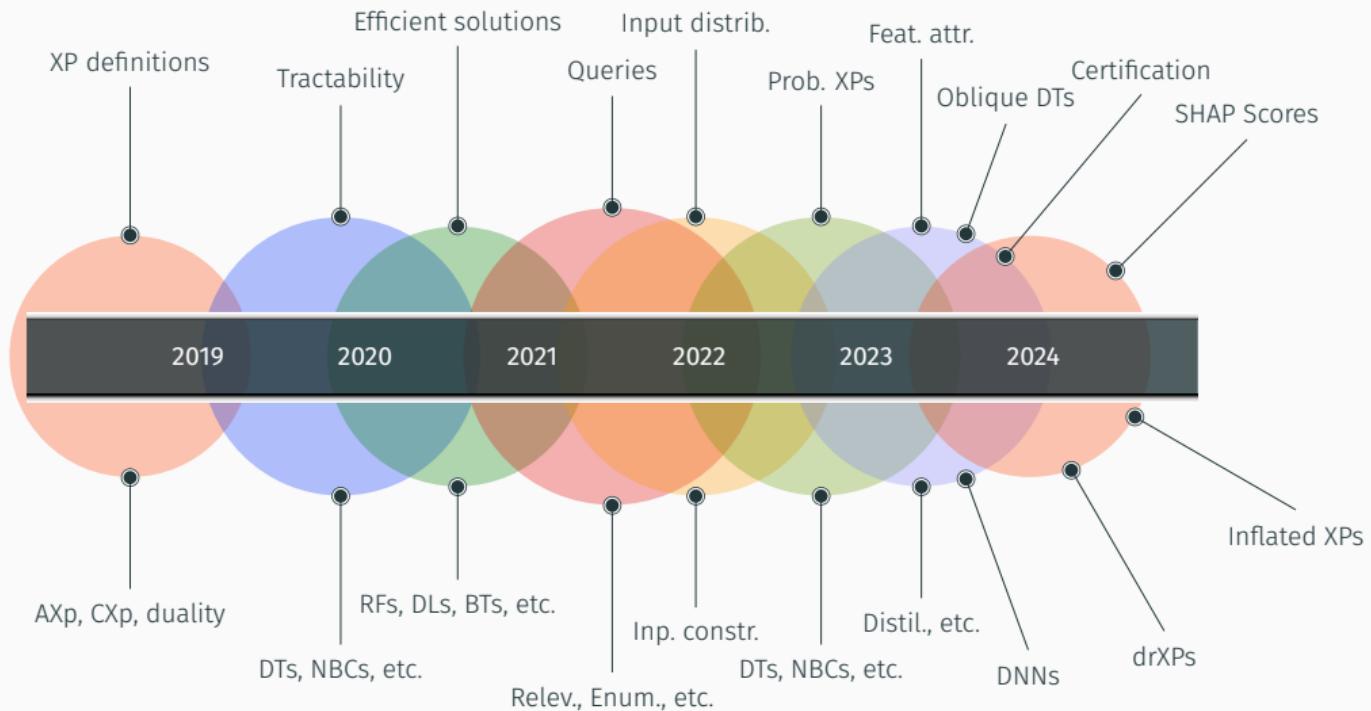
[MI22, Mar22, MS23, Mar24]

- Rigorous, logic-based, definitions of explanations
  - Relationship with abduction – abductive explanations (AXps)
  - Contrastive explanations (CXps) [Mil19]
- Duality between AXps & CXps
  - AXps are MHSes of CXps and vice-versa
- Tractability results
  - Devised efficient poly-time algorithms
- Intractability results
  - Devised efficient methods
  - Links with automated reasoners
- Wealth of computational problems related with AXps/CXps



# What have we been up to? 1. Created the field of symbolic (formal) XAI – II

[MI22, Mar22, MS23, Mar24]



# What have we been up to? 2. Uncovered key myths of non-symbolic XAI – I

[RSG16, LL17, RSG18, Rud19]

## LIME “Why Should I Trust You?” Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro  
University of Washington  
Seattle, WA 98105, USA  
marcotcr@cs.washington.edu

Sameer Singh  
University of Washington  
Seattle, WA 98105, USA  
sameer@cs.washington.edu

Carlos Guestrin  
University of Washington  
Seattle, WA 98105, USA  
guestrin@cs.washington.edu

## PERSPECTIVE

<https://doi.org/10.1038/s42256-019-0048-x>

nature  
machine intelligence

Stop explaining black box machine learning  
models for high stakes decisions and use  
interpretable models instead

Intrinsic Interpretability

Cynthia Rudin



## A Unified Approach to Interpreting Model Predictions

Scott M. Lundberg  
Paul G. Allen School of Computer Science  
University of Washington  
Seattle, WA 98105  
slund1@cs.washington.edu

Su-In Lee  
Paul G. Allen School of Computer Science  
Department of Genome Sciences  
University of Washington  
Seattle, WA 98105  
suinlee@cs.washington.edu



## Anchors: High-Precision Model-Agnostic Explanations

Marco Tulio Ribeiro  
University of Washington  
marcotcr@cs.washington.edu

Sameer Singh  
University of California, Irvine  
sameer@uci.edu

Carlos Guestrin  
University of Washington  
guestrin@cs.washington.edu



Anchor

[MSH24, HMS24, HM23c]

# research and advances



DOI:10.1145/3635301

**When the decisions of ML models impact people, one should expect explanations to offer the strongest guarantees of rigor. However, the most popular XAI approaches offer none.**

BY JOAO MARQUES-SILVA AND XUANXIANG HUANG

# Explainability Is Not a Game

## » key insights

- Shapley values find extensive uses in explaining machine learning models and serve to assign importance to the features of the model.
- Shapley values for explainability also find ever-increasing uses in high-risk and safety-critical domains, for example, medical diagnosis.
- This article proves that the existing definition of Shapley values for explainability can produce misleading information regarding feature importance, and so can induce human decision makers in error.

# Plan for this course

- Lecture 01 – unit(s):
  - #01: Foundations
- Lecture 02 – unit(s):
  - #02: Principles of symbolic XAI – **feature selection**
  - #03: Tractability in symbolic XAI (& myth of interpretability)
- Lecture 03 – unit(s):
  - #04: Intractability in symbolic XAI (& myth of model-agnostic XAI)
  - #05: Explainability queries
- Lecture 04 – unit(s):
  - #06: Recent, emerging & advanced topics
- Lecture 05 – unit(s):
  - #07: Principles of symbolic XAI – **feature attribution** (& myth of Shapley values in XAI)
  - #08: Corrected feature attribution – nuSHAP
  - #09: Conclusions & research directions

Unit #01

Foundations

## Classification problems

- Set of features  $\mathcal{F} = \{1, 2, \dots, m\}$ , each feature  $i$  taking values from domain  $D_i$ 
  - Features can be categorical, discrete or real-valued
  - Feature space:  $\mathbb{F} = \Pi_{i=1}^m D_i$
- Set of classes  $\mathcal{K} = \{c_1, \dots, c_K\}$

# Classification problems

- Set of features  $\mathcal{F} = \{1, 2, \dots, m\}$ , each feature  $i$  taking values from domain  $D_i$ 
  - Features can be categorical, discrete or real-valued
  - Feature space:  $\mathbb{F} = \Pi_{i=1}^m D_i$
- Set of classes  $\mathcal{K} = \{c_1, \dots, c_K\}$
- ML model  $\mathcal{M}_C$  computes a (non-constant) classification function  $\kappa : \mathbb{F} \rightarrow \mathcal{K}$ 
  - $\mathcal{M}_C$  is a tuple  $(\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$

# Classification problems

- Set of features  $\mathcal{F} = \{1, 2, \dots, m\}$ , each feature  $i$  taking values from domain  $D_i$ 
  - Features can be categorical, discrete or real-valued
  - Feature space:  $\mathbb{F} = \prod_{i=1}^m D_i$
- Set of classes  $\mathcal{K} = \{c_1, \dots, c_K\}$
- ML model  $\mathcal{M}_C$  computes a (non-constant) classification function  $\kappa : \mathbb{F} \rightarrow \mathcal{K}$ 
  - $\mathcal{M}_C$  is a tuple  $(\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$
- Instance  $(\mathbf{v}, c)$  for point  $\mathbf{v} = (v_1, \dots, v_m) \in \mathbb{F}$ , with prediction  $c = \kappa(\mathbf{v})$ ,  $c \in \mathcal{K}$ 
  - Goal: to compute explanations for  $(\mathbf{v}, c)$

# Regression problems

- For regression problems:
  - Codomain:  $\mathbb{V}$
  - Regression function:  $\rho : \mathbb{F} \rightarrow \mathbb{V}$  (non-constant)
  - ML model:  $\mathcal{M}_R$  is a tuple  $(\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$

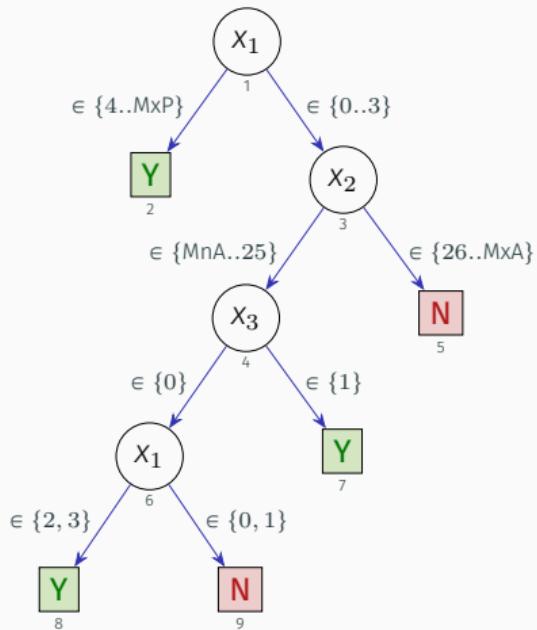
# Regression problems

- For regression problems:
  - Codomain:  $\mathbb{V}$
  - Regression function:  $\rho : \mathbb{F} \rightarrow \mathbb{V}$  (non-constant)
  - ML model:  $\mathcal{M}_R$  is a tuple  $(\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$
- General ML model:
  - $\mathbb{T}$ : range of possible predictions
  - Non-constant function  $\tau : \mathbb{F} \rightarrow \mathbb{T}$
  - ML model:  $\mathcal{M}$  is a tuple  $(\mathcal{F}, \mathbb{F}, \mathbb{T}, \tau)$

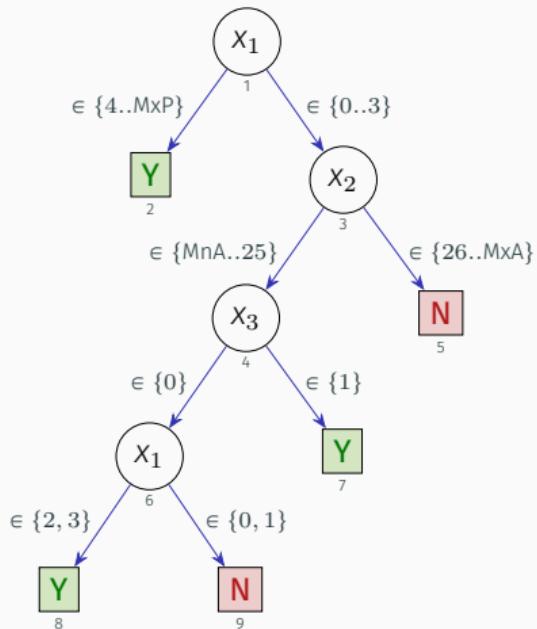
# Regression problems

- For regression problems:
  - Codomain:  $\mathbb{V}$
  - Regression function:  $\rho : \mathbb{F} \rightarrow \mathbb{V}$  (non-constant)
  - ML model:  $\mathcal{M}_R$  is a tuple  $(\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$
- General ML model:
  - $\mathbb{T}$ : range of possible predictions
  - Non-constant function  $\tau : \mathbb{F} \rightarrow \mathbb{T}$
  - ML model:  $\mathcal{M}$  is a tuple  $(\mathcal{F}, \mathbb{F}, \mathbb{T}, \tau)$
- Instance:  $(\mathbf{v}, q), q \in \mathbb{T}$

## Example ML models – classification – decision trees (DTs)

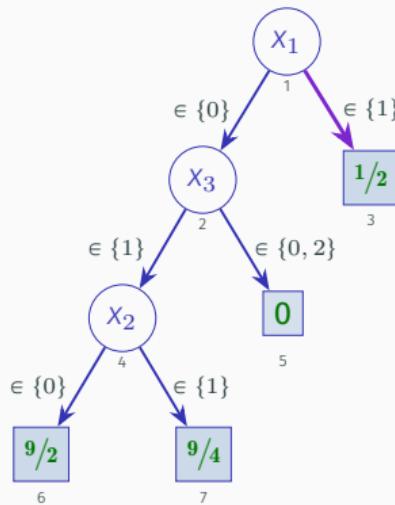


## Example ML models – classification – decision trees (DTs)



- Literals in DTs can use  $=$  or  $\in$

## Example ML models – regression – regression trees (RTs)



- Literals in RTs can use  $=$  or  $\in$

## Example ML models – classification – rules

- Ordered rules – decision lists (DLs):

```
IF       $x_1 \wedge x_2$  THEN predict Y  
ELSE IF  $\neg x_2 \vee x_3$  THEN predict N  
ELSE                THEN predict Y  
 $\mathcal{F} = \{1, 2, 3\}; \mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3 = \{0, 1\}; \mathcal{K} = \{Y, N\}$ 
```

## Example ML models – classification – rules

- Ordered rules – decision lists (DLs):

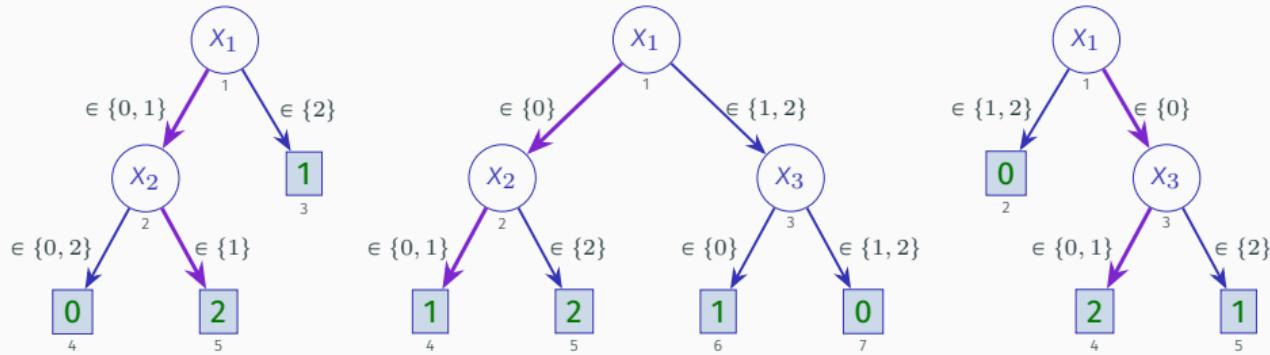
$$\begin{array}{lll} \text{IF } & x_1 \wedge x_2 & \text{THEN predict Y} \\ \text{ELSE IF } & \neg x_2 \vee x_3 & \text{THEN predict N} \\ \text{ELSE } & & \text{THEN predict Y} \\ \mathcal{F} = \{1, 2, 3\}; \mathcal{D}_1 = \mathcal{D}_2 = \mathcal{D}_3 = \{0, 1\}; \mathcal{K} = \{\text{Y}, \text{N}\} \end{array}$$

- Unordered rules – decision sets (DSs):

$$\begin{array}{lll} \text{IF } & x_1 + x_2 \geq 0 & \text{THEN predict } \blacksquare \\ \text{IF } & x_1 + x_2 < 0 & \text{THEN predict } \blacksquare \\ \mathcal{F} = \{1, 2\}; \mathcal{D}_1 = \mathcal{D}_2 = \mathbb{R}; \mathcal{K} = \{\blacksquare, \blacksquare\} \end{array}$$

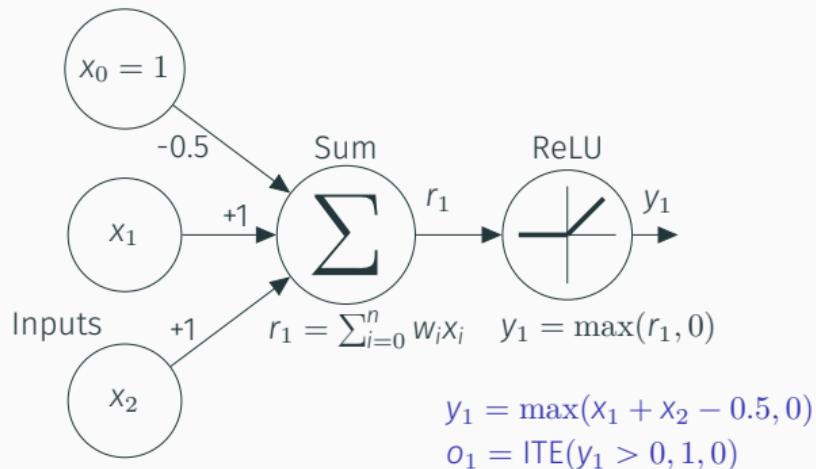
- Issues of DSs: overlap; incomplete coverage

## Example ML models – classification – random forests (RFs)



- For each input, each DT picks a class
- Result uses majority or weighted voting of the DTs

## Example ML models – classification – neural networks (NNs)



# Outline – Unit #01

ML Models: Classification & Regression Problems

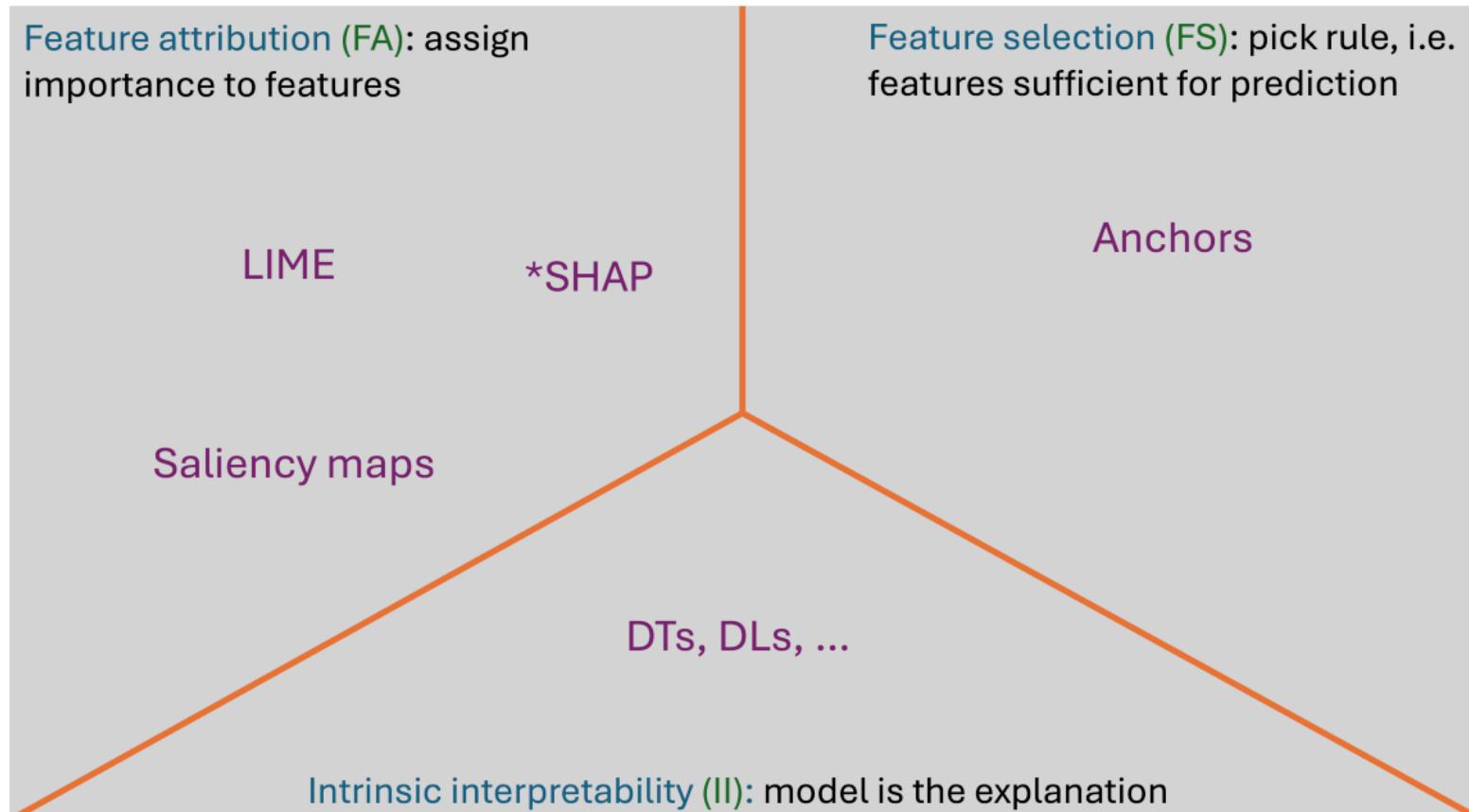
Basics of (non-symbolic) XAI

Motivation for Explanations

Brief Glimpse of Logic

Reasoning About ML Models

Understanding Intrinsic Interpretability

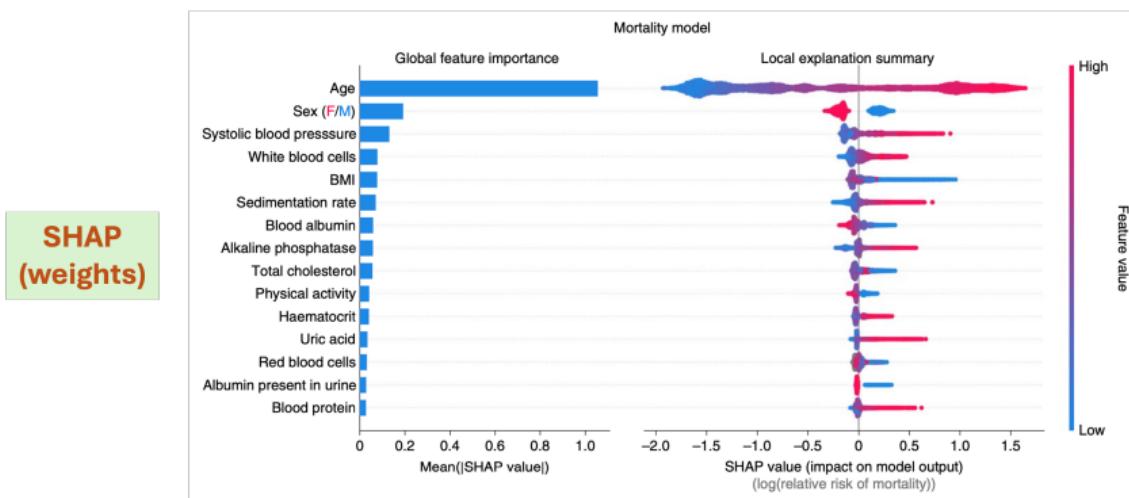


# Some non-symbolic XAI methods

[LL17, RSG18]

**IF Country = United-States AND Capital Loss = Low  
AND Race = White AND Relationship = Husband  
AND Married AND  $28 < \text{Age} \leq 37$   
AND Sex = Male AND High School grad  
AND Occupation = Blue-Collar  
THEN PREDICT Salary > \$50K**

**Anchors  
(rules)**



# Some non-symbolic XAI methods

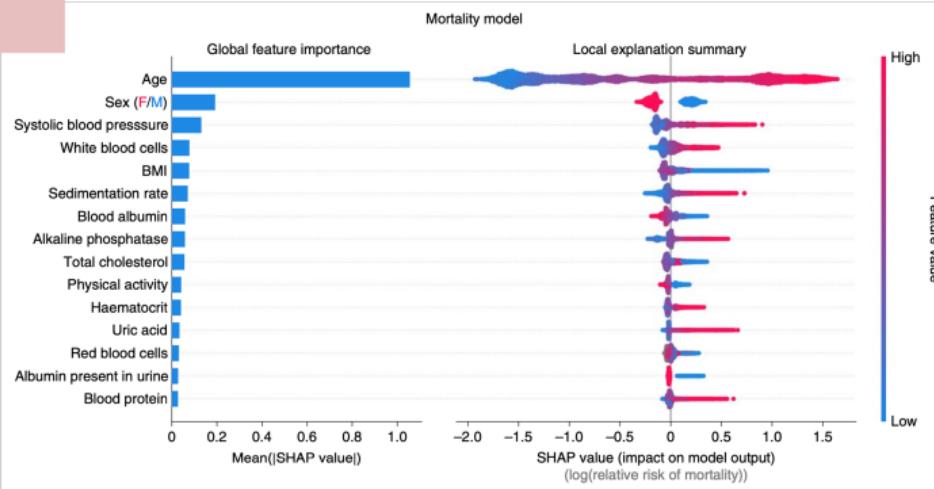
[LL17, RSG18]

**IF Country = United-States AND Capital Loss = Low  
AND Race = White AND Relationship = Husband  
AND Married AND  $28 < \text{Age} \leq 37$   
AND Sex = Male AND High School grad  
AND Occupation = Blue-Collar  
THEN PREDICT Salary > \$50K**

**Anchors  
(rules)**

model-agnostic, sample-based,  
& no guarantees of rigor...

**SHAP  
(weights)**



# Outline – Unit #01

ML Models: Classification & Regression Problems

Basics of (non-symbolic) XAI

Motivation for Explanations

Brief Glimpse of Logic

Reasoning About ML Models

Understanding Intrinsic Interpretability

## What explanations do we seek? I.e. how to answer **Why?** questions?

- How to answer a **Why?** question? I.e. “ Why (the prediction)? ”

## What explanations do we seek? I.e. how to answer **Why?** questions?

- How to answer a **Why?** question? I.e. “ Why (the prediction)? ”
  - Our answer to a **Why?** question is a **rule**:

IF <COND> THEN  $\kappa(\mathbf{x}) = \mathbf{c}$

# What explanations do we seek? I.e. how to answer **Why?** questions?

- How to answer a **Why?** question? I.e. “ Why (the prediction)? ”
  - Our answer to a **Why?** question is a **rule**:

IF <COND> THEN  $\kappa(x) = c$

- **Explanation:** set of **literals** (or just **features**) in <COND>; **irreducibility matters!**
  - <COND> is **sufficient** for the prediction

# What explanations do we seek? I.e. how to answer **Why?** questions?

- How to answer a **Why?** question? I.e. “ Why (the prediction)? ”
  - Our answer to a **Why?** question is a **rule**:

IF <COND> THEN  $\kappa(x) = c$

- **Explanation:** set of **literals** (or just **features**) in <COND>; **irreducibility matters!**
  - <COND> is **sufficient** for the prediction
- **Obs:** rules are used in tools like Anchors
  - An **anchor** is a “high-precision rule”

[RSG16]

[RSG16]

# What explanations do we seek? I.e. how to answer **Why?** questions?

- How to answer a **Why?** question? I.e. “ Why (the prediction)? ”
  - Our answer to a **Why?** question is a **rule**:

IF <COND> THEN  $\kappa(x) = c$

- **Explanation:** set of **literals** (or just **features**) in <COND>; **irreducibility matters!**
  - <COND> is **sufficient** for the prediction
- **Obs:** rules are used in tools like Anchors
  - An **anchor** is a “high-precision rule”
- We seek a **rigorous** definition of rules for answering **Why?** questions such that,

[RSG16]

[RSG16]

# What explanations do we seek? I.e. how to answer **Why?** questions?

- How to answer a **Why?** question? I.e. “ Why (the prediction)? ”
  - Our answer to a **Why?** question is a **rule**:

IF <COND> THEN  $\kappa(x) = c$

- **Explanation:** set of **literals** (or just **features**) in <COND>; **irreducibility matters!**
  - <COND> is **sufficient** for the prediction
- **Obs:** rules are used in tools like Anchors
  - An **anchor** is a “high-precision rule”
- We seek a **rigorous** definition of rules for answering **Why?** questions such that,
  - <COND> is **sufficient** for the prediction
  - <COND> is **irreducible**

[RSG16]

[RSG16]

# What explanations do we seek? I.e. how to answer **Why?** questions?

- How to answer a **Why?** question? I.e. “ Why (the prediction)? ”
  - Our answer to a **Why?** question is a **rule**:

IF   <COND>   THEN    $\kappa(x) = c$

- **Explanation:** set of **literals** (or just **features**) in **<COND>**; **irreducibility matters!**
  - **<COND>** is **sufficient** for the prediction
- **Obs:** rules are used in tools like Anchors
  - An **anchor** is a “high-precision rule”
- We seek a **rigorous** definition of rules for answering **Why?** questions such that,
  - **<COND>** is **sufficient** for the prediction
  - **<COND>** is **irreducible**
- We also seek the algorithms for the rigorous computation of such rules

[RSG16]

[RSG16]

## A decision list example

```
IF       $\neg x_1 \wedge x_2$  THEN predict Y
ELSE IF  $\neg x_1 \wedge x_3$  THEN predict Y
ELSE IF  $x_4 \wedge x_5$  THEN predict N
ELSE                THEN predict Y
```

## A decision list example

```
IF       $\neg x_1 \wedge x_2$  THEN predict Y  
ELSE IF  $\neg x_1 \wedge x_3$  THEN predict Y  
ELSE IF  $x_4 \wedge x_5$  THEN predict N  
ELSE                THEN predict Y
```

- Explanation for why  $\kappa(1, 1, 1, 1, 1) = \text{N}$ ?

## A decision list example

```
IF       $\neg x_1 \wedge x_2$  THEN predict Y  
ELSE IF  $\neg x_1 \wedge x_3$  THEN predict Y  
ELSE IF  $x_4 \wedge x_5$  THEN predict N  
ELSE                THEN predict Y
```

- Explanation for **why**  $\kappa(1, 1, 1, 1, 1) = \text{N}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ ,  
**IF**  $(x_1 = 1) \wedge (x_4 = 1) \wedge (x_5 = 1)$  **THEN**  $\kappa(\mathbf{x}) = \text{N}$
  - I.e.  $\{x_1 = 1, x_4 = 1, x_5 = 1\}$  suffice for DL to predict **N**

## A decision list example

```
IF       $\neg x_1 \wedge x_2$  THEN predict Y  
ELSE IF  $\neg x_1 \wedge x_3$  THEN predict Y  
ELSE IF  $x_4 \wedge x_5$  THEN predict N  
ELSE                THEN predict Y
```

- Explanation for **why**  $\kappa(1, 1, 1, 1, 1) = \text{N}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ ,  
**IF**  $(x_1 = 1) \wedge (x_4 = 1) \wedge (x_5 = 1)$  **THEN**  $\kappa(\mathbf{x}) = \text{N}$
  - I.e.  $\{x_1 = 1, x_4 = 1, x_5 = 1\}$  suffice for DL to predict **N**
- Explanation for **why**  $\kappa(1, 0, 0, 0, 0) = \text{Y}$ ?

## A decision list example

```
IF       $\neg x_1 \wedge x_2$  THEN predict Y  
ELSE IF  $\neg x_1 \wedge x_3$  THEN predict Y  
ELSE IF  $x_4 \wedge x_5$  THEN predict N  
ELSE                THEN predict Y
```

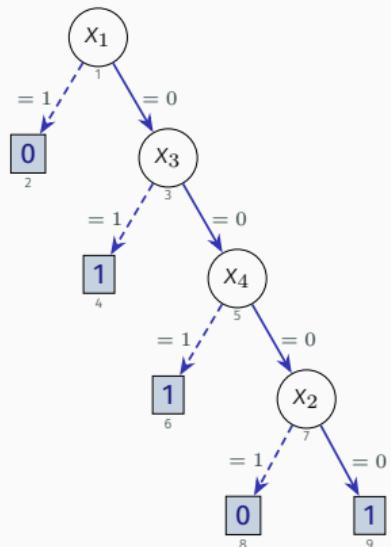
- Explanation for **why**  $\kappa(1, 1, 1, 1, 1) = \text{N}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ ,  
**IF**  $(x_1 = 1) \wedge (x_4 = 1) \wedge (x_5 = 1)$  **THEN**  $\kappa(\mathbf{x}) = \text{N}$
  - I.e.  $\{x_1 = 1, x_4 = 1, x_5 = 1\}$  suffice for DL to predict **N**
- Explanation for **why**  $\kappa(1, 0, 0, 0, 0) = \text{Y}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ ,  
**IF**  $(x_4 = 0)$  **THEN**  $\kappa(\mathbf{x}) = \text{Y}$
  - I.e.  $\{x_4 = 0\}$  suffices for DL to predict **Y**

## A decision list example

```
IF       $\neg x_1 \wedge x_2$  THEN predict Y  
ELSE IF  $\neg x_1 \wedge x_3$  THEN predict Y  
ELSE IF  $x_4 \wedge x_5$  THEN predict N  
ELSE                THEN predict Y
```

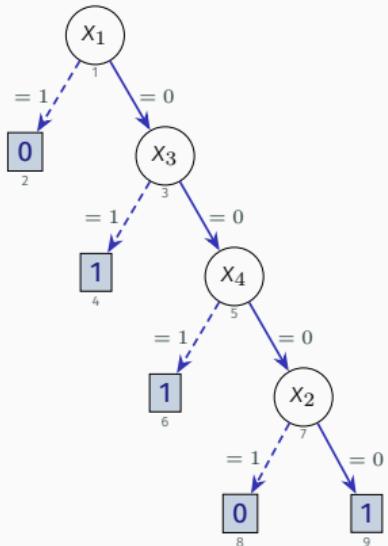
- Explanation for **why**  $\kappa(1, 1, 1, 1, 1) = \text{N}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ ,  
**IF**  $(x_1 = 1) \wedge (x_4 = 1) \wedge (x_5 = 1)$  **THEN**  $\kappa(\mathbf{x}) = \text{N}$
  - I.e.  $\{x_1 = 1, x_4 = 1, x_5 = 1\}$  suffice for DL to predict **N**
- Explanation for **why**  $\kappa(1, 0, 0, 0, 0) = \text{Y}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ ,  
**IF**  $(x_4 = 0)$  **THEN**  $\kappa(\mathbf{x}) = \text{Y}$
  - I.e.  $\{x_4 = 0\}$  suffices for DL to predict **Y**
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ ,  
**IF**  $(x_5 = 0)$  **THEN**  $\kappa(\mathbf{x}) = \text{Y}$
  - I.e.  $\{x_5 = 0\}$  also suffices for DL to predict **Y**

# A decision tree example



$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

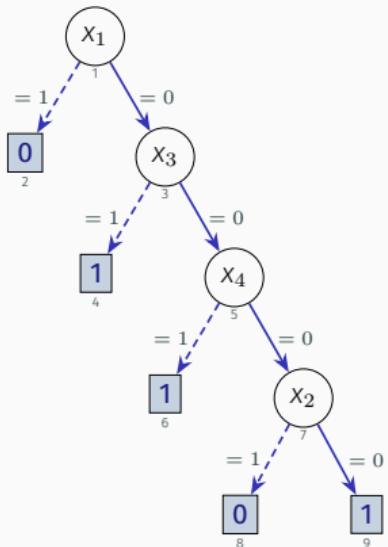
# A decision tree example



- Explanation for **why**  $\kappa(0, 0, 0, 0) = 1$ ?

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

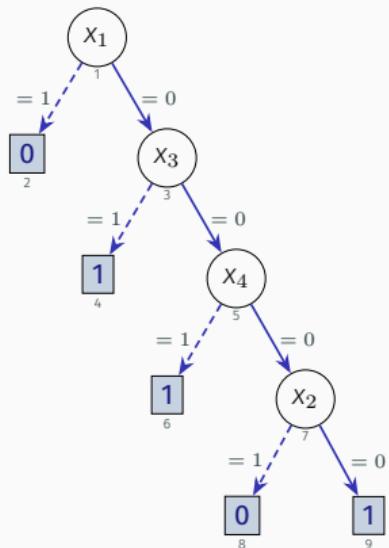
# A decision tree example



- Explanation for **why**  $\kappa(0, 0, 0, 0) = 1$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ ,
  - IF**  $(x_1 = 0) \wedge (x_2 = 0)$  **THEN**  $\kappa(\mathbf{x}) = 1$
  - i.e.  $\{x_1 = 0, x_2 = 0\}$  suffice for DT to predict 1

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

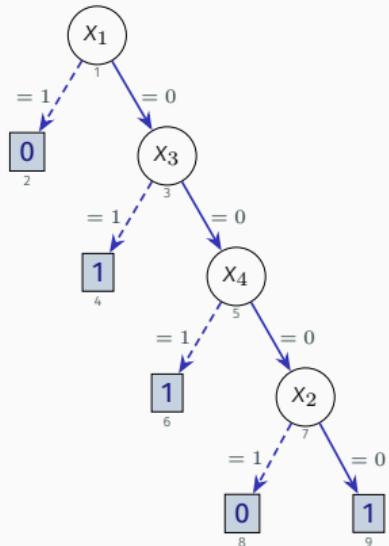
# A decision tree example



- Explanation for why  $\kappa(0, 0, 0, 0) = 1$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ ,
  - IF  $(x_1 = 0) \wedge (x_2 = 0)$  THEN  $\kappa(\mathbf{x}) = 1$
  - i.e.  $\{x_1 = 0, x_2 = 0\}$  suffice for DT to predict 1
- Explanation for why  $\kappa(1, 1, 1, 1) = 0$ ?

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

# A decision tree example

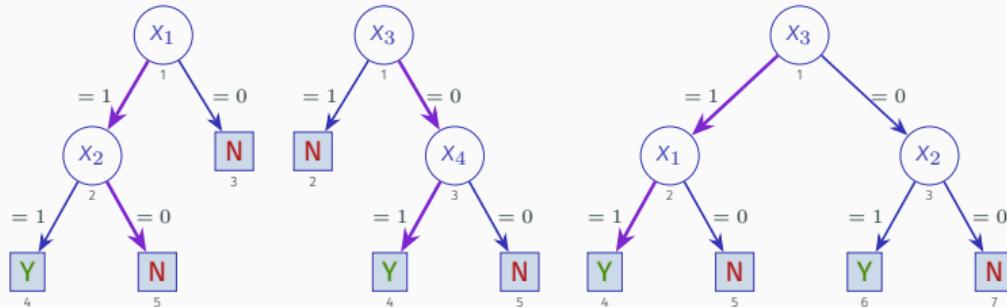


- Explanation for **why**  $\kappa(0, 0, 0, 0) = 1$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ ,  
**IF**  $(x_1 = 0) \wedge (x_2 = 0)$  **THEN**  $\kappa(\mathbf{x}) = 1$
  - i.e.  $\{x_1 = 0, x_2 = 0\}$  suffice for DT to predict **1**
- Explanation for **why**  $\kappa(1, 1, 1, 1) = 0$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ ,  
**IF**  $(x_1 = 1)$  **THEN**  $\kappa(\mathbf{x}) = 0$
  - i.e.  $\{x_1 = 1\}$  suffices for DT to predict **0**

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	<b>1</b>
0	0	0	1	<b>1</b>
0	0	1	0	<b>1</b>
0	0	1	1	<b>1</b>
0	1	0	0	<b>0</b>
0	1	0	1	<b>1</b>
0	1	1	0	<b>1</b>
0	1	1	1	<b>1</b>
1	0	0	0	<b>0</b>
1	0	0	1	<b>0</b>
1	0	1	0	<b>0</b>
1	0	1	1	<b>0</b>
1	1	0	0	<b>0</b>
1	1	0	1	<b>0</b>
1	1	1	0	<b>0</b>
1	1	1	1	<b>0</b>

# A random forest example

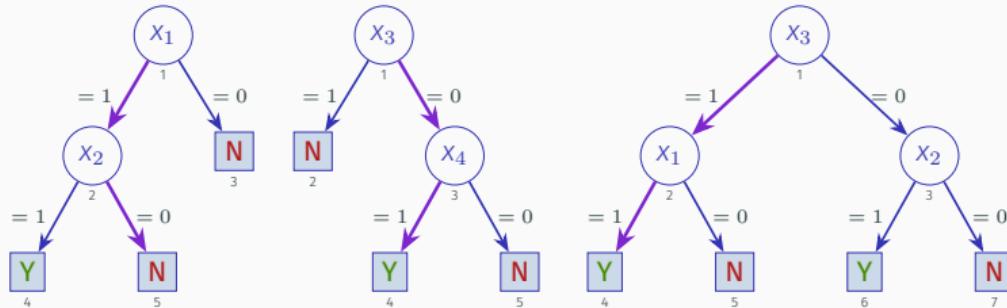
[IMS21]



$x_1$	$x_2$	$x_3$	$x_4$	$T_1$	$T_2$	$T_3$	$\kappa(\mathbf{x})$
0	0	0	0	N	N	N	N
0	0	0	1	N	Y	N	N
0	0	1	0	N	N	N	N
0	0	1	1	N	N	N	N
0	1	0	0	N	N	Y	N
0	1	0	1	N	Y	Y	Y
0	1	1	0	N	N	N	N
0	1	1	1	N	N	N	N
1	0	0	0	N	N	N	N
1	0	0	1	N	Y	N	N
1	0	1	0	N	N	Y	N
1	0	1	1	N	N	Y	N
1	1	0	0	Y	N	Y	Y
1	1	0	1	Y	Y	Y	Y
1	1	1	0	Y	N	Y	Y
1	1	1	1	Y	N	Y	Y

# A random forest example

[IMS21]

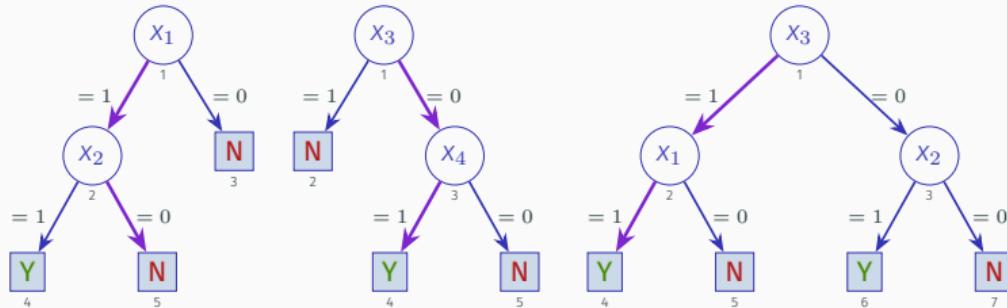


- Explanation for why  $\kappa(1, 0, 0, 1) = \text{N}$ ?

X <sub>1</sub>	X <sub>2</sub>	X <sub>3</sub>	X <sub>4</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	$\kappa(\mathbf{x})$
0	0	0	0	N	N	N	N
0	0	0	1	N	Y	N	N
0	0	1	0	N	N	N	N
0	0	1	1	N	N	N	N
0	1	0	0	N	N	Y	N
0	1	0	1	N	Y	Y	Y
0	1	1	0	N	N	N	N
0	1	1	1	N	N	N	N
1	0	0	0	N	N	N	N
1	0	0	1	N	Y	N	N
1	0	1	0	N	N	Y	N
1	0	1	1	N	N	Y	N
1	1	0	0	Y	N	Y	Y
1	1	0	1	Y	Y	Y	Y
1	1	1	0	Y	N	Y	Y
1	1	1	1	Y	N	Y	Y

# A random forest example

[IMS21]

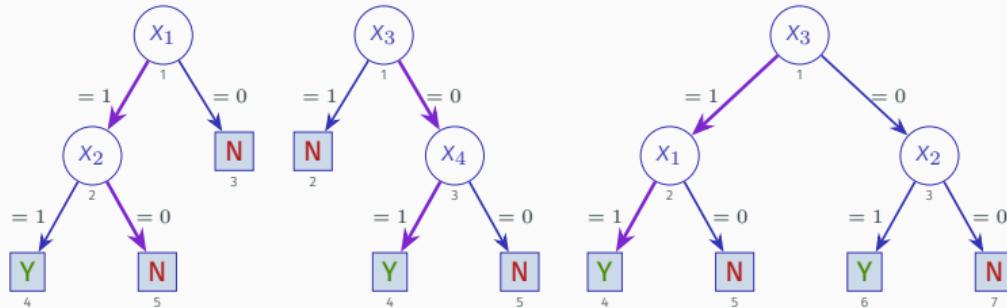


- Explanation for why  $\kappa(1, 0, 0, 1) = \mathbf{N}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , IF ( $x_2 = 0$ ) THEN  $\kappa(\mathbf{x}) = \mathbf{N}$
  - i.e.  $\{x_2 = 0\}$  suffices for RF to predict **N**

$x_1$	$x_2$	$x_3$	$x_4$	$T_1$	$T_2$	$T_3$	$\kappa(\mathbf{x})$
0	0	0	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	0	0	1	<b>N</b>	<b>Y</b>	<b>N</b>	<b>N</b>
0	0	1	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	0	1	1	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	1	0	0	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
0	1	0	1	<b>N</b>	<b>Y</b>	<b>Y</b>	<b>Y</b>
0	1	1	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	1	1	1	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
1	0	0	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
1	0	0	1	<b>N</b>	<b>Y</b>	<b>N</b>	<b>N</b>
1	0	1	0	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
1	0	1	1	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
1	1	0	0	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>
1	1	0	1	<b>Y</b>	<b>Y</b>	<b>Y</b>	<b>Y</b>
1	1	1	0	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>
1	1	1	1	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>

# A random forest example

[IMS21]

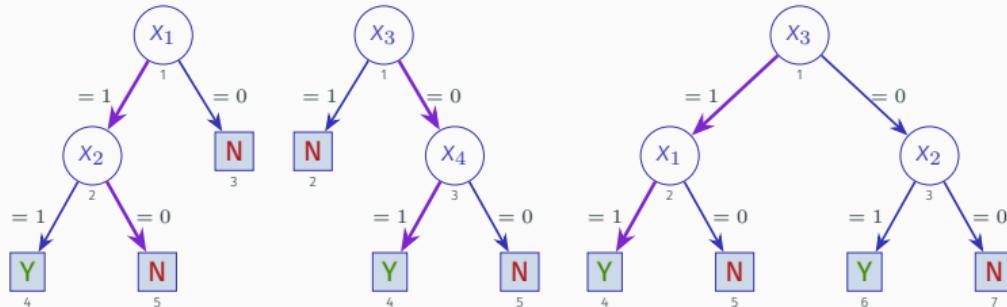


- Explanation for why  $\kappa(1, 0, 0, 1) = \text{N}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , IF ( $x_2 = 0$ ) THEN  $\kappa(\mathbf{x}) = \text{N}$
  - i.e.  $\{x_2 = 0\}$  suffices for RF to predict **N**
- Explanation for why  $\kappa(1, 1, 1, 1) = \text{Y}$ ?

$x_1$	$x_2$	$x_3$	$x_4$	$T_1$	$T_2$	$T_3$	$\kappa(\mathbf{x})$
0	0	0	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	0	0	1	<b>N</b>	<b>Y</b>	<b>N</b>	<b>N</b>
0	0	1	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	0	1	1	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	1	0	0	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
0	1	0	1	<b>N</b>	<b>Y</b>	<b>Y</b>	<b>Y</b>
0	1	1	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	1	1	1	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
1	0	0	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
1	0	0	1	<b>N</b>	<b>Y</b>	<b>N</b>	<b>N</b>
1	0	1	0	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
1	0	1	1	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
1	1	0	0	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>
1	1	0	1	<b>Y</b>	<b>Y</b>	<b>Y</b>	<b>Y</b>
1	1	1	0	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>
1	1	1	1	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>

# A random forest example

[IMS21]

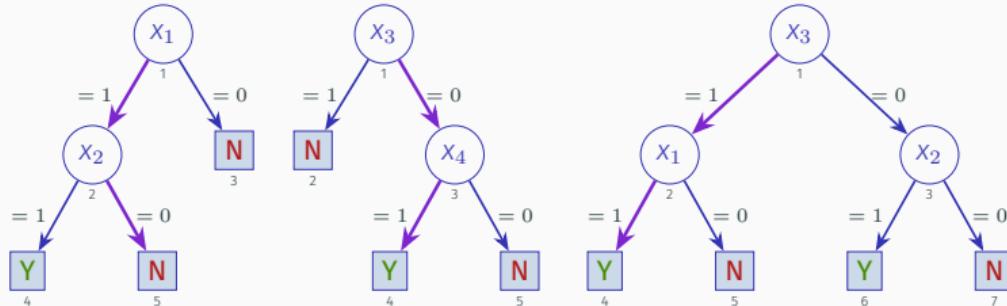


- Explanation for why  $\kappa(1, 0, 0, 1) = \text{N}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , IF  $(x_2 = 0)$  THEN  $\kappa(\mathbf{x}) = \text{N}$
  - I.e.  $\{x_2 = 0\}$  suffices for RF to predict **N**
- Explanation for why  $\kappa(1, 1, 1, 1) = \text{Y}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , IF  $(x_1 = 1) \wedge (x_2 = 1)$  THEN  $\kappa(\mathbf{x}) = \text{Y}$
  - I.e.  $\{x_1 = 1, x_2 = 1\}$  suffice for RF to predict **Y**

$x_1$	$x_2$	$x_3$	$x_4$	$T_1$	$T_2$	$T_3$	$\kappa(\mathbf{x})$
0	0	0	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	0	0	1	<b>N</b>	<b>Y</b>	<b>N</b>	<b>N</b>
0	0	1	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	0	1	1	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	1	0	0	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
0	1	0	1	<b>N</b>	<b>Y</b>	<b>Y</b>	<b>Y</b>
0	1	1	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	1	1	1	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
1	0	0	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
1	0	0	1	<b>N</b>	<b>Y</b>	<b>N</b>	<b>N</b>
1	0	1	0	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
1	0	1	1	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
1	1	0	0	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>
1	1	0	1	<b>Y</b>	<b>Y</b>	<b>Y</b>	<b>Y</b>
1	1	1	0	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>
1	1	1	1	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>

# A random forest example

[IMS21]

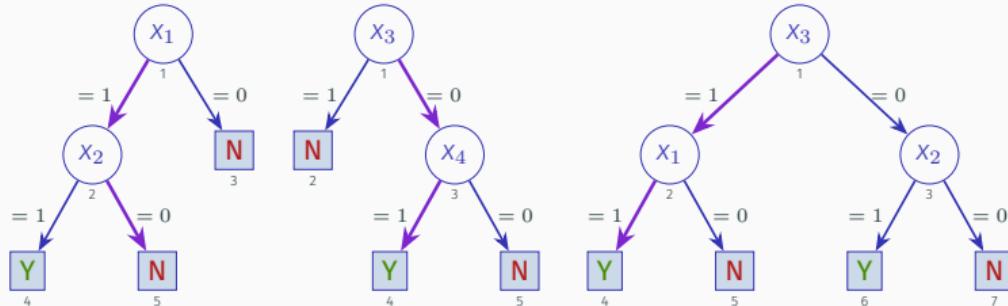


- Explanation for why  $\kappa(1, 0, 0, 1) = \text{N}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , IF  $(x_2 = 0)$  THEN  $\kappa(\mathbf{x}) = \text{N}$
  - I.e.  $\{x_2 = 0\}$  suffices for RF to predict **N**
- Explanation for why  $\kappa(1, 1, 1, 1) = \text{Y}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , IF  $(x_1 = 1) \wedge (x_2 = 1)$  THEN  $\kappa(\mathbf{x}) = \text{Y}$
  - I.e.  $\{x_1 = 1, x_2 = 1\}$  suffice for RF to predict **Y**
- Explanation for why  $\kappa(0, 1, 1, 1) = \text{N}$ ?

$x_1$	$x_2$	$x_3$	$x_4$	$T_1$	$T_2$	$T_3$	$\kappa(\mathbf{x})$
0	0	0	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	0	0	1	<b>N</b>	<b>Y</b>	<b>N</b>	<b>N</b>
0	0	1	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	0	1	1	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	1	0	0	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
0	1	0	1	<b>N</b>	<b>Y</b>	<b>Y</b>	<b>Y</b>
0	1	1	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
0	1	1	1	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
1	0	0	0	<b>N</b>	<b>N</b>	<b>N</b>	<b>N</b>
1	0	0	1	<b>N</b>	<b>Y</b>	<b>N</b>	<b>N</b>
1	0	1	0	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
1	0	1	1	<b>N</b>	<b>N</b>	<b>Y</b>	<b>N</b>
1	1	0	0	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>
1	1	0	1	<b>Y</b>	<b>Y</b>	<b>Y</b>	<b>Y</b>
1	1	1	0	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>
1	1	1	1	<b>Y</b>	<b>N</b>	<b>Y</b>	<b>Y</b>

# A random forest example

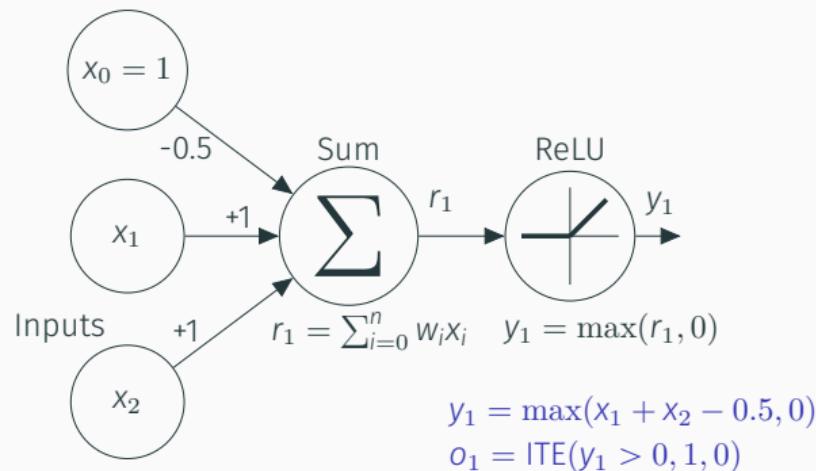
[IMS21]



- Explanation for why  $\kappa(1, 0, 0, 1) = \text{N}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , IF  $(x_2 = 0)$  THEN  $\kappa(\mathbf{x}) = \text{N}$
  - I.e.  $\{x_2 = 0\}$  suffices for RF to predict  $\text{N}$
- Explanation for why  $\kappa(1, 1, 1, 1) = \text{Y}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , IF  $(x_1 = 1) \wedge (x_2 = 1)$  THEN  $\kappa(\mathbf{x}) = \text{Y}$
  - I.e.  $\{x_1 = 1, x_2 = 1\}$  suffice for RF to predict  $\text{Y}$
- Explanation for why  $\kappa(0, 1, 1, 1) = \text{N}$ ?
  - Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ , IF  $(x_1 = 0) \wedge (x_2 = 1) \wedge (x_3 = 1)$  THEN  $\kappa(\mathbf{x}) = \text{N}$
  - I.e.  $\{x_1 = 0, x_2 = 1, x_3 = 1\}$  suffices for RF to predict  $\text{N}$

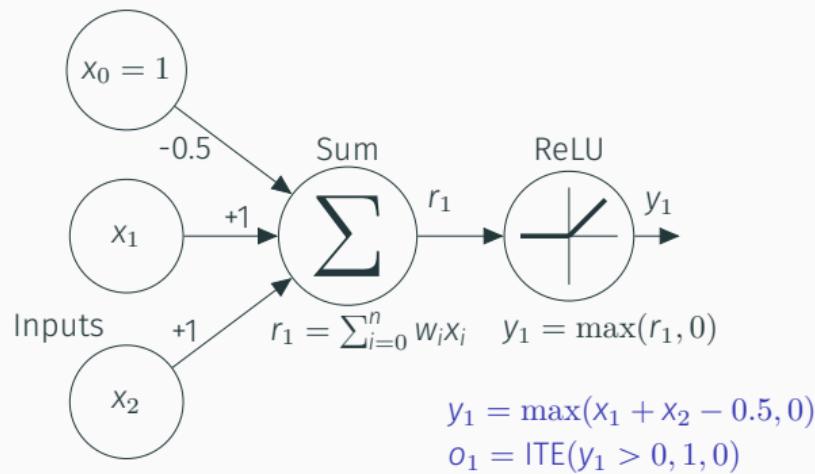
$x_1$	$x_2$	$x_3$	$x_4$	$T_1$	$T_2$	$T_3$	$\kappa(\mathbf{x})$
0	0	0	0	N	N	N	N
0	0	0	1	N	Y	N	N
0	0	1	0	N	N	N	N
0	0	1	1	N	N	N	N
0	1	0	0	N	N	Y	N
0	1	0	1	N	Y	Y	Y
0	1	1	0	N	N	N	N
0	1	1	1	N	N	N	N
1	0	0	0	N	N	N	N
1	0	0	1	N	Y	N	N
1	0	1	0	N	N	Y	N
1	0	1	1	N	N	Y	N
1	1	0	0	Y	N	Y	Y
1	1	0	1	Y	Y	Y	Y
1	1	1	0	Y	N	Y	Y
1	1	1	1	Y	N	Y	Y

# A neural network example



$x_1$	$x_2$	$r_1$	$y_1$	$\kappa(\mathbf{x})$
0	0	-0.5	0	0
0	1	0.5	0.5	1
1	0	0.5	0.5	1
1	1	1.5	1.5	1

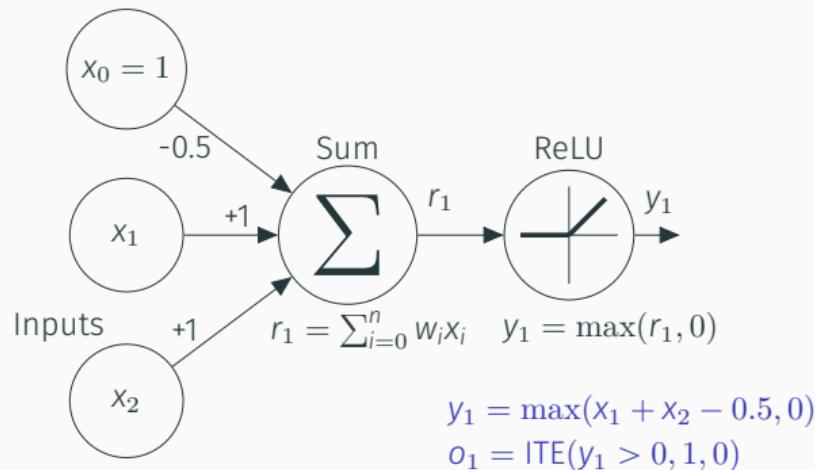
# A neural network example



$x_1$	$x_2$	$r_1$	$y_1$	$\kappa(\mathbf{x})$
0	0	-0.5	0	0
0	1	0.5	0.5	1
1	0	0.5	0.5	1
1	1	1.5	1.5	1

- Explanation for why  $\kappa(1, 1) = 1$ ?

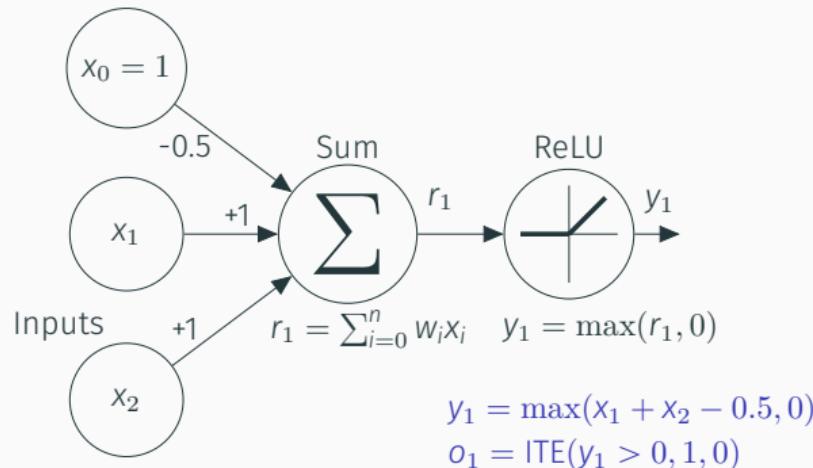
# A neural network example



$x_1$	$x_2$	$r_1$	$y_1$	$\kappa(\mathbf{x})$
0	0	-0.5	0	0
0	1	0.5	0.5	1
1	0	0.5	0.5	1
1	1	1.5	1.5	1

- Explanation for why  $\kappa(1, 1) = 1$ ?
  - Given  $\mathbf{x} = (x_1, x_2)$ , IF  $(x_1 = 1)$  THEN  $\kappa(\mathbf{x}) = 1$
  - i.e.  $\{x_1 = 1\}$  suffices for NN to predict 1

# A neural network example



$x_1$	$x_2$	$r_1$	$y_1$	$\kappa(\mathbf{x})$
0	0	-0.5	0	0
0	1	0.5	0.5	1
1	0	0.5	0.5	1
1	1	1.5	1.5	1

- Explanation for why  $\kappa(1, 1) = 1$ ?
  - Given  $\mathbf{x} = (x_1, x_2)$ , IF  $(x_1 = 1)$  THEN  $\kappa(\mathbf{x}) = 1$
  - i.e.  $\{x_1 = 1\}$  suffices for NN to predict 1
  - Given  $\mathbf{x} = (x_1, x_2)$ , IF  $(x_2 = 1)$  THEN  $\kappa(\mathbf{x}) = 1$
  - i.e.  $\{x_2 = 1\}$  suffices for NN to predict Y

# An arbitrary classifier

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

# An arbitrary classifier

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

- Instance:  $((0, 0, 0, 0), 1)$

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

# An arbitrary classifier

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

- Instance:  $((0, 0, 0, 0), 1)$

- Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ ,

IF  $(x_1 = 0) \wedge (x_3 = 0)$  THEN  $\kappa(\mathbf{x}) = 1$

- I.e.  $\{x_1 = 0, x_3 = 0\}$  suffices for RF to predict 1

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

# Outline – Unit #01

ML Models: Classification & Regression Problems

Basics of (non-symbolic) XAI

Motivation for Explanations

Brief Glimpse of Logic

Reasoning About ML Models

Understanding Intrinsic Interpretability

# Standard tools of the trade

- **SAT**: decision problem for propositional logic
  - Formulas most often represented in CNF
  - There are optimization variants: MaxSAT, PBO, MinSAT, etc.
  - There are quantified variants: QBF, QMaxSAT, etc.
- **SMT**: decision problem for (decidable) fragments of first-order logic (**FOL**)
  - There are optimization variants: MaxSMT, etc.
  - There are quantified variants
- **MILP**: decision/optimization problems defined on conjunctions of linear inequalities over integer & real-valued variables
- **CP**: constraint programming
  - There are optimization/quantified variants

# Standard tools of the trade

- **SAT**: decision problem for propositional logic
  - Formulas most often represented in CNF
  - There are optimization variants: MaxSAT, PBO, MinSAT, etc.
  - There are quantified variants: QBF, QMaxSAT, etc.
- **SMT**: decision problem for (decidable) fragments of first-order logic (**FOL**)
  - There are optimization variants: MaxSMT, etc.
  - There are quantified variants
- **MILP**: decision/optimization problems defined on conjunctions of linear inequalities over integer & real-valued variables
- **CP**: constraint programming
  - There are optimization/quantified variants
- Background on SAT/SMT:
  - <https://alexeyignatiev.github.io/ssa-school-2019/>
  - <https://alexeyignatiev.github.io/ijcai19tut/>

Basic knowledge on  
SAT & SMT assumed.  
See links below.

[BHvMW09]

## SAT/SMT/MILP/CP solvers used as oracles – more detail later

- Deciding satisfiability, entailment
- Computing prime implicants/implicates
- Computing MUSes, MCSes
  - Algorithms: Deletion, QuickXplain, Progression, Dichotomic, etc. [MM20]
- Enumeration of MUSes, MCSes
  - Algorithms: Marco, Camus, etc. [LS08, LPMM16]
- Solving MaxSAT, MaxSMT
  - Algorithms: Core-guided, Minimum hitting sets, branch&bound, etc. [MHL<sup>+</sup>13]
- Solving quantification problems, e.g. QBF
  - Algorithms: Abstraction refinement [JKMC16]

## Basic definitions in propositional logic

- Atoms ( $\{x, x_1, \dots\}$ ) & literals ( $x_1, \neg x_1$ )
- Well-formed formulas using  $\neg, \wedge, \vee, \dots$
- Clause: disjunction of literals
- Term: conjunction of literals
- Conjunctive normal form (CNF): conjunction of clauses
- Disjunctive normal form (DNF): disjunction of terms
- Simple to generalize to more expressive domains

# Basic definitions in propositional logic

- Atoms ( $\{x, x_1, \dots\}$ ) & literals ( $x_1, \neg x_1$ )
- Well-formed formulas using  $\neg, \wedge, \vee, \dots$
- Clause: disjunction of literals
- Term: conjunction of literals
- Conjunctive normal form (CNF): conjunction of clauses
- Disjunctive normal form (DNF): disjunction of terms
- Simple to generalize to more expressive domains
- $\text{CO}(\psi(\mathbf{x}))$  decides whether  $\psi(\mathbf{x})$  is **satisfiable** (i.e. whether it is **consistent**), using an oracle for SAT/SMT/MILP/CP/etc.

## Entailment

- Let  $\varphi$  represent some formula, defined on feature space  $\mathbb{F}$ , and representing a function  $\varphi : \mathbb{F} \rightarrow \{0, 1\}$
- Let  $\tau$  represent some other formula, also defined on  $\mathbb{F}$ , and with  $\tau : \mathbb{F} \rightarrow \{0, 1\}$

# Entailment

- Let  $\varphi$  represent some formula, defined on feature space  $\mathbb{F}$ , and representing a function  $\varphi : \mathbb{F} \rightarrow \{0, 1\}$
- Let  $\tau$  represent some other formula, also defined on  $\mathbb{F}$ , and with  $\tau : \mathbb{F} \rightarrow \{0, 1\}$ 
  - We say that  $\tau$  **entails**  $\varphi$ , written as  $\tau \vDash \varphi$ , if:

$$\forall(\mathbf{x} \in \mathbb{F}).[\tau(\mathbf{x}) \rightarrow \varphi(\mathbf{x})]$$

# Entailment

- Let  $\varphi$  represent some formula, defined on feature space  $\mathbb{F}$ , and representing a function  $\varphi : \mathbb{F} \rightarrow \{0, 1\}$
- Let  $\tau$  represent some other formula, also defined on  $\mathbb{F}$ , and with  $\tau : \mathbb{F} \rightarrow \{0, 1\}$ 
  - We say that  $\tau$  **entails**  $\varphi$ , written as  $\tau \vDash \varphi$ , if:

$$\forall(\mathbf{x} \in \mathbb{F}).[\tau(\mathbf{x}) \rightarrow \varphi(\mathbf{x})]$$

- We say that  $\tau(\mathbf{x})$  is **sufficient** for  $\varphi(\mathbf{x})$

# Entailment

- Let  $\varphi$  represent some formula, defined on feature space  $\mathbb{F}$ , and representing a function  $\varphi : \mathbb{F} \rightarrow \{0, 1\}$
- Let  $\tau$  represent some other formula, also defined on  $\mathbb{F}$ , and with  $\tau : \mathbb{F} \rightarrow \{0, 1\}$ 
  - We say that  $\tau$  entails  $\varphi$ , written as  $\tau \vDash \varphi$ , if:

$$\forall(\mathbf{x} \in \mathbb{F}).[\tau(\mathbf{x}) \rightarrow \varphi(\mathbf{x})]$$

- We say that  $\tau(\mathbf{x})$  is sufficient for  $\varphi(\mathbf{x})$
- To decide entailment:
  - $\tau \vDash \varphi$  if  $\tau(\mathbf{x}) \wedge \neg\varphi(\mathbf{x})$  is not consistent, i.e.  $\text{CO}(\tau(\mathbf{x}) \wedge \neg\varphi(\mathbf{x}))$  does not hold

# Entailment

- Let  $\varphi$  represent some formula, defined on feature space  $\mathbb{F}$ , and representing a function  $\varphi : \mathbb{F} \rightarrow \{0, 1\}$
- Let  $\tau$  represent some other formula, also defined on  $\mathbb{F}$ , and with  $\tau : \mathbb{F} \rightarrow \{0, 1\}$ 
  - We say that  $\tau$  entails  $\varphi$ , written as  $\tau \vDash \varphi$ , if:

$$\forall(\mathbf{x} \in \mathbb{F}).[\tau(\mathbf{x}) \rightarrow \varphi(\mathbf{x})]$$

- We say that  $\tau(\mathbf{x})$  is sufficient for  $\varphi(\mathbf{x})$
- To decide entailment:
  - $\tau \vDash \varphi$  if  $\tau(\mathbf{x}) \wedge \neg\varphi(\mathbf{x})$  is not consistent, i.e.  $\text{CO}(\tau(\mathbf{x}) \wedge \neg\varphi(\mathbf{x}))$  does not hold
- An example:
  - $\mathbb{F} = \{0, 1\}^2$
  - $\varphi(x_1, x_2) = x_1 \vee \neg x_2$
  - Clearly,  $x_1 \vDash \varphi$  and  $\neg x_2 \vDash \varphi$
  - Also,  $\text{CO}(x_1 \wedge (\neg x_1 \wedge x_2))$  does not hold

# Entailment

- Let  $\varphi$  represent some formula, defined on feature space  $\mathbb{F}$ , and representing a function  $\varphi : \mathbb{F} \rightarrow \{0, 1\}$
- Let  $\tau$  represent some other formula, also defined on  $\mathbb{F}$ , and with  $\tau : \mathbb{F} \rightarrow \{0, 1\}$ 
  - We say that  $\tau$  entails  $\varphi$ , written as  $\tau \vDash \varphi$ , if:

$$\forall(\mathbf{x} \in \mathbb{F}).[\tau(\mathbf{x}) \rightarrow \varphi(\mathbf{x})]$$

- We say that  $\tau(\mathbf{x})$  is sufficient for  $\varphi(\mathbf{x})$
- To decide entailment:
  - $\tau \vDash \varphi$  if  $\tau(\mathbf{x}) \wedge \neg\varphi(\mathbf{x})$  is not consistent, i.e.  $\text{CO}(\tau(\mathbf{x}) \wedge \neg\varphi(\mathbf{x}))$  does not hold

- An example:
  - $\mathbb{F} = \{0, 1\}^2$
  - $\varphi(x_1, x_2) = x_1 \vee \neg x_2$
  - Clearly,  $x_1 \vDash \varphi$  and  $\neg x_2 \vDash \varphi$
  - Also,  $\text{CO}(x_1 \wedge (\neg x_1 \wedge x_2))$  does not hold

- Another example:
  - $\mathbb{F} = \{0, 1\}^3$
  - $\varphi(x_1, x_2, x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3$
  - Clearly,  $x_1 \wedge x_2 \vDash \varphi$  and  $x_1 \wedge x_3 \vDash \varphi$
  - Also,  $\text{CO}(x_1 \wedge x_2 \wedge ((\neg x_1 \vee \neg x_2) \wedge (\neg x_1 \vee \neg x_3)))$  does not hold

## Entailment & explanations – how do we construct explanations?

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

- Instance:  $((0, 1, 0, 0), 1)$

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

## Entailment & explanations – how do we construct explanations?

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

- Instance:  $((0, 1, 0, 0), 1)$

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

- Localized explanation:** any irreducible conjunction of literals, consistent with  $\mathbf{v}$ , and that entails the prediction

## Entailment & explanations – how do we construct explanations?

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

- Instance:  $((0, 1, 0, 0), 1)$

- Localized explanation:** any irreducible conjunction of literals, consistent with  $\mathbf{v}$ , and that entails the prediction

- Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ ,

IF  $(x_1 = 0) \wedge (x_3 = 0)$  THEN  $\kappa(\mathbf{x}) = 1$

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

## Entailment & explanations – how do we construct explanations?

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

- Instance:  $((0, 1, 0, 0), 1)$

- Localized explanation:** any irreducible conjunction of literals, consistent with  $\mathbf{v}$ , and that entails the prediction

- Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ ,

IF  $(x_1 = 0) \wedge (x_3 = 0)$  THEN  $\kappa(\mathbf{x}) = 1$

- Global explanation:** any irreducible conjunction of literals, that is consistent, and that entails the prediction

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

# Entailment & explanations – how do we construct explanations?

- Classification function:

$$\kappa(x_1, x_2, x_3, x_4) = \neg x_1 \wedge \neg x_2 \vee x_1 \wedge x_2 \wedge x_4 \vee \neg x_1 \wedge x_2 \wedge \neg x_3 \vee \neg x_2 \wedge x_3 \wedge x_4$$

- Instance:  $((0, 1, 0, 0), 1)$

- Localized explanation:** any irreducible conjunction of literals, consistent with  $\mathbf{v}$ , and that entails the prediction

- Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ ,

IF  $(x_1 = 0) \wedge (x_3 = 0)$  THEN  $\kappa(\mathbf{x}) = 1$

- Global explanation:** any irreducible conjunction of literals, that is consistent, and that entails the prediction

- Given  $\mathbf{x} = (x_1, x_2, x_3, x_4)$ ,

IF  $(x_1 = 0) \wedge (x_2 = 0)$  THEN  $\kappa(\mathbf{x}) = 1$

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\mathbf{x})$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

# Outline – Unit #01

ML Models: Classification & Regression Problems

Basics of (non-symbolic) XAI

Motivation for Explanations

Brief Glimpse of Logic

Reasoning About ML Models

Understanding Intrinsic Interpretability

# Decision sets with boolean features

- Example ML model:

Features:  $x_1, x_2, x_3, x_4 \in \{0, 1\}$  (boolean)

Rules:

IF	$x_1 \wedge \neg x_2 \wedge x_3$	THEN	predict $\boxplus$
IF	$x_1 \wedge \neg x_3 \wedge x_4$	THEN	predict $\boxminus$
IF	$x_3 \wedge x_4$	THEN	predict $\boxminus$

## Decision sets with boolean features

- Example ML model:

Features:  $x_1, x_2, x_3, x_4 \in \{0, 1\}$  (boolean)

Rules:

IF	$x_1 \wedge \neg x_2 \wedge x_3$	THEN	predict $\blacksquare$
IF	$x_1 \wedge \neg x_3 \wedge x_4$	THEN	predict $\square$
IF	$x_3 \wedge x_4$	THEN	predict $\square$

- Q:** Can the model predict both  $\blacksquare$  and  $\square$  for some instance, i.e. is there overlap?

## Decision sets with boolean features

- Example ML model:

Features:  $x_1, x_2, x_3, x_4 \in \{0, 1\}$  (boolean)

Rules:

IF	$x_1 \wedge \neg x_2 \wedge x_3$	THEN	predict $\blacksquare$
IF	$x_1 \wedge \neg x_3 \wedge x_4$	THEN	predict $\square$
IF	$x_3 \wedge x_4$	THEN	predict $\square$

- Q:** Can the model predict both  $\blacksquare$  and  $\square$  for some instance, i.e. is there overlap?

- Yes, certainly: pick  $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$

# Decision sets with boolean features

- Example ML model:

Features:  $x_1, x_2, x_3, x_4 \in \{0, 1\}$  (boolean)

Rules:

IF	$x_1 \wedge \neg x_2 \wedge x_3$	THEN	predict $\blacksquare$
IF	$x_1 \wedge \neg x_3 \wedge x_4$	THEN	predict $\square$
IF	$x_3 \wedge x_4$	THEN	predict $\square$

- Q: Can the model predict both  $\blacksquare$  and  $\square$  for some instance, i.e. is there overlap?

- Yes, certainly: pick  $(x_1, x_2, x_3, x_4) = (1, 0, 1, 1)$
- A formalization:

$$\begin{aligned}y_{p,1} &\leftrightarrow (x_1 \wedge \neg x_2 \wedge x_3) \wedge \\y_{n,1} &\leftrightarrow (x_1 \wedge \neg x_3 \wedge x_4) \wedge \\y_{n,2} &\leftrightarrow (x_3 \wedge x_4) \wedge (y_p \leftrightarrow y_{p,1}) \wedge \\(y_n &\leftrightarrow (y_{n,1} \vee y_{n,2})) \wedge (y_p) \wedge (y_n)\end{aligned}$$

... and solve with SAT solver (after clausification)

Or use PySAT

[Tse68, PG86]

[IMM18]

$\therefore$  There exists a model iff there exists a point in feature space yielding both predictions

## Decision sets with ordinal features

- Example ML model:

Features:  $x_1, x_2 \in \{0, 1, 2\}$  (integer)

Rules:

IF  $2x_1 + x_2 > 0$  THEN predict  $\text{田}$

IF  $2x_1 - x_2 \leq 0$  THEN predict  $\text{□}$

## Decision sets with ordinal features

- Example ML model:

Features:  $x_1, x_2 \in \{0, 1, 2\}$  (integer)

Rules:

IF  $2x_1 + x_2 > 0$  THEN predict  $\text{■}$

IF  $2x_1 - x_2 \leq 0$  THEN predict  $\text{□}$

- Q:** Can the model predict both  $\text{■}$  and  $\text{□}$  for some instance, i.e. is there overlap?

## Decision sets with ordinal features

- Example ML model:

Features:  $x_1, x_2 \in \{0, 1, 2\}$  (integer)

Rules:

IF  $2x_1 + x_2 > 0$  THEN predict  $\text{■}$

IF  $2x_1 - x_2 \leq 0$  THEN predict  $\text{□}$

- Q:** Can the model predict both  $\text{■}$  and  $\text{□}$  for some instance, i.e. is there overlap?

- Yes, of course: pick  $x_1 = 0$  and  $x_2 = 1$

# Decision sets with ordinal features

- Example ML model:

Features:  $x_1, x_2 \in \{0, 1, 2\}$  (integer)

Rules:

IF  $2x_1 + x_2 > 0$  THEN predict  $\text{■}$

IF  $2x_1 - x_2 \leq 0$  THEN predict  $\text{□}$

- Q: Can the model predict both  $\text{■}$  and  $\text{□}$  for some instance, i.e. is there overlap?

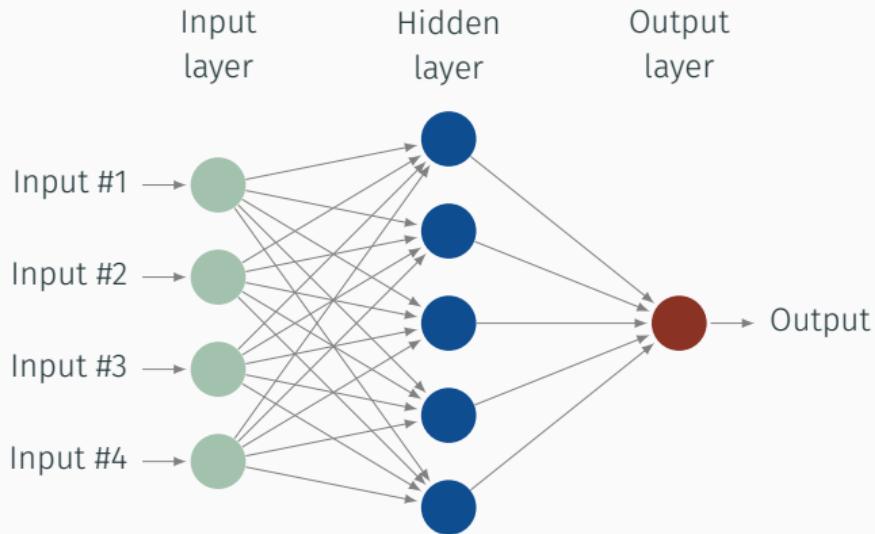
- Yes, of course: pick  $x_1 = 0$  and  $x_2 = 1$
- A formalization:

$$y_p \leftrightarrow (2x_1 + x_2 > 0) \wedge y_n \leftrightarrow (2x_1 - x_2 \leq 0) \wedge (y_p) \wedge (y_n)$$

... and solve with SMT solver (many alternatives)

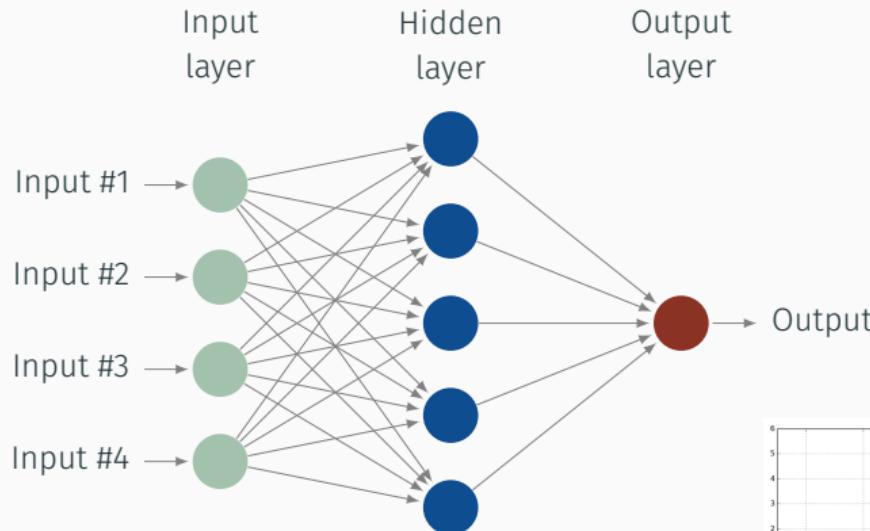
∴ There exists a model iff there exists a point in feature space yielding both predictions

# Neural networks

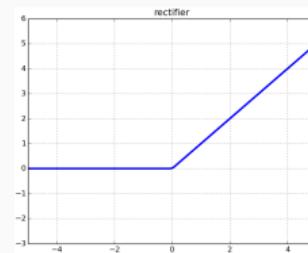


- Each layer (except first) viewed as a **block**, and
  - Compute  $\mathbf{x}'$  given input  $\mathbf{x}$ , weights matrix  $\mathbf{A}$ , and bias vector  $\mathbf{b}$
  - Compute output  $\mathbf{y}$  given  $\mathbf{x}'$  and activation function

# Neural networks



- Each layer (except first) viewed as a **block**, and
  - Compute  $\mathbf{x}'$  given input  $\mathbf{x}$ , weights matrix  $\mathbf{A}$ , and bias vector  $\mathbf{b}$
  - Compute output  $\mathbf{y}$  given  $\mathbf{x}'$  and activation function
- Each unit uses a **ReLU** activation function



[NH10]

## Encoding NNs using MILP

Computation for a NN ReLU **block**, in two steps:

$$\mathbf{x}' = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$$

$$\mathbf{y} = \max(\mathbf{x}', \mathbf{0})$$

# Encoding NNs using MILP

Computation for a NN ReLU **block**, in two steps:

$$\mathbf{x}' = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$$

$$\mathbf{y} = \max(\mathbf{x}', \mathbf{0})$$

Encoding each **block**:

[FJ18]

$$\sum_{j=1}^n a_{i,j}x_j + b_i = y_i - s_i$$

$$z_i = 1 \rightarrow y_i \leq 0$$

$$z_i = 0 \rightarrow s_i \leq 0$$

$$y_i \geq 0, s_i \geq 0, z_i \in \{0, 1\}$$

Simpler encodings exist, but **not** as effective

[KBD<sup>+</sup>17]

# Encoding NNs using MILP

Computation for a NN ReLU **block**, in two steps:

$$\mathbf{x}' = \mathbf{A} \cdot \mathbf{x} + \mathbf{b}$$

$$\mathbf{y} = \max(\mathbf{x}', \mathbf{0})$$

Modeling ML models  
with logic is not only  
possible but also **simple !**

Encoding each **block**:

[FJ18]

$$\sum_{j=1}^n a_{i,j}x_j + b_i = y_i - s_i$$

$$z_i = 1 \rightarrow y_i \leq 0$$

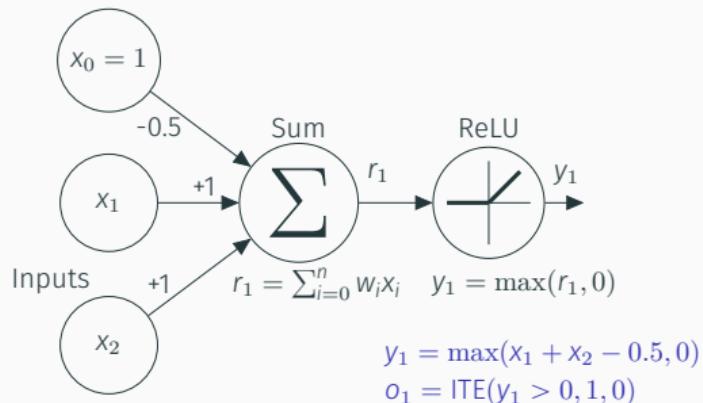
$$z_i = 0 \rightarrow s_i \leq 0$$

$$y_i \geq 0, s_i \geq 0, z_i \in \{0, 1\}$$

Simpler encodings exist, but **not** as effective

[KBD<sup>+</sup>17]

## Example – encoding a simple NN in MILP



$x_1$	$x_2$	$r_1$	$y_1$	$o_1$
0	0	-0.5	0	0
1	0	0.5	0.5	1
0	1	0.5	0.5	1
1	1	1.5	1.5	1

MILP encoding:

$$x_1 + x_2 - 0.5 = y_1 - s_1$$

$$z_1 = 1 \rightarrow y_1 \leq 0$$

$$z_1 = 0 \rightarrow s_1 \leq 0$$

$$o_1 = (y_1 > 0)$$

$$x_1, x_2, z_1, o_1 \in \{0, 1\}$$

$$y_1, s_1 \geq 0$$

Instance:  $(\mathbf{x}, c) = ((1, 0), 1)$

$$1 + 0 - 0.5 = 0.5 - 0$$

$$1 \vee 0.5 \leq 0$$

$$0 \vee 0 \leq 0$$

$$1 = (0.5 > 0)$$

$$x_1 = 1, x_2 = 0, z_1 = 0, o_1 = 1$$

$$y_1 = 0.5, s_1 = 0$$

Checking:  $\mathbf{x} = (0, 0)$

$$0 + 0 - 0.5 = 0 - 0.5$$

$$0 \vee 0 \leq 0$$

$$1 \vee 0.5 \leq 0$$

$$0 = (0 > 0)$$

$$x_1 = 0, x_2 = 0, z_1 = 1, o_1 = 0$$

$$y_1 = 0, s_1 = 0.5$$

# Outline – Unit #01

ML Models: Classification & Regression Problems

Basics of (non-symbolic) XAI

Motivation for Explanations

Brief Glimpse of Logic

Reasoning About ML Models

Understanding Intrinsic Interpretability

# What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*

[Rud19, Mol20, RCC<sup>+</sup>22, Rud22]

# What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But:** definition of *interpretability* is rather *subjective*...

[Rud19, Mol20, RCC<sup>+</sup>22, Rud22]

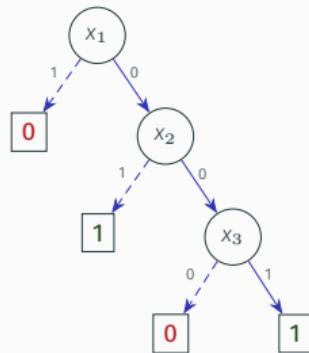
[Lip18]

# What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But:** definition of *interpretability* is rather *subjective*...

[Rud19, Mol20, RCC<sup>+</sup>22, Rud22]

[Lip18]



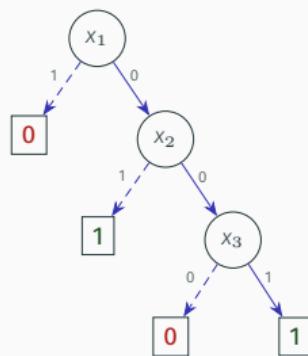
# What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But:** definition of *interpretability* is rather *subjective*...

[Rud19, Mol20, RCC<sup>+</sup>22, Rud22]

[Lip18]

- What is an explanation for  $((0, 0, 1), 1)$ ?

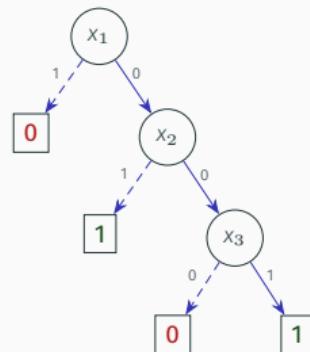


# What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But:** definition of *interpretability* is rather *subjective*...

[Rud19, Mol20, RCC<sup>+</sup>22, Rud22]

[Lip18]



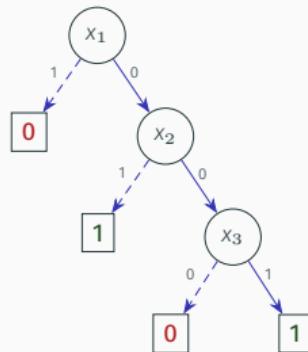
- What is an explanation for  $((0, 0, 1), 1)$ ?
- Clearly, IF  $\neg x_1 \wedge \neg x_2 \wedge x_3$  THEN  $\kappa(\mathbf{x}) = 1$

# What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But:** definition of *interpretability* is rather *subjective*...

[Rud19, Mol20, RCC<sup>+</sup>22, Rud22]

[Lip18]



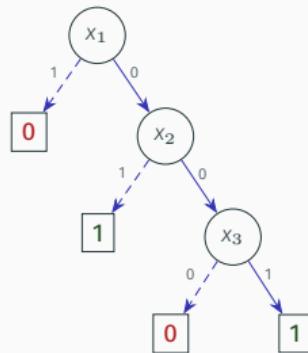
- What is an explanation for  $((0, 0, 1), 1)$ ?
- Clearly, IF  $\neg x_1 \wedge \neg x_2 \wedge x_3$  THEN  $\kappa(\mathbf{x}) = 1$ 
  - $\{\neg x_1, \neg x_2, x_3\}$  or  $\{1, 2, 3\}$  is an explanation

# What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But:** definition of *interpretability* is rather *subjective...*

[Rud19, Mol20, RCC<sup>+</sup>22, Rud22]

[Lip18]



- What is an explanation for  $((0, 0, 1), 1)$ ?
- Clearly, IF  $\neg x_1 \wedge \neg x_2 \wedge x_3$  THEN  $\kappa(\mathbf{x}) = 1$ 
  - $\{\neg x_1, \neg x_2, x_3\}$  or  $\{1, 2, 3\}$  is an explanation

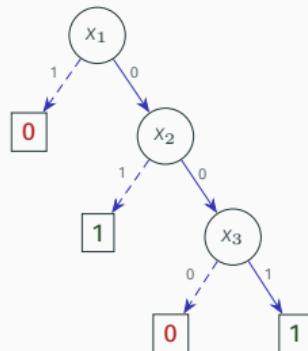
Really?

# What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But:** definition of *interpretability* is rather *subjective*...

[Rud19, Mol20, RCC<sup>+</sup>22, Rud22]

[Lip18]



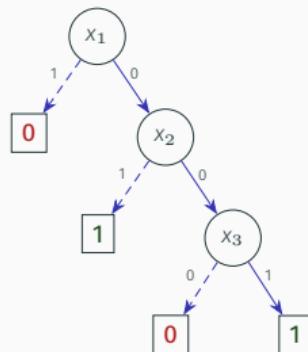
- What is an explanation for  $((0, 0, 1), 1)$ ?
- Clearly, IF  $\neg x_1 \wedge \neg x_2 \wedge x_3$  THEN  $\kappa(\mathbf{x}) = 1$ 
  - $\{\neg x_1, \neg x_2, x_3\}$  or  $\{1, 2, 3\}$  is a *weak* explanation!
- It is the case that: IF  $\neg x_1 \wedge x_3$  THEN  $\kappa(\mathbf{x}) = 1$   
 $\therefore \{1, 3\}$  is also *sufficient* for the prediction!

# What is intrinsic interpretability?

- Goal is to deploy *interpretable* ML models
  - E.g. Decision trees, decision lists, decision sets, etc.
- The explanation is the model itself, because it is *interpretable*
- **But:** definition of *interpretability* is rather *subjective*...

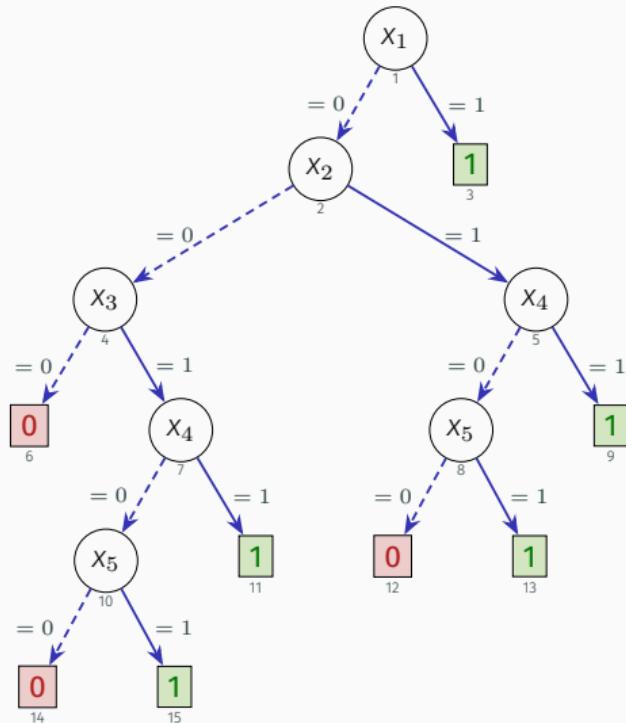
[Rud19, Mol20, RCC<sup>+</sup>22, Rud22]

[Lip18]



- What is an explanation for  $((0, 0, 1), 1)$ ?
- Clearly, IF  $\neg x_1 \wedge \neg x_2 \wedge x_3$  THEN  $\kappa(\mathbf{x}) = 1$ 
  - $\{\neg x_1, \neg x_2, x_3\}$  or  $\{1, 2, 3\}$  is a **weak** explanation!
- It is the case that: IF  $\neg x_1 \wedge x_3$  THEN  $\kappa(\mathbf{x}) = 1$ 
  - $\{1, 3\}$  is also **sufficient** for the prediction!
  - $\{1, 3\}$  is easier to grasp; also, it is **irreducible**

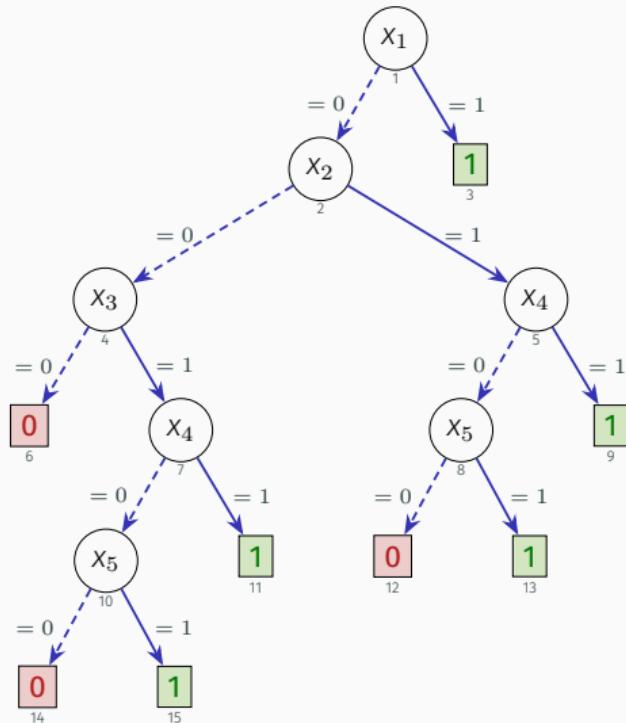
# Are *interpretable* models really interpretable? – DTs



- Case of **optimal** decision tree (DT)
- Explanation for  $(0, 0, 1, 0, 1)$ , with prediction 1?

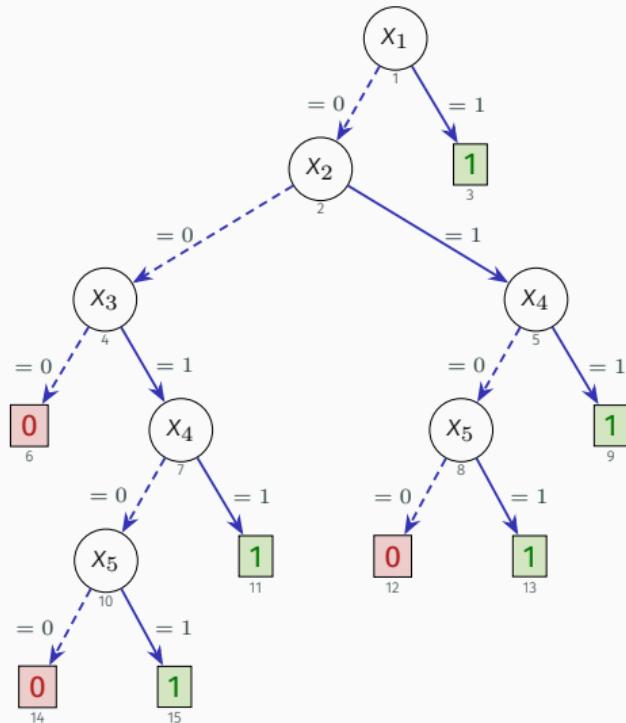
[HRS19]

# Are *interpretable* models really interpretable? – DTs



- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for  $(0, 0, 1, 0, 1)$ , with prediction 1?
  - Clearly, IF  $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5$  THEN  $\kappa(x) = 1$

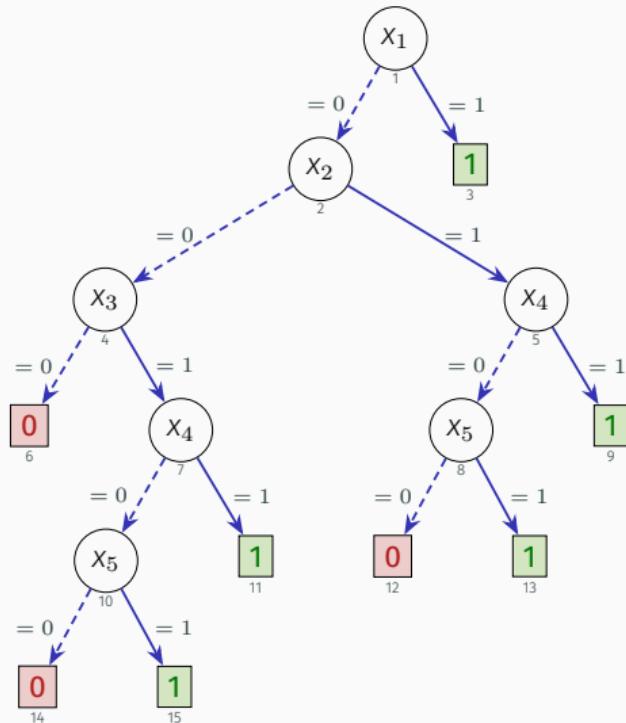
# Are *interpretable* models really interpretable? – DTs



- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for  $(0, 0, 1, 0, 1)$ , with prediction 1?
  - Clearly, IF  $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5$  THEN  $\kappa(\mathbf{x}) = 1$
  - But,  $x_1, x_2, x_4$  are **irrelevant** for the prediction:

$x_3$	$x_5$	$x_1$	$x_2$	$x_4$	$\kappa(\mathbf{x})$
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

# Are *interpretable* models really interpretable? – DTs



- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for  $(0, 0, 1, 0, 1)$ , with prediction 1?
  - Clearly, IF  $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5$  THEN  $\kappa(\mathbf{x}) = 1$
  - But,  $x_1, x_2, x_4$  are **irrelevant** for the prediction:

$x_3$	$x_5$	$x_1$	$x_2$	$x_4$	$\kappa(\mathbf{x})$
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

$\therefore$  fixing  $\{3, 5\}$  suffices for the prediction  
Compare with  $\{1, 2, 3, 4, 5\}...$

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires
- What is an explanation for the prediction?

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires
- What is an explanation for the prediction?
- Fixing  $\{3, 4, 6\}$  suffices for the prediction

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires
- What is an explanation for the prediction?
- Fixing  $\{3, 4, 6\}$  suffices for the prediction
  - Why?
    - We need 3 (or 1) so that  $R_1$  cannot fire
    - With 3, we do not need 2, since with 4 and 6 fixed, then  $R_4$  is guaranteed to fire

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires
- What is an explanation for the prediction?
- Fixing  $\{3, 4, 6\}$  suffices for the prediction
  - Why?
    - We need 3 (or 1) so that  $R_1$  cannot fire
    - With 3, we do not need 2, since with 4 and 6 fixed, then  $R_4$  is guaranteed to fire
  - Some questions:
    - Would average human decision maker be able to understand the irreducible set  $\{3, 4, 6\}$ ?

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires
- What is an explanation for the prediction?
- Fixing  $\{3, 4, 6\}$  suffices for the prediction
  - Why?
    - We need 3 (or 1) so that  $R_1$  cannot fire
    - With 3, we do not need 2, since with 4 and 6 fixed, then  $R_4$  is guaranteed to fire
  - Some questions:
    - Would average human decision maker be able to understand the irreducible set  $\{3, 4, 6\}$ ?
    - Would he/she be able to compute the set  $\{3, 4, 6\}$ , by manual inspection?

# Questions?

# Lecture 02

## Recapitulate first lecture

- ML models: classification & regression

## Recapitulate first lecture

- ML models: classification & regression
- Glimpse of heuristic XAI

## Recapitulate first lecture

- ML models: classification & regression
- Glimpse of heuristic XAI
- Answers to **Why?** questions as logic rules

## Recapitulate first lecture

- ML models: classification & regression
- Glimpse of heuristic XAI
- Answers to **Why?** questions as logic rules
- Logic-based reasoning of ML models

## Recapitulate first lecture

- ML models: classification & regression
- Glimpse of heuristic XAI
- Answers to **Why?** questions as logic rules
- Logic-based reasoning of ML models
- Apparent difficulties with explaining interpretable models

# Plan for this course

- Lecture 01 – unit(s):
  - #01: Foundations
- Lecture 02 – unit(s):
  - #02: Principles of symbolic XAI – **feature selection**
  - #03: Tractability in symbolic XAI (& myth of interpretability)
- Lecture 03 – unit(s):
  - #04: Intractability in symbolic XAI (& myth of model-agnostic XAI)
  - #05: Explainability queries
- Lecture 04 – unit(s):
  - #06: Recent, emerging & advanced topics
- Lecture 05 – unit(s):
  - #07: Principles of symbolic XAI – **feature attribution** (& myth of Shapley values in XAI)
  - #08: Corrected feature attribution – nuSHAP
  - #09: Conclusions & research directions

Unit #02

## Principles of Symbolic XAI – Feature Selection

## Outline – Unit #02

---

Definitions of Explanations

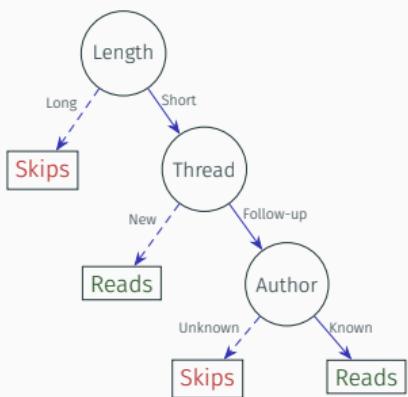
Duality Properties

Computational Problems

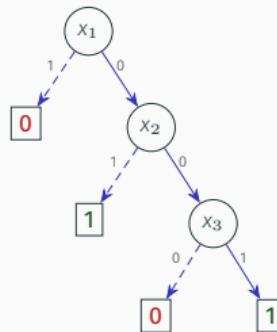
# What is an explanation?

- Notation:

Original DT [PM17]



Rewritten DT



## Mapping

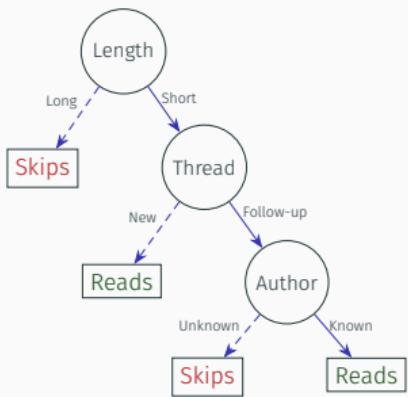
$x_1 = 1$  iff Length = Long  
 $x_2 = 1$  iff Thread = New  
 $x_3 = 1$  iff Author = Known  
 $\kappa(\cdot) = 1$  iff  $\kappa'(\dots) = \text{Reads}$   
 $\kappa(\cdot) = 0$  iff  $\kappa'(\dots) = \text{Skips}$

- What is an explanation?

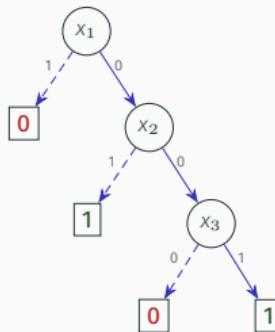
# What is an explanation?

- Notation:

Original DT [PM17]



Rewritten DT



---

Mapping

---

$x_1 = 1$  iff Length = Long  
 $x_2 = 1$  iff Thread = New  
 $x_3 = 1$  iff Author = Known  
 $\kappa(\cdot) = 1$  iff  $\kappa'(\dots) = \text{Reads}$   
 $\kappa(\cdot) = 0$  iff  $\kappa'(\dots) = \text{Skips}$

---

- What is an explanation?

- Answer to question “**Why** (the prediction)?” is a rule:

IF <COND> THEN  $\kappa(x) = c$

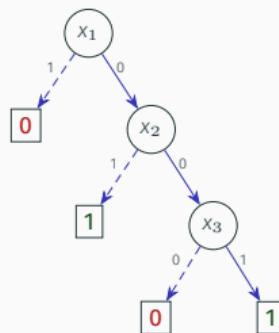
# What is an explanation?

- Notation:

Original DT [PM17]



Rewritten DT



## Mapping

$x_1 = 1$  iff Length = Long  
 $x_2 = 1$  iff Thread = New  
 $x_3 = 1$  iff Author = Known  
 $\kappa(\cdot) = 1$  iff  $\kappa'(\cdot \cdot \cdot) = \text{Reads}$   
 $\kappa(\cdot) = 0$  iff  $\kappa'(\cdot \cdot \cdot) = \text{Skips}$

- What is an explanation?

- Answer to question “**Why** (the prediction)?” is a rule:  
IF <COND> THEN  $\kappa(x) = c$
- Explanation:** set of literals (or just features) in <COND>; irreducibility matters!

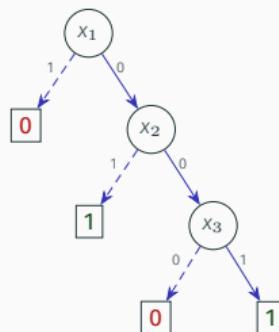
# What is an explanation?

- Notation:

Original DT [PM17]



Rewritten DT



## Mapping

$x_1 = 1$  iff Length = Long  
 $x_2 = 1$  iff Thread = New  
 $x_3 = 1$  iff Author = Known  
 $\kappa(\cdot) = 1$  iff  $\kappa'(\cdot \cdot \cdot) = \text{Reads}$   
 $\kappa(\cdot) = 0$  iff  $\kappa'(\cdot \cdot \cdot) = \text{Skips}$

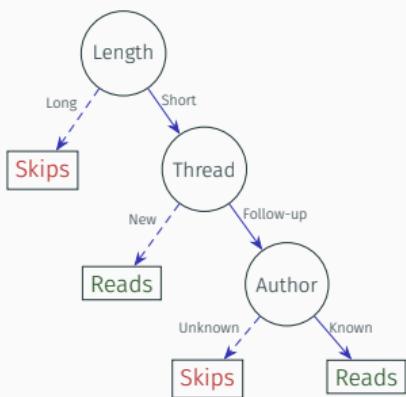
- What is an explanation?

- Answer to question “**Why** (the prediction)?” is a **rule**:  $\text{IF } <\text{COND}> \text{ THEN } \kappa(x) = c$
- Explanation**: set of **literals** (or just **features**) in **<COND>**; **irreducibility matters!**
- E.g.: explanation for  $v = (\neg x_1, \neg x_2, x_3)$ ?

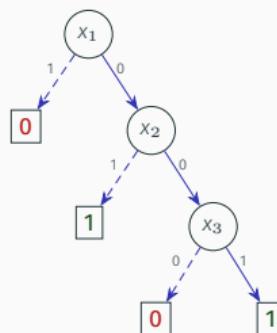
# What is an explanation?

- Notation:

Original DT [PM17]



Rewritten DT



## Mapping

$x_1 = 1$  iff Length = Long  
 $x_2 = 1$  iff Thread = New  
 $x_3 = 1$  iff Author = Known  
 $\kappa(\cdot) = 1$  iff  $\kappa'(\cdot \cdot \cdot) = \text{Reads}$   
 $\kappa(\cdot) = 0$  iff  $\kappa'(\cdot \cdot \cdot) = \text{Skips}$

- What is an explanation?

- Answer to question “**Why** (the prediction)?” is a **rule**:  $\text{IF } <\text{COND}> \text{ THEN } \kappa(\mathbf{x}) = c$
- Explanation**: set of **literals** (or just **features**) in **<COND>**; **irreducibility matters!**
- E.g.: explanation for  $\mathbf{v} = (\neg x_1, \neg x_2, x_3)$ ?
  - It is the case that, **IF**  $\neg x_1 \wedge \neg x_2 \wedge x_3$  **THEN**  $\kappa(\mathbf{x}) = 1$

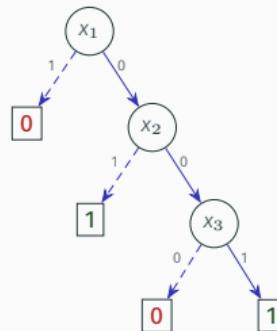
# What is an explanation?

- Notation:

Original DT [PM17]



Rewritten DT



## Mapping

$x_1 = 1$  iff Length = Long  
 $x_2 = 1$  iff Thread = New  
 $x_3 = 1$  iff Author = Known  
 $\kappa(\cdot) = 1$  iff  $\kappa'(\cdot \cdot \cdot) = \text{Reads}$   
 $\kappa(\cdot) = 0$  iff  $\kappa'(\cdot \cdot \cdot) = \text{Skips}$

- What is an explanation?

- Answer to question “**Why** (the prediction)?” is a **rule**:  $\text{IF } <\text{COND}> \text{ THEN } \kappa(\mathbf{x}) = c$
- Explanation**: set of **literals** (or just **features**) in **<COND>**; **irreducibility matters!**
- E.g.: explanation for  $\mathbf{v} = (\neg x_1, \neg x_2, x_3)$ ?
  - It is the case that, **IF**  $\neg x_1 \wedge \neg x_2 \wedge x_3$  **THEN**  $\kappa(\mathbf{x}) = 1$
  - One possible explanation is  $\{\neg x_1, \neg x_2, x_3\}$  or simply  $\{1, 2, 3\}$

# The similarity predicate

[Mar24]

- Recall ML models for classification & regression:

- Classification:  $\mathcal{M}_C = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$
- Regression:  $\mathcal{M}_R = (\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$
- General:  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathbb{T}, \tau)$

# The similarity predicate

[Mar24]

- Recall ML models for classification & regression:
  - Classification:  $\mathcal{M}_C = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$
  - Regression:  $\mathcal{M}_R = (\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$
  - General:  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathbb{T}, \tau)$
- **Similarity predicate:**  $\sigma : \mathbb{F} \rightarrow \{\top, \perp\}$ 
  - Classification:  $\sigma(\mathbf{x}) := [\kappa(\mathbf{x}) = \kappa(\mathbf{v})]$ 
    - **Obs:** For boolean classifiers, no need for  $\sigma$
  - Regression:  $\sigma(\mathbf{x}) := [|\rho(\mathbf{x}) - \rho(\mathbf{v})| \leq \delta]$ , where  $\delta$  is user-specified

# The similarity predicate

[Mar24]

- Recall ML models for classification & regression:
  - Classification:  $\mathcal{M}_C = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$
  - Regression:  $\mathcal{M}_R = (\mathcal{F}, \mathbb{F}, \mathbb{V}, \rho)$
  - General:  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathbb{T}, \tau)$
- **Similarity predicate:**  $\sigma : \mathbb{F} \rightarrow \{\top, \perp\}$ 
  - Classification:  $\sigma(\mathbf{x}) := [\kappa(\mathbf{x}) = \kappa(\mathbf{v})]$ 
    - **Obs:** For boolean classifiers, no need for  $\sigma$
  - Regression:  $\sigma(\mathbf{x}) := [|\rho(\mathbf{x}) - \rho(\mathbf{v})| \leq \delta]$ , where  $\delta$  is user-specified
- Bottom line:  
Reason about symbolic explainability by abstracting away type of ML model

## Abductive explanations – answering Why? questions

- Instance  $(\mathbf{v}, q)$ , i.e.  $c = \tau(\mathbf{v})$

## Abductive explanations – answering Why? questions

- Instance  $(\mathbf{v}, q)$ , i.e.  $c = \tau(\mathbf{v})$
- **Abductive explanation** ( $\text{AX}_p$ , PI-explanation):
  - Subset-minimal set of features  $\mathcal{X} \subseteq \mathcal{F}$  sufficient for ensuring prediction

[SCD18, INM19a]

$$\forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$$

## Abductive explanations – answering Why? questions

- Instance  $(\mathbf{v}, q)$ , i.e.  $c = \tau(\mathbf{v})$
- **Abductive explanation** ( $\text{AXp}$ , PI-explanation):
  - Subset-minimal set of features  $\mathcal{X} \subseteq \mathcal{F}$  sufficient for ensuring prediction

[SCD18, INM19a]

$$\text{WAXp}(\mathcal{X}) := \forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$$

- Defining  $\text{AXp}$  (from weak  $\text{AXps}$ ,  $\text{WAXps}$ ):

$$\text{AXp}(\mathcal{X}) := \text{WAXp}(\mathcal{X}) \wedge \forall (\mathcal{X}' \subsetneq \mathcal{X}). \neg \text{WAXp}(\mathcal{X}')$$

## Abductive explanations – answering Why? questions

- Instance  $(\mathbf{v}, q)$ , i.e.  $c = \tau(\mathbf{v})$
- **Abductive explanation** ( $\text{AXp}$ , PI-explanation):
  - Subset-minimal set of features  $\mathcal{X} \subseteq \mathcal{F}$  sufficient for ensuring prediction

[SCD18, INM19a]

$$\text{WAXp}(\mathcal{X}) \quad := \quad \forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$$

- Defining  $\text{AXp}$  (from weak  $\text{AXps}$ ,  $\text{WAXps}$ ):

$$\text{AXp}(\mathcal{X}) := \text{WAXp}(\mathcal{X}) \wedge \forall (\mathcal{X}' \subsetneq \mathcal{X}). \neg \text{WAXp}(\mathcal{X}')$$

- But,  $\text{WAXp}$  is **monotone**; hence,

$$\text{AXp}(\mathcal{X}) := \text{WAXp}(\mathcal{X}) \wedge \forall (t \in \mathcal{X}). \neg \text{WAXp}(\mathcal{X} \setminus \{t\})$$

# Abductive explanations – answering Why? questions

- Instance  $(\mathbf{v}, q)$ , i.e.  $c = \tau(\mathbf{v})$
- Abductive explanation (AXp, PI-explanation):
  - Subset-minimal set of features  $\mathcal{X} \subseteq \mathcal{F}$  sufficient for ensuring prediction

[SCD18, INM19a]

$$\text{WAXp}(\mathcal{X}) \quad := \quad \forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$$

- Defining AXp (from weak AXps, WAXps):

$$\text{AXp}(\mathcal{X}) := \text{WAXp}(\mathcal{X}) \wedge \forall (\mathcal{X}' \subsetneq \mathcal{X}). \neg \text{WAXp}(\mathcal{X}')$$

- But, WAXp is monotone; hence,

$$\text{AXp}(\mathcal{X}) := \text{WAXp}(\mathcal{X}) \wedge \forall (t \in \mathcal{X}). \neg \text{WAXp}(\mathcal{X} \setminus \{t\})$$

- Finding one AXp (example algorithm; many more exist):

[MM20]

- Let  $\mathcal{X} = \mathcal{F}$ , i.e. fix all features
- Invariant:  $\text{WAXp}(\mathcal{X})$  must hold. Why?
- Analyze features in any order, one feature  $i$  at a time
  - If  $\text{WAXp}(\mathcal{X} \setminus \{i\})$  holds, then remove  $i$  from  $\mathcal{X}$ , i.e.  $i$  becomes free

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)?$

Recap weak AXp:  $\forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**

Recap weak AXp:  $\forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ?

Recap weak AXp:  $\forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**

Recap weak AXp:  $\forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ?

Recap weak AXp:  $\forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**

Recap weak AXp:  $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 4 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \top \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ?

Recap weak AXp:  $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 4 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \top \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **No**

Recap weak AXp:  $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 4 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \top \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **No**
- AXp  $\mathcal{X} = \{4\}$

Recap weak AXp:  $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 4 be removed, i.e.  $\forall(\mathbf{x} \in \{0, 1\}^4). \top \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **No**
- AXp  $\mathcal{X} = \{4\}$
- In general, **validity/consistency checked with SAT/SMT/MILP/CP reasoners**

Recap weak AXp:  $\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## A simple example – AXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$ . **AXp?**
- Define  $\mathcal{X} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \neg x_2 \wedge \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \neg x_3 \wedge x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). x_4 \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 4 be removed, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^4). \top \rightarrow \kappa(x_1, x_2, x_3, x_4)$ ? **No**
- AXp  $\mathcal{X} = \{4\}$
- In general, **validity/consistency checked with SAT/SMT/MILP/CP reasoners**
  - **Obs:** for some classes of classifiers, poly-time algorithms exist

Recap weak AXp:  $\forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$

## More notation

- Notation  $\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}$ :

$$[\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \equiv \bigwedge_{i \in \mathcal{S}} (x_i = v_i)$$

## More notation

- Notation  $\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}$ :

$$[\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \equiv \bigwedge_{i \in \mathcal{S}} (x_i = v_i)$$

- Definition of  $\Upsilon(\mathcal{S})$ :

$$\Upsilon(\mathcal{S}) := \{\mathbf{x} \in \mathbb{F} \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}\}$$

## More notation

- Notation  $\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}$ :

$$[\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \equiv \bigwedge_{i \in \mathcal{S}} (x_i = v_i)$$

- Definition of  $\Upsilon(\mathcal{S})$ :

$$\Upsilon(\mathcal{S}) := \{\mathbf{x} \in \mathbb{F} \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}\}$$

- Expected value, non-real-valued features:

$$\mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] := \frac{1}{|\Upsilon(\mathcal{S}; \mathbf{v})|} \sum_{\mathbf{x} \in \Upsilon(\mathcal{S}; \mathbf{v})} \tau(\mathbf{x})$$

## More notation

- Notation  $\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}$ :

$$[\mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] \equiv \bigwedge_{i \in \mathcal{S}} (x_i = v_i)$$

- Definition of  $\Upsilon(\mathcal{S})$ :

$$\Upsilon(\mathcal{S}) := \{\mathbf{x} \in \mathbb{F} \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}\}$$

- Expected value, non-real-valued features:

$$\mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] := \frac{1}{|\Upsilon(\mathcal{S}; \mathbf{v})|} \sum_{\mathbf{x} \in \Upsilon(\mathcal{S}; \mathbf{v})} \tau(\mathbf{x})$$

- Expected value, real-valued features:

$$\mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] := \frac{1}{|\Upsilon(\mathcal{S}; \mathbf{v})|} \int_{\Upsilon(\mathcal{S}; \mathbf{v})} \tau(\mathbf{x}) d\mathbf{x}$$

## Other definitions of WAXps/AXps

- Using probabilities, non-real-valued features:

[WMHK21, IH<sup>+</sup>22, ABOS22, IH<sup>+</sup>23]

$$\text{WAXp}(\mathcal{S}) \quad := \quad \Pr(\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) = 1$$

## Other definitions of WAXps/AXps

- Using probabilities, non-real-valued features:

[WMHK21, IH<sup>+</sup>22, ABOS22, IH<sup>+</sup>23]

$$\text{WAXp}(\mathcal{S}) := \Pr(\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) = 1$$

- Using expected values:

$$\text{WAXp}(\mathcal{S}) := \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1$$

## Other definitions of WAXps/AXps

- Using probabilities, non-real-valued features:

[WMHK21, IH<sup>+</sup>22, ABOS22, IH<sup>+</sup>23]

$$\text{WAXp}(\mathcal{S}) := \Pr(\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) = 1$$

- Using expected values:

$$\text{WAXp}(\mathcal{S}) := \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1$$

- Definition of AXp remains unchanged
  - This is true when comparing against 1

## Constrastive explanations – answering Why not? questions

- Instance  $(\mathbf{v}, c)$ , i.e.  $c = \kappa(\mathbf{v})$

## Constrastive explanations – answering Why not? questions

- Instance  $(\mathbf{v}, c)$ , i.e.  $c = \kappa(\mathbf{v})$
- **Contrastive explanation** (**CXp**):
  - Subset-minimal set of features  $\mathcal{Y} \subseteq \mathcal{F}$  sufficient for **changing** prediction

[Mil19, INAM20]

$$\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$$

## Constrastive explanations – answering Why not? questions

- Instance  $(\mathbf{v}, c)$ , i.e.  $c = \kappa(\mathbf{v})$
- **Contrastive explanation** ( $\text{CXp}$ ):

- Subset-minimal set of features  $\mathcal{Y} \subseteq \mathcal{F}$  sufficient for **changing** prediction

[Mil19, INAM20]

$$\text{WCXp}(\mathcal{Y}) := \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$$

- Defining  $\text{CXp}$ :

$$\text{CXp}(\mathcal{Y}) := \text{WCXp}(\mathcal{Y}) \wedge \forall (\mathcal{Y}' \subsetneq \mathcal{Y}). \neg \text{WCXp}(\mathcal{Y}')$$

# Constrastive explanations – answering Why not? questions

- Instance  $(\mathbf{v}, c)$ , i.e.  $c = \kappa(\mathbf{v})$
- Contrastive explanation ( $\text{CXp}$ ):

- Subset-minimal set of features  $\mathcal{Y} \subseteq \mathcal{F}$  sufficient for **changing** prediction

[Mil19, INAM20]

$$\text{WCXp}(\mathcal{Y}) := \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$$

- Defining  $\text{CXp}$ :

$$\text{CXp}(\mathcal{Y}) := \text{WCXp}(\mathcal{Y}) \wedge \forall (\mathcal{Y}' \subsetneq \mathcal{Y}). \neg \text{WCXp}(\mathcal{Y}')$$

- But,  $\text{WCXp}$  is also **monotone**; hence,

$$\text{CXp}(\mathcal{Y}) := \text{WCXp}(\mathcal{Y}) \wedge \forall (t \in \mathcal{Y}). \neg \text{WCXp}(\mathcal{Y} \setminus \{t\})$$

# Constrastive explanations – answering Why not? questions

- Instance  $(\mathbf{v}, c)$ , i.e.  $c = \kappa(\mathbf{v})$
- Contrastive explanation (CXp):**

[Mil19, INAM20]

- Subset-minimal set of features  $\mathcal{Y} \subseteq \mathcal{F}$  sufficient for **changing** prediction

$$\text{WCXp}(\mathcal{Y}) := \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$$

- Defining CXp:

$$\text{CXp}(\mathcal{Y}) := \text{WCXp}(\mathcal{Y}) \wedge \forall (\mathcal{Y}' \subsetneq \mathcal{Y}). \neg \text{WCXp}(\mathcal{Y}')$$

- But, WCXp is also **monotone**; hence,

$$\text{CXp}(\mathcal{Y}) := \text{WCXp}(\mathcal{Y}) \wedge \forall (t \in \mathcal{Y}). \neg \text{WCXp}(\mathcal{Y} \setminus \{t\})$$

- Finding one CXp:

[MM20]

- Let  $\mathcal{Y} = \mathcal{F}$ , i.e. **free all features**
- Invariant:  $\text{WCXp}(\mathcal{Y})$  must hold. **Why?**
- Analyze features in any order, one feature  $i$  at a time
  - If  $\text{WCXp}(\mathcal{Y} \setminus \{i\})$  holds, then remove  $i$  from  $\mathcal{Y}$ , i.e.  $i$  becomes **fixed**

## A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$
- Define  $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$

## A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$
- Define  $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg \kappa(x_1, x_2, x_3, x_4)?$

Recap weak CXp:  $\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$

## A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$
- Define  $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? Yes

Recap weak CXp:  $\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$

## A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$
- Define  $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ?

Recap weak CXp:  $\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$

## A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$
- Define  $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**

Recap weak CXp:  $\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$

## A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$
- Define  $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ?

Recap weak CXp:  $\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$

## A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$
- Define  $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**

Recap weak CXp:  $\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$

## A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$
- Define  $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 4 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ?

Recap weak CXp:  $\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$

## A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$
- Define  $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 4 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **No**

Recap weak CXp:  $\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$

## A simple example – CXp's

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- Point  $\mathbf{v} = (0, 0, 0, 1)$  with prediction  $\kappa(\mathbf{v}) = 1$
- Define  $\mathcal{Y} = \{1, 2, 3, 4\} = \mathcal{F}$
- Can feature 1 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 2 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 3 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **Yes**
- Can feature 4 be removed, i.e.  $\exists(\mathbf{x} \in \{0, 1\}^4). \neg x_1 \wedge \neg x_2 \wedge \neg x_3 \wedge x_4 \wedge \neg \kappa(x_1, x_2, x_3, x_4)$ ? **No**
- CXp  $\mathcal{Y} = \{4\}$
- **Obs:** AXp is MHS of CXp and vice-versa...

Recap weak CXp:  $\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x}))$

## Other definitions of WCXps/CXps

- Using probabilities, non-real-valued features:

$$\text{WCXp}(\mathcal{S}) := \Pr(\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) < 1$$

## Other definitions of WCXps/CXps

- Using probabilities, non-real-valued features:

$$\text{WCXp}(\mathcal{S}) := \Pr(\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) < 1$$

- Using expected values:

$$\text{WCXp}(\mathcal{S}) := \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] < 1$$

## Other definitions of WCXps/CXps

- Using probabilities, non-real-valued features:

$$\text{WCXp}(\mathcal{S}) := \Pr(\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}) < 1$$

- Using expected values:

$$\text{WCXp}(\mathcal{S}) := \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] < 1$$

- Definition of CXp remains unchanged

# Detour: global explanations

[INM19b]

- AXps and CXps are defined locally (because of  $\mathbf{v}$ ) but hold globally
  - Localized explanations
  - Can be viewed as attempt at formalizing local explanations
- One can define explanations without picking a given point in feature space
  - Let  $q \in \mathbb{T}$ , and refine the similarity predicate:
    - Classification:  $\sigma(\mathbf{x}) = [\kappa(\mathbf{x}) = q]$
    - Regression:  $\sigma(\mathbf{x}) = [|\kappa(\mathbf{x}) - q| \leq \delta]$ ,  $\delta$  is user-specified
  - Let  $\mathbb{L} = \{(x_i = v_i) \mid i \in \mathcal{F} \wedge v_i \in \mathbb{V}\}$
  - Let  $\mathcal{S} \subsetneq \mathbb{L}$  be a subset of literals that does not repeat features, i.e.  $\mathcal{S}$  is not inconsistent
  - Then,  $\mathcal{S}$  is a global AXp if,

$$\forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{(x_i = v_i) \in \mathcal{S}} (x_i = v_i) \rightarrow (\sigma(\mathbf{x}))$$

- Counterexamples are minimal hitting sets of global AXps and vice-versa

[INM19b]

## Outline – Unit #02

---

Definitions of Explanations

Duality Properties

Computational Problems

# Duality in explainability – basic results

[INAM20, Mar22]

## Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is an AXp iff it is a minimal hitting set  
(MHS) of the set of CXps

# Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is an AXp iff it is a minimal hitting set  
(MHS) of the set of CXps

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is a CXp iff it is a minimal hitting set (MHS)  
of the set of AXps

# Duality in explainability – basic results

[INAM20, Mar22]

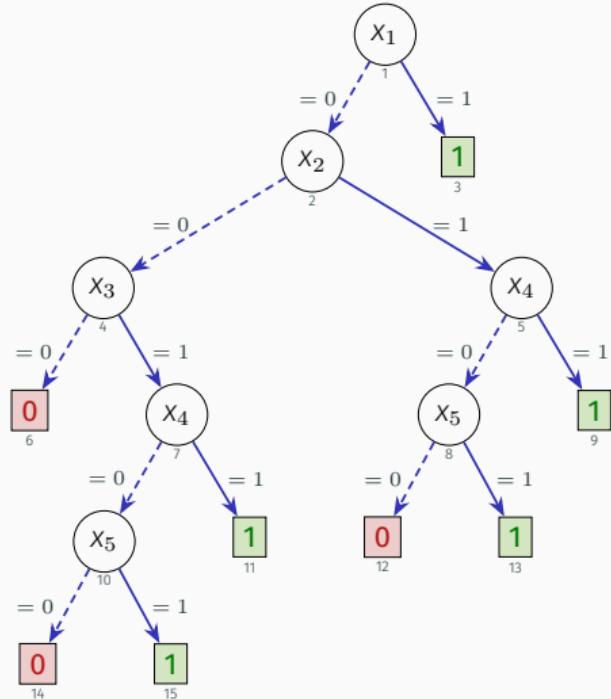
- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is an AXp iff it is a minimal hitting set (MHS) of the set of CXps

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is a CXp iff it is a minimal hitting set (MHS) of the set of AXps

- An example,  $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$ :



# Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**

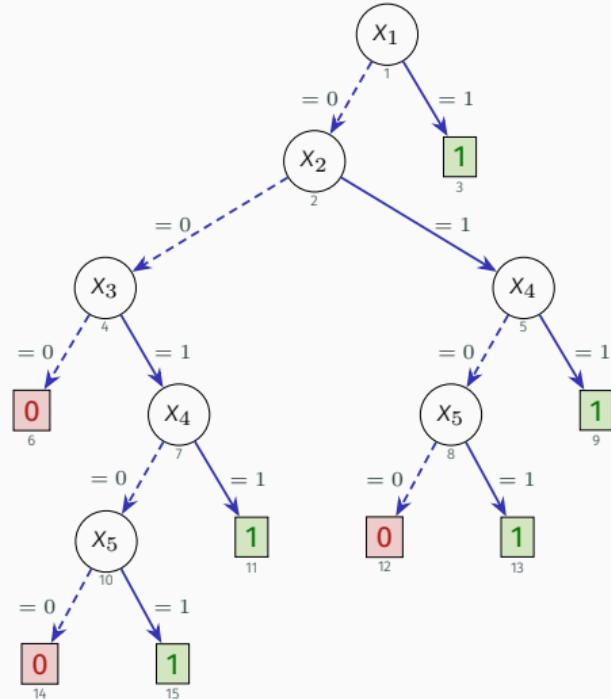
$\mathcal{S} \subseteq \mathcal{F}$  is an AXp iff it is a minimal hitting set (MHS) of the set of CXps

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is a CXp iff it is a minimal hitting set (MHS) of the set of AXps

- An example,  $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$ :

- AXps:



# Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**

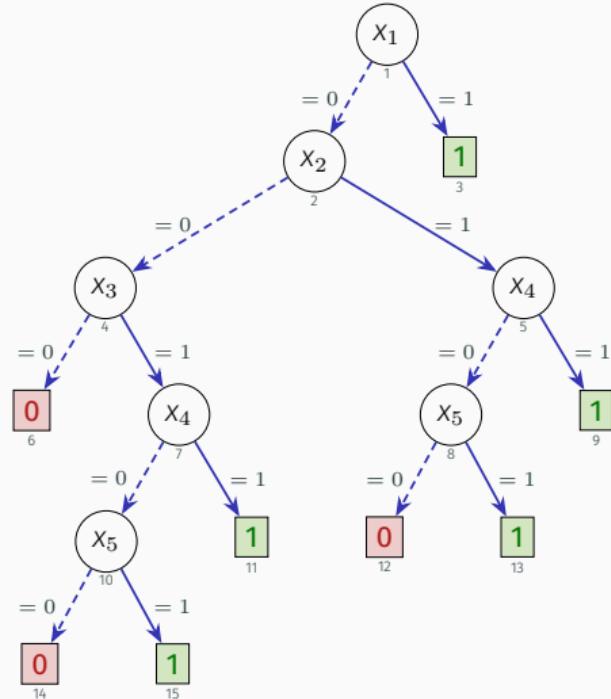
$\mathcal{S} \subseteq \mathcal{F}$  is an AXp iff it is a minimal hitting set (MHS) of the set of CXps

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is a CXp iff it is a minimal hitting set (MHS) of the set of AXps

- An example,  $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$ :

- AXps:  $\{\{3, 5\}\}$



# Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**

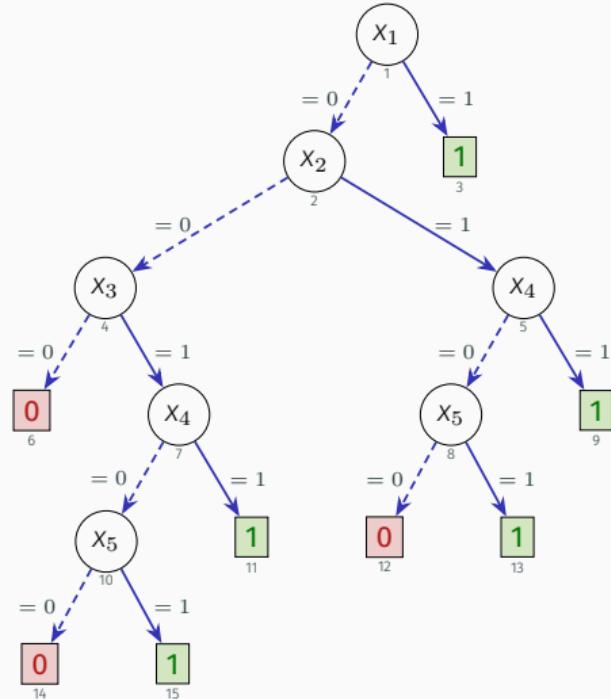
$\mathcal{S} \subseteq \mathcal{F}$  is an AXp iff it is a minimal hitting set (MHS) of the set of CXps

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is a CXp iff it is a minimal hitting set (MHS) of the set of AXps

- An example,  $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$ :

- AXps:  $\{\{3, 5\}\}$
- CXps:



# Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**

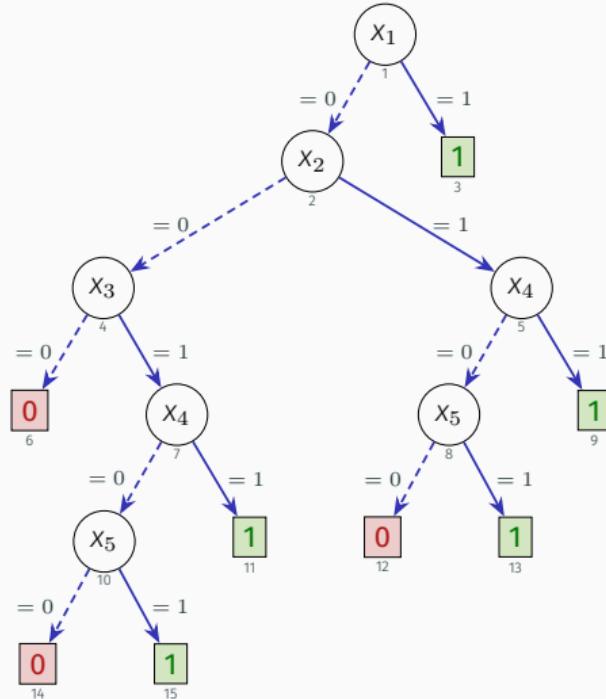
$\mathcal{S} \subseteq \mathcal{F}$  is an AXp iff it is a minimal hitting set (MHS) of the set of CXps

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is a CXp iff it is a minimal hitting set (MHS) of the set of AXps

- An example,  $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$ :

- AXps:  $\{\{3, 5\}\}$
- CXps:  $\{\{3\}, \{5\}\}$



# Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**

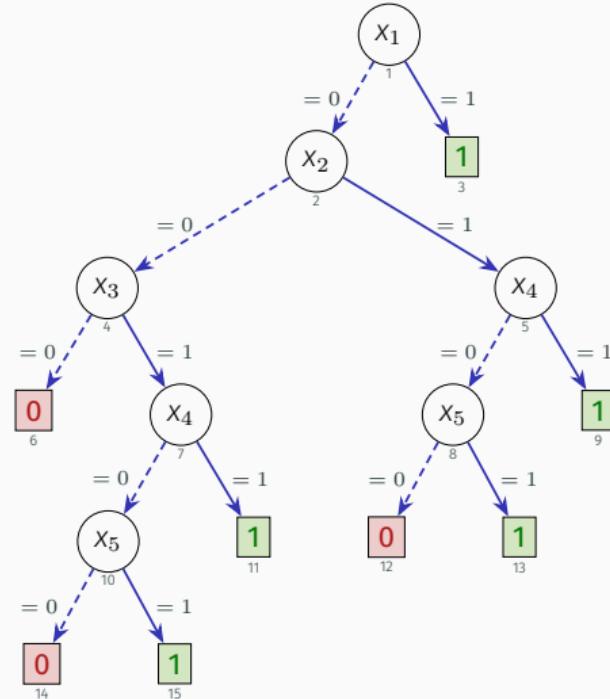
$\mathcal{S} \subseteq \mathcal{F}$  is an AXp iff it is a minimal hitting set (MHS) of the set of CXps

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is a CXp iff it is a minimal hitting set (MHS) of the set of AXps

- An example,  $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$ :

- AXps:  $\{\{3, 5\}\}$
- CXps:  $\{\{3\}, \{5\}\}$
- Each AXp is an MHS of the set of CXps
- Each CXp is an MHS of the set of AXps



# Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**

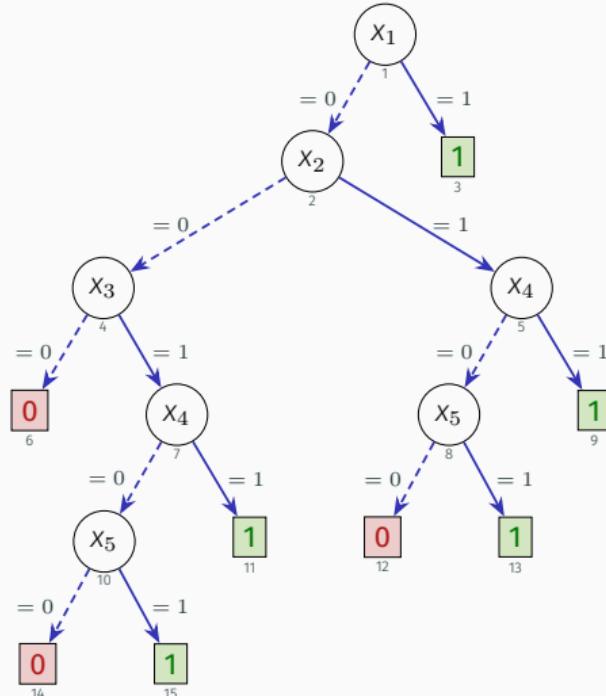
$\mathcal{S} \subseteq \mathcal{F}$  is an AXp iff it is a minimal hitting set (MHS) of the set of CXps

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is a CXp iff it is a minimal hitting set (MHS) of the set of AXps

- An example,  $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$ :

- AXps:  $\{\{3, 5\}\}$
- CXps:  $\{\{3\}, \{5\}\}$
- Each AXp is an MHS of the set of CXps
- Each CXp is an MHS of the set of AXps
- BTW,
  - $\{2, 5\}$  is **not** a CXp
  - $\{1, 2, 3, 4, 5\}, \{1, 2, 3, 5\}$  and  $\{1, 3, 5\}$  are **not** AXps



# Duality in explainability – basic results

[INAM20, Mar22]

- **Claim:**

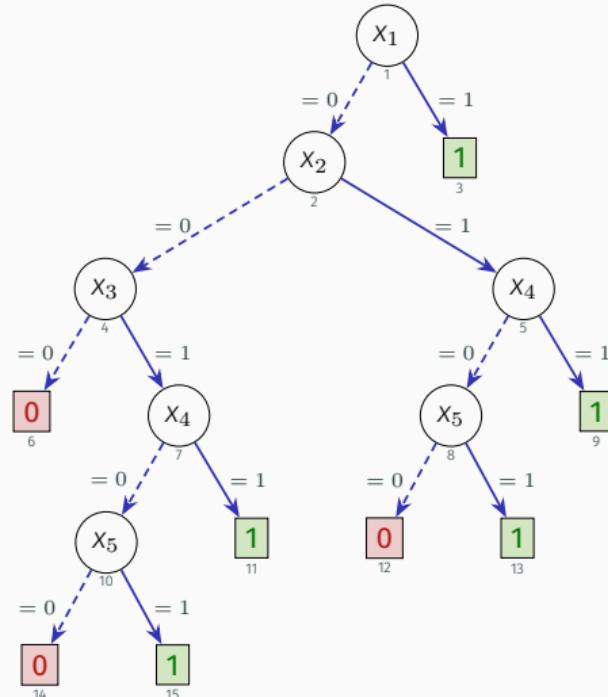
$\mathcal{S} \subseteq \mathcal{F}$  is an AXp iff it is a minimal hitting set (MHS) of the set of CXps

- **Claim:**

$\mathcal{S} \subseteq \mathcal{F}$  is a CXp iff it is a minimal hitting set (MHS) of the set of AXps

- An example,  $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$ :

- AXps:  $\{\{3, 5\}\}$
- CXps:  $\{\{3\}, \{5\}\}$
- Each AXp is an MHS of the set of CXps
- Each CXp is an MHS of the set of AXps
- BTW,
  - $\{2, 5\}$  is **not** a CXp
  - $\{1, 2, 3, 4, 5\}, \{1, 2, 3, 5\}$  and  $\{1, 3, 5\}$  are **not** AXps
  - **Why?**



## Outline – Unit #02

---

Definitions of Explanations

Duality Properties

Computational Problems

# Computational problems in (formal) explainability

- Compute one abductive/contrastive explanation

# Computational problems in (formal) explainability

- Compute **one** abductive/contrastive explanation
- Enumerate **all** abductive/contrastive explanations

# Computational problems in (formal) explainability

- Compute **one** abductive/contrastive explanation
- Enumerate **all** abductive/contrastive explanations
- Decide whether feature included in **all** abductive/contrastive explanations

# Computational problems in (formal) explainability

- Compute **one** abductive/contrastive explanation
- Enumerate **all** abductive/contrastive explanations
- Decide whether feature included in **all** abductive/contrastive explanations
- Decide whether feature included in **some** abductive/contrastive explanation

## Computing one AXp/CXp

- Encode classifier into suitable logic representation  $\mathcal{T}$  & pick suitable reasoner

## Computing one AXp/CXp

- Encode classifier into suitable logic representation  $\mathcal{T}$  & pick suitable reasoner
- For **AXp**: start from  $\mathcal{S} = \mathcal{F}$  and drop (i.e. free) features from  $\mathcal{S}$  while WAXp condition holds

## Computing one AXp/CXp

- Encode classifier into suitable logic representation  $\mathcal{T}$  & pick suitable reasoner
- For **AXp**: start from  $\mathcal{S} = \mathcal{F}$  and drop (i.e. free) features from  $\mathcal{S}$  while WAXp condition holds
- For **CXp**: start from  $\mathcal{S} = \mathcal{F}$  and drop (i.e. fix) features from  $\mathcal{S}$  while WCXp condition holds

## Computing one AXp/CXp

- Encode classifier into suitable logic representation  $\mathcal{T}$  & pick suitable reasoner
- For **AXp**: start from  $\mathcal{S} = \mathcal{F}$  and drop (i.e. free) features from  $\mathcal{S}$  while WAXp condition holds
- For **CXp**: start from  $\mathcal{S} = \mathcal{F}$  and drop (i.e. fix) features from  $\mathcal{S}$  while WCXp condition holds
- **Monotone** predicates for WAXp & WCXp:

$$\mathbb{P}_{\text{axp}}(\mathcal{S}) \triangleq \neg \text{CO} (\llbracket (\bigwedge_{i \in \mathcal{S}} (x_i = v_i)) \wedge (\neg \sigma(\mathbf{x})) \rrbracket) \quad \mathbb{P}_{\text{cxp}}(\mathcal{S}) \triangleq \text{CO} \left( \llbracket \left( \bigwedge_{i \in \mathcal{F} \setminus \mathcal{S}} (x_i = v_i) \right) \wedge (\neg \sigma(\mathbf{x})) \rrbracket \right)$$

# Computing one AXp/CXp

- Encode classifier into suitable logic representation  $\mathcal{T}$  & pick suitable reasoner
- For **AXp**: start from  $\mathcal{S} = \mathcal{F}$  and drop (i.e. free) features from  $\mathcal{S}$  while WAXp condition holds
- For **CXp**: start from  $\mathcal{S} = \mathcal{F}$  and drop (i.e. fix) features from  $\mathcal{S}$  while WCXp condition holds
- **Monotone** predicates for WAXp & WCXp:

$$\mathbb{P}_{\text{axp}}(\mathcal{S}) \triangleq \neg \text{CO} (\llbracket (\bigwedge_{i \in \mathcal{S}} (x_i = v_i)) \wedge (\neg \sigma(\mathbf{x})) \rrbracket) \quad \mathbb{P}_{\text{cxp}}(\mathcal{S}) \triangleq \text{CO} (\llbracket (\bigwedge_{i \in \mathcal{F} \setminus \mathcal{S}} (x_i = v_i)) \wedge (\neg \sigma(\mathbf{x})) \rrbracket)$$

**Input:** Predicate  $\mathbb{P}$ , parameterized by  $\mathcal{T}, \mathcal{M}$

**Output:** One XP  $\mathcal{S}$

```
1: procedure oneXP( $\mathbb{P}$ )
2:    $\mathcal{S} \leftarrow \mathcal{F}$                                  $\triangleright$  Initialization:  $\mathbb{P}(\mathcal{S})$  holds
3:   for  $i \in \mathcal{F}$  do
4:     if  $\mathbb{P}(\mathcal{S} \setminus \{i\})$  then                 $\triangleright$  Loop invariant:  $\mathbb{P}(\mathcal{S})$  holds
5:        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
6:   return  $\mathcal{S}$                                  $\triangleright$  Update  $\mathcal{S}$  only if  $\mathbb{P}(\mathcal{S} \setminus \{i\})$  holds
                                                 $\triangleright$  Returned set  $\mathcal{S}$ :  $\mathbb{P}(\mathcal{S})$  holds
```

# Computing one AXp/CXp

- Encode classifier into suitable logic representation  $\mathcal{T}$  & pick suitable reasoner
- For **AXp**: start from  $\mathcal{S} = \mathcal{F}$  and drop (i.e. free) features from  $\mathcal{S}$  while WAXp condition holds
- For **CXp**: start from  $\mathcal{S} = \mathcal{F}$  and drop (i.e. fix) features from  $\mathcal{S}$  while WCXp condition holds
- **Monotone** predicates for WAXp & WCXp:

$$\mathbb{P}_{\text{axp}}(\mathcal{S}) \triangleq \neg \text{CO} (\llbracket (\bigwedge_{i \in \mathcal{S}} (x_i = v_i)) \wedge (\neg \sigma(\mathbf{x})) \rrbracket) \quad \mathbb{P}_{\text{cxp}}(\mathcal{S}) \triangleq \text{CO} (\llbracket (\bigwedge_{i \in \mathcal{F} \setminus \mathcal{S}} (x_i = v_i)) \wedge (\neg \sigma(\mathbf{x})) \rrbracket)$$

**Input:** Predicate  $\mathbb{P}$ , parameterized by  $\mathcal{T}, \mathcal{M}$

**Output:** One XP  $\mathcal{S}$

```
1: procedure oneXP( $\mathbb{P}$ )
2:    $\mathcal{S} \leftarrow \mathcal{F}$ 
3:   for  $i \in \mathcal{F}$  do
4:     if  $\mathbb{P}(\mathcal{S} \setminus \{i\})$  then
5:        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
6:   return  $\mathcal{S}$ 
```

Exploiting MSMP, i.e.  
basic algorithm used  
for different problems.

- ▷ Initialization:  $\mathbb{P}(\mathcal{S})$  holds
- ▷ Loop invariant:  $\mathbb{P}(\mathcal{S})$  holds
- ▷ Update  $\mathcal{S}$  only if  $\mathbb{P}(\mathcal{S} \setminus \{i\})$  holds
- ▷ Returned set  $\mathcal{S}$ :  $\mathbb{P}(\mathcal{S})$  holds

## Detour: More Connections with Automated Reasoning

## Prime implicants & implicants

- A conjunction of literals  $\pi$  (which will be viewed as a set of literals where convenient) is a **prime implicant** of some function  $\varphi$  if,
  1.  $\pi \vDash \varphi$
  2. For any  $\pi' \subsetneq \pi$ ,  $\pi' \not\vDash \varphi$

## Prime implicants & implicants

- A conjunction of literals  $\pi$  (which will be viewed as a set of literals where convenient) is a **prime implicant** of some function  $\varphi$  if,
  1.  $\pi \vDash \varphi$
  2. For any  $\pi' \subsetneq \pi$ ,  $\pi' \not\vDash \varphi$
- Example:
  - $\mathbb{F} = \{0, 1\}^3$
  - $\varphi(x_1, x_2, x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3$
  - Clearly,  $x_1 \wedge x_2 \vDash \varphi$
  - Also,  $x_1 \not\vDash \varphi$  and  $x_2 \not\vDash \varphi$

# Prime implicants & implicants

- A conjunction of literals  $\pi$  (which will be viewed as a set of literals where convenient) is a prime implicant of some function  $\varphi$  if,
  1.  $\pi \vDash \varphi$
  2. For any  $\pi' \subsetneq \pi$ ,  $\pi' \not\vDash \varphi$
  - Example:
    - $\mathbb{F} = \{0, 1\}^3$
    - $\varphi(x_1, x_2, x_3) = x_1 \wedge x_2 \vee x_1 \wedge x_3$
    - Clearly,  $x_1 \wedge x_2 \vDash \varphi$
    - Also,  $x_1 \not\vDash \varphi$  and  $x_2 \not\vDash \varphi$
- A disjunction of literals  $\eta$  (also viewed as a set of literals where convenient) is a prime implicate of some function  $\varphi$  if
  1.  $\varphi \vDash \eta$
  2. For any  $\eta' \subsetneq \eta$ ,  $\varphi \not\vDash \eta'$

- Formula  $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$ , with
  - $\mathcal{B}$ : background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$ : additional (inconsistent) knowledge, i.e. soft constraints
  - And,  $\mathcal{T} \vDash \perp$
  - E.g.  $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$ ,  $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$

- Formula  $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$ , with
  - $\mathcal{B}$ : background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$ : additional (inconsistent) knowledge, i.e. soft constraints
  - And,  $\mathcal{T} \vDash \perp$
  - E.g.  $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$ ,  $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set  $\mathcal{U} \subseteq \mathcal{S}$ , s.t.  $\mathcal{B} \cup \mathcal{U} \vDash \perp$
  - E.g.  $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$

- Formula  $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$ , with
  - $\mathcal{B}$ : background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$ : additional (inconsistent) knowledge, i.e. soft constraints
  - And,  $\mathcal{T} \vDash \perp$
  - E.g.  $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$ ,  $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set  $\mathcal{U} \subseteq \mathcal{S}$ , s.t.  $\mathcal{B} \cup \mathcal{U} \vDash \perp$
  - E.g.  $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$
- Minimal correction subset (MCS):
  - Subset-minimal set  $\mathcal{C} \subseteq \mathcal{S}$ , s.t.  $\mathcal{B} \cup (\mathcal{S} \setminus \mathcal{C}) \not\vDash \perp$
  - E.g.  $\mathcal{C} = \{(\neg x_1)\}$

- Formula  $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$ , with
  - $\mathcal{B}$ : background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$ : additional (inconsistent) knowledge, i.e. soft constraints
  - And,  $\mathcal{T} \models \perp$
  - E.g.  $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$ ,  $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set  $\mathcal{U} \subseteq \mathcal{S}$ , s.t.  $\mathcal{B} \cup \mathcal{U} \models \perp$
  - E.g.  $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$
- Minimal correction subset (MCS):
  - Subset-minimal set  $\mathcal{C} \subseteq \mathcal{S}$ , s.t.  $\mathcal{B} \cup (\mathcal{S} \setminus \mathcal{C}) \not\models \perp$
  - E.g.  $\mathcal{C} = \{(\neg x_1)\}$
- Duality:
  - MUSes are minimal-hitting sets (MHSes) of the MCSes, and vice-versa

[Rei87]

- Formula  $\mathcal{T} = \mathcal{B} \cup \mathcal{S}$ , with
  - $\mathcal{B}$ : background knowledge (base), i.e. hard constraints
  - $\mathcal{S}$ : additional (inconsistent) knowledge, i.e. soft constraints
  - And,  $\mathcal{T} \vDash \perp$
  - E.g.  $\mathcal{B} = \{(x_1 \vee x_2), (x_1 \vee \neg x_3)\}$ ,  $\mathcal{S} = \{(\neg x_1), (\neg x_2), (x_3)\}$
- Minimal unsatisfiable subset (MUS):
  - Subset-minimal set  $\mathcal{U} \subseteq \mathcal{S}$ , s.t.  $\mathcal{B} \cup \mathcal{U} \vDash \perp$
  - E.g.  $\mathcal{U} = \{(\neg x_1), (\neg x_2)\}$
- Minimal correction subset (MCS):
  - Subset-minimal set  $\mathcal{C} \subseteq \mathcal{S}$ , s.t.  $\mathcal{B} \cup (\mathcal{S} \setminus \mathcal{C}) \not\vDash \perp$
  - E.g.  $\mathcal{C} = \{(\neg x_1)\}$
- Duality:
  - MUSes are minimal-hitting sets (MHSes) of the MCSes, and vice-versa
- Variants:
  - Smallest(-cost) MCS, i.e. complement of maximum(-cost) satisfiability (MaxSAT)
  - Smallest(-cost) MUS

[Rei87]

## Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\text{WAXp}(\mathcal{X}) \quad := \quad \forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x}))$$

$$\text{WCXp}(\mathcal{Y}) \quad := \quad \exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg\sigma(\mathbf{x}))$$

## Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned} \text{WAXp}(\mathcal{X}) &:= \neg \left[ \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \right] \\ \text{WCXp}(\mathcal{Y}) &:= \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \end{aligned}$$

## Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned} \text{WAXp}(\mathcal{X}) &:= \neg \left[ \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \right] \\ \text{WCXp}(\mathcal{Y}) &:= \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \end{aligned}$$

- Let,

- Hard constraints,  $\mathcal{B}$ :

$$\mathcal{B} := \wedge_{i \in \mathcal{F}} (s_i \rightarrow (x_i = v_i)) \wedge \text{Encode}_{\mathcal{T}}(\neg \sigma(\mathbf{x}))$$

## Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned} \text{WAXp}(\mathcal{X}) &:= \neg \left[ \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \right] \\ \text{WCXp}(\mathcal{Y}) &:= \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \end{aligned}$$

- Let,

- Hard constraints,  $\mathcal{B}$ :

$$\mathcal{B} := \wedge_{i \in \mathcal{F}} (s_i \rightarrow (x_i = v_i)) \wedge \text{Encode}_{\mathcal{T}}(\neg \sigma(\mathbf{x}))$$

- Soft constraints:  $\mathcal{S} = \{s_i \mid i \in \mathcal{F}\}$

## Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned} \text{WAXp}(\mathcal{X}) &:= \neg \left[ \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \right] \\ \text{WCXp}(\mathcal{Y}) &:= \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \end{aligned}$$

- Let,

- Hard constraints,  $\mathcal{B}$ :

$$\mathcal{B} := \wedge_{i \in \mathcal{F}} (s_i \rightarrow (x_i = v_i)) \wedge \text{Encode}_{\mathcal{T}}(\neg \sigma(\mathbf{x}))$$

- Soft constraints:  $\mathcal{S} = \{s_i \mid i \in \mathcal{F}\}$
- **Claim:** Each MUS of  $(\mathcal{B}, \mathcal{S})$  is an AXp & each MCS of  $(\mathcal{B}, \mathcal{S})$  is a CXp

## Computing AXps (resp. CXps) as MUSes (resp. MCSes)

- Recap:

$$\begin{aligned} \text{WAXp}(\mathcal{X}) &:= \neg \left[ \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \right] \\ \text{WCXp}(\mathcal{Y}) &:= \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \end{aligned}$$

- Let,

- Hard constraints,  $\mathcal{B}$ :

$$\mathcal{B} := \wedge_{i \in \mathcal{F}} (s_i \rightarrow (x_i = v_i)) \wedge \text{Encode}_{\mathcal{T}}(\neg \sigma(\mathbf{x}))$$

- Soft constraints:  $\mathcal{S} = \{s_i \mid i \in \mathcal{F}\}$
- **Claim:** Each MUS of  $(\mathcal{B}, \mathcal{S})$  is an AXp & each MCS of  $(\mathcal{B}, \mathcal{S})$  is a CXp
  - Can use MUS/MCS algorithms for finding AXps/CXps

Unit #03

## Tractability in Symbolic XAI

# Outline – Unit #03

Explanations for Decision Trees

XAI Queries for DTs

Myth #01: Intrinsic Interpretability

Detour: From Decision Trees to Explained Decision Sets

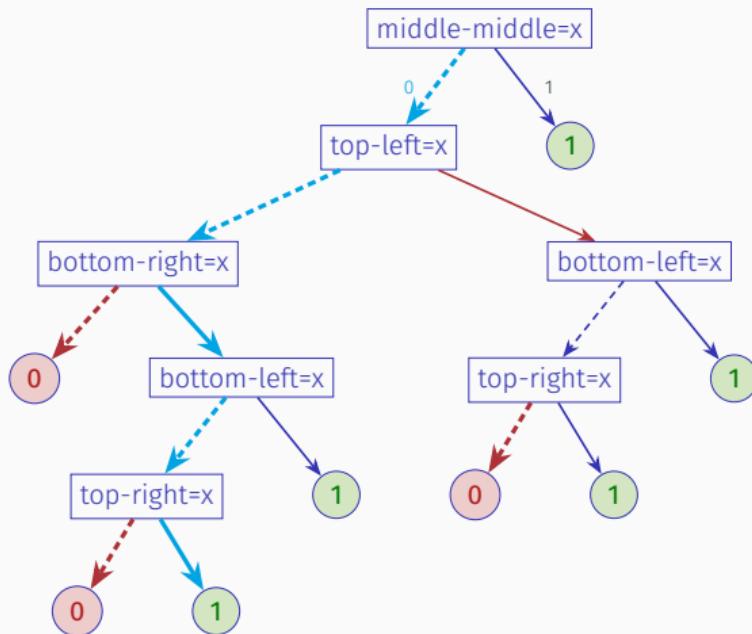
Explanations for Decision Graphs

Explanations for Monotonic Classifiers

Review examples

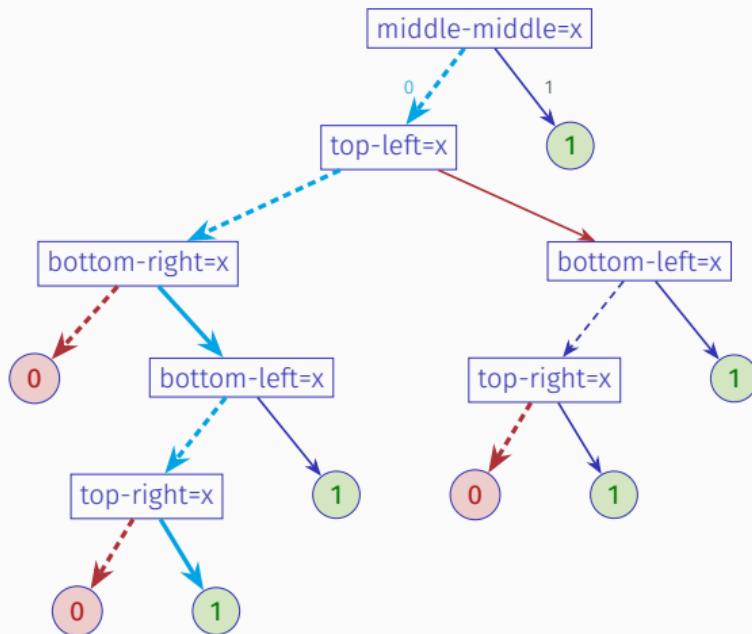
# DT explanations

[IIM20]



# DT explanations

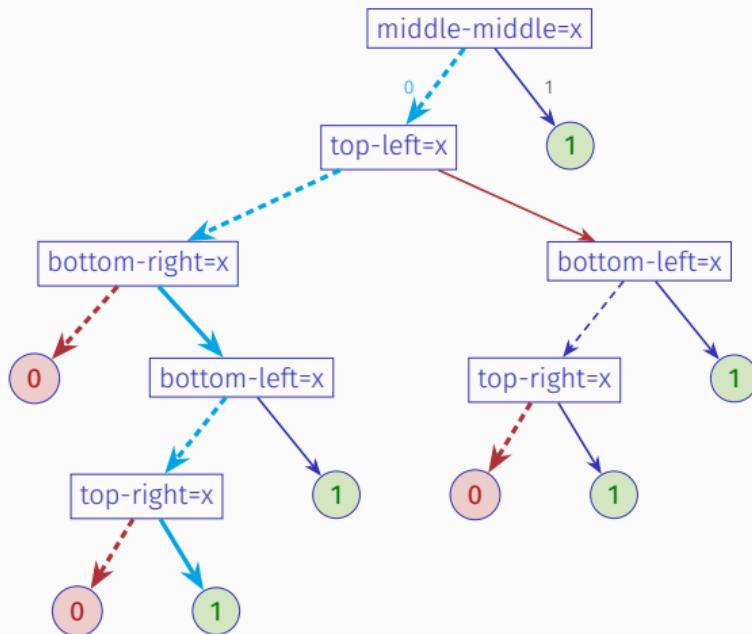
[IIM20]



- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time

# DT explanations

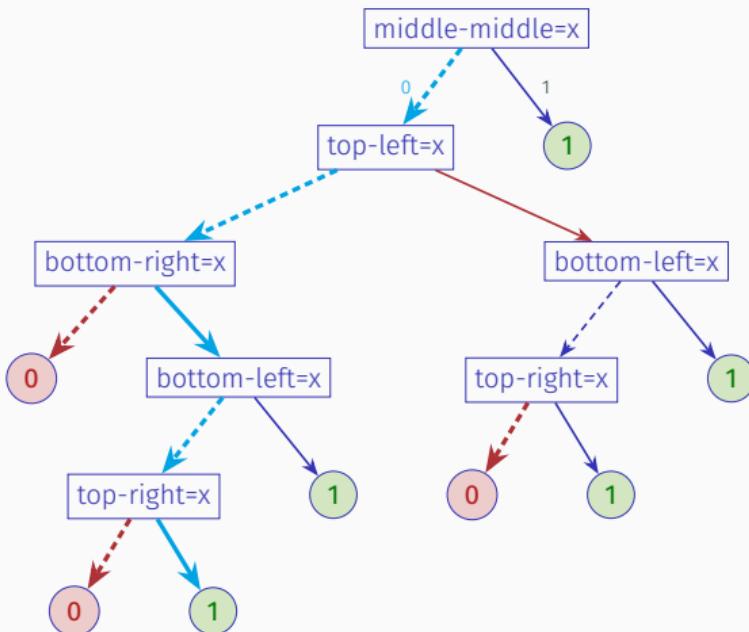
[IIM20]



- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time
- For prediction **1**, it suffices to ensure **all** paths with prediction **0** remain inconsistent

# DT explanations in polynomial time

[IIM20]



- Run PI-explanation algorithm based on NP-oracles
  - Worst-case exponential time
- For prediction **1**, it suffices to ensure **all** paths with prediction **0** remain inconsistent
  - I.e. find a **subset-minimal hitting set** of **all 0** paths; **these are the features to keep**
    - E.g. BR and TR suffice for prediction
  - Well-known to be solvable in **polynomial time**

[EG95]

# Outline – Unit #03

Explanations for Decision Trees

XAI Queries for DTs

Myth #01: Intrinsic Interpretability

Detour: From Decision Trees to Explained Decision Sets

Explanations for Decision Graphs

Explanations for Monotonic Classifiers

Review examples

## Answering queries in DTs

- Finding one  $AXp$  in polynomial-time – covered

## Answering queries in DTs

- Finding one AXp in polynomial-time – covered
- Finding one CXp in polynomial-time

## Answering queries in DTs

- Finding one AXp in polynomial-time – covered
- Finding one CXp in polynomial-time
- Finding all CXps in polynomial-time

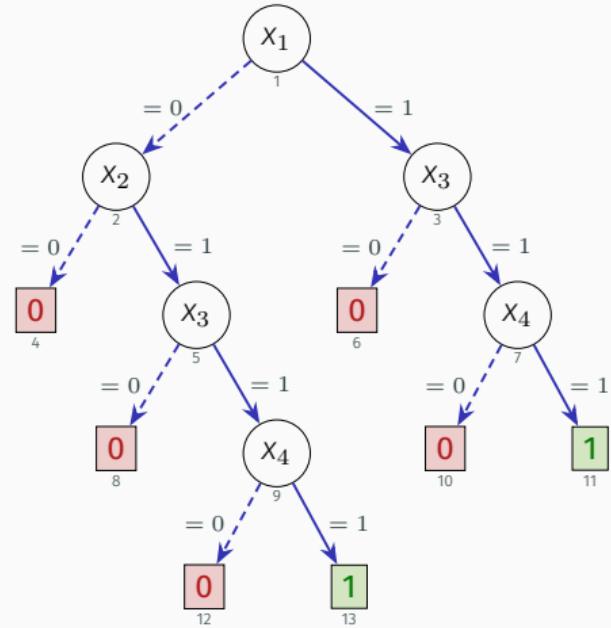
- Finding one AXp in polynomial-time – covered
- Finding one CXp in polynomial-time
- Finding all CXps in polynomial-time; hence, finding one CXp also in polynomial-time

## Answering queries in DTs

- Finding one AXp in polynomial-time – covered
- Finding one CXp in polynomial-time
- Finding all CXps in polynomial-time; hence, finding one CXp also in polynomial-time
- Practically efficient enumeration of AXps – later

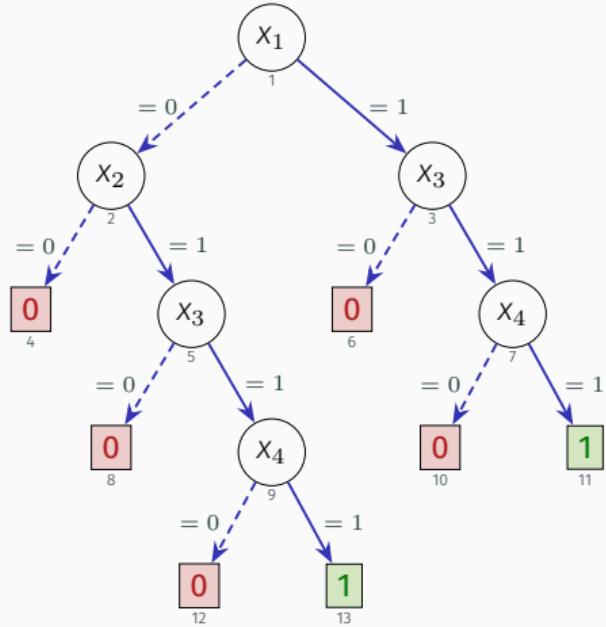
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$



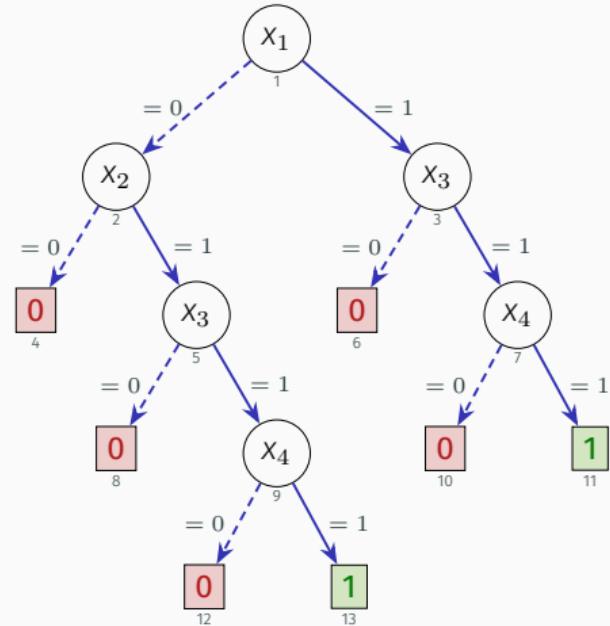
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :



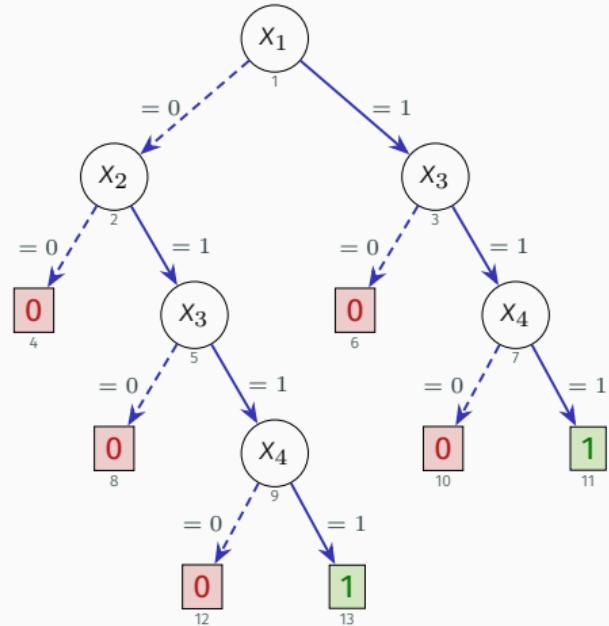
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$



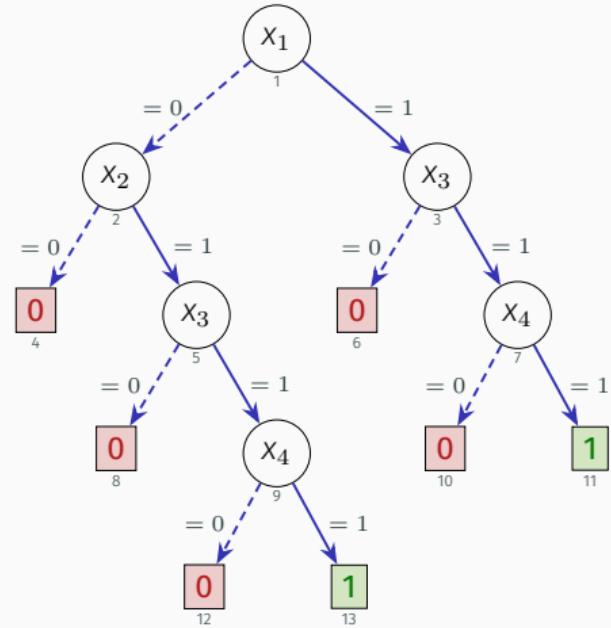
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$



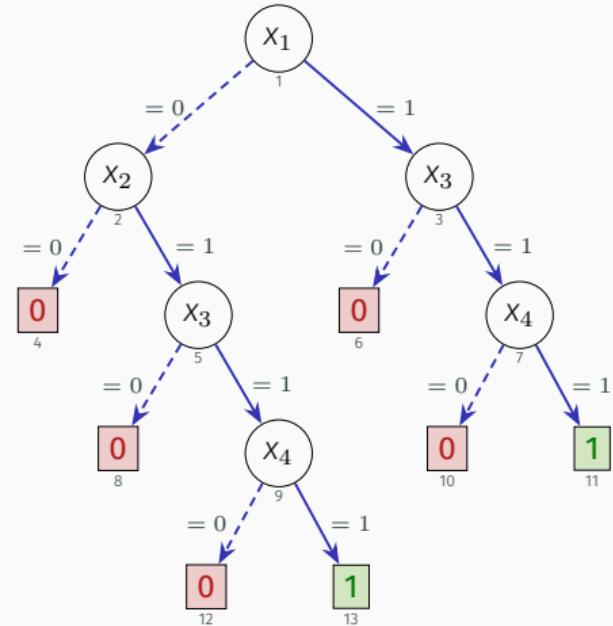
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets



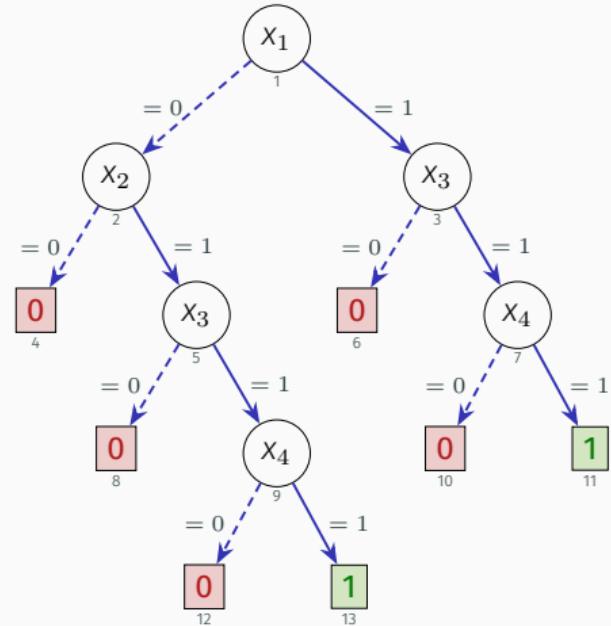
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets
  - $\mathcal{L}$  contains all the CXps of the DT



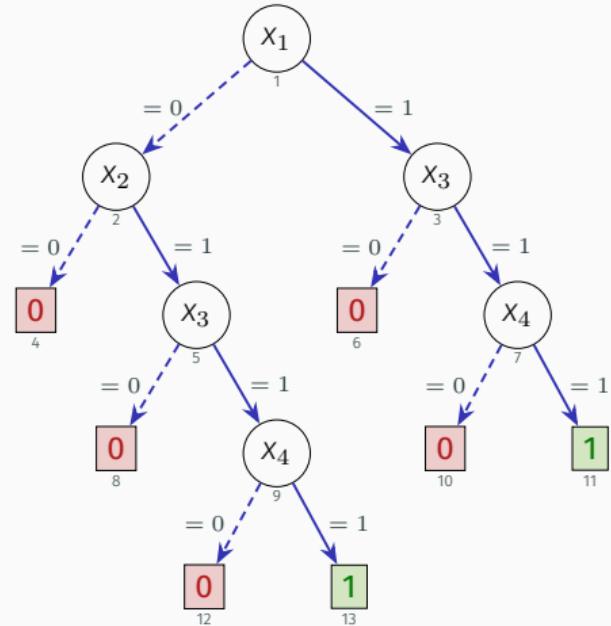
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets
  - $\mathcal{L}$  contains all the CXps of the DT
- Example: instance is  $((1, 1, 1, 1), 1)$



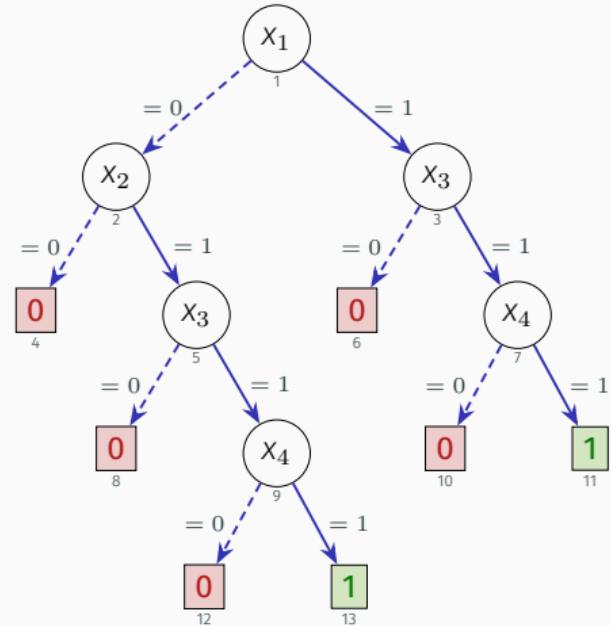
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets
  - $\mathcal{L}$  contains all the CXps of the DT
- Example: instance is  $((1, 1, 1, 1), 1)$ 
  - Add  $\{1, 2\}$  to  $\mathcal{L}$



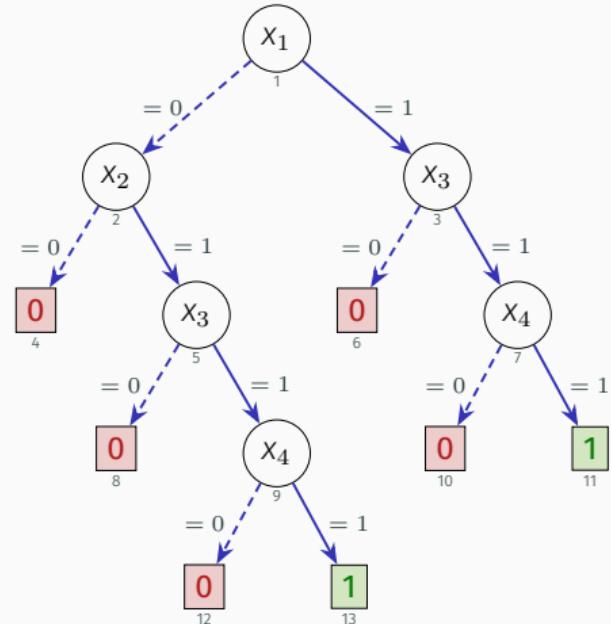
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets
  - $\mathcal{L}$  contains all the CXps of the DT
- Example: instance is  $((1, 1, 1, 1), 1)$ 
  - Add  $\{1, 2\}$  to  $\mathcal{L}$
  - Add  $\{1, 3\}$  to  $\mathcal{L}$



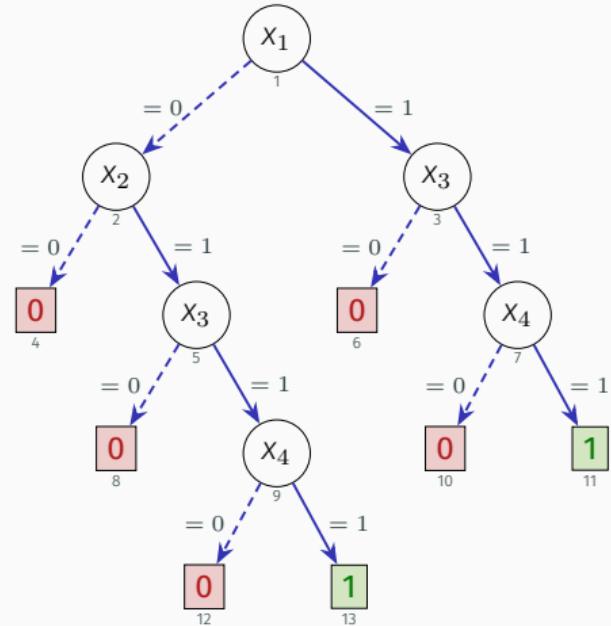
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets
  - $\mathcal{L}$  contains all the CXps of the DT
- Example: instance is  $((1, 1, 1, 1), 1)$ 
  - Add  $\{1, 2\}$  to  $\mathcal{L}$
  - Add  $\{1, 3\}$  to  $\mathcal{L}$
  - Add  $\{1, 4\}$  to  $\mathcal{L}$



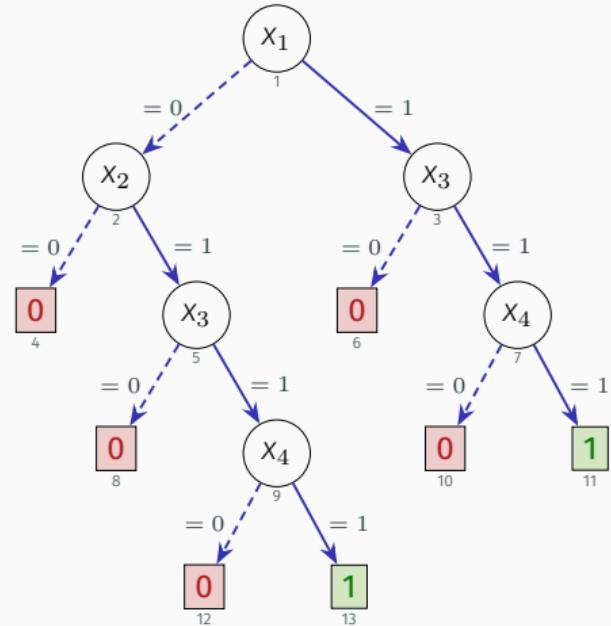
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets
  - $\mathcal{L}$  contains all the CXps of the DT
- Example: instance is  $((1, 1, 1, 1), 1)$ 
  - Add  $\{1, 2\}$  to  $\mathcal{L}$
  - Add  $\{1, 3\}$  to  $\mathcal{L}$
  - Add  $\{1, 4\}$  to  $\mathcal{L}$
  - Add  $\{3\}$  to  $\mathcal{L}$



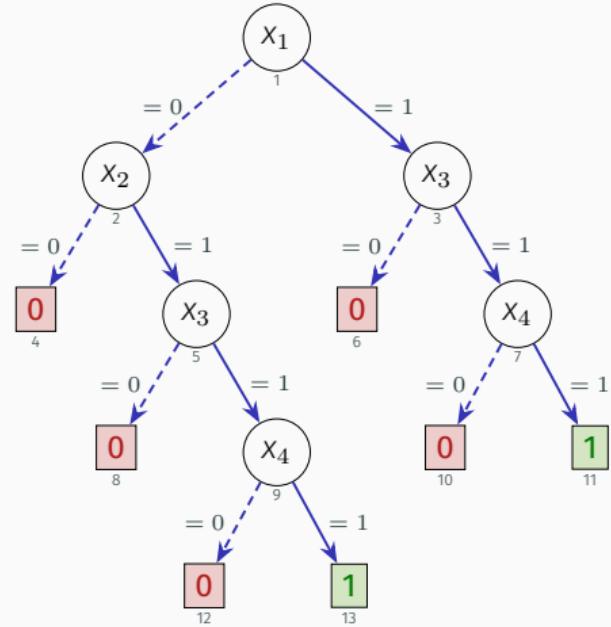
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets
  - $\mathcal{L}$  contains all the CXps of the DT
- Example: instance is  $((1, 1, 1, 1), 1)$ 
  - Add  $\{1, 2\}$  to  $\mathcal{L}$
  - Add  $\{1, 3\}$  to  $\mathcal{L}$
  - Add  $\{1, 4\}$  to  $\mathcal{L}$
  - Add  $\{3\}$  to  $\mathcal{L}$
  - Add  $\{4\}$  to  $\mathcal{L}$



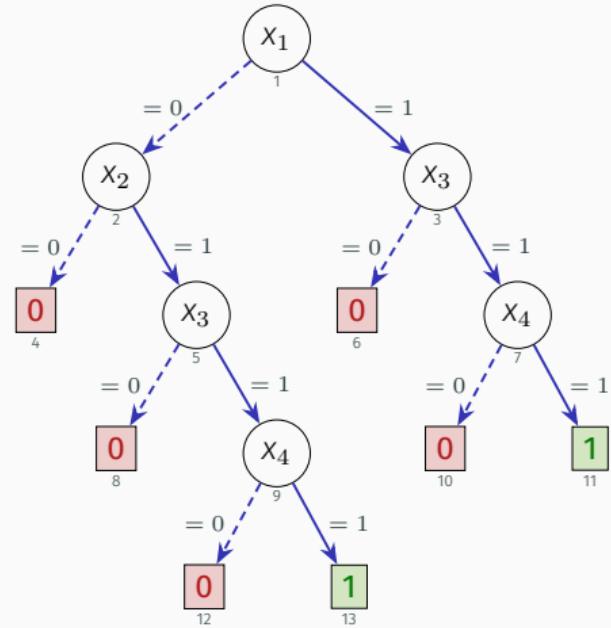
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets
  - $\mathcal{L}$  contains all the CXps of the DT
- Example: instance is  $((1, 1, 1, 1), 1)$ 
  - Add  $\{1, 2\}$  to  $\mathcal{L}$
  - Add  $\{1, 3\}$  to  $\mathcal{L}$
  - Add  $\{1, 4\}$  to  $\mathcal{L}$
  - Add  $\{3\}$  to  $\mathcal{L}$
  - Add  $\{4\}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$ :  $\{1, 3\}$  and  $\{1, 4\}$



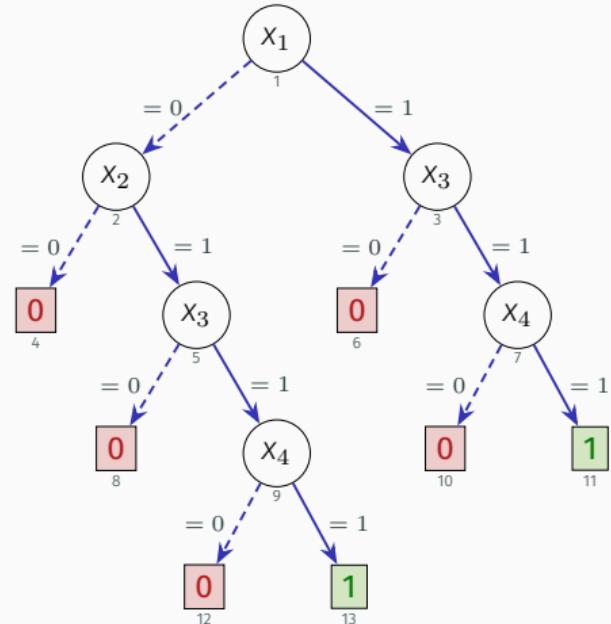
# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets
  - $\mathcal{L}$  contains all the CXps of the DT
- Example: instance is  $((1, 1, 1, 1), 1)$ 
  - Add  $\{1, 2\}$  to  $\mathcal{L}$
  - Add  $\{1, 3\}$  to  $\mathcal{L}$
  - Add  $\{1, 4\}$  to  $\mathcal{L}$
  - Add  $\{3\}$  to  $\mathcal{L}$
  - Add  $\{4\}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$ :  $\{1, 3\}$  and  $\{1, 4\}$
  - CXps:  $\{\{1, 2\}, \{3\}, \{4\}\}$



# Finding all CXps in polynomial-time

- Basic algorithm:
  - $\mathcal{L} = \emptyset$
  - For each leaf node not predicting  $q$ :
    - $\mathcal{I}$ : features with literals inconsistent with  $\mathbf{v}$
    - Add  $\mathcal{I}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$  non-minimal sets
  - $\mathcal{L}$  contains all the CXps of the DT
- Example: instance is  $((1, 1, 1, 1), 1)$ 
  - Add  $\{1, 2\}$  to  $\mathcal{L}$
  - Add  $\{1, 3\}$  to  $\mathcal{L}$
  - Add  $\{1, 4\}$  to  $\mathcal{L}$
  - Add  $\{3\}$  to  $\mathcal{L}$
  - Add  $\{4\}$  to  $\mathcal{L}$
  - Remove from  $\mathcal{L}$ :  $\{1, 3\}$  and  $\{1, 4\}$
  - CXps:  $\{\{1, 2\}, \{3\}, \{4\}\}$
  - AXps:  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$ , by computing all MHSes



# Outline – Unit #03

Explanations for Decision Trees

XAI Queries for DTs

Myth #01: Intrinsic Interpretability

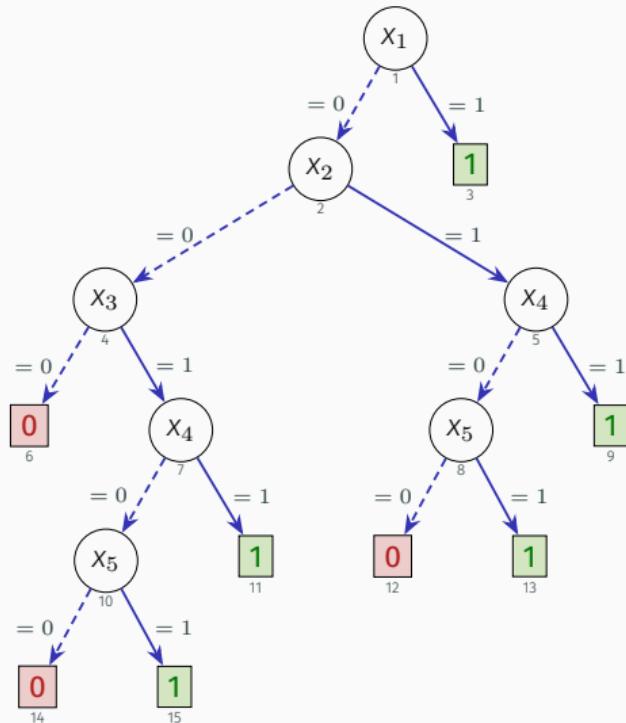
Detour: From Decision Trees to Explained Decision Sets

Explanations for Decision Graphs

Explanations for Monotonic Classifiers

Review examples

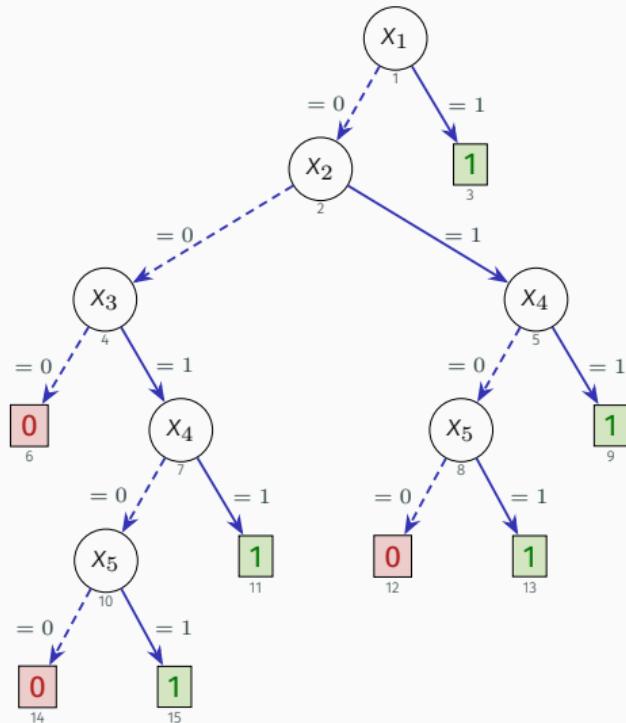
# Are *interpretable* models really interpretable? – DTs



- Case of **optimal** decision tree (DT)
- Explanation for  $(0, 0, 1, 0, 1)$ , with prediction 1?

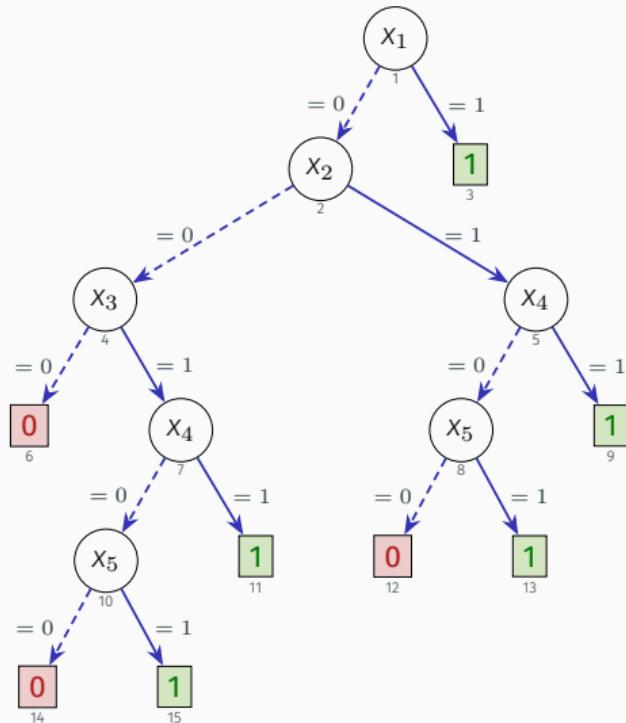
[HRS19]

# Are *interpretable* models really interpretable? – DTs



- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for  $(0, 0, 1, 0, 1)$ , with prediction 1?
  - Clearly, IF  $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5$  THEN  $\kappa(\mathbf{x}) = 1$

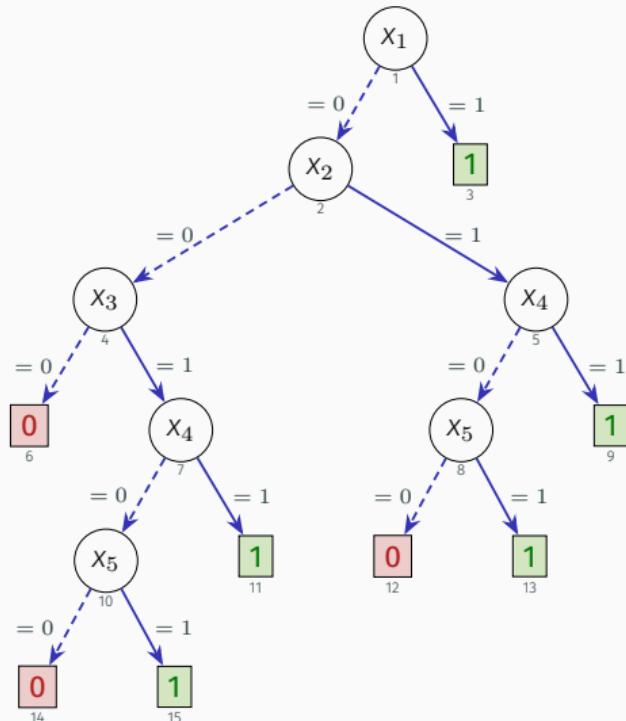
# Are *interpretable* models really interpretable? – DTs



- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for  $(0, 0, 1, 0, 1)$ , with prediction 1?
  - Clearly, IF  $\neg X_1 \wedge \neg X_2 \wedge X_3 \wedge \neg X_4 \wedge X_5$  THEN  $\kappa(\mathbf{x}) = 1$
  - But,  $X_1, X_2, X_4$  are **irrelevant** for the prediction:

$X_3$	$X_5$	$X_1$	$X_2$	$X_4$	$\kappa(\mathbf{x})$
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

# Are *interpretable* models really interpretable? – DTs

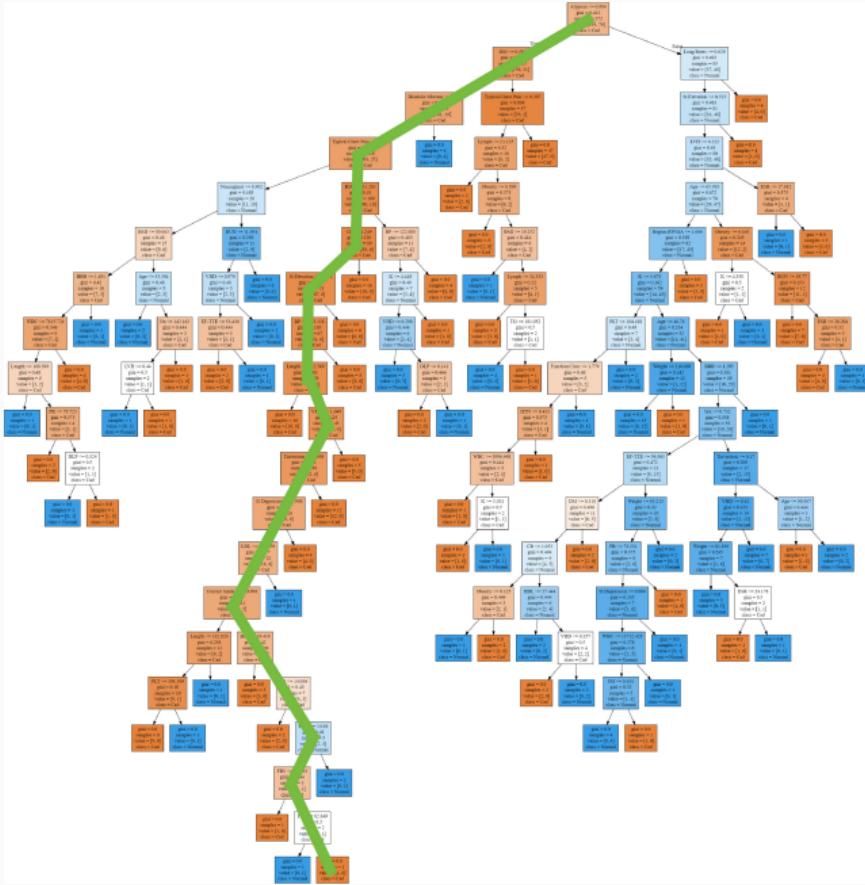


- Case of **optimal** decision tree (DT) [HRS19]
- Explanation for (0, 0, 1, 0, 1), with prediction 1?
  - Clearly, IF  $\neg x_1 \wedge \neg x_2 \wedge x_3 \wedge \neg x_4 \wedge x_5$  THEN  $\kappa(\mathbf{x}) = 1$
  - But,  $x_1, x_2, x_4$  are **irrelevant** for the prediction:

$x_3$	$x_5$	$x_1$	$x_2$	$x_4$	$\kappa(\mathbf{x})$
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

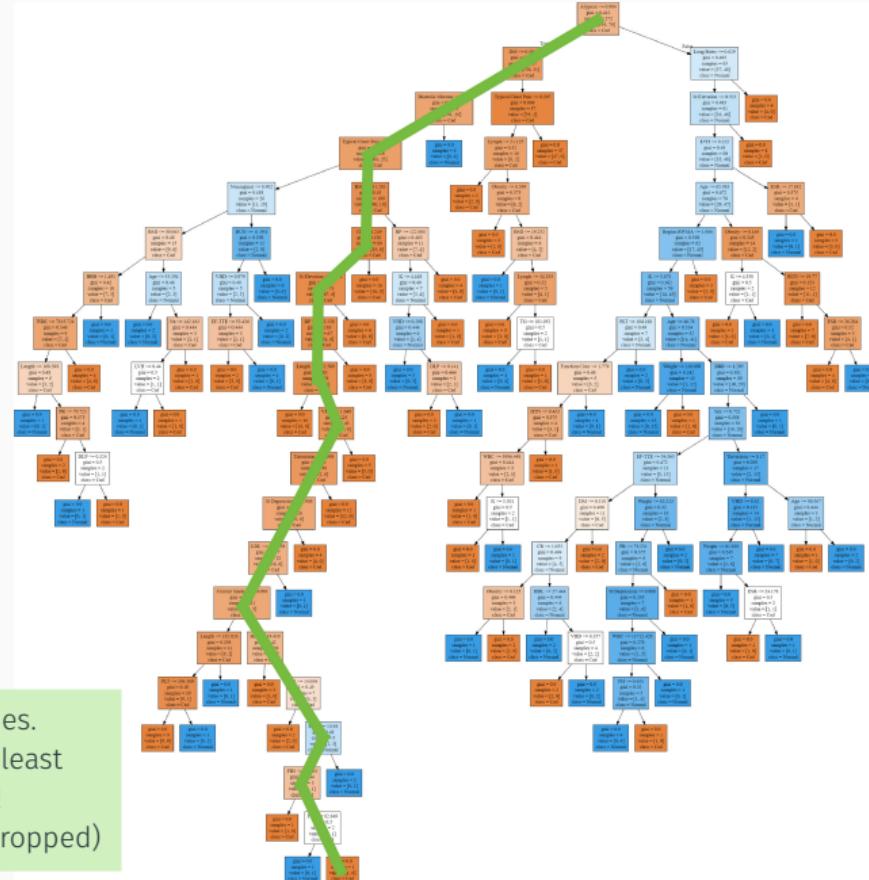
$\therefore$  one AXp is {3, 5}  
Compare with {1, 2, 3, 4, 5}...

# Are *interpretable* models really interpretable? – large DTs



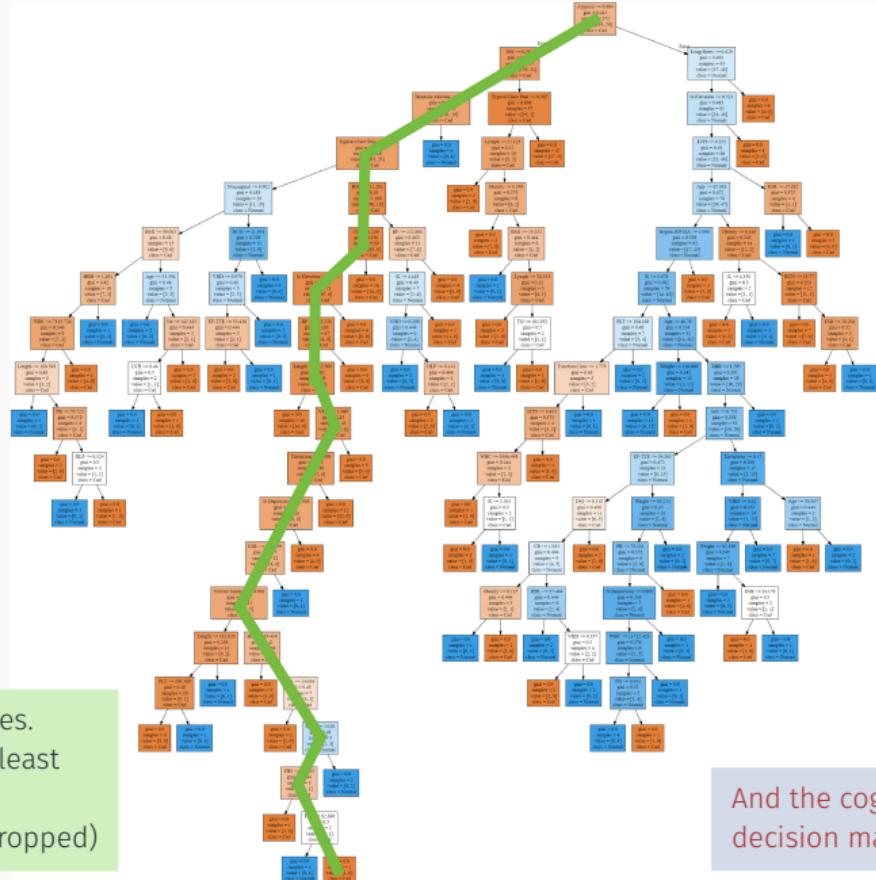
[GZM20]

# Are *interpretable* models really interpretable? – large DTs



[GZM20]

# Are *interpretable* models really interpretable? – large DTs

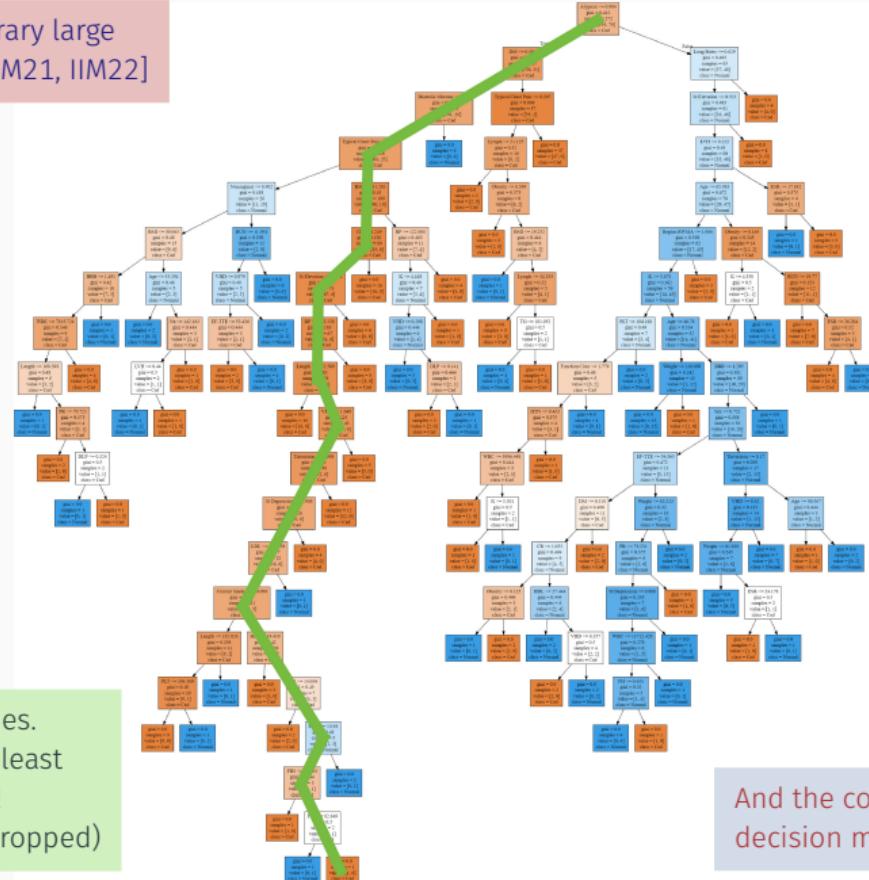


[GZM20]

# Are *interpretable* models really interpretable? – large DTs

Redundancy can be arbitrary large  
on path length [IIM20, HIIM21, IIM22]

[GZM20]



Path with 19 internal nodes.  
By manual inspection, at least  
10 literals are redundant!  
(And at least 9 features dropped)

And the cognitive limits of human  
decision makers are well-known [Mil56]

# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with  $x_1, \dots, x_m \in \{0, 1\}$ :

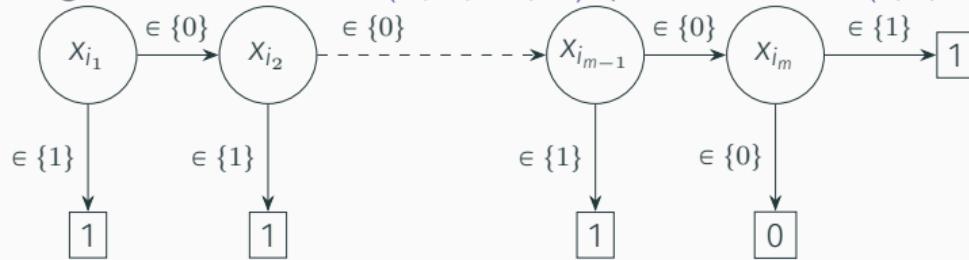
$$\kappa(x_1, x_2, \dots, x_{m-1}, x_m) = \bigvee_{i=1}^m x_i$$

# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with  $x_1, \dots, x_m \in \{0, 1\}$ :

$$\kappa(x_1, x_2, \dots, x_{m-1}, x_m) = \bigvee_{i=1}^m x_i$$

- Build DT, by picking variables in order  $\langle i_1, i_2, \dots, i_m \rangle$ , permutation of  $\langle 1, 2, \dots, m \rangle$ :

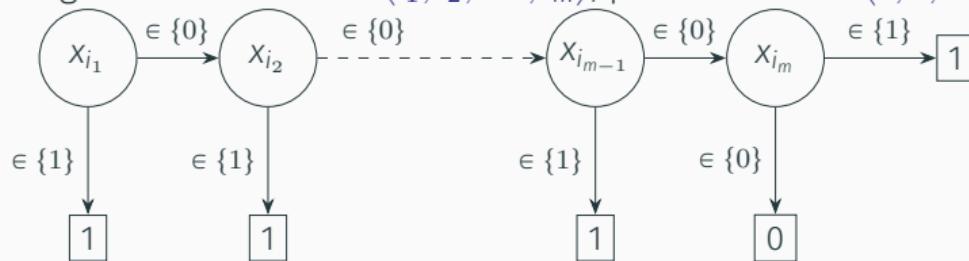


# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with  $x_1, \dots, x_m \in \{0, 1\}$ :

$$\kappa(x_1, x_2, \dots, x_{m-1}, x_m) = \bigvee_{i=1}^m x_i$$

- Build DT, by picking variables in order  $\langle i_1, i_2, \dots, i_m \rangle$ , permutation of  $\langle 1, 2, \dots, m \rangle$ :



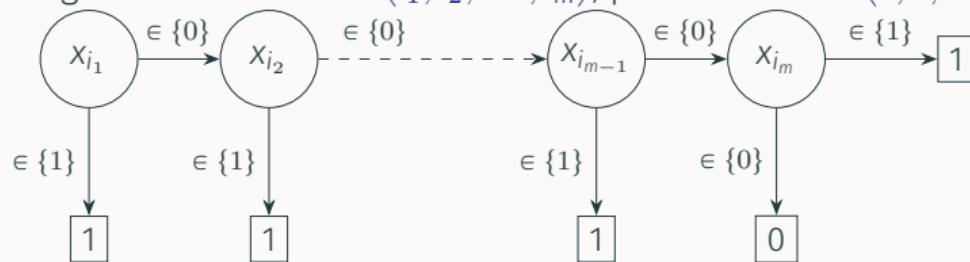
- Point:  $(x_{i_1}, x_{i_2}, \dots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \dots, 0, 1)$ , and prediction 1

# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with  $x_1, \dots, x_m \in \{0, 1\}$ :

$$\kappa(x_1, x_2, \dots, x_{m-1}, x_m) = \bigvee_{i=1}^m x_i$$

- Build DT, by picking variables in order  $\langle i_1, i_2, \dots, i_m \rangle$ , permutation of  $\langle 1, 2, \dots, m \rangle$ :



- Point:  $(x_{i_1}, x_{i_2}, \dots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \dots, 0, 1)$ , and prediction 1
- Explanation using path in DT:  $\{i_1, i_2, \dots, i_m\}$ , i.e.

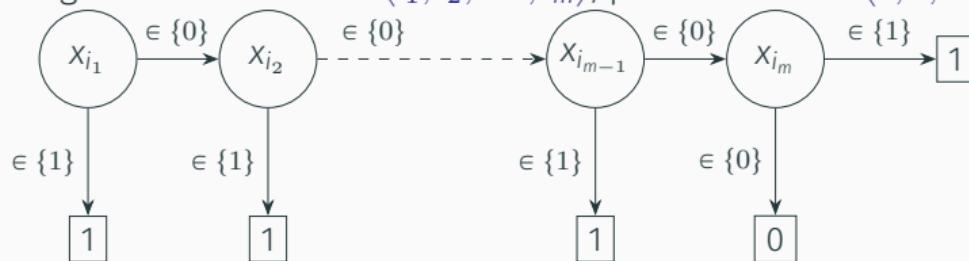
$$(x_{i_1} = 0) \wedge (x_{i_2} = 0) \wedge \dots \wedge (x_{i_{m-1}} = 0) \wedge (x_{i_m} = 1) \rightarrow \kappa(x_1, \dots, x_m)$$

# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, HIIM21, IIM22]

- Classifier, with  $x_1, \dots, x_m \in \{0, 1\}$ :

$$\kappa(x_1, x_2, \dots, x_{m-1}, x_m) = \bigvee_{i=1}^m x_i$$

- Build DT, by picking variables in order  $\langle i_1, i_2, \dots, i_m \rangle$ , permutation of  $\langle 1, 2, \dots, m \rangle$ :



- Point:  $(x_{i_1}, x_{i_2}, \dots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \dots, 0, 1)$ , and prediction 1
- Explanation using path in DT:  $\{i_1, i_2, \dots, i_m\}$ , i.e.

$$(x_{i_1} = 0) \wedge (x_{i_2} = 0) \wedge \dots \wedge (x_{i_{m-1}} = 0) \wedge (x_{i_m} = 1) \rightarrow \kappa(x_1, \dots, x_m)$$

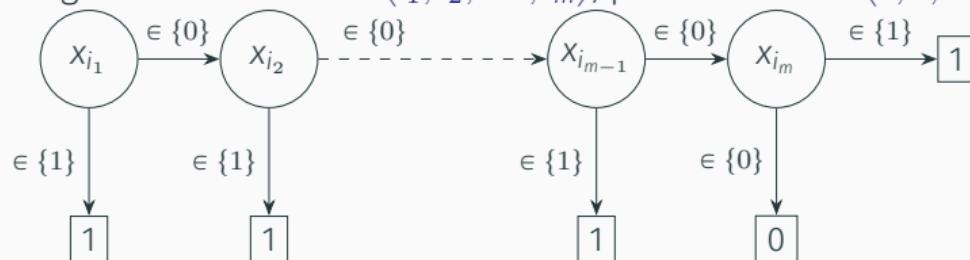
- But  $\{i_m\}$  suffices for prediction, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^m). (x_{i_m}) \rightarrow \kappa(\mathbf{x})$

# Are *interpretable* models really interpretable? – arbitrary redundancy [IIM20, IIM21, IIM22]

- Classifier, with  $x_1, \dots, x_m \in \{0, 1\}$ :

$$\kappa(x_1, x_2, \dots, x_{m-1}, x_m) = \bigvee_{i=1}^m x_i$$

- Build DT, by picking variables in order  $\langle i_1, i_2, \dots, i_m \rangle$ , permutation of  $\langle 1, 2, \dots, m \rangle$ :



- Point:  $(x_{i_1}, x_{i_2}, \dots, x_{i_{m-1}}, x_{i_m}) = (0, 0, \dots, 0, 1)$ , and prediction 1
- Explanation using path in DT:  $\{i_1, i_2, \dots, i_m\}$ , i.e.

$$(x_{i_1} = 0) \wedge (x_{i_2} = 0) \wedge \dots \wedge (x_{i_{m-1}} = 0) \wedge (x_{i_m} = 1) \rightarrow \kappa(x_1, \dots, x_m)$$

- But  $\{i_m\}$  suffices for prediction, i.e.  $\forall (\mathbf{x} \in \{0, 1\}^m). (x_{i_m}) \rightarrow \kappa(\mathbf{x})$

- AXp's can be arbitrarily smaller than paths in (optimal) DTs!

[IIM20, IIM22]

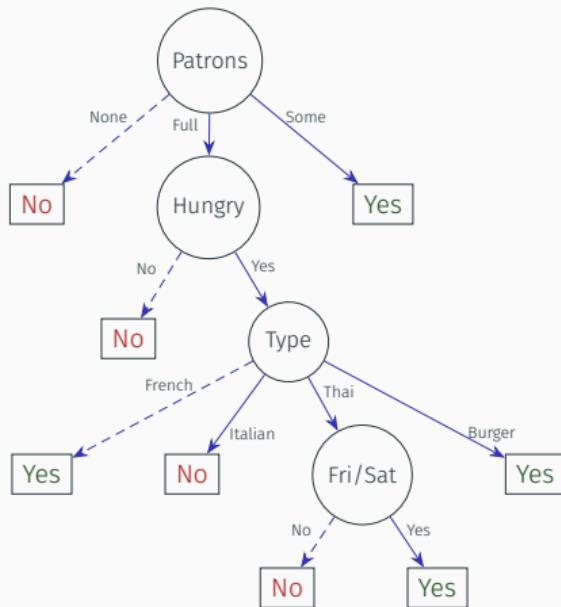
# Explanation redundancy in DTs is ubiquitous – published DT examples

[IIM22]

DT Ref	D	#N	#P	%R	%C	%m	%M	%avg
[Alp14, Ch. 09, Fig. 9.1]	2	5	3	33	25	50	50	50
[Alp16, Ch. 03, Fig. 3.2]	2	5	3	33	25	50	50	50
[Bra20, Ch. 01, Fig. 1.3]	4	9	5	60	25	25	50	36
[BA97, Figure 1]	3	12	7	14	8	33	33	33
[BBHK10, Ch. 08, Fig. 8.2]	3	7	4	25	12	50	50	50
[BFOS84, Ch. 01, Fig. 1.1]	3	7	4	50	25	33	33	33
[DL01, Ch. 01, Fig. 1.2a]	2	5	3	33	25	33	33	33
[DL01, Ch. 01, Fig. 1.2b]	2	5	3	33	25	33	33	33
[KMND20, Ch. 04, Fig. 4.14]	3	7	4	25	12	50	50	50
[KMND20, Sec. 4.7, Ex. 4]	2	5	3	33	25	50	50	50
[Qui93, Ch. 01, Fig. 1.3]	3	12	7	28	17	33	50	41
[RM08, Ch. 01, Fig. 1.5]	3	9	5	20	12	33	33	33
[RM08, Ch. 01, Fig. 1.4]	3	7	4	50	25	33	33	33
[WFHP17, Ch. 01, Fig. 1.2]	3	7	4	25	12	50	50	50
[VLE <sup>+</sup> 16, Figure 4]	6	39	20	65	63	20	40	33
[Fla12, Ch. 02, Fig. 2.1(right)]	2	5	3	33	25	50	50	50
[Kot13, Figure 1]	3	10	6	33	11	33	33	33
[Mor82, Figure 1]	3	9	5	80	75	33	50	41
[PM17, Ch. 07, Fig. 7.4]	3	7	4	50	25	33	33	33
[RN10, Ch. 18, Fig. 18.6]	4	12	8	25	6	25	33	29
[SB14, Ch. 18, Page 212]	2	5	3	33	25	50	50	50
[Zho12, Ch. 01, Fig. 1.3]	2	5	3	33	25	33	33	33
[BHO09, Figure 1b]	4	13	7	71	50	33	50	36
[Zho21, Ch. 04, Fig. 4.3]	4	14	9	11	2	25	25	25

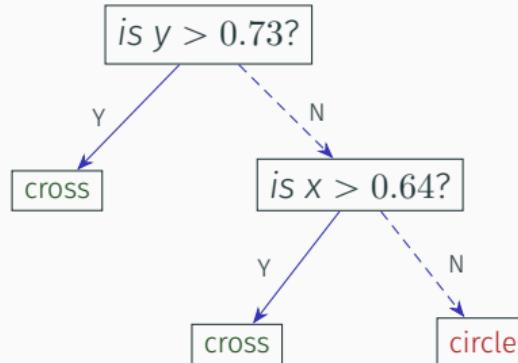
# Many DTs have paths that are not minimal XPs – Russell&Norvig's book

[RN10]



- Explanation for  $(P, H, T, W) = (\text{Full}, \text{Yes}, \text{Thai}, \text{No})$ ?

## Many DTs have paths that are not minimal XPs – Zhou's book



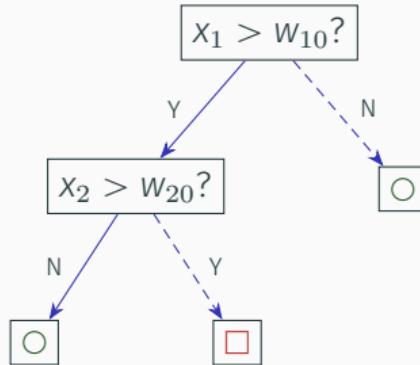
[Zho12]

- Explanation for  $(x, y) = (1.25, -1.13)$ ?

**Obs:** True explanations can be computed for categorical, integer or real-valued features !

## Many DTs have paths that are not minimal XPs – Alpaydin's book

[Alp14]

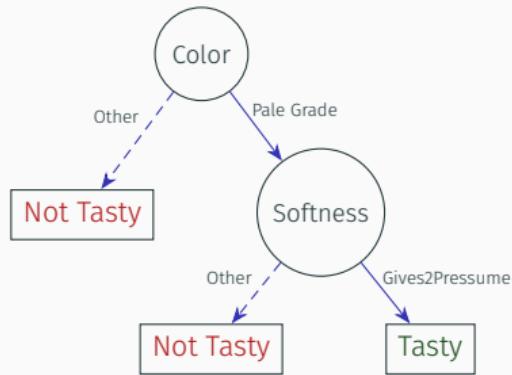


- Explanation for  $(x_1, x_2) = (\alpha, \beta)$ , with  $\alpha > w_{10}$  and  $\beta \leq w_{20}$ ?

**Obs:** True explanations can be computed for categorical, integer or real-valued features !

## Many DTs have paths that are not minimal XPs – S.-S.&B.-D.'s book

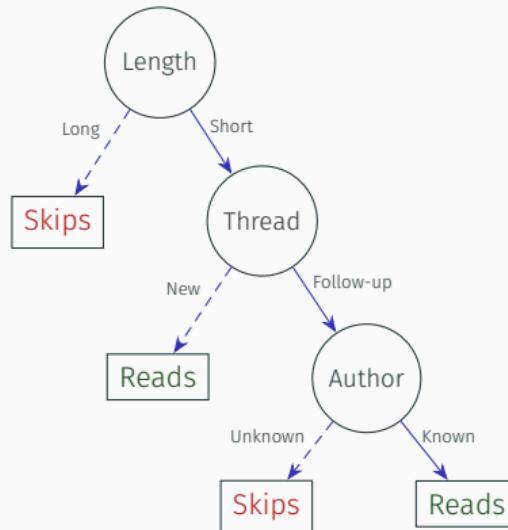
[SB14]



- Explanation for  $(\text{color}, \text{softness}) = (\text{Pale Grade}, \text{Other})$ ?

## Many DTs have paths that are not minimal XPs – Poole&Mackworth's book

[PM17]



- Explanation for  $(L, T, A) = (\text{Short}, \text{Follow-Up}, \text{Unknown})$ ?
- Explanation for  $(L, T, A) = (\text{Short}, \text{Follow-Up}, \text{Known})$ ?

# Explanation redundancy in DTs is ubiquitous – DTs from datasets

[IIM20, HIIM21, IIM22]

Dataset	(#F #S)	IAI										ITI									
		D	#N	%A	#P	%R	%C	%m	%M	%avg	D	#N	%A	#P	%R	%C	%m	%M	%avg		
adult	( 12 6061)	6	83	78	42	33	25	20	40	25	17	509	73	255	75	91	10	66	22		
anneal	( 38 886)	6	29	99	15	26	16	16	33	21	9	31	100	16	25	4	12	20	16		
backache	( 32 180)	4	17	72	9	33	39	25	33	30	3	9	91	5	80	87	50	66	54		
bank	( 19 36293)	6	113	88	57	5	12	16	20	18	19	1467	86	734	69	64	7	63	27		
biodegradation	( 41 1052)	5	19	65	10	30	1	25	50	33	8	71	76	36	50	8	14	40	21		
cancer	( 9 449)	6	37	87	19	36	9	20	25	21	5	21	84	11	54	10	25	50	37		
car	( 6 1728)	6	43	96	22	86	89	20	80	45	11	57	98	29	65	41	16	50	30		
colic	( 22 357)	6	55	81	28	46	6	16	33	20	4	17	80	9	33	27	25	25	25		
compas	( 11 1155)	6	77	34	39	17	8	16	20	17	15	183	37	92	66	43	12	60	27		
contraceptive	( 9 1425)	6	99	49	50	8	2	20	60	37	17	385	48	193	27	32	12	66	21		
dermatology	( 34 366)	6	33	90	17	23	3	16	33	21	7	17	95	9	22	0	14	20	17		
divorce	( 54 150)	5	15	90	8	50	19	20	33	24	2	5	96	3	33	16	50	50	50		
german	( 21 1000)	6	25	61	13	38	10	20	40	29	10	99	72	50	46	13	12	40	22		
heart-c	( 13 302)	6	43	65	22	36	18	20	33	22	4	15	75	8	87	81	25	50	34		
heart-h	( 13 293)	6	37	59	19	31	4	20	40	24	8	25	77	13	61	60	20	50	32		
kr-vs-kp	( 36 3196)	6	49	96	25	80	75	16	60	33	13	67	99	34	79	43	7	70	35		
lending	( 9 5082)	6	45	73	23	73	80	16	50	25	14	507	65	254	69	80	12	75	25		
letter	( 16 18668)	6	127	58	64	1	0	20	20	20	46	4857	68	2429	6	7	6	25	9		
lymphography	( 18 148)	6	61	76	31	35	25	16	33	21	6	21	86	11	9	0	16	16	16		
mortality	(118 13442)	6	111	74	56	8	14	16	20	17	26	865	76	433	61	61	7	54	19		
mushroom	( 22 8124)	6	39	100	20	80	44	16	33	24	5	23	100	12	50	31	20	40	25		
pendigits	( 16 10992)	6	121	88	61	0	0	—	—	—	38	937	85	469	25	86	6	25	11		
promoters	( 58 106)	1	3	90	2	0	0	—	—	—	3	9	81	5	20	14	33	33	33		
recidivism	( 15 3998)	6	105	61	53	28	22	16	33	18	15	611	51	306	53	38	9	44	16		
seismic_bumps	( 18 2578)	6	37	89	19	42	19	20	33	24	8	39	93	20	60	79	20	60	42		
shuttle	( 9 58000)	6	63	99	32	28	7	20	33	23	23	159	99	80	33	9	14	50	30		
soybean	( 35 623)	6	63	88	32	9	5	25	25	25	16	71	89	36	22	1	9	12	10		
spambase	( 57 4210)	6	63	75	32	37	12	16	33	19	15	143	91	72	76	98	7	58	25		
spect	( 22 228)	6	45	82	23	60	51	20	50	35	6	15	86	8	87	98	50	83	65		
splice	( 2 3178)	3	7	50	4	0	0	—	—	—	88	177	55	89	0	0	—	—	—		

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires
- What is the abductive explanation?

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires
- What is the abductive explanation?
- Recall: one AXp is  $\{3, 4, 6\}$

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

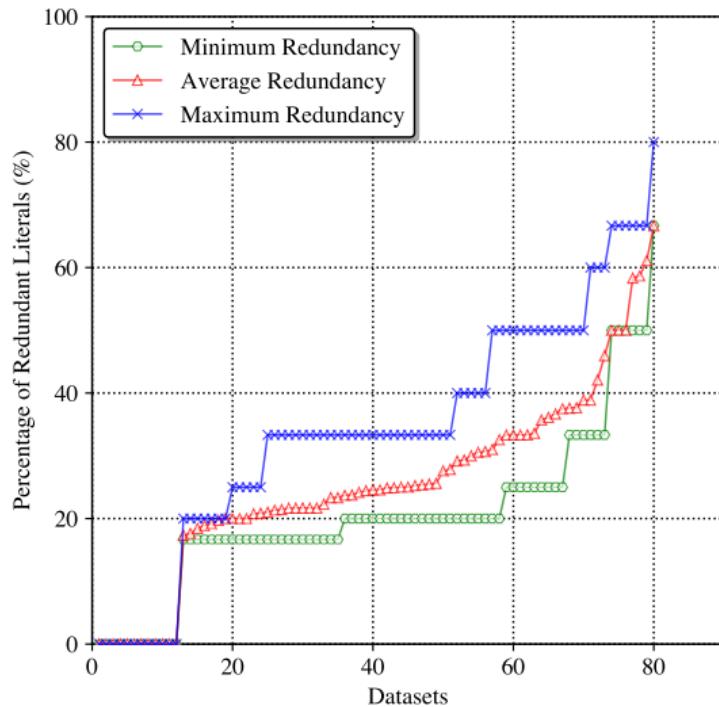
- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires
- What is the abductive explanation?
- Recall: one AXp is  $\{3, 4, 6\}$ 
  - Why?
    - We need 3 (or 1) so that  $R_1$  cannot fire
    - With 3, we do not need 2, since with 4 and 6 fixed, then  $R_4$  is guaranteed to fire
  - Some questions:
    - Would average human decision maker be able to understand the AXp?
    - Would he/she be able to compute one AXp, by manual inspection?

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_2 :$	ELSE IF	$(x_2 \wedge x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(\neg x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_1 \wedge \neg x_3)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(x_6)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 1$

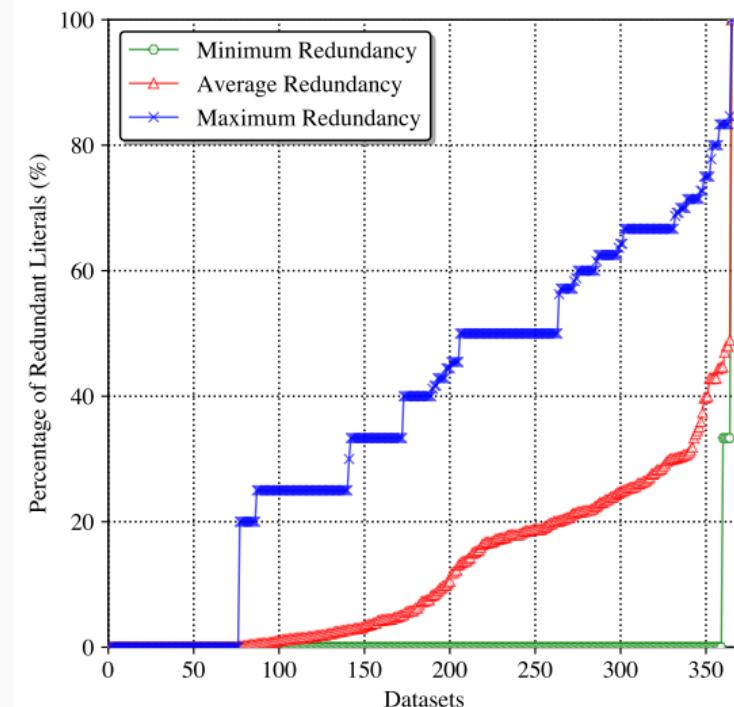
- Instance:  $((0, 1, 0, 1, 0, 1), 0)$ , i.e. rule  $R_2$  fires
  - What is the abductive explanation?
  - Recall: one AXp is  $\{3, 4, 6\}$ 
    - Why?
      - We need 3 (or 1) so that  $R_1$  cannot fire
      - With 3, we do not need 2, since with 4 and 6 fixed, then  $R_4$  is guaranteed to fire
    - Some questions:
      - Would average human decision maker be able to understand the AXp?
      - Would he/she be able to compute one AXp, by manual inspection?
- (BTW, we have proved that computing one AXp for DLs is computationally hard...)

# Are *interpretable* models really interpretable? – DTs/DLs in practice

[MSI23]



DTs learned with Interpretable AI, max depth 6



DLs learned with CN2

# Outline – Unit #03

Explanations for Decision Trees

XAI Queries for DTs

Myth #01: Intrinsic Interpretability

Detour: From Decision Trees to Explained Decision Sets

Explanations for Decision Graphs

Explanations for Monotonic Classifiers

Review examples

## From DTs to explained DSs

[HM23a]

- Decision sets raise a number of issues:
  - **Overlap**: Two rules with different predictions can fire on the same input
  - **Incomplete coverage**: For some inputs, no rule may fire
    - A default rule defeats the purpose of unordered rules

# From DTs to explained DSs

[HM23a]

- Decision sets raise a number of issues:
  - **Overlap**: Two rules with different predictions can fire on the same input
  - **Incomplete coverage**: For some inputs, no rule may fire
    - A default rule defeats the purpose of unordered rules
  - A DS without overlap and complete coverage computes a classification function

# From DTs to explained DSs

[HM23a]

- Decision sets raise a number of issues:
  - **Overlap**: Two rules with different predictions can fire on the same input
  - **Incomplete coverage**: For some inputs, no rule may fire
    - A default rule defeats the purpose of unordered rules
  - A DS without overlap and complete coverage computes a classification function
- And explaining DSs is computationally hard...

# From DTs to explained DSs

[HM23a]

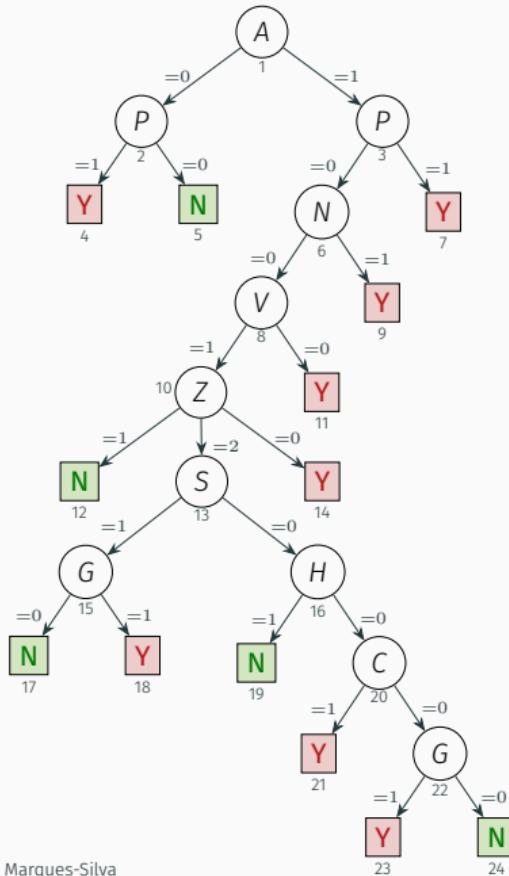
- Decision sets raise a number of issues:
  - **Overlap**: Two rules with different predictions can fire on the same input
  - **Incomplete coverage**: For some inputs, no rule may fire
    - A default rule defeats the purpose of unordered rules
  - A DS without overlap and complete coverage computes a classification function
- And explaining DSs is computationally hard...
- One can extract explained DSs from DTs

# From DTs to explained DSs

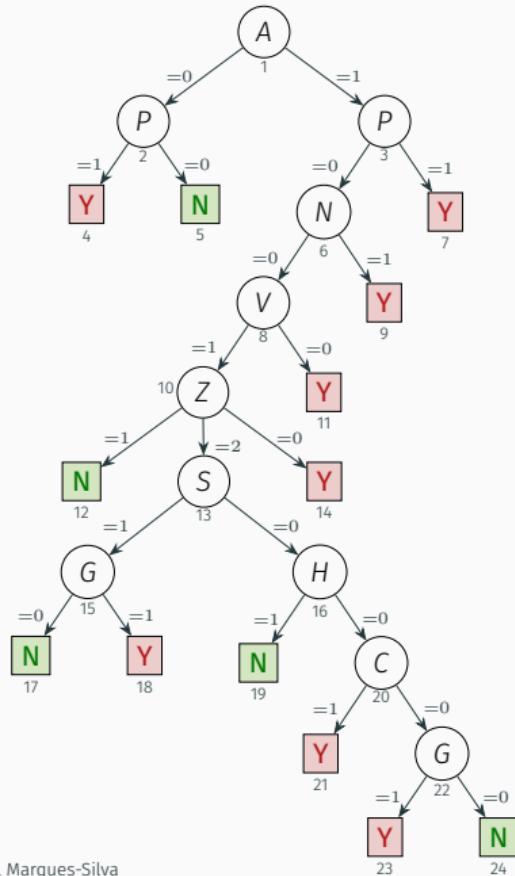
[HM23a]

- Decision sets raise a number of issues:
  - **Overlap**: Two rules with different predictions can fire on the same input
  - **Incomplete coverage**: For some inputs, no rule may fire
    - A default rule defeats the purpose of unordered rules
  - A DS without overlap and complete coverage computes a classification function
- And explaining DSs is computationally hard...
- One can extract explained DSs from DTs
  - Extract one AXp (viewed as a logic rule) from each path in DT
  - Resulting rules are non-overlapping, and cover feature space

# Example



# Example



- R<sub>01</sub>: IF [P] THEN  $\kappa(\cdot) = \mathbf{Y}$
- R<sub>02</sub>: IF [ $\bar{A} \wedge \bar{P}$ ] THEN  $\kappa(\cdot) = \mathbf{N}$
- R<sub>03</sub>: IF [ $\bar{P} \wedge \bar{N} \wedge V \wedge Z = 1$ ] THEN  $\kappa(\cdot) = \mathbf{N}$
- R<sub>04</sub>: IF [ $\bar{P} \wedge \bar{N} \wedge V \wedge Z = 2 \wedge S \wedge \bar{G}$ ] THEN  $\kappa(\cdot) = \mathbf{N}$
- R<sub>05</sub>: IF [A  $\wedge$  Z = 2  $\wedge$  S  $\wedge$  G] THEN  $\kappa(\cdot) = \mathbf{Y}$
- R<sub>06</sub>: IF [ $\bar{P} \wedge \bar{N} \wedge V \wedge Z = 2 \wedge \bar{S} \wedge H$ ] THEN  $\kappa(\cdot) = \mathbf{N}$
- R<sub>07</sub>: IF [A  $\wedge$  Z = 2  $\wedge$   $\bar{S}$   $\wedge$   $\bar{H}$   $\wedge$  C] THEN  $\kappa(\cdot) = \mathbf{Y}$
- R<sub>08</sub>: IF [A  $\wedge$  Z = 2  $\wedge$   $\bar{H}$   $\wedge$  G] THEN  $\kappa(\cdot) = \mathbf{Y}$
- R<sub>09</sub>: IF [ $\bar{P} \wedge \bar{N} \wedge V \wedge Z = 2 \wedge \bar{C} \wedge \bar{G}$ ] THEN  $\kappa(\cdot) = \mathbf{N}$
- R<sub>10</sub>: IF [A  $\wedge$  Z = 0] THEN  $\kappa(\cdot) = \mathbf{Y}$
- R<sub>11</sub>: IF [A  $\wedge$   $\bar{V}$ ] THEN  $\kappa(\cdot) = \mathbf{Y}$
- R<sub>12</sub>: IF [A  $\wedge$  N] THEN  $\kappa(\cdot) = \mathbf{Y}$

# Outline – Unit #03

Explanations for Decision Trees

XAI Queries for DTs

Myth #01: Intrinsic Interpretability

Detour: From Decision Trees to Explained Decision Sets

Explanations for Decision Graphs

Explanations for Monotonic Classifiers

Review examples

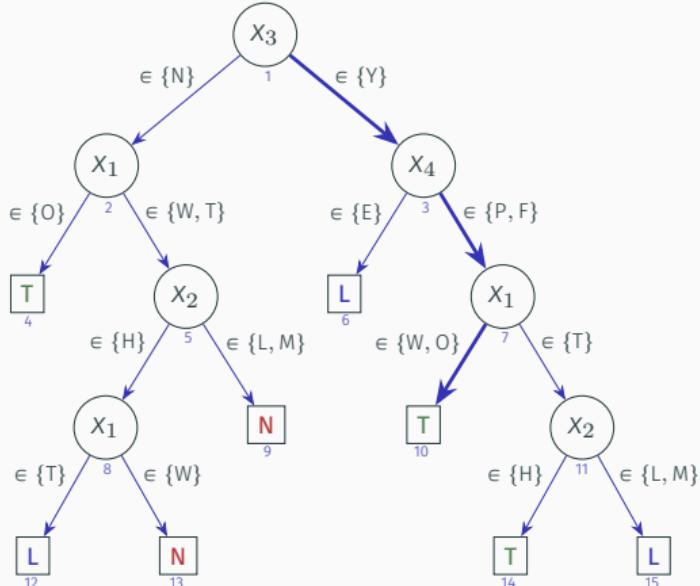
# Explanation graphs – overview of results

[HIIM21]

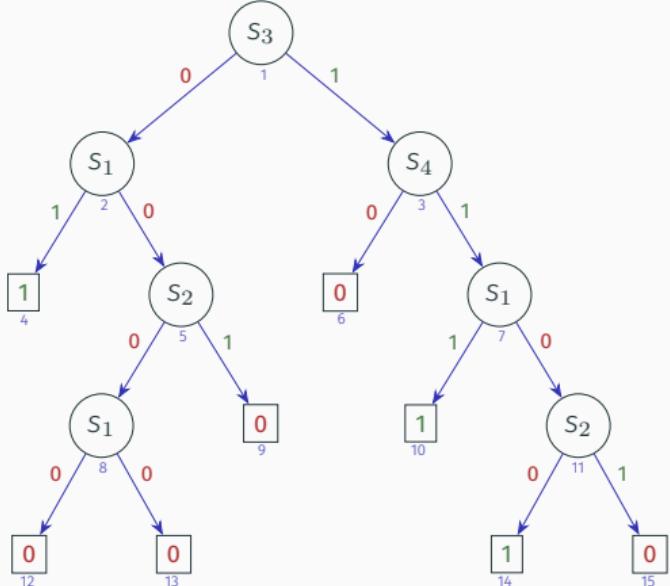
- Concept of explanation graph (XpG)
- Explanations of decision trees reducible to XpG's
- Explanations of decision graphs reducible to XpG's
- Explanations of OBDDs reducible to XpG's
- Explanations of OMDDs reducible to XpG's
- Explanations (AXp's and CXp's) of XpG's computed in polynomial time

# Example of XpG – DTs

- DT; point:  $(O, L, Y, P)$ ; prediction  $T$ :

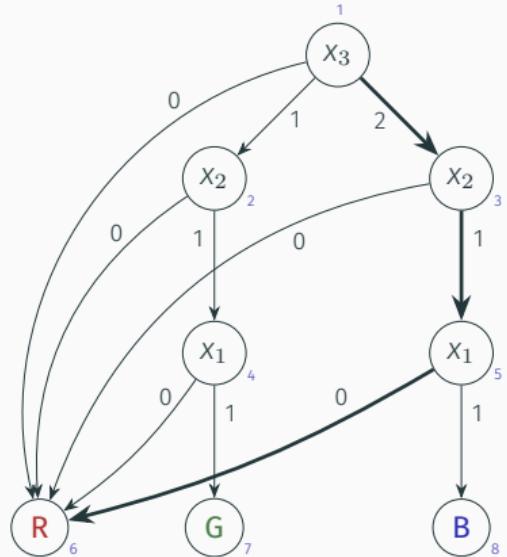


- XpG:

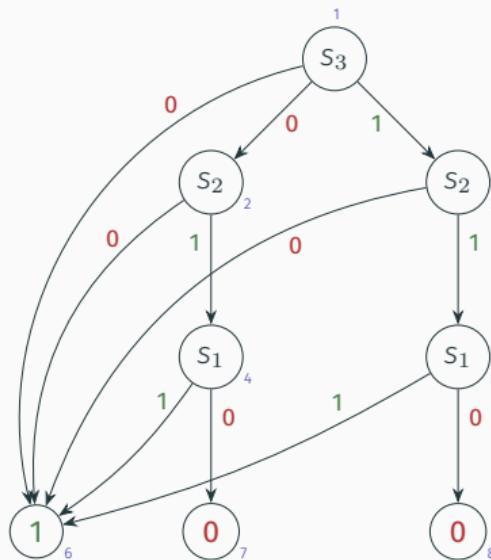


## Example of XpG – OMDDs

- OMBBD; point:  $(0, 1, 2)$ ; prediction  $R$ :



- **XpG:**



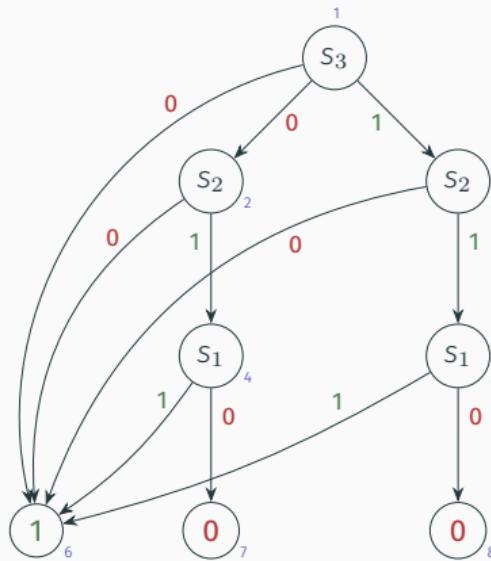
## Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

$$S \leftarrow \mathcal{F}$$

For each feature  $i$  in  $\mathcal{F}$

- XpG:



## Finding one AXp for XpGs – polynomial time

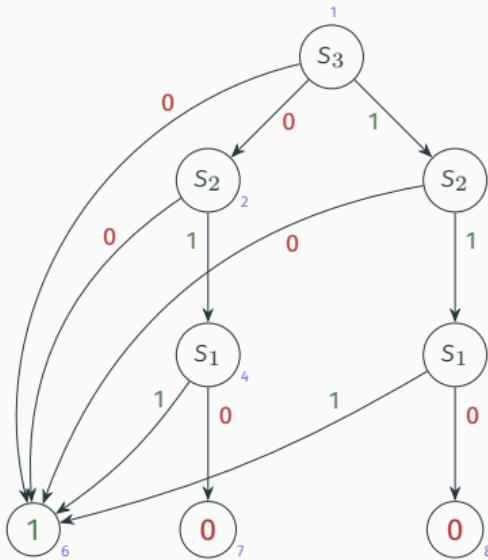
- Algorithm (with no inconsistent paths):

$$\mathcal{S} \leftarrow \mathcal{F}$$

For each feature  $i$  in  $\mathcal{F}$

Drop feature  $i$  from  $\mathcal{S}$ , i.e.  $i$  is free

- XpG:



# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

$\mathcal{S} \leftarrow \mathcal{F}$

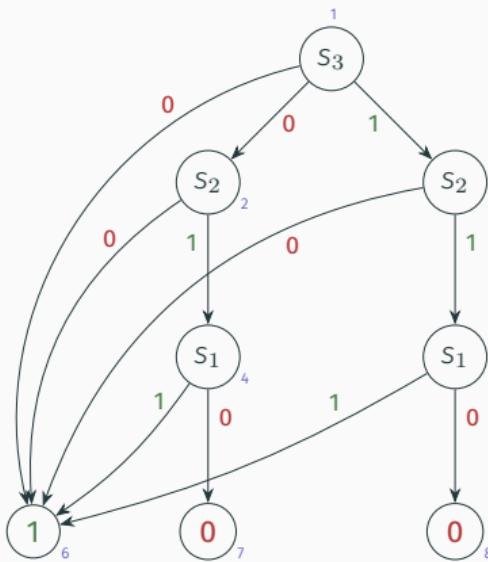
For each feature  $i$  in  $\mathcal{F}$

Drop feature  $i$  from  $\mathcal{S}$ , i.e.  $i$  is free

If path to some **0** not blocked by 0-valued literals, then

Add feature  $i$  back to  $\mathcal{S}$

- XpG:



# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

$\mathcal{S} \leftarrow \mathcal{F}$

For each feature  $i$  in  $\mathcal{F}$

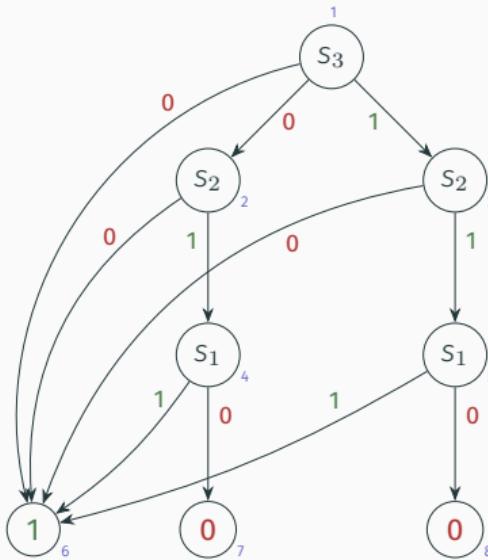
Drop feature  $i$  from  $\mathcal{S}$ , i.e.  $i$  is free

If path to some **0** not blocked by 0-valued literals, then

Add feature  $i$  back to  $\mathcal{S}$

Return  $\mathcal{S}$

- XpG:



# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

$\mathcal{S} \leftarrow \mathcal{F}$

For each feature  $i$  in  $\mathcal{F}$

Drop feature  $i$  from  $\mathcal{S}$ , i.e.  $i$  is free

If path to some **0** not blocked by 0-valued literals, then

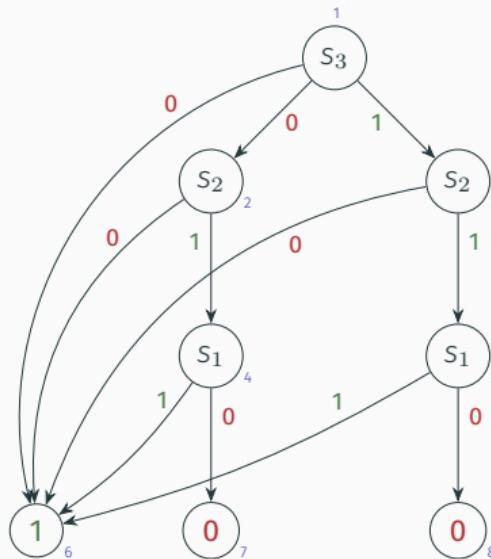
Add feature  $i$  back to  $\mathcal{S}$

Return  $\mathcal{S}$

- Example:

- $\mathcal{S} = \{1, 2, 3\}$

- XpG:



# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

$$\mathcal{S} \leftarrow \mathcal{F}$$

For each feature  $i$  in  $\mathcal{F}$

Drop feature  $i$  from  $\mathcal{S}$ , i.e.  $i$  is free  
If path to some **0** not blocked by  
0-valued literals, then

Add feature  $i$  back to  $\mathcal{S}$

Return  $\mathcal{S}$

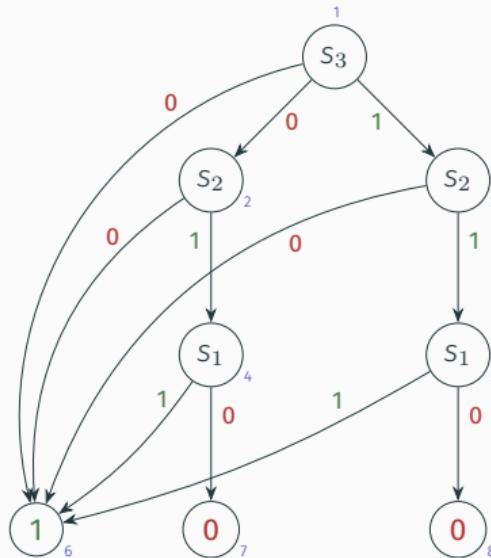
- Example:

- $\mathcal{S} = \{1, 2, 3\}$

- Feature 1 cannot be dropped, e.g.

$s_3 \rightarrow s_2 \rightarrow s_1 \rightarrow 0$

- XpG:



# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

$\mathcal{S} \leftarrow \mathcal{F}$

For each feature  $i$  in  $\mathcal{F}$

Drop feature  $i$  from  $\mathcal{S}$ , i.e.  $i$  is free

If path to some **0** not blocked by 0-valued literals, then

Add feature  $i$  back to  $\mathcal{S}$

Return  $\mathcal{S}$

- Example:

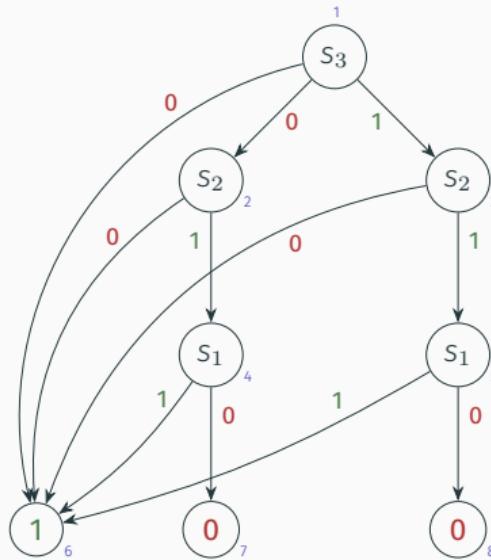
- $\mathcal{S} = \{1, 2, 3\}$

- Feature 1 cannot be dropped, e.g.

$s_3 \rightarrow s_2 \rightarrow s_1 \rightarrow 0$

- Both features 2 and 3 dropped from  $\mathcal{S}$

- XpG:



# Finding one AXp for XpGs – polynomial time

- Algorithm (with no inconsistent paths):

$\mathcal{S} \leftarrow \mathcal{F}$

For each feature  $i$  in  $\mathcal{F}$

Drop feature  $i$  from  $\mathcal{S}$ , i.e.  $i$  is free

If path to some **0** not blocked by 0-valued literals, then

Add feature  $i$  back to  $\mathcal{S}$

Return  $\mathcal{S}$

- Example:

- $\mathcal{S} = \{1, 2, 3\}$

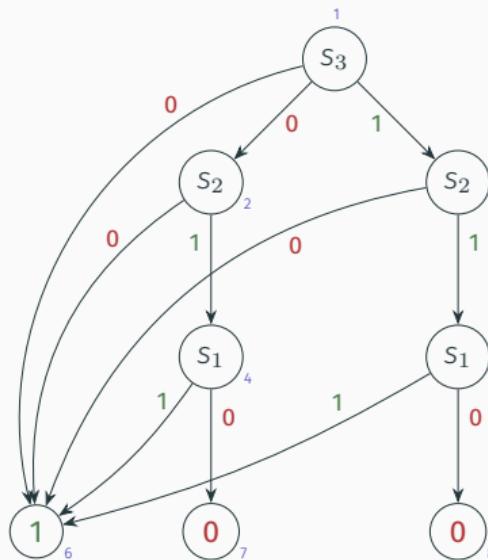
- Feature 1 cannot be dropped, e.g.

$s_3 \rightarrow s_2 \rightarrow s_1 \rightarrow 0$

- Both features 2 and 3 dropped from  $\mathcal{S}$

- Return  $\mathcal{S} = \{1\}$

- XpG:



# Outline – Unit #03

Explanations for Decision Trees

XAI Queries for DTs

Myth #01: Intrinsic Interpretability

Detour: From Decision Trees to Explained Decision Sets

Explanations for Decision Graphs

Explanations for Monotonic Classifiers

Review examples

## Example monotonic classifier – $(\mathbf{v}, c) = ((10, 10, 5, 0), A)$

[MGC<sup>+</sup>21]

Variable	Meaning		Range
$\kappa(\cdot) \triangleq M$	Student grade		$\in \{A, B, C, D, E, F\}$
$S$	Final score		$\in \{0, \dots, 10\}$
Feat. id	Feat. var.	Feat. name	Domain
1	$Q$	Quiz	$\{0, \dots, 10\}$
2	$X$	Exam	$\{0, \dots, 10\}$
3	$H$	Homework	$\{0, \dots, 10\}$
4	$R$	Project	$\{0, \dots, 10\}$

$$M = \text{ITE}(S \geq 9, A, \text{ITE}(S \geq 7, B, \text{ITE}(S \geq 5, C, \text{ITE}(S \geq 4, D, \text{ite}(S \geq 2, E, F)))))$$

$$S = \max [0.3 \times Q + 0.6 \times X + 0.1 \times H, R]$$

$$\text{Also, } F \leq E \leq D \leq C \leq B \leq A$$

$$\text{And, } \kappa(\mathbf{x}_1) \leq \kappa(\mathbf{x}_2) \text{ if } \mathbf{x}_1 \leq \mathbf{x}_2$$

# Explaining monotonic classifiers

- Instance  $(\mathbf{v}, c)$
- Domain for  $i \in \mathcal{F}$ :  $\lambda(i) \leq x_i \leq \mu(i)$
- Idea: refine lower and upper bounds on the prediction
  - $\mathbf{v}_L$  and  $\mathbf{v}_U$
- Utilities:
  - **FixAttr( $i$ ):**

```
 $\mathbf{v}_L \leftarrow (v_{L_1}, \dots, v_i, \dots, v_{L_N})$ 
 $\mathbf{v}_U \leftarrow (v_{U_1}, \dots, v_i, \dots, v_{U_N})$ 
 $(\mathcal{A}, \mathcal{B}) \leftarrow (\mathcal{A} \setminus \{i\}, \mathcal{B} \cup \{i\})$ 
return  $(\mathbf{v}_L, \mathbf{v}_U, \mathcal{A}, \mathcal{B})$ 
```

- **FreeAttr( $i$ ):**

```
 $\mathbf{v}_L \leftarrow (v_{L_1}, \dots, \lambda(i), \dots, v_{L_N})$ 
 $\mathbf{v}_U \leftarrow (v_{U_1}, \dots, \mu(i), \dots, v_{U_N})$ 
 $(\mathcal{A}, \mathcal{B}) \leftarrow (\mathcal{A} \setminus \{i\}, \mathcal{B} \cup \{i\})$ 
return  $(\mathbf{v}_L, \mathbf{v}_U, \mathcal{A}, \mathcal{B})$ 
```

## Computing one AXp

```
1:  $\mathbf{v}_L \leftarrow (v_1, \dots, v_N)$ 
2:  $\mathbf{v}_U \leftarrow (v_1, \dots, v_N)$ 
3:  $(\mathcal{C}, \mathcal{D}, \mathcal{P}) \leftarrow (\mathcal{F}, \emptyset, \emptyset)$ 
4: for all  $i \in \mathcal{S}$  do
5:    $(\mathbf{v}_L, \mathbf{v}_U, \mathcal{C}, \mathcal{D}) \leftarrow \text{FreeAttr}(i, \mathbf{v}, \mathbf{v}_L, \mathbf{v}_U, \mathcal{C}, \mathcal{D})$ 
6: for all  $i \in \mathcal{F} \setminus \mathcal{S}$  do
7:    $(\mathbf{v}_L, \mathbf{v}_U, \mathcal{C}, \mathcal{D}) \leftarrow \text{FreeAttr}(i, \mathbf{v}, \mathbf{v}_L, \mathbf{v}_U, \mathcal{C}, \mathcal{D})$ 
8:   if  $\kappa(\mathbf{v}_L) \neq \kappa(\mathbf{v}_U)$  then
9:      $(\mathbf{v}_L, \mathbf{v}_U, \mathcal{D}, \mathcal{P}) \leftarrow \text{FixAttr}(i, \mathbf{v}, \mathbf{v}_L, \mathbf{v}_U, \mathcal{D}, \mathcal{P})$ 
10: return  $\mathcal{P}$ 
```

- ▷ Ensures:  $\kappa(\mathbf{v}_L) = \kappa(\mathbf{v}_U)$
- ▷  $\mathcal{S}$ : Some possible seed
- ▷ Require:  $\kappa(\mathbf{v}_L) = \kappa(\mathbf{v}_U)$ , given  $\mathcal{S}$
- ▷ Loop inv.:  $\kappa(\mathbf{v}_L) = \kappa(\mathbf{v}_U)$
- ▷ If invariant broken, fix it

- Obs:  $\mathcal{S} = \emptyset$  for computing a single AXp/CXp

## Computing one AXp – example

- $\lambda(i) = 0$  and  $\mu(i) = 10$
- $\mathbf{v} = (10, 10, 5, 0)$ , with  $\kappa(\mathbf{v}) = A$
- **Q:** find one AXp (CXp is similar)

Feat.	Initial values		Changed values		Predictions		Dec.	Resulting values	
	$\mathbf{v}_L$	$\mathbf{v}_U$	$\mathbf{v}_L$	$\mathbf{v}_U$	$\kappa(\mathbf{v}_L)$	$\kappa(\mathbf{v}_U)$		$\mathbf{v}_L$	$\mathbf{v}_U$
1	(10,10,5,0)	(10,10,5,0)	(0,10,5,0)	(10,10,5,0)	C	A	✓	(10,10,5,0)	(10,10,5,0)
2	(10,10,5,0)	(10,10,5,0)	(10,0,5,0)	(10,10,5,0)	E	A	✓	(10,10,5,0)	(10,10,5,0)
3	(10,10,5,0)	(10,10,5,0)	(10,10,0,0)	(10,10,10,0)	A	A	✗	(10,10,0,0)	(10,10,10,0)
4	(10,10,0,0)	(10,10,10,0)	(10,10,0,0)	(10,10,10,10)	A	A	✗	(10,10,0,0)	(10,10,10,10)

# Outline – Unit #03

Explanations for Decision Trees

XAI Queries for DTs

Myth #01: Intrinsic Interpretability

Detour: From Decision Trees to Explained Decision Sets

Explanations for Decision Graphs

Explanations for Monotonic Classifiers

Review examples

## Recap computation of (W)AXps/(W)CXps

$$\begin{aligned} \text{WAXp}(\mathcal{X}) &:= \forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x})) \\ \text{WCXp}(\mathcal{Y}) &:= \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \end{aligned}$$

## Recap computation of $(W)AXps/(W)CXps$

$$\begin{aligned} WAXp(\mathcal{X}) &:= \forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\sigma(\mathbf{x})) \\ WCXp(\mathcal{Y}) &:= \exists (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \notin \mathcal{Y}} (x_j = v_j) \wedge (\neg \sigma(\mathbf{x})) \end{aligned}$$

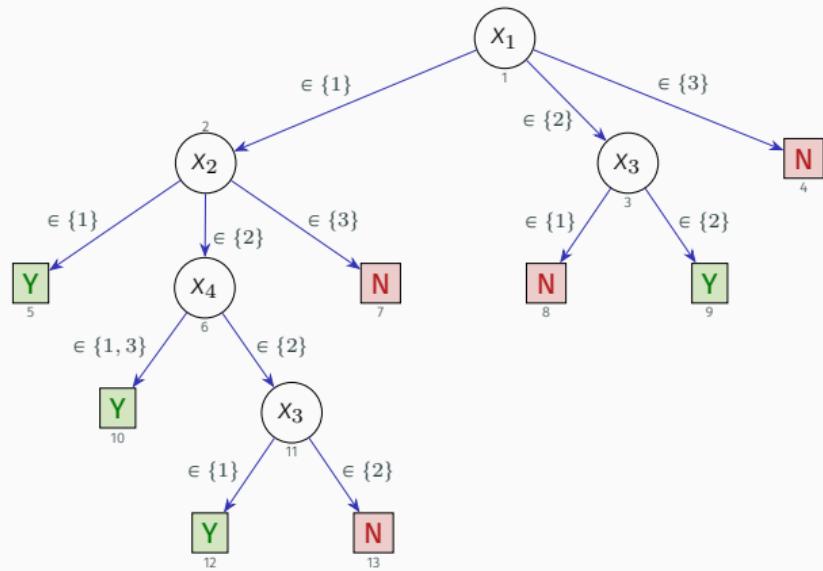
**Input:** Predicate  $\mathbb{P}$ , parameterized by  $\mathcal{T}, \mathcal{M}$

**Output:** One XP  $\mathcal{S}$

```
1: procedure oneXP( $\mathbb{P}$ )
2:    $\mathcal{S} \leftarrow \mathcal{F}$                                  $\triangleright$  Initialization:  $\mathbb{P}(\mathcal{S})$  holds
3:   for  $i \in \mathcal{F}$  do
4:     if  $\mathbb{P}(\mathcal{S} \setminus \{i\})$  then                 $\triangleright$  Loop invariant:  $\mathbb{P}(\mathcal{S})$  holds
5:        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
6:   return  $\mathcal{S}$                                  $\triangleright$  Update  $\mathcal{S}$  only if  $\mathbb{P}(\mathcal{S} \setminus \{i\})$  holds
                                                  $\triangleright$  Returned set  $\mathcal{S}$ :  $\mathbb{P}(\mathcal{S})$  holds
```

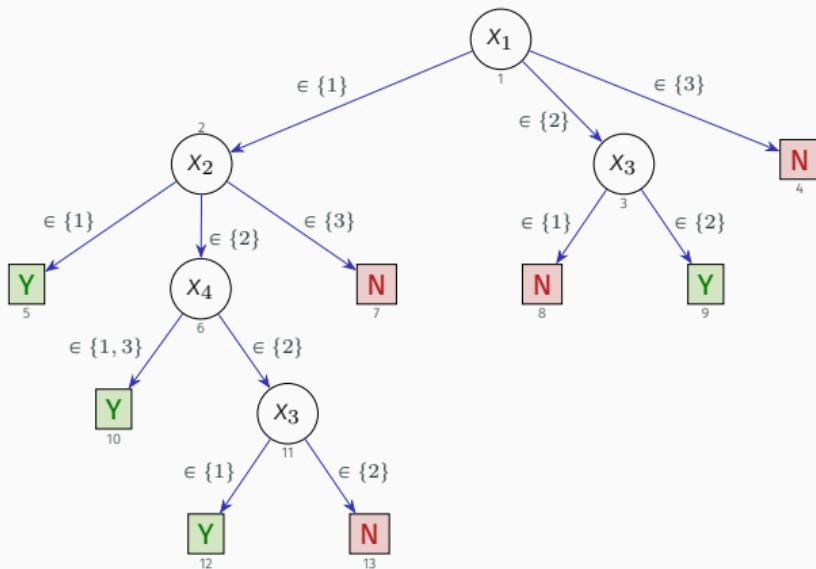
## Review exercise – one AXp for example DT

- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$



## Review exercise – one AXp for example DT

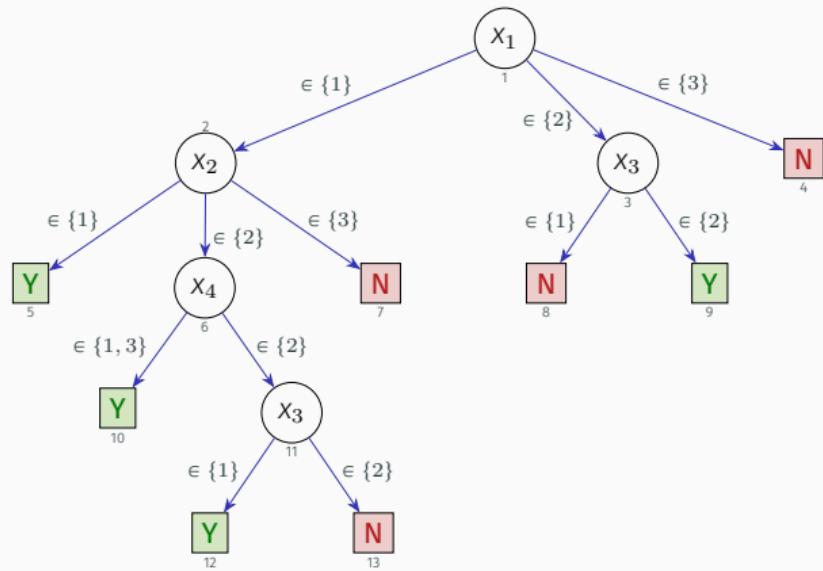
- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$



- Finding on AXp:

## Review exercise – one AXp for example DT

- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$

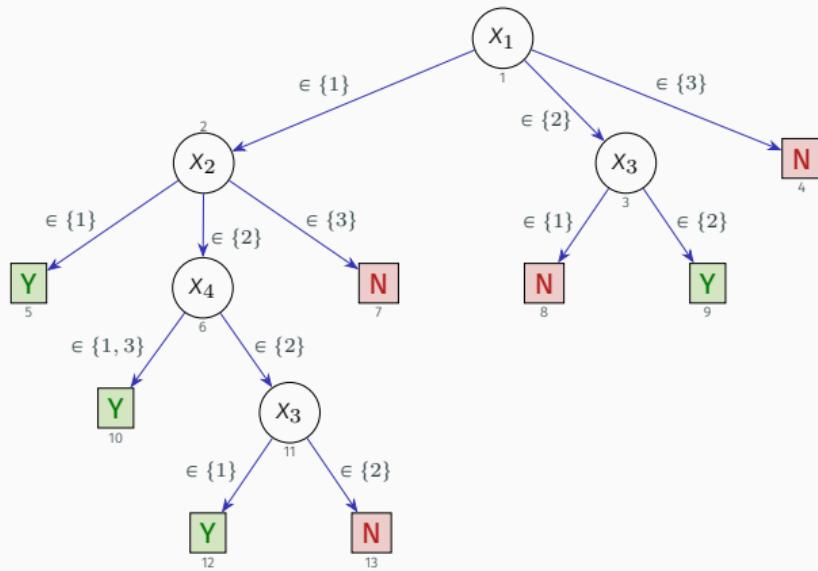


- Finding on AXp:

- 1st path inconsistent:  $H_1 = \{3\}$

## Review exercise – one AXp for example DT

- Instance:  $(v, c) = ((1, 2, 1, 2), \text{Y})$

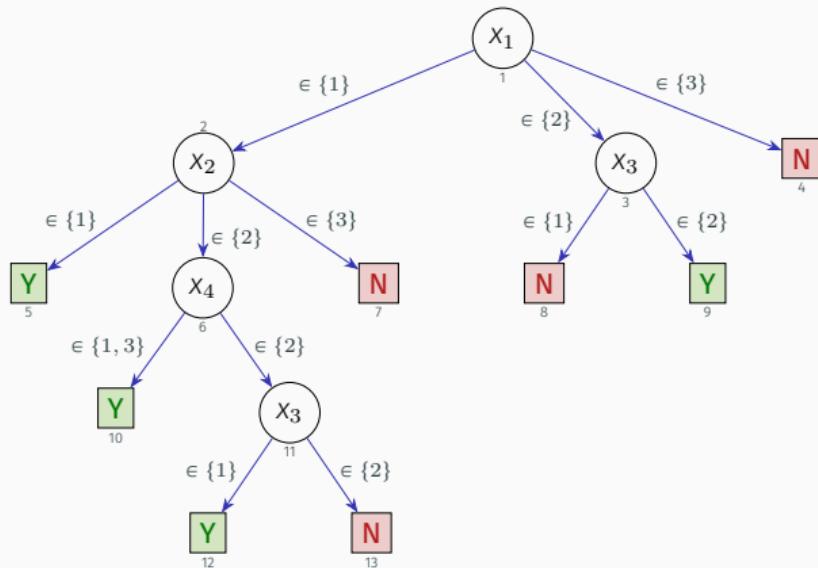


- Finding on AXp:

- 1st path inconsistent:  $H_1 = \{3\}$
- 2nd path inconsistent:  $H_2 = \{2\}$

## Review exercise – one AXp for example DT

- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$

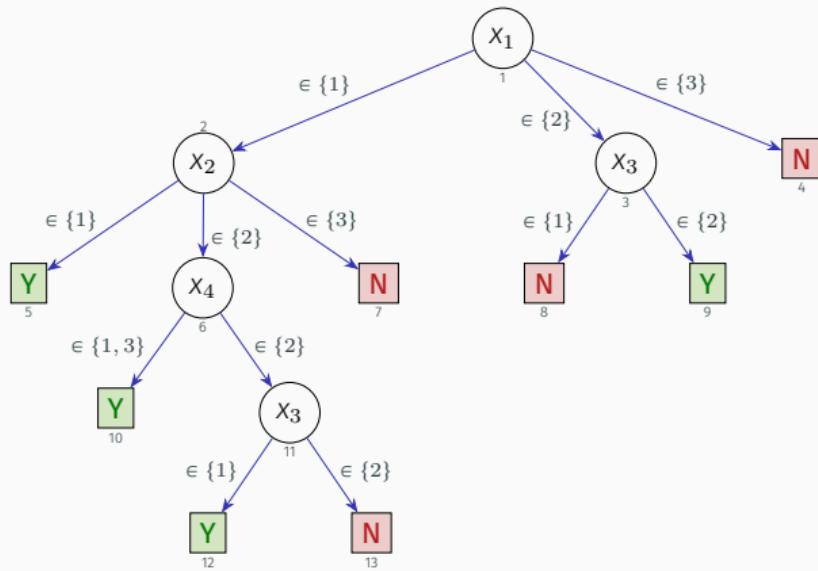


- Finding on AXp:

- 1st path inconsistent:  $H_1 = \{3\}$
- 2nd path inconsistent:  $H_2 = \{2\}$
- 3rd path inconsistent:  $H_3 = \{1\}$

## Review exercise – one AXp for example DT

- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$

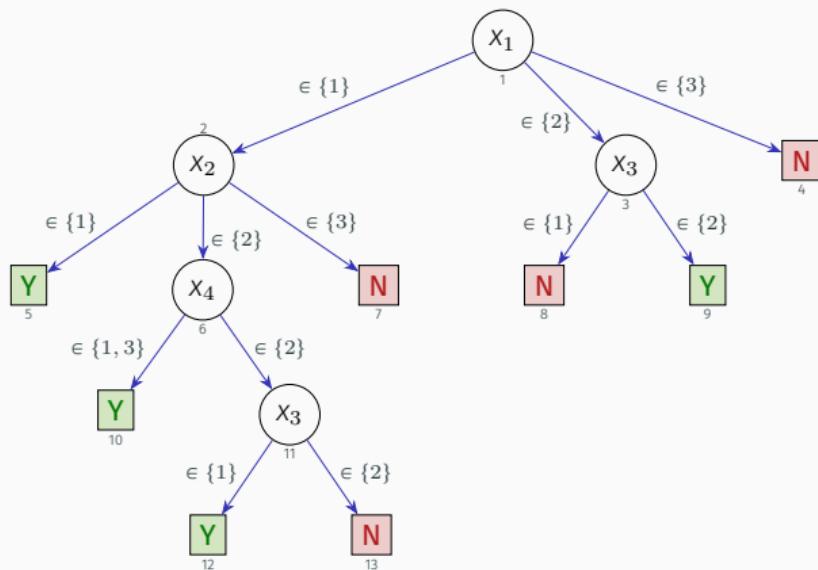


- Finding on AXp:

- 1st path inconsistent:  $H_1 = \{3\}$
- 2nd path inconsistent:  $H_2 = \{2\}$
- 3rd path inconsistent:  $H_3 = \{1\}$
- 4th path inconsistent:  $H_4 = \{1\}$

## Review exercise – one AXp for example DT

- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$



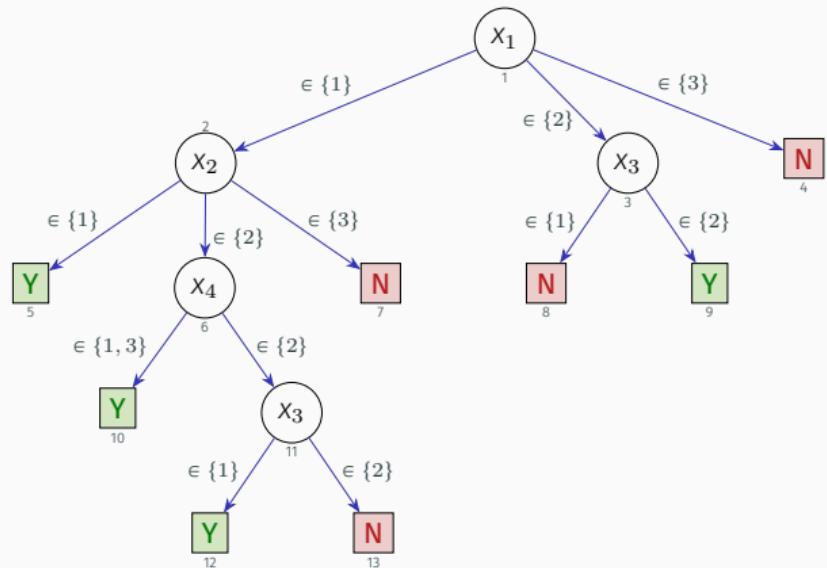
- Finding on AXp:

- 1st path inconsistent:  $H_1 = \{3\}$
- 2nd path inconsistent:  $H_2 = \{2\}$
- 3rd path inconsistent:  $H_3 = \{1\}$
- 4th path inconsistent:  $H_4 = \{1\}$

- AXp is MHS of  $H_j$  sets:  $\{1, 2, 3\}$

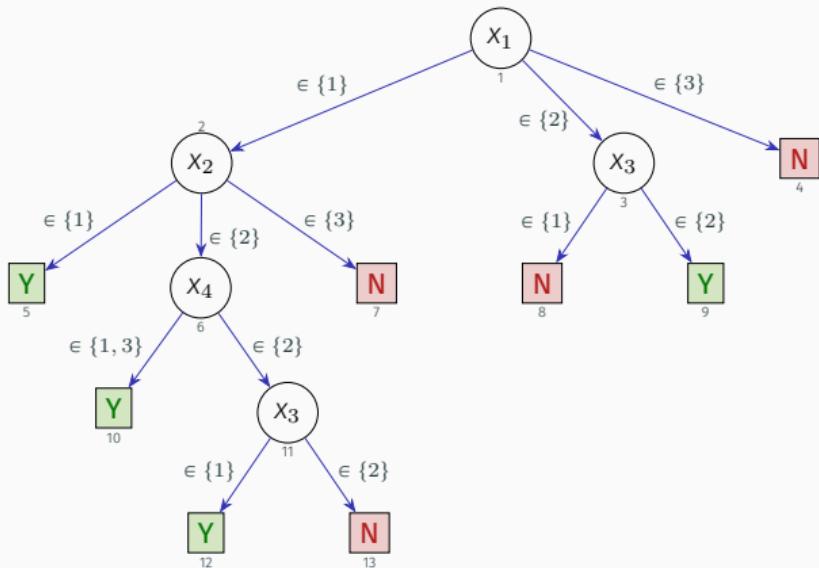
## Review exercise – all CXps & AXps for example DT

- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$



## Review exercise – all CXps & AXps for example DT

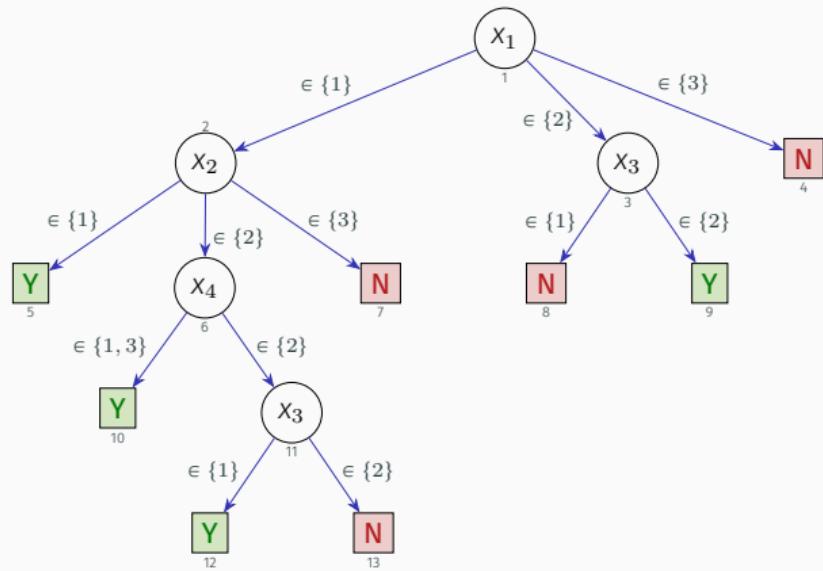
- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$



- Finding CXps:

## Review exercise – all CXps & AXps for example DT

- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$

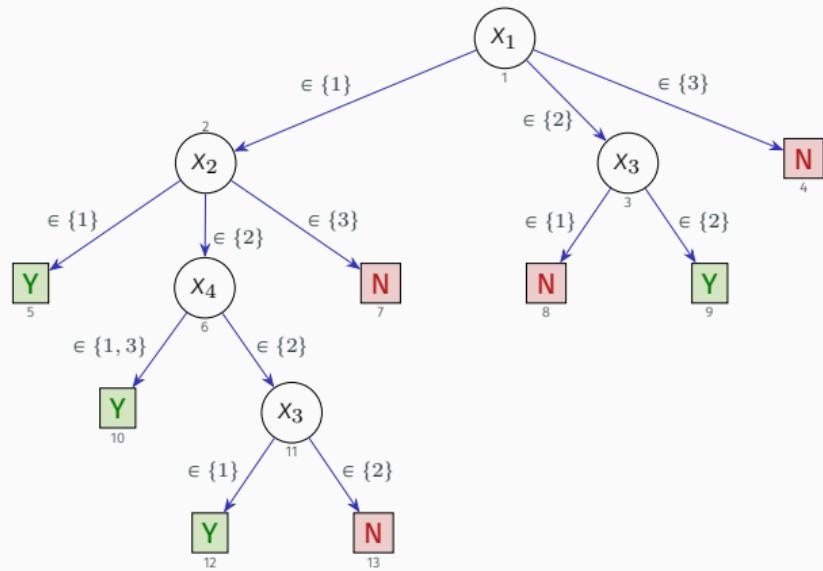


- Finding CXps:

- 1st path:  $l_1 = \{3\}$

## Review exercise – all CXps & AXps for example DT

- Instance:  $(v, c) = ((1, 2, 1, 2), Y)$

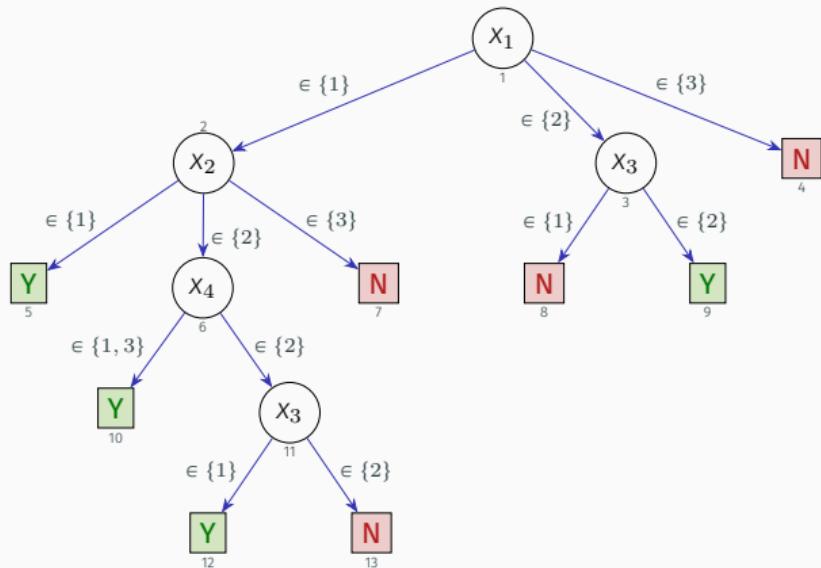


### Finding CXps:

- 1st path:  $I_1 = \{3\}$
- 2nd path:  $I_2 = \{2\}$

## Review exercise – all CXps & AXps for example DT

- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$

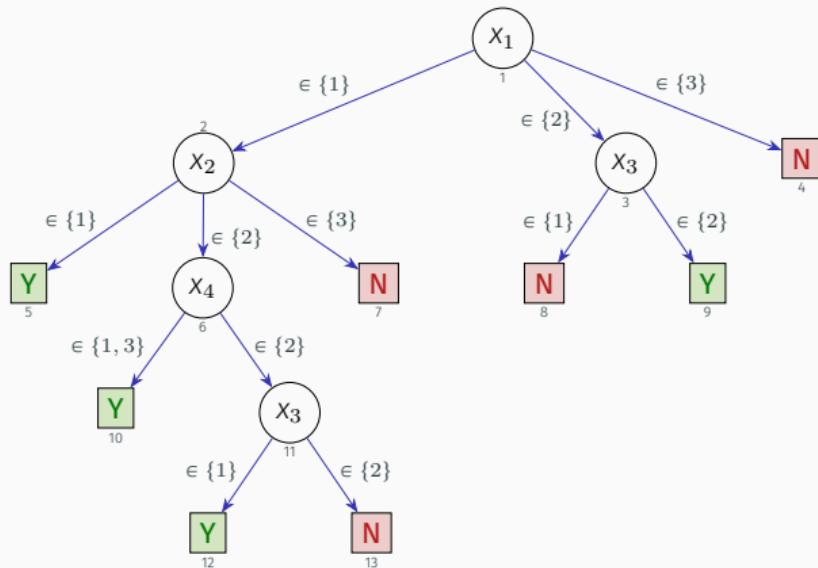


### Finding CXps:

- 1st path:  $I_1 = \{3\}$
- 2nd path:  $I_2 = \{2\}$
- 3rd path:  $I_3 = \{1\}$

## Review exercise – all CXps & AXps for example DT

- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$

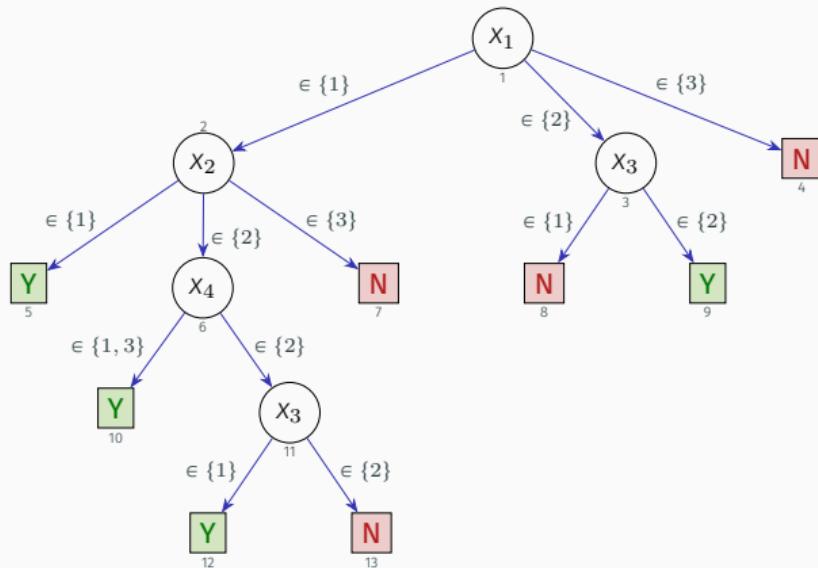


### Finding CXps:

- 1st path:  $I_1 = \{3\}$
- 2nd path:  $I_2 = \{2\}$
- 3rd path:  $I_3 = \{1\}$
- 4th path:  $I_4 = \{1\}$

## Review exercise – all CXps & AXps for example DT

- Instance:  $(v, c) = ((1, 2, 1, 2), \text{Y})$

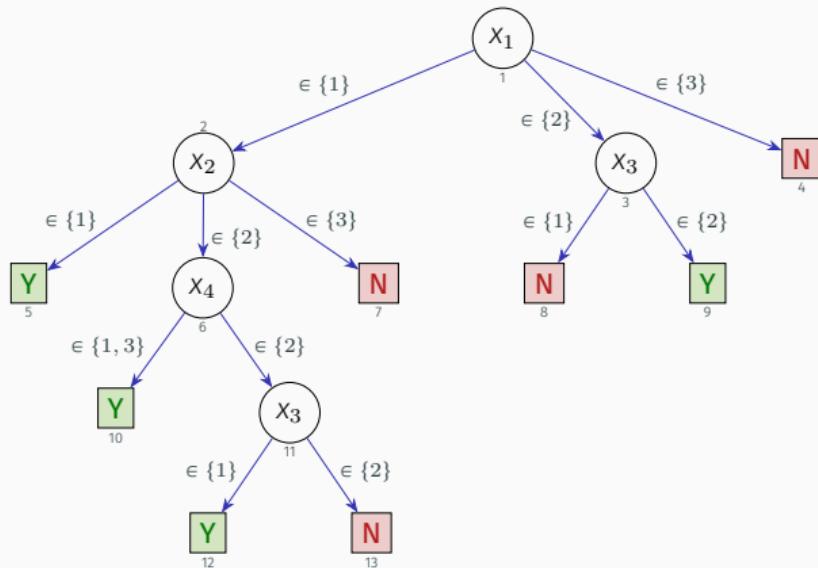


### Finding CXps:

- 1st path:  $I_1 = \{3\}$
- 2nd path:  $I_2 = \{2\}$
- 3rd path:  $I_3 = \{1\}$
- 4th path:  $I_4 = \{1\}$
- $\mathcal{L} = \{\{1\}, \{2\}, \{3\}\} = \mathbb{C}$

# Review exercise – all CXps & AXps for example DT

- Instance:  $(\mathbf{v}, c) = ((1, 2, 1, 2), \text{Y})$



- Finding CXps:

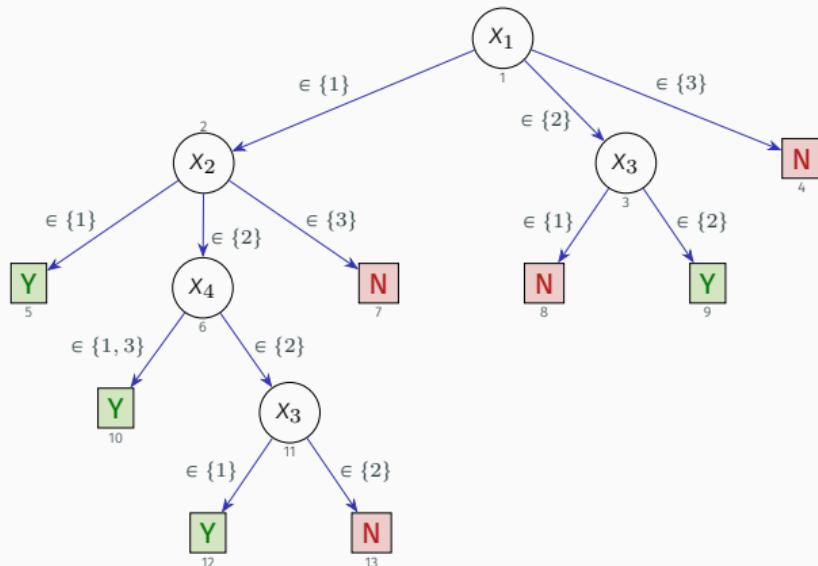
- 1st path:  $I_1 = \{3\}$
- 2nd path:  $I_2 = \{2\}$
- 3rd path:  $I_3 = \{1\}$
- 4th path:  $I_4 = \{1\}$
- $\mathcal{L} = \{\{1\}, \{2\}, \{3\}\} = \mathbb{C}$

- Finding AXps:

(i.e. all MHSes of sets in  $\mathbb{C}$ )

# Review exercise – all CXps & AXps for example DT

- Instance:  $(v, c) = ((1, 2, 1, 2), \text{Y})$



- Finding CXps:

- 1st path:  $I_1 = \{3\}$
- 2nd path:  $I_2 = \{2\}$
- 3rd path:  $I_3 = \{1\}$
- 4th path:  $I_4 = \{1\}$
- $\mathcal{L} = \{\{1\}, \{2\}, \{3\}\} = \mathbb{C}$

- Finding AXps:

(i.e. all MHSes of sets in  $\mathbb{C}$ )

- $\mathbb{A} = \{\{1, 2, 3\}\}$

## Another review exercise – one AXp for example DL

- DL:

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_2 :$	ELSE IF	$(x_1 \wedge x_5)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(x_2 \wedge x_4)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_1 \wedge x_7)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(\neg x_4 \wedge \neg x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_7 :$	ELSE IF	$(\neg x_2 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 0$

## Another review exercise – one AXp for example DL

- DL:

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_2 :$	ELSE IF	$(x_1 \wedge x_5)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(x_2 \wedge x_4)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_1 \wedge x_7)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(\neg x_4 \wedge \neg x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_7 :$	ELSE IF	$(\neg x_2 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 0$

- Instance:  $\mathbf{v} = (0, 1, 0, 1, 0, 1, 0)$ 
  - The prediction is 1, due to  $R_3$

## Another review exercise – one AXp for example DL

- DL:

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_2 :$	ELSE IF	$(x_1 \wedge x_5)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(x_2 \wedge x_4)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_1 \wedge x_7)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(\neg x_4 \wedge \neg x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_7 :$	ELSE IF	$(\neg x_2 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 0$

- Instance:  $\mathbf{v} = (0, 1, 0, 1, 0, 1, 0)$ 
  - The prediction is 1, due to  $R_3$
- AXp:

## Another review exercise – one AXp for example DL

- DL:

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_2 :$	ELSE IF	$(x_1 \wedge x_5)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(x_2 \wedge x_4)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_1 \wedge x_7)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(\neg x_4 \wedge \neg x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_7 :$	ELSE IF	$(\neg x_2 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 0$

- Instance:  $\mathbf{v} = (0, 1, 0, 1, 0, 1, 0)$ 
  - The prediction is 1, due to  $R_3$
  - AXp: {1, 2}

## Another review exercise – one AXp for example DL

- DL:

$R_1 :$	IF	$(x_1 \wedge x_3)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_2 :$	ELSE IF	$(x_1 \wedge x_5)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_3 :$	ELSE IF	$(x_2 \wedge x_4)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_4 :$	ELSE IF	$(x_1 \wedge x_7)$	THEN	$\kappa(\mathbf{x}) = 0$
$R_5 :$	ELSE IF	$(\neg x_4 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_6 :$	ELSE IF	$(\neg x_4 \wedge \neg x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_7 :$	ELSE IF	$(\neg x_2 \wedge x_6)$	THEN	$\kappa(\mathbf{x}) = 1$
$R_{DEF} :$	ELSE			$\kappa(\mathbf{x}) = 0$

- Instance:  $\mathbf{v} = (0, 1, 0, 1, 0, 1, 0)$ 
  - The prediction is 1, due to  $R_3$
- AXp: {1, 2}
- Quiz: write down the constraints and confirm AXp with SAT solver

Questions?

# Lecture 03

## Recapitulate second lecture

- Rigorous definitions of abductive and contrastive explanations

## Recapitulate second lecture

- Rigorous definitions of abductive and contrastive explanations
- Example algorithm for finding one AXp/CXp

## Recapitulate second lecture

- Rigorous definitions of abductive and contrastive explanations
- Example algorithm for finding one AXp/CXp
- Explanations for DTs

## Recapitulate second lecture

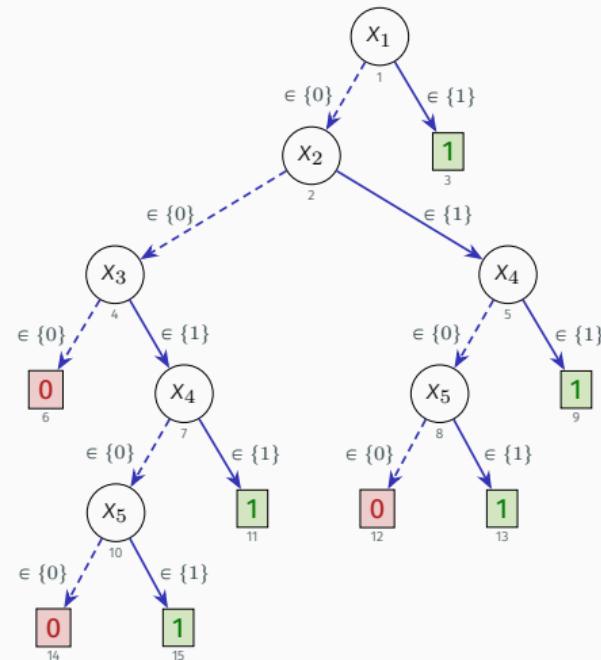
- Rigorous definitions of abductive and contrastive explanations
- Example algorithm for finding one AXp/CXp
- Explanations for DTs
- Explanations for XpGs

## Recapitulate second lecture

- Rigorous definitions of abductive and contrastive explanations
- Example algorithm for finding one AXp/CXp
- Explanations for DTs
- Explanations for XpGs
- Explanations for monotonic classifiers

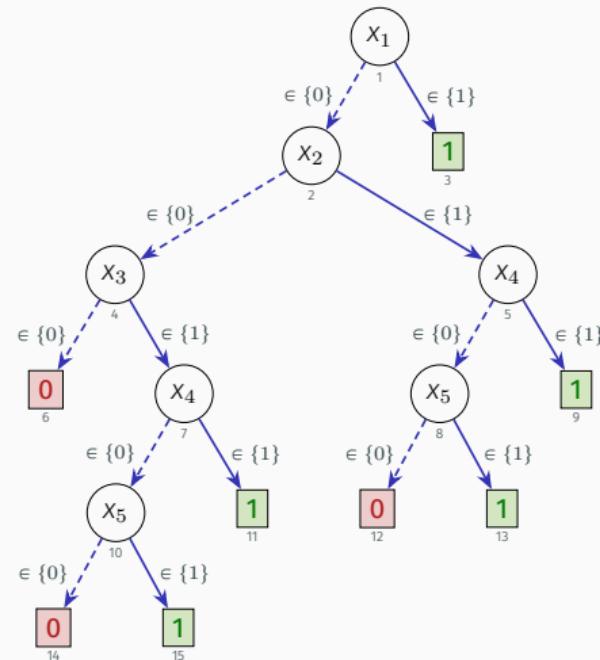
## Recap AXps/CXps: DT example

- Instance:  $((0, 0, 1, 0, 0), 0)$



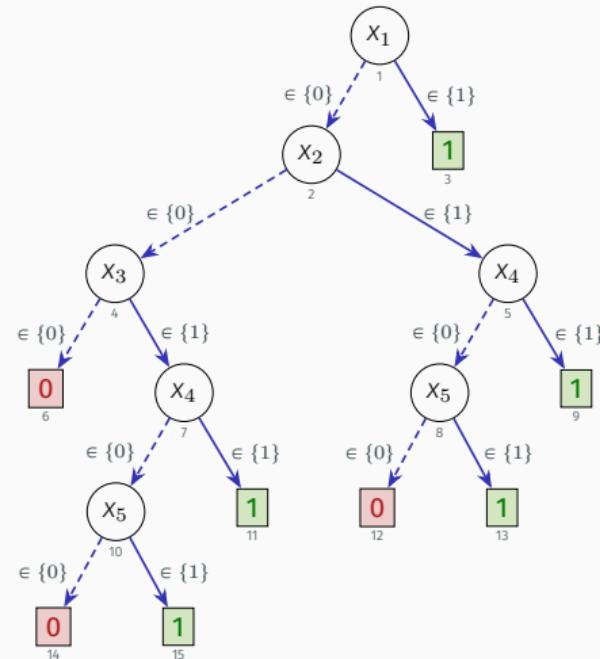
## Recap AXps/CXps: DT example

- Instance:  $((0, 0, 1, 0, 0), 0)$
- One AXp:  $\{1, 4, 5\}$



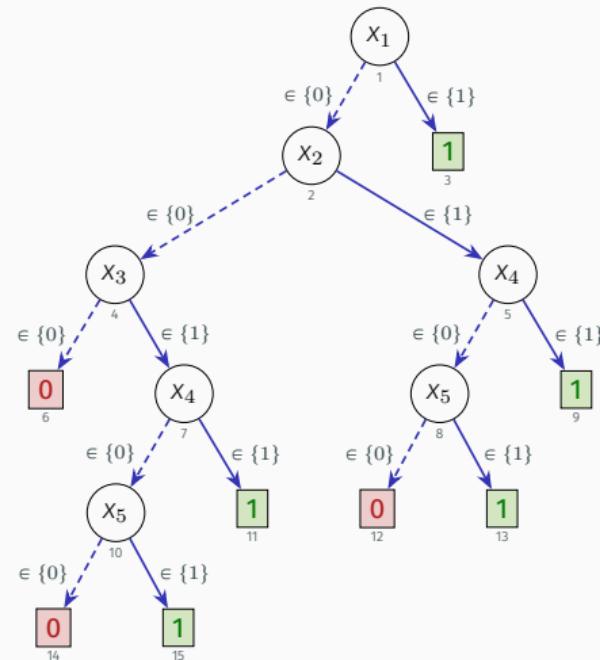
## Recap AXps/CXps: DT example

- Instance:  $((0, 0, 1, 0, 0), 0)$
- One AXp:  $\{1, 4, 5\}$
- All CXps:



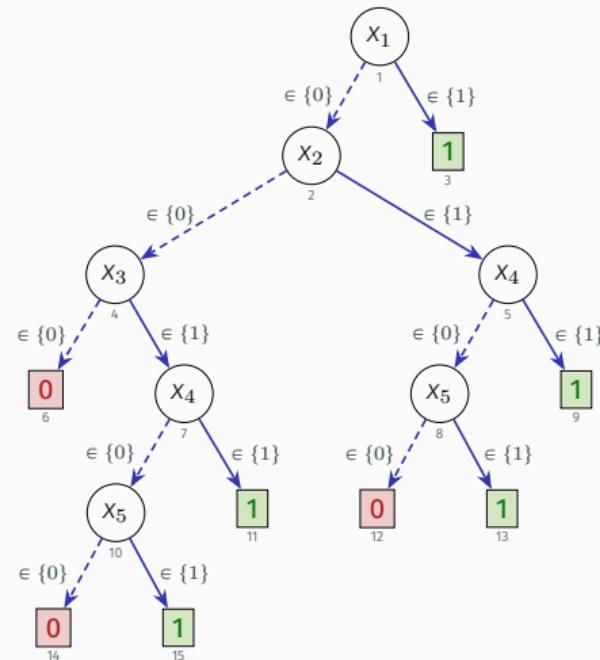
## Recap AXps/CXps: DT example

- Instance:  $((0, 0, 1, 0, 0), 0)$
- One AXp:  $\{1, 4, 5\}$
- All CXps:
  - $I_1: \{5\}$



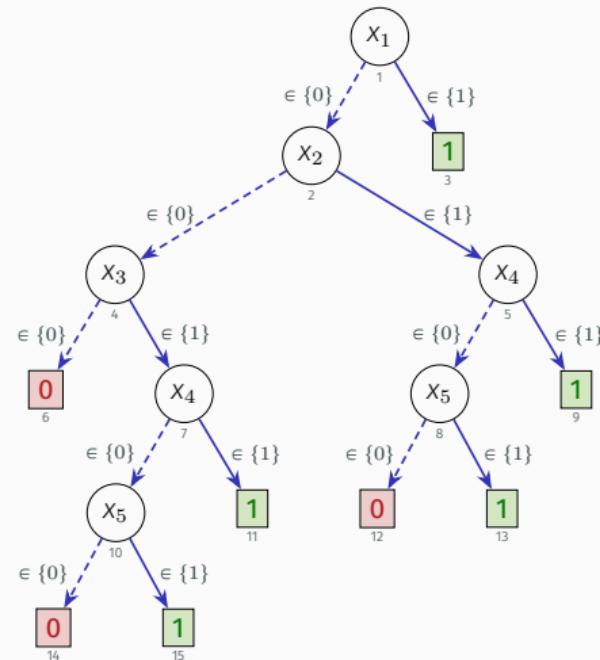
## Recap AXps/CXps: DT example

- Instance:  $((0, 0, 1, 0, 0), 0)$
- One AXp:  $\{1, 4, 5\}$
- All CXps:
  - $I_1: \{5\}$
  - $I_2: \{4\}$



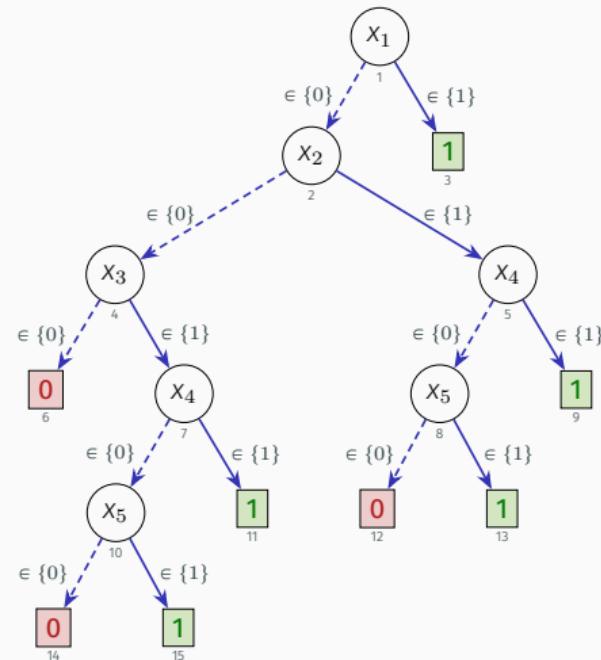
## Recap AXps/CXps: DT example

- Instance:  $((0, 0, 1, 0, 0), 0)$
- One AXp:  $\{1, 4, 5\}$
- All CXps:
  - $l_1: \{5\}$
  - $l_2: \{4\}$
  - $l_3: \{2, 5\}$



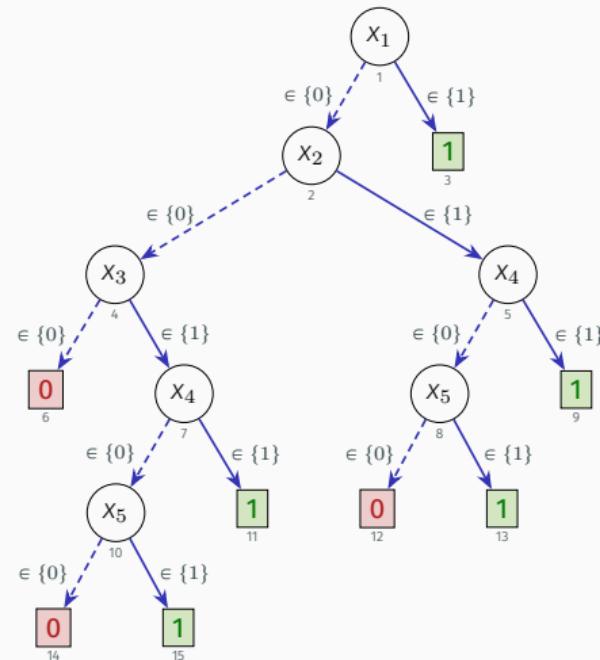
## Recap AXps/CXps: DT example

- Instance:  $((0, 0, 1, 0, 0), 0)$
- One AXp:  $\{1, 4, 5\}$
- All CXps:
  - $l_1: \{5\}$
  - $l_2: \{4\}$
  - $l_3: \{2, 5\}$
  - $l_4: \{2, 4\}$



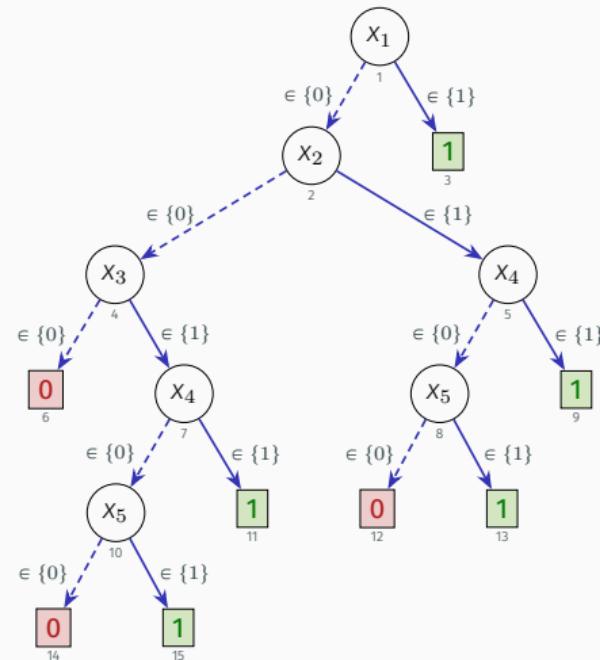
## Recap AXps/CXps: DT example

- Instance:  $((0, 0, 1, 0, 0), 0)$
- One AXp:  $\{1, 4, 5\}$
- All CXps:
  - $l_1: \{5\}$
  - $l_2: \{4\}$
  - $l_3: \{2, 5\}$
  - $l_4: \{2, 4\}$
  - $l_5: \{1\}$



## Recap AXps/CXps: DT example

- Instance:  $((0, 0, 1, 0, 0), 0)$
- One AXp:  $\{1, 4, 5\}$
- All CXps:
  - $l_1: \{5\}$
  - $l_2: \{4\}$
  - $l_3: \{2, 5\}$
  - $l_4: \{2, 4\}$
  - $l_5: \{1\}$
  - $\mathcal{L} = \{\{1\}, \{4\}, \{5\}\}$



## Recap AXps/CXps: DL example

- R<sub>1</sub>: IF  $(x_1 = 1)$  THEN 0  
 R<sub>2</sub>: ELSE IF  $(x_2 = 1)$  THEN 1  
 R<sub>3</sub>: ELSE IF  $(x_4 = 1)$  THEN 0  
 R<sub>DEF</sub>: ELSE THEN 1

Entry	$x_1$	$x_2$	$x_3$	$x_4$	Rule	$\kappa_1(\mathbf{x})$
00	0	0	0	0	R <sub>DEF</sub>	1
01	0	0	0	1	R <sub>3</sub>	0
02	0	0	0	2	R <sub>DEF</sub>	1
03	0	0	1	0	R <sub>DEF</sub>	1
04	0	0	1	1	R <sub>3</sub>	0
05	0	0	1	2	R <sub>DEF</sub>	1
06	0	1	0	0	R <sub>2</sub>	1
07	0	1	0	1	R <sub>2</sub>	1
08	0	1	0	2	R <sub>2</sub>	1
09	0	1	1	0	R <sub>2</sub>	1
10	0	1	1	1	R <sub>2</sub>	1
11	0	1	1	2	R <sub>2</sub>	1
12	1	0	0	0	R <sub>1</sub>	0
13	1	0	0	1	R <sub>1</sub>	0
14	1	0	0	2	R <sub>1</sub>	0
15	1	0	1	0	R <sub>1</sub>	0
16	1	0	1	1	R <sub>1</sub>	0
17	1	0	1	2	R <sub>1</sub>	0
18	1	1	0	0	R <sub>1</sub>	0
19	1	1	0	1	R <sub>1</sub>	0
20	1	1	0	2	R <sub>1</sub>	0
21	1	1	1	0	R <sub>1</sub>	0
22	1	1	1	1	R <sub>1</sub>	0
23	1	1	1	2	R <sub>1</sub>	0

## Recap AXps/CXps: DL example

- $R_1:$  IF  $(x_1 = 1)$  THEN 0
- $R_2:$  ELSE IF  $(x_2 = 1)$  THEN 1
- $R_3:$  ELSE IF  $(x_4 = 1)$  THEN 0
- $R_{DEF}:$  ELSE THEN 1

- Instance:  $(\mathbf{v}, c) = ((0, 0, 1, 2), 1)$
- AXp's:  $\{1, 4\}$  (prediction unchanged)
- CXp's:
  - $\{1\}$ , by flipping the value of feature 1
  - $\{4\}$ , by flipping the value of feature 4
  - But also,  $\{\{1\}, \{4\}\}$  by MHS duality

Entry	$x_1$	$x_2$	$x_3$	$x_4$	Rule	$\kappa_1(\mathbf{x})$
00	0	0	0	0	$R_{DEF}$	1
01	0	0	0	1	$R_3$	0
02	0	0	0	2	$R_{DEF}$	1
03	0	0	1	0	$R_{DEF}$	1
04	0	0	1	1	$R_3$	0
05	0	0	1	2	$R_{DEF}$	1
06	0	1	0	0	$R_2$	1
07	0	1	0	1	$R_2$	1
08	0	1	0	2	$R_2$	1
09	0	1	1	0	$R_2$	1
10	0	1	1	1	$R_2$	1
11	0	1	1	2	$R_2$	1
12	1	0	0	0	$R_1$	0
13	1	0	0	1	$R_1$	0
14	1	0	0	2	$R_1$	0
15	1	0	1	0	$R_1$	0
16	1	0	1	1	$R_1$	0
17	1	0	1	2	$R_1$	0
18	1	1	0	0	$R_1$	0
19	1	1	0	1	$R_1$	0
20	1	1	0	2	$R_1$	0
21	1	1	1	0	$R_1$	0
22	1	1	1	1	$R_1$	0
23	1	1	1	2	$R_1$	0

# Plan for this course

- Lecture 01 – unit(s):
  - #01: Foundations
- Lecture 02 – unit(s):
  - #02: Principles of symbolic XAI – **feature selection**
  - #03: Tractability in symbolic XAI (& myth of interpretability)
- Lecture 03 – unit(s):
  - #04: Intractability in symbolic XAI (& myth of model-agnostic XAI)
  - #05: Explainability queries
- Lecture 04 – unit(s):
  - #06: Recent, emerging & advanced topics
- Lecture 05 – unit(s):
  - #07: Principles of symbolic XAI – **feature attribution** (& myth of Shapley values in XAI)
  - #08: Corrected feature attribution – nuSHAP
  - #09: Conclusions & research directions



## Some necessary comments...

- Std question: Can we apply symbolic XAI to this highly complex ML model XYZ?

## Some necessary comments...

- Std question: Can we apply symbolic XAI to this highly complex ML model **XYZ**?
  - Most likely answer: **No!**

## Some necessary comments...

- Std question: Can we apply symbolic XAI to this highly complex ML model XYZ?
  - Most likely answer: No! But ...

## Some necessary comments...

- Std question: Can we apply symbolic XAI to this highly complex ML model XYZ?
  - Most likely answer: No! But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?

## Some necessary comments...

- Std question: Can we apply symbolic XAI to this highly complex ML model XYZ?
  - Most likely answer: No! But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?
  - fly with a airliner whose planes crash in about 1% of its flights?

## Some necessary comments...

- Std question: Can we apply symbolic XAI to this highly complex ML model XYZ?
  - Most likely answer: No! But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?
  - fly with a airliner whose planes crash in about 1% of its flights?
  - undergo an optional surgery that might be life-threatening in about 5% of the cases?

## Some necessary comments...

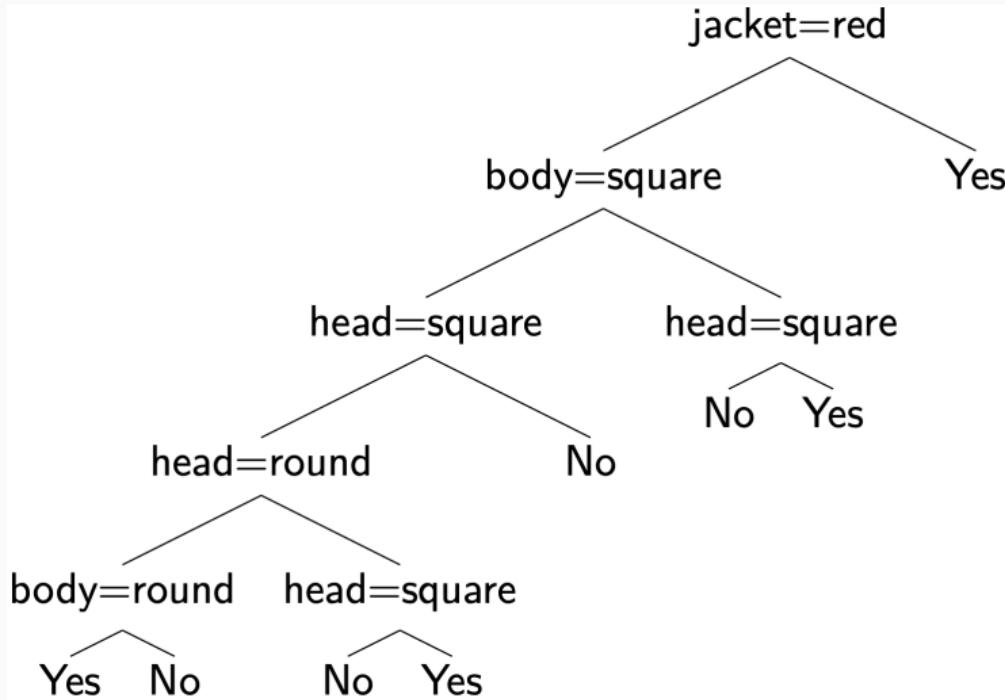
- Std question: Can we apply symbolic XAI to this highly complex ML model XYZ?
  - Most likely answer: No! But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?
  - fly with a airliner whose planes crash in about 1% of its flights?
  - undergo an optional surgery that might be life-threatening in about 5% of the cases?
- For high-risk and safety-critical domains:
  - Would you use an ML model that you cannot explain with rigor, and whose heuristic explanations can be incorrect, and so debugging/understanding with rigor is all but impossible?

## Some necessary comments...

- Std question: Can we apply symbolic XAI to this highly complex ML model **XYZ**?
  - Most likely answer: **No!** But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?
  - fly with a airliner whose planes crash in about 1% of its flights?
  - undergo an optional surgery that might be life-threatening in about 5% of the cases?
- For high-risk and safety-critical domains:
  - Would you use an ML model that you cannot explain with rigor, and whose heuristic explanations can be incorrect, and so debugging/understanding with rigor is all but impossible?
- What is the bottom line?
  - For high-risk and safety-critical domains, one **ought** to deploy models that can be explained with rigor
  - If that means using a fairly unexciting NN with up to 100K neurons, that is the cost of trust; **for anything else, one is trying his/her luck, in situations that could become catastrophic!**

## Some necessary comments...

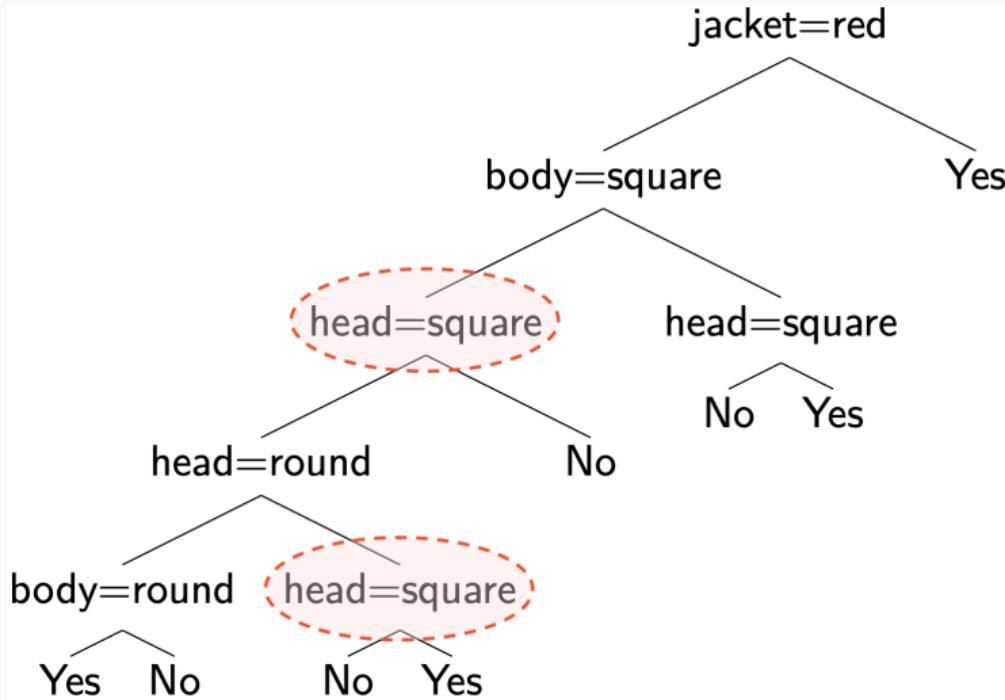
- Std question: Can we apply symbolic XAI to this highly complex ML model **XYZ**?
  - Most likely answer: **No!** But ...
- Would you...
  - ride in a car that fails to break 10% of the time, or that fails to turn 20% of the time?
  - fly with a airliner whose planes crash in about 1% of its flights?
  - undergo an optional surgery that might be life-threatening in about 5% of the cases?
- For high-risk and safety-critical domains:
  - Would you use an ML model that you cannot explain with rigor, and whose heuristic explanations can be incorrect, and so debugging/understanding with rigor is all but impossible?
- What is the bottom line?
  - For high-risk and safety-critical domains, one **ought** to deploy models that can be explained with rigor
  - If that means using a fairly unexciting NN with up to 100K neurons, that is the cost of trust; **for anything else, one is trying his/her luck, in situations that could become catastrophic!**
  - More examples next...



Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer:

Optimal Sparse Decision Trees.

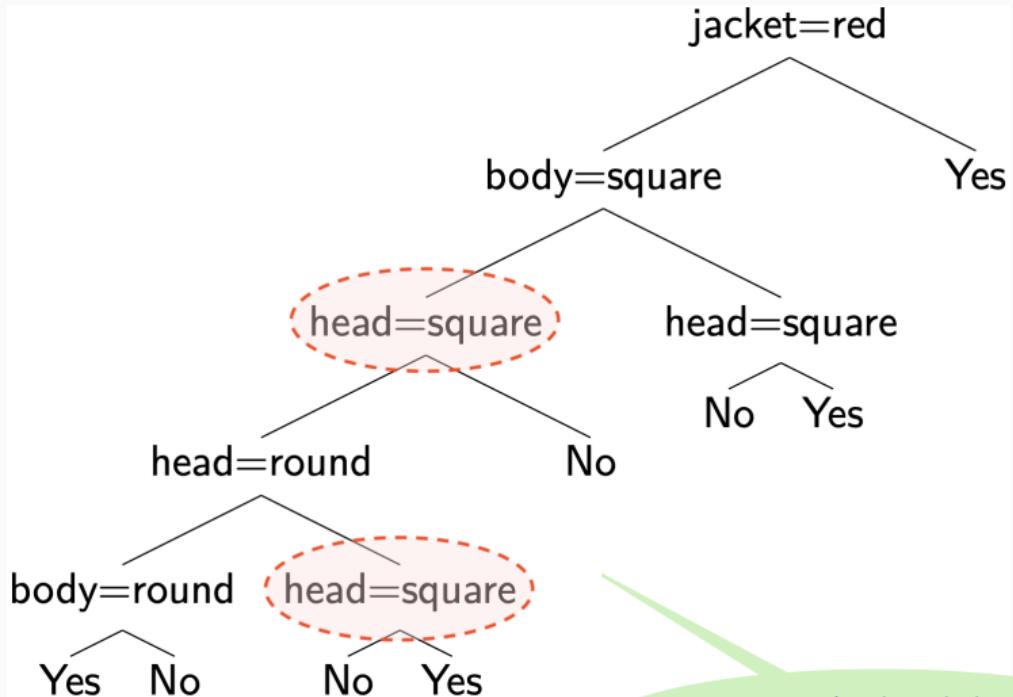
NeurIPS 2019: 7265-7273



Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer:

Optimal Sparse Decision Trees.

NeurIPS 2019: 7265-7273



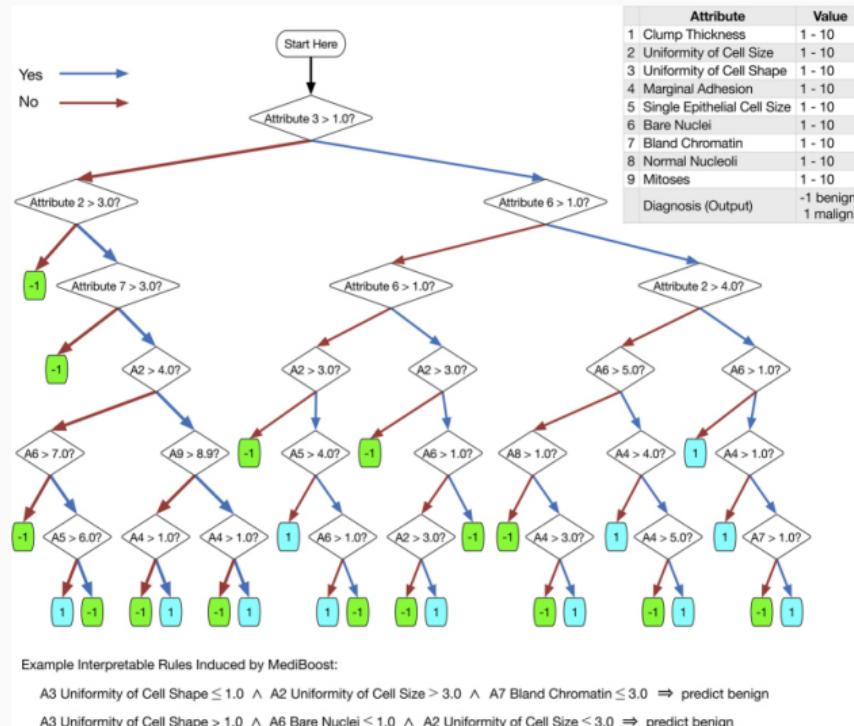
An optimal tool that  
produces non-optimal DTs...!?

Source: Xiyang Hu, Cynthia Rudin, Margo I. Seltzer:

Optimal Sparse Decision Trees.

NeurIPS 2019: 7265-7273

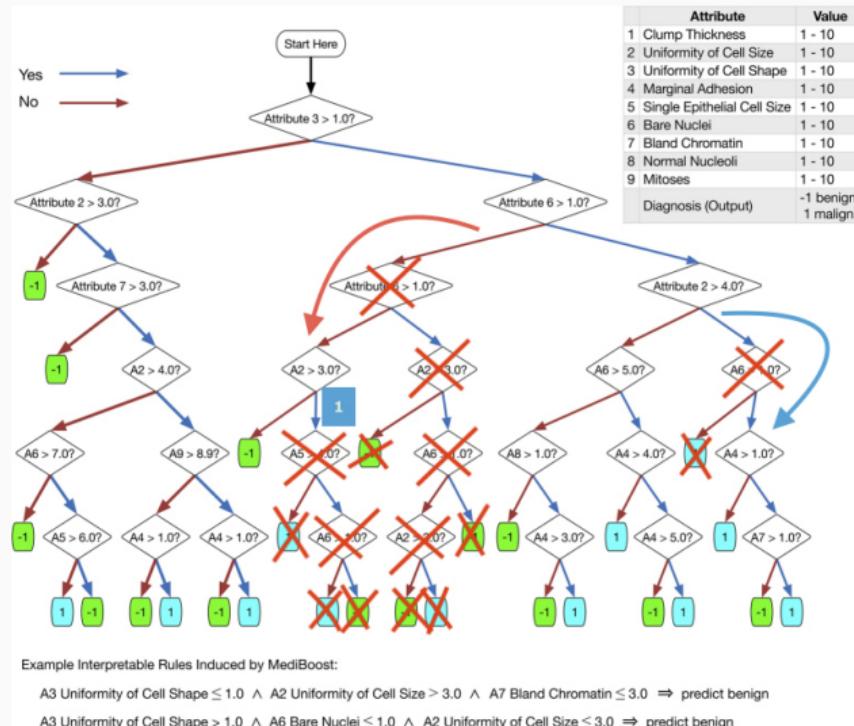
BTW, highly problematic decision trees also in precision medicine...



**Source:** G. Valdes, J.M. Luna, E. Eaton, C.B. Simone, L.H. Ungar, & T.D. Solberg.

**MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine.**  
*Scientific reports*, 6(1):1-8, 2016.

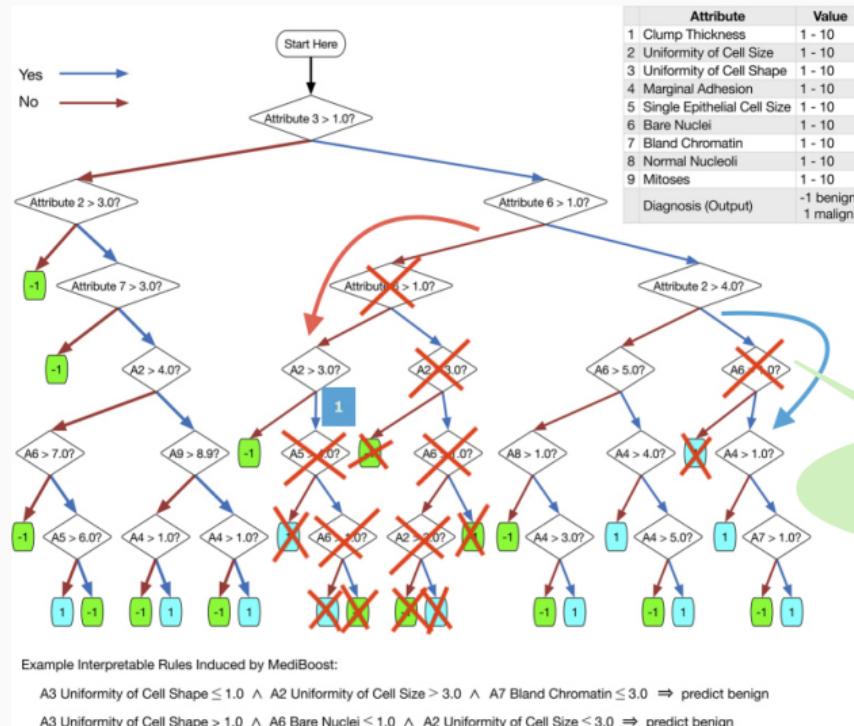
BTW, highly problematic decision trees also in precision medicine...



**Source:** G. Valdes, J.M. Luna, E. Eaton, C.B. Simone, L.H. Ungar, & T.D. Solberg.

**MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine.**  
*Scientific reports*, 6(1):1-8, 2016.

BTW, highly problematic decision trees also in precision medicine...



And massive  
path redundancy!

**Source:** G. Valdes, J.M. Luna, E. Eaton, C.B. Simone, L.H. Ungar, & T.D. Solberg.

MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine. *Scientific reports*, 6(1):1-8, 2016.

And more comments...

- Previous slides: two examples of obviously buggy DTs

## And more comments...

- Previous slides: two examples of obviously buggy DTs
- However, it is relatively simple to implement tree learners

## And more comments...

---

- Previous slides: two examples of obviously buggy DTs
- However, it is relatively simple to implement tree learners
- Can one really trust the operation of more complex ML models, even those subject to extensive testing?

## And more comments...

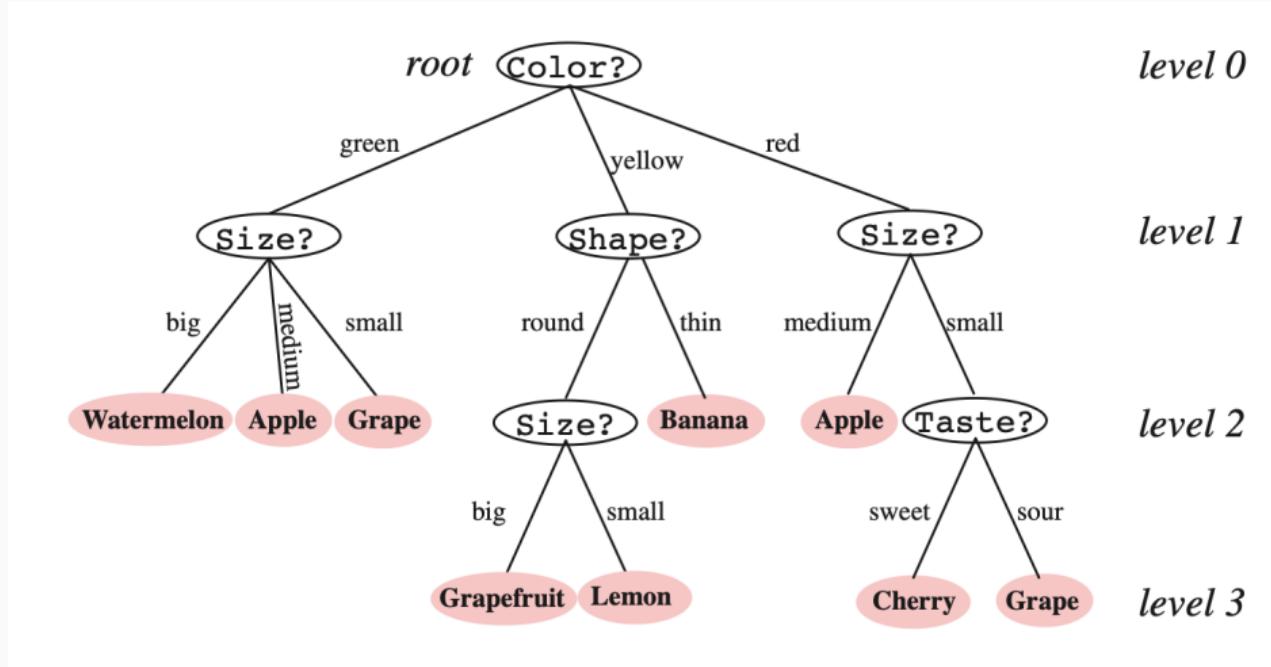
- Previous slides: two examples of obviously buggy DTs
- However, it is relatively simple to implement tree learners
- Can one really trust the operation of more complex ML models, even those subject to extensive testing?
- And how to debug complex ML models if heuristic explanations are also incorrect (more later)?

## And more comments...

- Previous slides: two examples of obviously buggy DTs
- However, it is relatively simple to implement tree learners
- Can one really trust the operation of more complex ML models, even those subject to extensive testing?
- And how to debug complex ML models if heuristic explanations are also incorrect (more later)?
- **For trustworthy AI, there exists no alternative to rigorous logic-based explanations!**

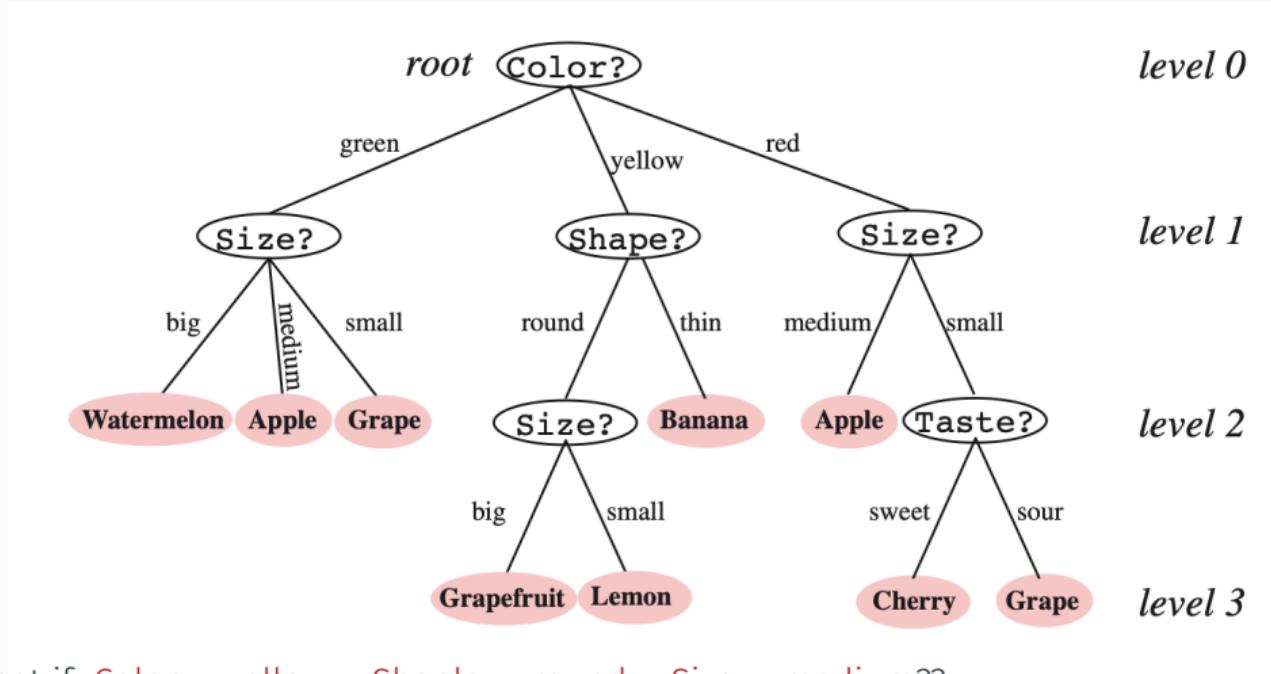
BTW, problematic DTs even in books...

[dud01]



BTW, problematic DTs even in books...

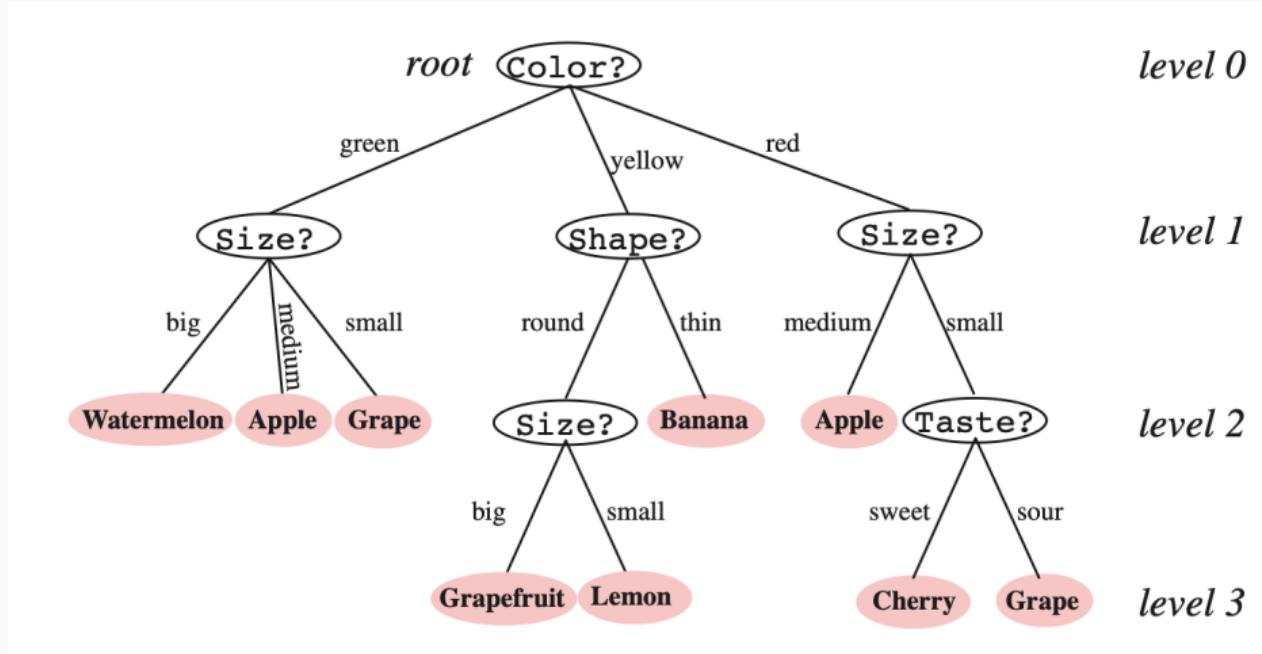
[dud01]



- What if  $\text{Color} = \text{yellow} \wedge \text{Shape} = \text{round} \wedge \text{Size} = \text{medium}??$

BTW, problematic DTs even in books...

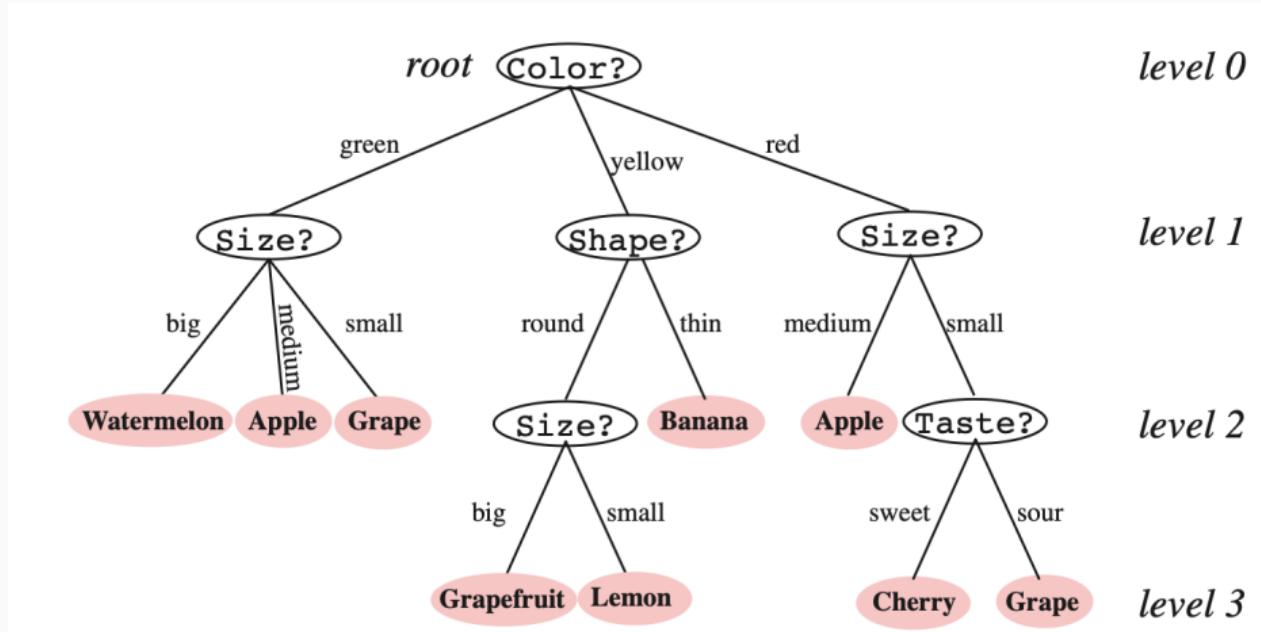
[dud01]



- What if  $\text{Color} = \text{yellow} \wedge \text{Shape} = \text{round} \wedge \text{Size} = \text{medium}$ ??
- Or, what if  $\text{Color} = \text{red} \wedge \text{Size} = \text{big}$ ??

BTW, problematic DTs even in books...

[dud01]



- What if  $\text{Color} = \text{yellow} \wedge \text{Shape} = \text{round} \wedge \text{Size} = \text{medium}$ ??
- Or, what if  $\text{Color} = \text{red} \wedge \text{Size} = \text{big}$ ??
- Easy to envision more serious use-cases...

Unit #04

## (Efficient) Intractability in Symbolic XAI

# Outline – Unit #04

Explaining Decision Lists

Myth #02: Model-Agnostic Explainability

Progress Report on Symbolic XAI

## An encoding for DLs – components

R <sub>1</sub> :	IF	( $\tau_1$ )	THEN	$d_1$
R <sub>2</sub> :	ELSE IF	( $\tau_2$ )	THEN	$d_2$
	...			
R <sub>j</sub> :	ELSE IF	( $\tau_j$ )	THEN	$d_j$
	...			
R <sub>n</sub> :	ELSE IF	( $\tau_n$ )	THEN	$d_n$
R <sub>DEF</sub> :	ELSE		THEN	$d_{n+1}$

## An encoding for DLs – components

R <sub>1</sub> :	IF	( $\tau_1$ )	THEN	$d_1$
R <sub>2</sub> :	ELSE IF	( $\tau_2$ )	THEN	$d_2$
...				
R <sub>j</sub> :	ELSE IF	( $\tau_j$ )	THEN	$d_j$
...				
R <sub>n</sub> :	ELSE IF	( $\tau_n$ )	THEN	$d_n$
R <sub>DEF</sub> :	ELSE		THEN	$d_{n+1}$

- Clauses for encoding  $\phi$ :  $\mathfrak{E}_\phi(z_1, \dots)$ , such that  $z_1 = 1$  iff  $\phi = 1$
- For  $\tau_j$ :  $\mathfrak{E}_{\tau_j}(t_j, \dots)$
- For  $x_i = v_i$ :  $\mathfrak{E}_{x_i=v_i}(l_i, \dots)$
- Let  $e_j = 1$  iff  $d_j$  matches  $c$
- Prediction change with rule up to R<sub>j</sub> (with  $d_j \neq c$ ), if  $\tau_j \not\models \perp$  and  $\tau_k \models \perp$ , for  $1 \leq k < j$ , with  $e_k = 1$ :

$$\left[ f_j \leftrightarrow \left( t_j \wedge \bigwedge_{1 \leq k < j, e_k=1} \neg t_k \right) \right]$$

## An encoding for DLs – components

R <sub>1</sub> :	IF	( $\tau_1$ )	THEN	$d_1$
R <sub>2</sub> :	ELSE IF	( $\tau_2$ )	THEN	$d_2$
...				
R <sub>j</sub> :	ELSE IF	( $\tau_j$ )	THEN	$d_j$
...				
R <sub>n</sub> :	ELSE IF	( $\tau_n$ )	THEN	$d_n$
R <sub>DEF</sub> :	ELSE		THEN	$d_{n+1}$

- Clauses for encoding  $\phi$ :  $\mathfrak{E}_\phi(z_1, \dots)$ , such that  $z_1 = 1$  iff  $\phi = 1$
- For  $\tau_j$ :  $\mathfrak{E}_{\tau_j}(t_j, \dots)$
- For  $x_i = v_i$ :  $\mathfrak{E}_{x_i=v_i}(l_i, \dots)$
- Let  $e_j = 1$  iff  $d_j$  matches  $c$
- Require that at least one  $f_j$ , with  $e_j = 0$  and  $1 \leq j \leq n$ , to be consistent (i.e. some rule up to  $j$  with prediction other than  $c$  to fire):

$$\left( \bigvee_{1 \leq j \leq n, e_j=0} f_j \right)$$

## An encoding for DLs – components

R <sub>1</sub> :	IF	( $\tau_1$ )	THEN	$d_1$
R <sub>2</sub> :	ELSE IF	( $\tau_2$ )	THEN	$d_2$
...				
R <sub>j</sub> :	ELSE IF	( $\tau_j$ )	THEN	$d_j$
...				
R <sub>n</sub> :	ELSE IF	( $\tau_n$ )	THEN	$d_n$
R <sub>DEF</sub> :	ELSE		THEN	$d_{n+1}$

- The set of soft clauses is given by:  $\mathcal{S} \triangleq \{(l_i), i = 1, \dots, m\}$
- The set of hard clauses is given by:

$$\begin{aligned}\mathcal{B} \triangleq & \bigwedge\nolimits_{1 \leqslant i \leqslant m} \mathfrak{E}_{x_i=v_i}(l_i, \dots) \wedge \bigwedge\nolimits_{1 \leqslant j \leqslant n} \mathfrak{E}_{\tau_j}(t_j, \dots) \wedge \\ & \bigwedge\nolimits_{1 \leqslant j \leqslant n, e_j=0} \left( f_j \leftrightarrow \left( t_j \wedge \bigwedge\nolimits_{1 \leqslant k < j, e_k=1} \neg t_k \right) \right) \wedge \left( \bigvee\nolimits_{1 \leqslant j \leqslant n, e_j=0} f_j \right)\end{aligned}$$

- $\mathcal{B} \cup \mathcal{S} \models \perp$ 
  - MUSes are AXp's & MCSes are CXp's

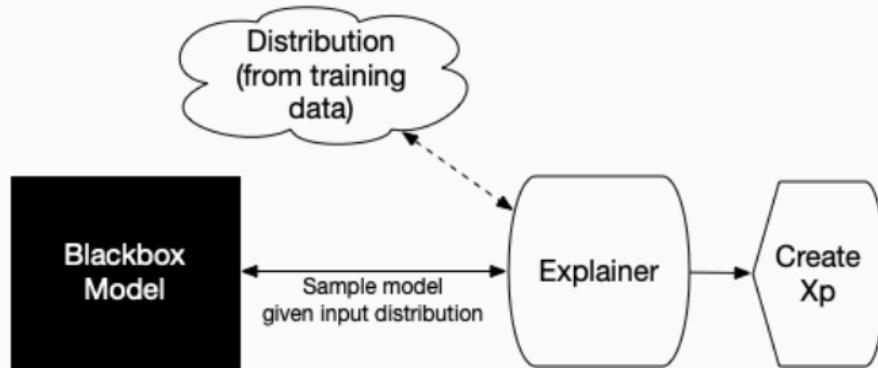
## Outline – Unit #04

Explaining Decision Lists

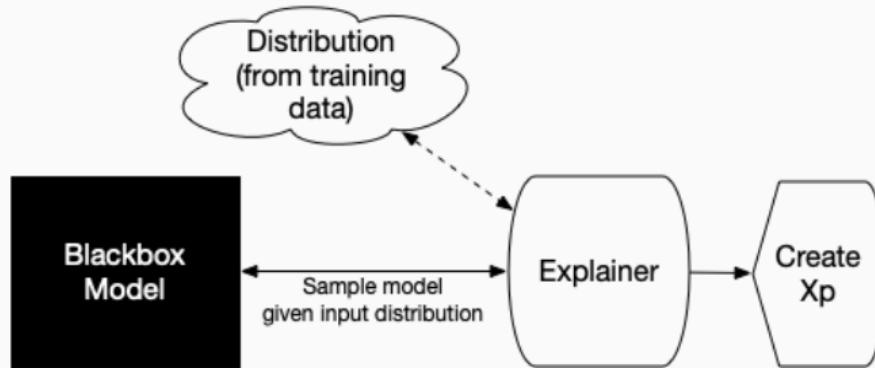
Myth #02: Model-Agnostic Explainability

Progress Report on Symbolic XAI

# What is model-agnostic explainability?



# What is model-agnostic explainability?



- Wildly popular XAI approach

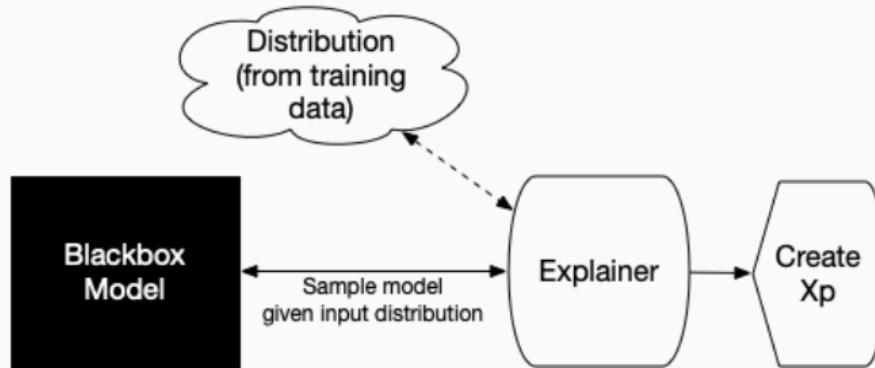
[RSG16, LL17, RSG18]

- **Feature attribution:** LIME, SHAP, ...
- **Feature selection:** Anchors, ...

[RSG16, LL17]

[RSG18]

# What is model-agnostic explainability?



- Wildly popular XAI approach
  - **Feature attribution:** LIME, SHAP, ...
  - **Feature selection:** Anchors, ...
- **Q:** Are model-agnostic explanations rigorous?

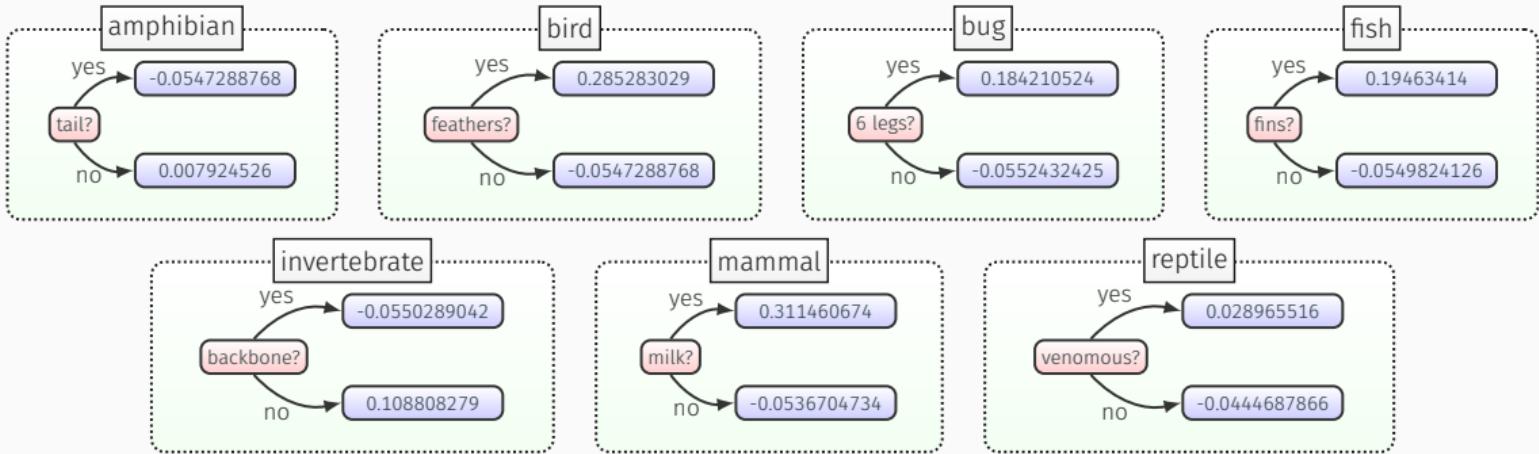
[RSG16, LL17, RSG18]

[RSG16, LL17]

[RSG18]

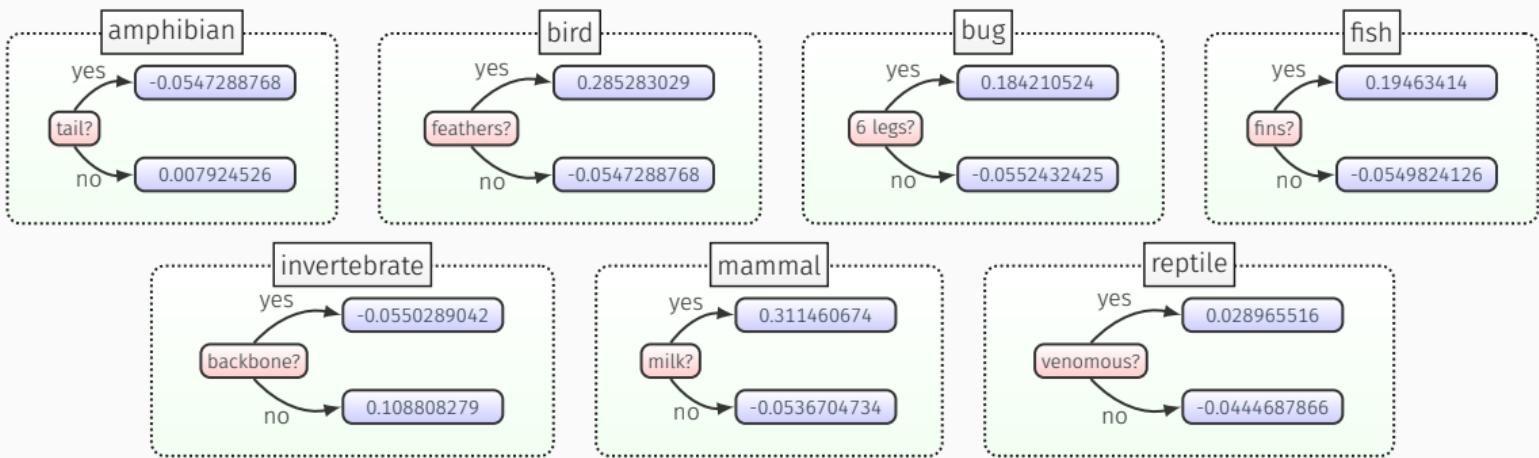
# Easy to spot problems – BT for zoo dataset

[INM19c, Ign20]



# Easy to spot problems – BT for zoo dataset

[INM19c, Ign20]

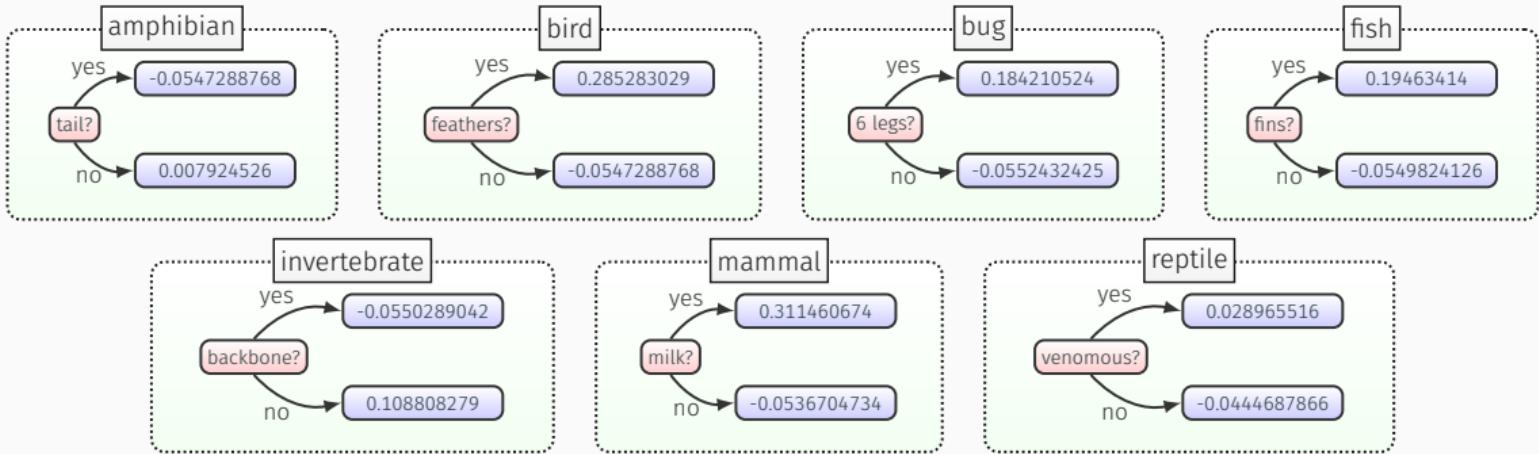


- Example instance:

**IF**       $(\text{animal\_name} = \text{pitviper}) \wedge \neg \text{hair} \wedge \neg \text{feathers} \wedge \text{eggs} \wedge \neg \text{milk} \wedge \neg \text{airborne} \wedge \neg \text{aquatic} \wedge \text{predator} \wedge \neg \text{toothed} \wedge \text{backbone} \wedge \text{breathes} \wedge \text{venomous} \wedge \neg \text{fins} \wedge (\text{legs} = 0) \wedge \text{tail} \wedge \neg \text{domestic} \wedge \neg \text{catsize}$   
**THEN**     $(\text{class} = \text{reptile})$

# Easy to spot problems – BT for zoo dataset & Anchor

[INM19c, Ign20]



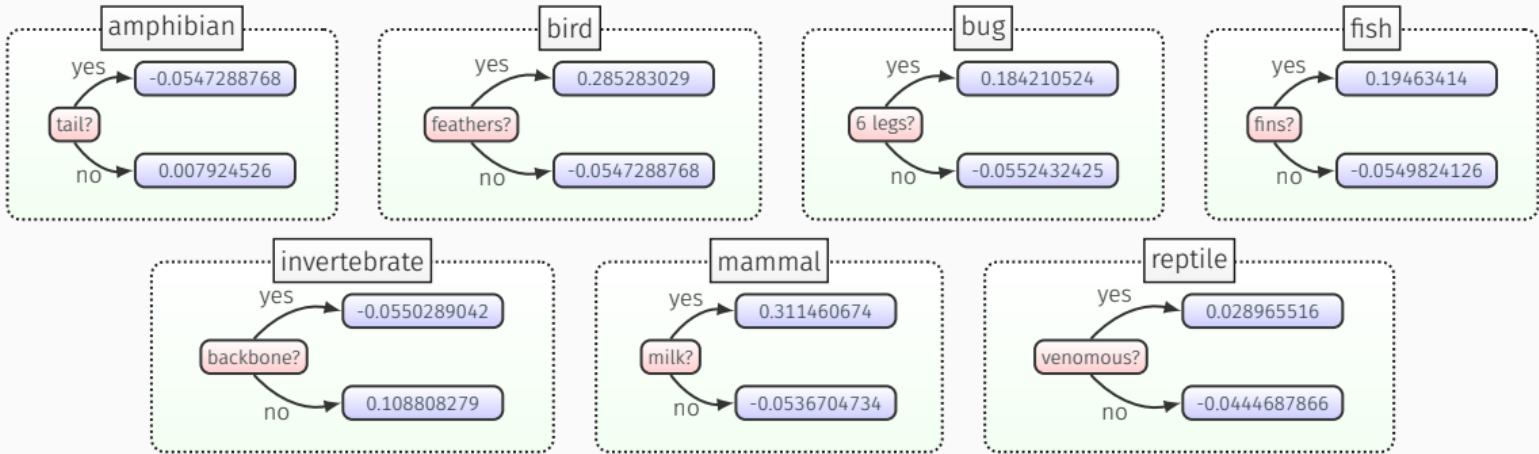
- Example instance (& Anchor picks):

[RSG18]

**IF**      (*animal\_name* = pitviper)  $\wedge \neg$ *hair*  $\wedge \neg$ *feathers*  $\wedge$  eggs  $\wedge \neg$ *milk*  $\wedge$   
 $\neg$ *airborne*  $\wedge \neg$ *aquatic*  $\wedge$  predator  $\wedge \neg$ *toothed*  $\wedge$  backbone  $\wedge$  breathes  $\wedge$   
*venomous*  $\wedge \neg$ *fins*  $\wedge$  (legs = 0)  $\wedge$  tail  $\wedge \neg$ *domestic*  $\wedge \neg$ *catsize*  
**THEN**    (*class* = *reptile*)

# Easy to spot problems – BT for zoo dataset & Anchor

[INM19c, Ign20]



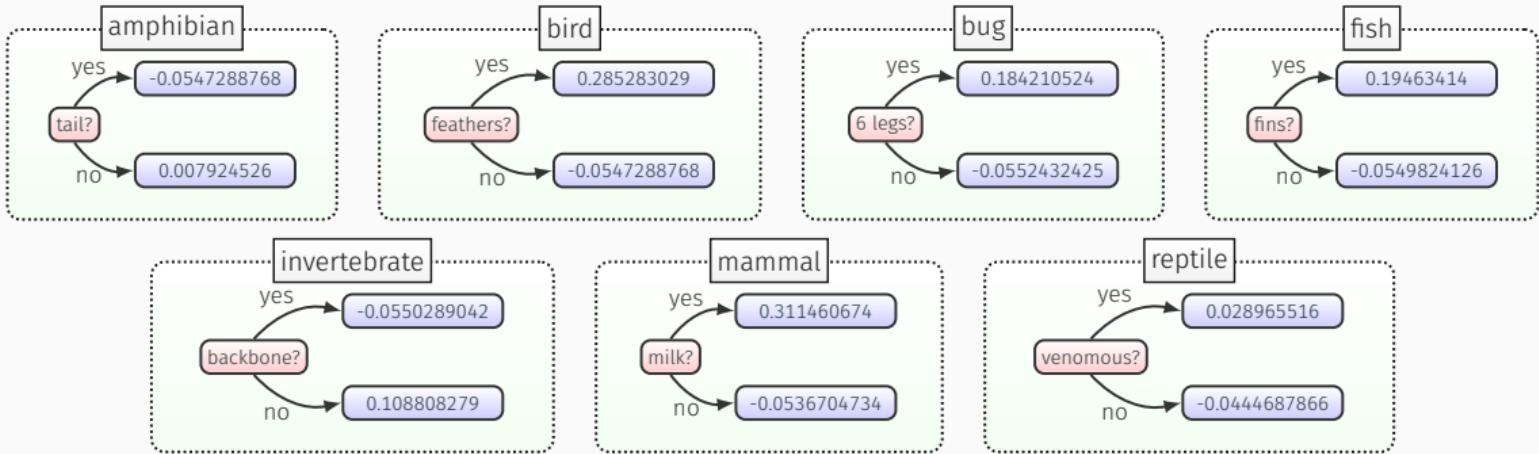
- Explanation obtained with Anchor:

[RSG18]

IF       $\neg \text{hair} \wedge \neg \text{milk} \wedge \neg \text{toothed} \wedge \neg \text{fins}$   
THEN    (class = **reptile**)

# Easy to spot problems – BT for zoo dataset & Anchor

[INM19c, Ign20]



- But, explanation **incorrectly “explains”** another instance (from training data!)

**IF**  $(\text{animal\_name} = \text{toad}) \wedge \neg \text{hair} \wedge \neg \text{feathers} \wedge \text{eggs} \wedge \neg \text{milk} \wedge \neg \text{airborne} \wedge \neg \text{aquatic} \wedge \neg \text{predator} \wedge \neg \text{toothed} \wedge \text{backbone} \wedge \text{breathes} \wedge \neg \text{venomous} \wedge \neg \text{fins} \wedge (\text{legs} = 4) \wedge \neg \text{tail} \wedge \neg \text{domestic} \wedge \neg \text{catsize}$

**THEN**  $(\text{class} = \text{amphibian})$

## Incorrect explanations:

Classifier for deciding bank loans

## Incorrect explanations:

Classifier for deciding bank loans

Two samples: Bessie  $\coloneqq (v_1, \text{Y})$  and Clive  $\coloneqq (v_2, \text{N})$

## Incorrect explanations:

Classifier for deciding bank loans

Two samples: Bessie  $\coloneqq (v_1, \text{Y})$  and Clive  $\coloneqq (v_2, \text{N})$

Explanation X: age = 45, salary = 50K

## Incorrect explanations:

Classifier for deciding bank loans

Two samples: Bessie  $\coloneqq (v_1, \text{Y})$  and Clive  $\coloneqq (v_2, \text{N})$

Explanation X: age = 45, salary = 50K

And,

X is consistent with Bessie  $\coloneqq (v_1, \text{Y})$

X is consistent with Clive  $\coloneqq (v_2, \text{N})$

## Incorrect explanations:

Classifier for deciding bank loans

Two samples: Bessie  $\coloneqq (v_1, \text{Y})$  and Clive  $\coloneqq (v_2, \text{N})$

Explanation X: age = 45, salary = 50K

And,

X is consistent with Bessie  $\coloneqq (v_1, \text{Y})$

X is consistent with Clive  $\coloneqq (v_2, \text{N})$

∴ different outcomes & same explanation !?

## How to validate model-agnostic explanations

- For feature selection, checking rigor is *easy*

## How to validate model-agnostic explanations

- For feature selection, checking rigor is *easy*
- Let  $\mathcal{X}$  be the features reported by model-agnostic tool

# How to validate model-agnostic explanations

- For feature selection, checking rigor is *easy*
- Let  $\mathcal{X}$  be the features reported by model-agnostic tool
- Check whether  $\mathcal{X}$  is a (*rigorous*) (W)AXp:
  1.  $\mathcal{X}$  is sufficient for prediction:

$$\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

2. And,  $\mathcal{X}$  is subset-minimal:

$$\forall(t \in \mathcal{X}). \exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in (\mathcal{X} \setminus \{t\})} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) \neq c)$$

Depending on logic encoding used for classifier, different automated reasoners can be employed

# How to validate model-agnostic explanations

- For feature selection, checking rigor is *easy*
- Let  $\mathcal{X}$  be the features reported by model-agnostic tool
- Check whether  $\mathcal{X}$  is a (*rigorous*) (W)AXp:
  1.  $\mathcal{X}$  is sufficient for prediction:

$$\forall(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

2. And,  $\mathcal{X}$  is subset-minimal:

$$\forall(t \in \mathcal{X}). \exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in (\mathcal{X} \setminus \{t\})} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) \neq c)$$

Depending on logic encoding used for classifier, different automated reasoners can be employed

- Approach is bounded by scalability of rigorous explanations...

# How serious is the lack of rigor of model-agnostic explanations?

- Obs: Lack of rigor of model-agnostic explanations known since 2019

[INM19c, Ign20, YIS<sup>+</sup>23]

## How serious is the lack of rigor of model-agnostic explanations?

- Obs: Lack of rigor of model-agnostic explanations known since 2019
- Results for boosted trees, due to non-scalability with NNs

[INM19c, Ign20, YIS<sup>+</sup>23]

[CG16]

# How serious is the lack of rigor of model-agnostic explanations?

- Obs: Lack of rigor of model-agnostic explanations known since 2019

[INM19c, Ign20, YIS+23]

- Results for boosted trees, due to non-scalability with NNs

[CG16]

- Some results for Anchors

[RSG18]

Dataset	% Incorrect	% Redundant	% Correct
adult	80.5%	1.6%	17.9%
lending	3.0%	0.0%	97.0%
rcdv	99.4%	0.4%	0.2%
compas	84.4%	1.7%	13.9%
german	99.7%	0.2%	0.1%

# How serious is the lack of rigor of model-agnostic explanations?

- Obs: Lack of rigor of model-agnostic explanations known since 2019

[INM19c, Ign20, YIS<sup>+</sup>23]

- Results for boosted trees, due to non-scalability with NNs

[CG16]

- Some results for Anchors

[RSG18]

Dataset	% Incorrect	% Redundant	% Correct
adult	80.5%	1.6%	17.9%
lending	3.0%	0.0%	97.0%
rcdv	99.4%	0.4%	0.2%
compas	84.4%	1.7%	13.9%
german	99.7%	0.2%	0.1%

- Obs: Results are not positive even if we count how often prediction changes
  - In this case, BNNs were used, to allow for model counting...

[NSM<sup>+</sup>19]

# How serious is the lack of rigor of model-agnostic explanations?

- Obs: Lack of rigor of model-agnostic explanations known since 2019

[INM19c, Ign20, YIS<sup>+</sup>23]

- Results for boosted trees, due to non-scalability with NNs

[CG16]

- Some results for Anchors

[RSG18]

Dataset	% Incorrect	% Redundant	% Correct
adult	80.5%	1.6%	17.9%
lending	3.0%	0.0%	97.0%
rcdv	99.4%	0.4%	0.2%
compas	84.4%	1.7%	13.9%
german	99.7%	0.2%	0.1%

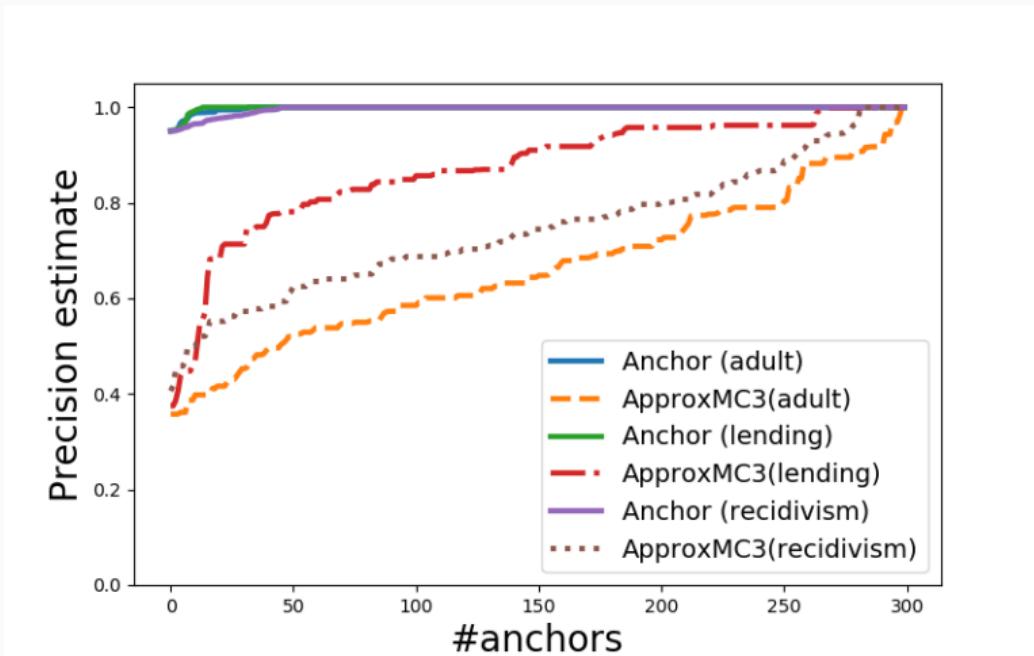
- Obs: Results are not positive even if we count how often prediction changes

[NSM<sup>+</sup>19]

- In this case, BNNs were used, to allow for model counting...

- For feature attribution we proposed different ways of assessing rigor

[INM19c, NSM<sup>+</sup>19, Ign20, YIS<sup>+</sup>23]

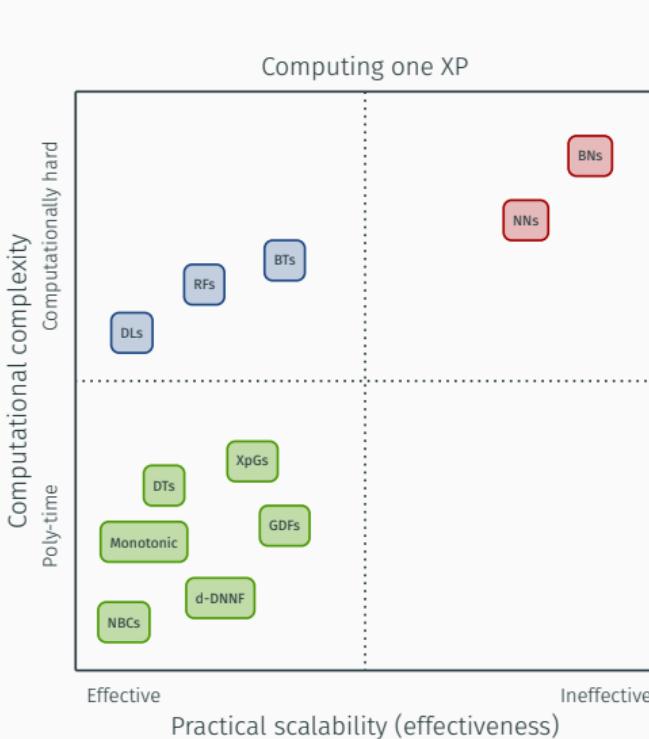


# Outline – Unit #04

Explaining Decision Lists

Myth #02: Model-Agnostic Explainability

Progress Report on Symbolic XAI



[INM19c, Ign20, IIM20, MGC<sup>+</sup>20, MGC<sup>+</sup>21, HIIM21, IMS21, IM21, CM21, HII<sup>+</sup>22, IISMS22]

## • Formal explanations efficient for several families of classifiers

### • Polynomial-time:

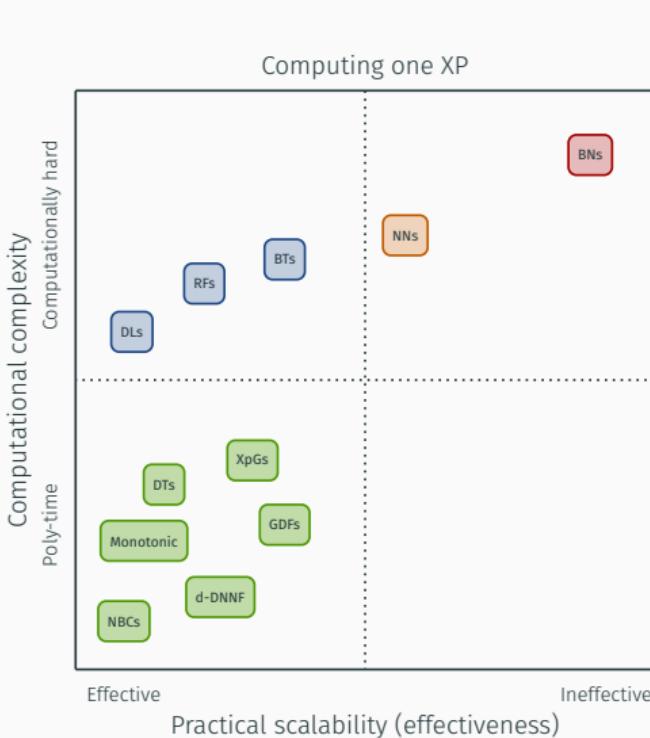
- Naive-Bayes classifiers (**NBCs**) [MGC<sup>+</sup>20]
- Decision trees (**DTs**) [IIM20, HIIM21]
- **XpG's:** DTs, OBDDs, OMDDs, etc. [HIIM21]
- **Monotonic** classifiers [MGC<sup>+</sup>21]
- Propositional languages (e.g. d-DNNF, ...) [HII<sup>+</sup>22]
- Additional results [CM21, HII<sup>+</sup>22]

### • Comp. hard, but **effective** (efficient in practice):

- Random forests (**RFs**) [IMS21]
- Decision lists (**DLS**) [IM21]
- Boosted trees (**BTs**) [INM19c, Ign20, IISMS22]

### • Comp. hard, and **ineffective** (hard in practice):

- Neural networks (**NNs**) [INM19a]
- Bayesian networks (**BNs**) [SCD18]



[INM19c, Ign20, IIM20, MGC<sup>+</sup>20, MGC<sup>+</sup>21, HIIM21, IMS21, IM21, CM21, HII<sup>+</sup>22, IISMS22]

## • Formal explanations efficient for several families of classifiers

### • Polynomial-time:

- Naive-Bayes classifiers (**NBCs**) [MGC<sup>+</sup>20]
- Decision trees (**DTs**) [IIM20, HIIM21]
- **XpG's**: DTs, OBDDs, OMDDs, etc. [HIIM21]
- **Monotonic** classifiers [MGC<sup>+</sup>21]
- Propositional languages (e.g. d-DNNF, ...) [HII<sup>+</sup>22]
- Additional results [CM21, HII<sup>+</sup>22]

### • Comp. hard, but **effective** (efficient in practice):

- Random forests (**RFs**) [IMS21]
- Decision lists (**DLS**) [IM21]
- Boosted trees (**BTs**) [INM19c, Ign20, IISMS22]

### • Comp. hard, but some practical **scalability**:

- Neural networks (**NNs**) [HM23b]

### • Comp. hard, and **ineffective** (hard in practice):

- Bayesian networks (**BNs**) [SCD18]

# Results for RFs in 2021 (with SAT)

[IMS21]

Dataset	#F #C #I			RF			CNF		SAT oracle				AXp (RFexpl)				Anchor	
	D	#N	%A	#var	#cl	MxS	MxU	#S	#U	Mx	m	avg	%w	avg	%w	avg	%w	
ann-thyroid	( 21 3 718) 4	2192	98	17854	29230	0.12	0.15	2	18	0.36	0.05	0.13	96	0.32	4			
appendicitis	( 7 2 43) 6	1920	90	5181	10085	0.02	0.02	4	3	0.05	0.01	0.03	100	0.48	0			
banknote	( 4 2 138) 5	2772	97	8068	16776	0.01	0.01	2	2	0.03	0.02	0.02	100	0.19	0			
biodegradation	( 41 2 106 5	4420	88	11007	23842	0.31	1.05	17	22	2.27	0.04	0.29	97	4.07	3			
heart-c	( 13 2 61) 5	3910	85	5594	11963	0.04	0.02	6	7	0.07	0.01	0.04	100	0.85	0			
ionosphere	( 34 2 71) 5	2096	87	7174	14406	0.02	0.02	22	11	0.11	0.02	0.03	100	12.43	0			
karhunen	( 64 10 200) 5	6198	91	36708	70224	1.06	1.41	35	29	14.64	0.65	2.78	100	28.15	0			
letter	( 16 26 398 8	44304	82	28991	68148	1.97	3.31	8	8	6.91	0.24	1.61	70	2.48	30			
magic	( 10 2 381) 6	9840	84	29530	66776	0.51	1.84	6	4	2.13	0.07	0.14	99	0.91	1			
new-thyroid	( 5 3 43) 5	1766	100	17443	28134	0.03	0.01	3	2	0.08	0.03	0.05	100	0.36	0			
pendigits	( 16 10 220) 6	12004	95	30522	59922	2.40	1.32	10	6	4.11	0.14	0.94	96	3.68	4			
ring	( 20 2 740 6	6188	89	19114	42362	0.27	0.44	11	9	1.25	0.05	0.25	92	7.25	8			
segmentation	( 19 7 42) 4	1966	90	21288	35381	0.11	0.17	8	10	0.53	0.11	0.31	100	4.13	0			
shuttle	( 9 7 116 3	1460	99	18669	29478	0.11	0.08	2	7	0.34	0.05	0.14	99	0.42	1			
sonar	( 60 2 42) 5	2614	88	9938	20537	0.04	0.06	36	24	0.43	0.04	0.09	100	23.02	0			
spectf	( 44 2 54) 5	2306	88	6707	13449	0.07	0.06	20	24	0.34	0.02	0.07	100	8.12	0			
texture	( 40 11 550) 5	5724	87	34293	64187	0.79	0.63	23	17	3.24	0.19	0.93	100	28.13	0			
twonorm	( 20 2 740 5	6266	94	21198	46901	0.08	0.08	12	8	0.28	0.06	0.10	100	5.73	0			
vowel	( 13 11 198) 6	10176	90	44523	88696	1.66	2.11	8	5	4.52	0.15	1.15	66	1.67	34			
waveform-40	( 40 3 500 5	6232	83	30438	58380	0.50	0.86	15	25	7.07	0.11	0.88	100	11.93	0			
wpbc	( 33 2 78) 5	2432	76	9078	18675	1.00	1.53	20	13	5.33	0.03	0.65	79	3.91	21			

# Results for NNs in 2019 (with SMT/MILP)

[INM19a]

Dataset		Minimal explanation			Minimum explanation		
		size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	m	1	0.03	0.05	—	—
		a	8.79	1.38	0.33	—	—
		M	14	17.00	1.43	—	—
backache	(32)	m	13	0.13	0.14	—	—
		a	19.28	5.08	0.85	—	—
		M	26	22.21	2.75	—	—
breast-cancer	(9)	m	3	0.02	0.04	3	0.02
		a	5.15	0.65	0.20	4.86	0.41
		M	9	6.11	0.41	9	24.80
cleve	(13)	m	4	0.05	0.07	4	0.07
		a	8.62	3.32	0.32	7.89	5.14
		M	13	60.74	0.60	13	39.06
hepatitis	(19)	m	6	0.02	0.04	4	0.04
		a	11.42	0.07	0.06	9.39	2.89
		M	19	0.26	0.20	19	22.23
voting	(16)	m	3	0.01	0.02	3	0.02
		a	4.56	0.04	0.13	3.46	0.25
		M	11	0.10	0.37	11	1.77
spect	(22)	m	3	0.02	0.02	3	0.04
		a	7.31	0.13	0.07	6.44	0.67
		M	20	0.88	0.29	20	10.73

# Results for NNs in 2019 (with SMT/MILP)

[INM19a]

First rigorous approach  
for explaining NNs !

			Minimal explanation			Minimum explanation		
			size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	m	1	0.03	0.05	—	—	—
		a	8.79	1.38	0.33	—	—	—
		M	14	17.00	1.43	—	—	—
backache	(32)	m	13	0.13	0.14	—	—	—
		a	19.28	5.08	0.85	—	—	—
		M	26	22.21	2.75	—	—	—
breast-cancer	(9)	m	3	0.02	0.04	3	0.02	0.03
		a	5.15	0.65	0.20	4.86	2.18	0.41
		M	9	6.11	0.41	9	24.80	1.81
cleve	(13)	m	4	0.05	0.07	4	—	0.07
		a	8.62	3.32	0.32	7.89	—	5.14
		M	13	60.74	0.60	13	—	39.06
hepatitis	(19)	m	6	0.02	0.04	4	0.01	0.04
		a	11.42	0.07	0.06	9.39	4.07	2.89
		M	19	0.26	0.20	19	27.05	22.23
voting	(16)	m	3	0.01	0.02	3	0.01	0.02
		a	4.56	0.04	0.13	3.46	0.3	0.25
		M	11	0.10	0.37	11	1.25	1.77
spect	(22)	m	3	0.02	0.02	3	0.02	0.04
		a	7.31	0.13	0.07	6.44	1.61	0.67
		M	20	0.88	0.29	20	8.97	10.73

# Results for NNs in 2019 (with SMT/MILP)

[INM19a]

First rigorous approach  
for explaining NNs !

			Minimal explanation			Minimum explanation		
			size	SMT (s)	MILP (s)	size	SMT (s)	MILP (s)
australian	(14)	m	1	0.03	0.05	—	—	—
		a	8.79	1.38	0.33	—	—	—
		M	14	17.00	1.43	—	—	—
backache	(32)	m	13	0.13	0.14	—	—	—
		a	19.28	5.08	0.85	—	—	—
		M	26	22.21	2.75	—	—	—
breast-cancer	(9)	m	3	0.02	0.04	3	0.02	0.03
		a	5.15	0.65	0.20	4.86	2.18	0.41
		M	9	6.11	0.41	9	24.80	1.81
cleve	(13)	m	4	0.05	0.07	4	—	0.07
		a	8.62	3.32	0.32	7.89	—	5.14
		M	13	60.74	0.60	13	—	39.06
hepatitis	(19)	m	6	0.02	0.04	4	0.01	0.04
		a	11.42	0.07	0.06	9.39	4.07	2.89
		M	19	0.26	0.20	19	27.05	22.23
voting	(16)	m	3	0.01	0.02	3	0.01	0.02
		a	4.56	0.04	0.13	3.46	0.3	0.25
		M	11	0.10	0.37	11	1.25	1.77
spect	(22)	m	3	0.02	0.02	3	0.02	0.04
		a	7.31	0.13	0.07	6.44	1.61	0.67
		M	20	0.88	0.29	20	8.97	10.78

Scales to (a few)  
tens of neurons...

# Results for NNs in 2023 (using Marabou [KHI<sup>+</sup>19])

[HM23b]

DNN	points	AXp	#Calls	Time	#TO	AXp	#Calls	Time	#TO
$\epsilon = 0.1$					$\epsilon = 0.05$				
ACASXU_1_5	#1	3	5	185.9	0	2	5	113.8	0
	#2	2	5	273.8	0	1	5	33.2	0
	#3	0	5	714.2	0	0	5	4.3	0
ACASXU_3_1	#1	0	5	2219.3	0	0	5	14.2	0
	#2	2	5	4263.5	1	0	5	1853.1	0
	#3	1	5	581.8	0	0	5	355.9	0
ACASXU_3_2	#1	3	5	13739.3	2	1	5	6890.1	1
	#2	3	5	226.4	0	2	5	125.1	0
	#3	2	5	1740.6	0	2	5	173.6	0
ACASXU_3_5	#1	4	5	43.6	0	2	5	59.4	0
	#2	3	5	5039.4	0	2	5	4303.8	1
	#3	2	5	5574.9	1	2	5	2660.3	0
ACASXU_3_6	#1	1	5	6225.0	1	0	5	51.0	0
	#2	3	5	4957.2	1	2	5	1897.3	0
	#3	1	5	196.1	0	1	5	919.2	0
ACASXU_3_7	#1	3	5	6256.2	0	4	5	26.9	0
	#2	4	5	311.3	0	1	5	6958.6	1
	#3	2	5	7756.5	1	1	5	7807.6	1
ACASXU_4_1	#1	2	5	12413.0	2	1	5	5090.5	1
	#2	1	5	5035.1	1	0	5	2335.6	0
	#3	4	5	1237.3	0	4	5	1143.4	0
ACASXU_4_2	#1	4	5	15.9	0	4	5	12.1	0
	#2	3	5	1507.6	0	1	5	111.3	0
	#3	2	5	5641.6	2	0	5	1639.1	0

# Results for NNs in 2023 (using Marabou [KHI<sup>+</sup>19])

[HM23b]

DNN	points	AXp	#Calls	Time	#TO	AXp	#Calls	Time	#TO
$\epsilon = 0.1$					$\epsilon = 0.05$				
ACASXU_1_5	#1	3	5	185.9	0	2	5	113.8	0
	#2	2	5	273.8	0	1	5	33.2	0
	#3	0	5	714.2	0	0	5	4.3	0
ACASXU_3_1	#1	0	5	2219.3	0	0	5	14.2	0
	#2	2	5	4263.5	1	0	5	1853.1	0
	#3	1	5	581.8	0	0	5	355.9	0
ACASXU_3_2	#1	3	5	13739.3	2	1	5	6890.1	1
	#2	3	5	226.4	0	2	5	125.1	0
	#3	2	5	1740.6	0	2	5	173.6	0
ACASXU_3_5	#1	4	5	43.6	0	2	5	59.4	0
	#2	3	5	5039.4	0	2	5	4303.8	1
	#3	2	5	5574.9	1	2	5	2660.3	0
ACASXU_3_6	#1	1	5	6225.0	1	0	5	51.0	0
	#2	3	5	4957.2	1	2	5	1897.3	0
	#3	1	5	196.1	0	1	5	919.2	0
ACASXU_3_7	#1	3	5	6256.2	0	4	5	26.9	0
	#2	4	5	311.3	0	1	5	6958.6	1
	#3	2	5	7756.5	1	1	5	7807.6	1
ACASXU_4_1	#1	2	5	12413.0	2	1	5	5090.5	1
	#2	1	5	5035.1	1	0	5	2335.6	0
	#3	4	5	1237.3	0	4	5	1143.4	0
ACASXU_4_2	#1	4	5	15.9	0	4	5	12.1	0
	#2	3	5	1507.6	0	1	5	111.3	0
	#3	2	5	5641.6	2	0	5	1639.1	0

Scales to a few hundred neurons

# More recent results (from 2024)...

[IHM<sup>+</sup>24a]

Model	Deletion							SwiftXplain						
	avgC	nCalls	Len	Mn	Mx	avg	TO	avgC	nCalls	Len	FD%	Mn	Mx	avg
gtsrb-dense	0.06	1024	448	52.0	76.3	63.1	0	0.23	54	447	77.4	10.8	14.0	12.2
gtsrb-convSmall	0.06	1024	309	59.2	82.6	65.1	0	0.22	74	313	39.7	15.1	19.5	16.2
gtsrb-conv	—	—	—	—	—	—	100	96.49	45	174	33.2	3858.7	6427.7	4449.4
mnist-denseSmall	0.28	784	177	190.9	420.3	220.4	0	0.77	111	180	15.5	77.6	104.4	85.1
mnist-dense	0.19	784	231	138.1	179.9	150.6	0	0.75	183	229	11.5	130.1	145.5	136.8
mnist-convSmall	—	—	—	—	—	—	100	98.56	52	116	21.3	4115.2	6858.3	5132.8

## More recent results (from 2024)...

[IHM<sup>+</sup> 24a]

Model	Deletion								SwiftXplain							
	avgC	nCalls	Len	Mn	Mx	avg	TO	avgC	nCalls	Len	FD%	Mn	Mx	avg		
gtsrb-dense	0.06	1024	448	52.0	76.3	63.1	0	0.23	54	447	77.4	10.8	14.0	12.2		
gtsrb-convSmall	0.06	1024	309	59.2	82.6	65.1	0	0.22	74	313	39.7	15.1	19.5	16.2		
gtsrb-conv	—	—	—	—	—	—	100	96.49	45	174	33.2	3858.7	6427.7	4449.4		
mnist-denseSmall	0.28	784	177	190.9	420.3	220.4	0	0.77	111	180	15.5	77.6	104.4	85.1		
mnist-dense	0.19	784	231	138.1	179.9	150.6	0	0.75	183	229	11.5	130.1	145.5	136.8		
mnist-convSmall	—	—	—	—	—	—	100	98.56	52	116	21.3	4115.2	6858.3	5132.8		

Scales to **tens of thousands** of neurons!

Model	Deletion							SwiftXplain						
	avgC	nCalls	Len	Mn	Mx	avg	TO	avgC	nCalls	Len	FD%	Mn	Mx	avg
gtsrb-dense	0.06	1024	448	52.0	76.3	63.1	0	0.23	54	447	77.4	10.8	14.0	12.2
gtsrb-convSmall	0.06	1024	309	59.2	82.6	65.1	0	0.22	74	313	39.7	15.1	19.5	16.2
gtsrb-conv	—	—	—	—	—	—	100	96.49	45	174	33.2	3858.7	6427.7	4449.4
mnist-denseSmall	0.28	784	177	190.9	420.3	220.4	0	0.77	111	180	15.5	77.6	104.4	85.1
mnist-dense	0.19	784	231	138.1	179.9	150.6	0	0.75	183	229	11.5	130.1	145.5	136.8
mnist-convSmall	—	—	—	—	—	—	100	98.56	52	116	21.3	4115.2	6858.3	5132.8

Scales to **tens of thousands** of neurons!

Largest for MNIST: 10142 neurons  
Largest for GSTRB: 94308 neurons

Unit #05

Queries in Symbolic XAI

## Outline – Unit #05

Enumeration of Explanations

Feature Necessity & Relevancy

## How to navigate the space of XPs?

- **Goal:** iteratively list yet unlisted XPs (either AXp's or CXp's)

# How to navigate the space of XPs?

- **Goal:** iteratively list yet unlisted XPs (either AXp's or CXp's)
- Complexity results:
  - For NBCs: enumeration with polynomial delay [MGC<sup>+</sup>20]
  - For monotonic classifiers: enumeration is computationally hard [MGC<sup>+</sup>21]
  - Recall: for DTs, enumeration of CXp's is in P [HIIM21, IIM22]

# How to navigate the space of XPs?

- **Goal:** iteratively list yet unlisted XPs (either AXp's or CXp's)
- Complexity results:
  - For NBCs: enumeration with polynomial delay
  - For monotonic classifiers: enumeration is computationally hard
  - Recall: for DTs, enumeration of CXp's is in P
- There are algorithms for direct enumeration of CXp's
  - Akin to enumerating MCSes

[MGC<sup>+</sup>20]

[MGC<sup>+</sup>21]

[HIIM21, IIM22]

# How to navigate the space of XPs?

- **Goal:** iteratively list yet unlisted XPs (either AXp's or CXp's)
- Complexity results:
  - For NBCs: enumeration with polynomial delay
  - For monotonic classifiers: enumeration is computationally hard
  - Recall: for DTs, enumeration of CXp's is in P
- There are algorithms for direct enumeration of CXp's
  - Akin to enumerating MCSes
- No known algorithms for **direct** enumeration of AXp's
  - Akin to enumerating MUSes

[MGC<sup>+</sup>20]

[MGC<sup>+</sup>21]

[HIIM21, IIM22]

[MM20]

# How to navigate the space of XPs?

- **Goal:** iteratively list yet unlisted XPs (either AXp's or CXp's)
- Complexity results:
  - For NBCs: enumeration with polynomial delay [MGC<sup>+</sup>20]
  - For monotonic classifiers: enumeration is computationally hard [MGC<sup>+</sup>21]
  - Recall: for DTs, enumeration of CXp's is in P [HIIM21, IIM22]
- There are algorithms for direct enumeration of CXp's
  - Akin to enumerating MCSes
- No known algorithms for **direct** enumeration of AXp's
  - Akin to enumerating MUSes
- Enumeration of MCSes + dualization often not realistic
  - There can be too many CXp's...

# How to navigate the space of XPs?

- **Goal:** iteratively list yet unlisted XPs (either AXp's or CXp's)
- Complexity results:
  - For NBCs: enumeration with polynomial delay [MGC<sup>+</sup>20]
  - For monotonic classifiers: enumeration is computationally hard [MGC<sup>+</sup>21]
  - Recall: for DTs, enumeration of CXp's is in P [HIIM21, IIM22]
- There are algorithms for direct enumeration of CXp's
  - Akin to enumerating MCSes
- No known algorithms for **direct** enumeration of AXp's [MM20]
  - Akin to enumerating MUSes
- Enumeration of MCSes + dualization often not realistic [LS08, FK96]
  - There can be too many CXp's...
- Best solution is a MARCO-like algorithm (for enumerating MUSes) [LPMM16]
  - On-demand enumeration of AXp's/CXp's

## Recall computing one AXp/CXp – oneXP

**Input:** Predicate  $\mathbb{P}$ , parameterized by  $\mathcal{T}, \mathcal{M}$

**Output:** One XP  $\mathcal{S}$

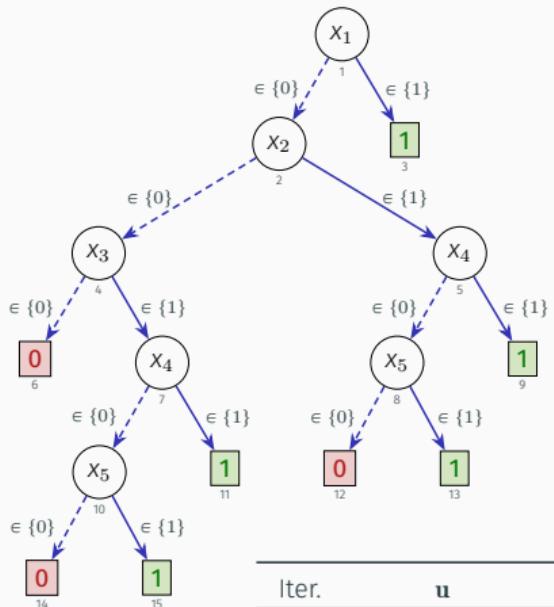
```
1: procedure oneXP( $\mathbb{P}$ )
2:    $\mathcal{S} \leftarrow \mathcal{F}$                                  $\triangleright$  Initialization:  $\mathbb{P}(\mathcal{S})$  holds
3:   for  $i \in \mathcal{F}$  do
4:     if  $\mathbb{P}(\mathcal{S} \setminus \{i\})$  then            $\triangleright$  Loop invariant:  $\mathbb{P}(\mathcal{S})$  holds
5:        $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
6:   return  $\mathcal{S}$                                  $\triangleright$  Update  $\mathcal{S}$  only if  $\mathbb{P}(\mathcal{S} \setminus \{i\})$  holds
                                                 $\triangleright$  Returned set  $\mathcal{S}$ :  $\mathbb{P}(\mathcal{S})$  holds
```

# Generic oracle-based enumeration algorithm

Input: Parameters  $\mathbb{P}_{\text{axp}}, \mathbb{P}_{\text{cxp}}, \mathcal{T}, \mathcal{F}, \kappa, \mathbf{v}$

```
1:  $\mathcal{H} \leftarrow \emptyset$                                  $\triangleright \mathcal{H}$  defined on set  $U = \{u_1, \dots, u_m\}$ ; initially no constraints
2: repeat
3:    $(\text{outc}, \mathbf{u}) \leftarrow \text{SAT}(\mathcal{H})$            $\triangleright$  Use SAT oracle to pick assignment s.t. known constraints in  $\mathcal{H}$ 
4:   if  $\text{outc} = \text{true}$  then
5:      $\mathcal{S} \leftarrow \{i \in \mathcal{F} \mid u_i = 0\}$             $\triangleright \mathcal{S}$ : fixed features
6:      $\mathcal{U} \leftarrow \{i \in \mathcal{F} \mid u_i = 1\}$             $\triangleright \mathcal{U}$ : universal features;  $\mathcal{F} = \mathcal{S} \cup \mathcal{U}$ 
7:     if  $\mathbb{P}_{\text{cxp}}(\mathcal{U}; \mathcal{T}, \mathcal{F}, \kappa, \mathbf{v})$  then       $\triangleright \mathcal{U} = \mathcal{F} \setminus \mathcal{S} \supseteq$  some Cxp
8:        $\mathcal{P} \leftarrow \text{oneXP}(\mathcal{U}; \mathbb{P}_{\text{cxp}}, \mathcal{T}, \mathcal{F}, \kappa, \mathbf{v})$ 
9:       reportCxp( $\mathcal{P}$ )
10:       $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\vee_{i \in \mathcal{P}} \neg u_i)\}$          $\triangleright \mathcal{P} \subseteq \mathcal{U}$ : one 1-value variable must be 0 in future iterations
11:    else                                               $\triangleright \mathcal{S} \supseteq$  some Axp
12:       $\mathcal{P} \leftarrow \text{oneXP}(\mathcal{S}; \mathbb{P}_{\text{axp}}, \mathcal{T}, \mathcal{F}, \kappa, \mathbf{v})$ 
13:      reportAxp( $\mathcal{P}$ )
14:       $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\vee_{i \in \mathcal{P}} u_i)\}$          $\triangleright \mathcal{P} \subseteq \mathcal{S}$ : one 0-value variable must be 1 in future iterations
15: until  $\text{outc} = \text{false}$ 
```

# DT classifier – example run of enumerator



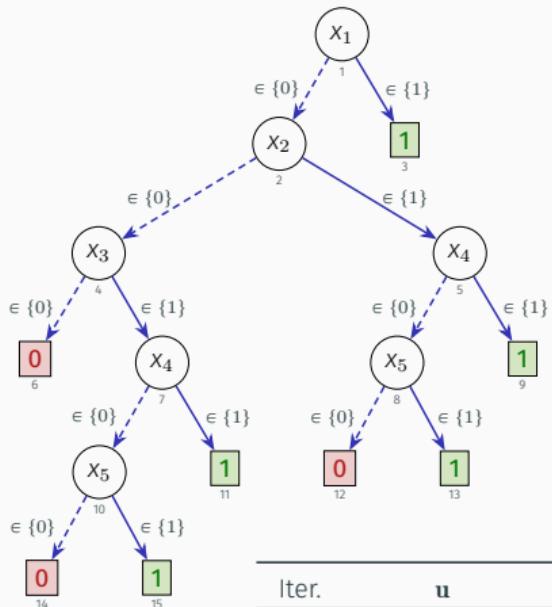
- Instance:  $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$

$x_3$	$x_5$	$x_1$	$x_2$	$x_4$	$\kappa_2(\mathbf{x})$
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

$x_3$	$x_5$	$x_1$	$x_2$	$x_4$	$\kappa_2(\mathbf{x})$
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
1	1	0	0	0	1

Iter.	$\mathbf{u}$	$\mathcal{S}$	$\mathbb{P}_{\text{exp}}(\cdot)$	AXp	CXp	Clause	Resulting $\mathcal{H}$
1	(1, 1, 1, 1, 1)	$\emptyset$	1	-	{3}	$(\neg u_3)$	$\{(\neg u_3)\}$
2	(1, 1, 0, 1, 1)	{3}	1	-	{5}	$(\neg u_5)$	$\{(\neg u_3), (\neg u_5)\}$
3	(1, 1, 0, 1, 0)	{3, 5}	0	{3, 5}	-	$(u_3 \vee u_5)$	$\{(\neg u_3), (\neg u_5), (u_3 \vee u_5)\}$
5	[outc = false]	-	-	-	-	-	$\{(\neg u_3), (\neg u_5), (u_3 \vee u_5)\}$

# DT classifier – another example run of enumerator



• Instance:  $(\mathbf{v}, c) = ((0, 0, 1, 0, 1), 1)$

$x_3$	$x_5$	$x_1$	$x_2$	$x_4$	$\kappa_2(\mathbf{x})$
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

$x_3$	$x_5$	$x_1$	$x_2$	$x_4$	$\kappa_2(\mathbf{x})$
0	0	0	0	0	0
0	1	0	0	0	0
1	0	0	0	0	0
1	1	0	0	0	1

Iter.	$\mathbf{u}$	$\mathcal{S}$	$\mathbb{P}_{\text{exp}}(\cdot)$	AXp	CXp	Clause	Resulting $\mathcal{H}$
1	(0, 0, 0, 0, 0)	{1, 2, 3, 4, 5}	0	{3, 5}	-	$(u_3 \vee u_5)$	$\{(u_3 \vee u_5)\}$
2	(0, 0, 1, 0, 0)	{1, 2, 4, 5}	1	-	{3}	$(\neg u_3)$	$\{(u_3 \vee u_5), (\neg u_3)\}$
3	(0, 0, 0, 0, 1)	{1, 2, 3, 4}	1	-	{5}	$(\neg u_5)$	$\{(u_3 \vee u_5), (\neg u_3), (\neg u_5)\}$
5	[outc = false]	-	-	-	-	-	$\{(u_3 \vee u_5), (\neg u_3), (\neg u_5)\}$

## DTs admit more efficient algorithms

- Recall:
  - Given instance  $(\mathbf{v}, c)$ , create set  $\mathcal{I}$
  - For each path  $P_k$  with prediction  $d \neq c$ :
    - Let  $I_k$  denote the features with literals inconsistent with  $\mathbf{v}$
    - Add  $I_k$  to  $\mathcal{I}$
  - Remove from  $\mathcal{I}$  the sets that have a proper subset in  $\mathcal{I}$ , and duplicates
- $\mathcal{I}$  is the set of CXp's – algorithm runs in poly-time

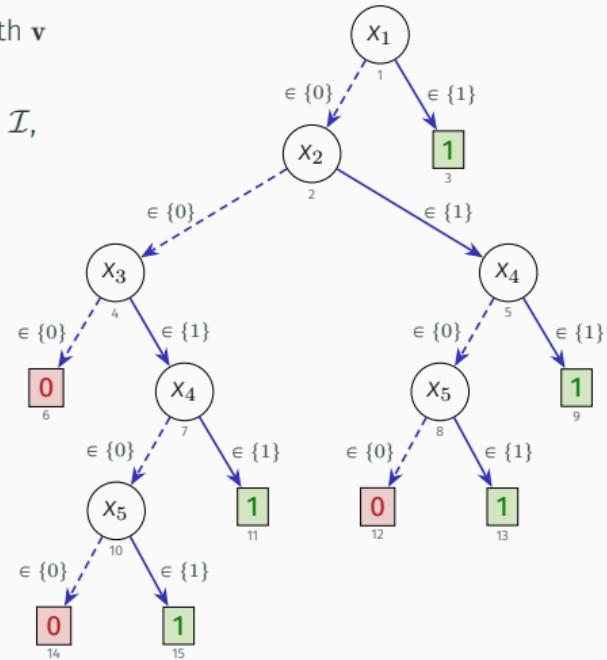
## DTs admit more efficient algorithms

- Recall:
  - Given instance  $(\mathbf{v}, c)$ , create set  $\mathcal{I}$
  - For each path  $P_k$  with prediction  $d \neq c$ :
    - Let  $I_k$  denote the features with literals inconsistent with  $\mathbf{v}$
    - Add  $I_k$  to  $\mathcal{I}$
  - Remove from  $\mathcal{I}$  the sets that have a proper subset in  $\mathcal{I}$ , and duplicates
- $\mathcal{I}$  is the set of CXp's – algorithm runs in poly-time
- For AXp's: run std dualization algorithm [FK96]
  - Obs: starting hypergraph is poly-size!
  - **And each MHS is an AXp**

# DTs admit more efficient algorithms

- Recall:

- Given instance  $(\mathbf{v}, c)$ , create set  $\mathcal{I}$
- For each path  $P_k$  with prediction  $d \neq c$ :
  - Let  $I_k$  denote the features with literals inconsistent with  $\mathbf{v}$
  - Add  $I_k$  to  $\mathcal{I}$
- Remove from  $\mathcal{I}$  the sets that have a proper subset in  $\mathcal{I}$ , and duplicates
- $\mathcal{I}$  is the set of CXp's – algorithm runs in poly-time
- For AXp's: run std dualization algorithm [FK96]
  - Obs: starting hypergraph is poly-size!
  - **And each MHS is an AXp**
- Example:
  - $I_1 = \{3\}$
  - $I_2 = \{5\}$
  - $I_3 = \{2, 5\}$
  - $\therefore$  keep  $I_1$  and  $I_2$
  - AXp's: MHSes yield  $\{\{3, 5\}\}$



## Outline – Unit #05

Enumeration of Explanations

Feature Necessity & Relevancy

# (Conditioned) Classifier Decision Problem ((C)CDP)

[HCM + 23]

## (Conditioned) Classifier Decision Problem ((C)CDP)

[HCM<sup>+</sup>23]

- Given  $c \in \mathcal{K}$ , CDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}).(\kappa(\mathbf{x}) = c)$$

## (Conditioned) Classifier Decision Problem ((C)CDP)

[HCM<sup>+</sup>23]

- Given  $c \in \mathcal{K}$ , CDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}).(\kappa(\mathbf{x}) = c)$$

- Given  $\mathcal{S} \subseteq \mathcal{F}$ , instance  $(\mathbf{v}, c)$ , CCDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{i \in \mathcal{S}} (x_i = v_i) \wedge (\kappa(\mathbf{x}) = c)$$

## (Conditioned) Classifier Decision Problem ((C)CDP)

[HCM<sup>+</sup>23]

- Given  $c \in \mathcal{K}$ , CDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}).(\kappa(\mathbf{x}) = c)$$

- Given  $\mathcal{S} \subseteq \mathcal{F}$ , instance  $(\mathbf{v}, c)$ , CCDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{i \in \mathcal{S}} (x_i = v_i) \wedge (\kappa(\mathbf{x}) = c)$$

- Claim:** (C)CDP is in polynomial-time for DTs, decision graphs, monotonic classifiers, among others

## (Conditioned) Classifier Decision Problem ((C)CDP)

[HCM<sup>+</sup>23]

- Given  $c \in \mathcal{K}$ , CDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}).(\kappa(\mathbf{x}) = c)$$

- Given  $\mathcal{S} \subseteq \mathcal{F}$ , instance  $(\mathbf{v}, c)$ , CCDP is to decide whether the following statement holds:

$$\exists(\mathbf{x} \in \mathbb{F}). \bigwedge_{i \in \mathcal{S}} (x_i = v_i) \wedge (\kappa(\mathbf{x}) = c)$$

- Claim:** (C)CDP is in polynomial-time for DTs, decision graphs, monotonic classifiers, among others
- Claim:** (C)CDP is in NP-complete for DLs, RFs, BTs, boolean NNs and BNNs

## Feature necessity

- Consider instance  $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{X})\}$$

$\mathbb{A}$ : encodes the set of all **irreducible rules** for prediction  $c$  given  $\mathbf{v}$

## Feature necessity

- Consider instance  $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{X})\}$$

$\mathbb{A}$ : encodes the set of all **irreducible rules** for prediction  $c$  given  $\mathbf{v}$

- Features common to all AXps in  $\mathbb{A}$  and all CXps in  $\mathbb{C}$ :

$$N_{\mathbb{A}} := \bigcap_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$N_{\mathbb{C}} := \bigcap_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

## Feature necessity

- Consider instance  $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{X})\}$$

$\mathbb{A}$ : encodes the set of all **irreducible rules** for prediction  $c$  given  $\mathbf{v}$

- Features common to all AXps in  $\mathbb{A}$  and all CXps in  $\mathbb{C}$ :

$$N_{\mathbb{A}} := \bigcap_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$N_{\mathbb{C}} := \bigcap_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- $N_{\mathbb{A}}$  and  $N_{\mathbb{C}}$  need not be equal

- $\mathbb{A} = \{\{1\}, \{2, 3\}\}$

## Feature necessity

- Consider instance  $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{X})\}$$

$\mathbb{A}$ : encodes the set of all **irreducible rules** for prediction  $c$  given  $\mathbf{v}$

- Features common to all AXps in  $\mathbb{A}$  and all CXps in  $\mathbb{C}$ :

$$N_{\mathbb{A}} := \bigcap_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$N_{\mathbb{C}} := \bigcap_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- $N_{\mathbb{A}}$  and  $N_{\mathbb{C}}$  need not be equal
  - $\mathbb{A} = \{\{1\}, \{2, 3\}\}$
- A feature  $i$  is **necessary** for abductive explanations (AXp-necessary) if  $i \in N_{\mathbb{A}}$

## Feature necessity

- Consider instance  $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{X})\}$$

$\mathbb{A}$ : encodes the set of all **irreducible rules** for prediction  $c$  given  $\mathbf{v}$

- Features common to all AXps in  $\mathbb{A}$  and all CXps in  $\mathbb{C}$ :

$$N_{\mathbb{A}} := \bigcap_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$N_{\mathbb{C}} := \bigcap_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- $N_{\mathbb{A}}$  and  $N_{\mathbb{C}}$  need not be equal
  - $\mathbb{A} = \{\{1\}, \{2, 3\}\}$
- A feature  $i$  is **necessary** for abductive explanations (AXp-necessary) if  $i \in N_{\mathbb{A}}$
- A feature  $i$  is **necessary** for contrastive explanations (CXp-necessary) if  $i \in N_{\mathbb{C}}$

## More on feature necessity

[HCM<sup>+</sup> 23]

## More on feature necessity

[HCM<sup>+</sup> 23]

- **Claim #01:**  $t \in \mathcal{F}$  is AXp-necessary iff  $\{t\}$  is a CXp

## More on feature necessity

[HCM<sup>+</sup> 23]

- **Claim #01:**  $t \in \mathcal{F}$  is AXp-necessary iff  $\{t\}$  is a CXp
- **Claim #02:**  $t \in \mathcal{F}$  is CXp-necessary iff  $\{t\}$  is a AXp

## More on feature necessity

[HCM<sup>+</sup> 23]

- **Claim #01:**  $t \in \mathcal{F}$  is AXp-necessary iff  $\{t\}$  is a CXp
- **Claim #02:**  $t \in \mathcal{F}$  is CXp-necessary iff  $\{t\}$  is a AXp
- **Claim #03:** CXp-necessity is in P if CCDP is in P
  - I.e. this is the case for DTs, DGs, and monotonic classifiers, among others

## More on feature necessity

[HCM<sup>+</sup> 23]

- **Claim #01:**  $t \in \mathcal{F}$  is AXp-necessary iff  $\{t\}$  is a CXp
- **Claim #02:**  $t \in \mathcal{F}$  is CXp-necessary iff  $\{t\}$  is a AXp
- **Claim #03:** CXp-necessity is in P if CCDP is in P
  - I.e. this is the case for DTs, DGs, and monotonic classifiers, among others
- **Claim #04:** AXp-necessity of  $t \in \mathcal{F}$  is in P if  $t$  has a domain size which is polynomially-bounded on instance size

## More on feature necessity

[HCM<sup>+</sup> 23]

- **Claim #01:**  $t \in \mathcal{F}$  is AXp-necessary iff  $\{t\}$  is a CXp
- **Claim #02:**  $t \in \mathcal{F}$  is CXp-necessary iff  $\{t\}$  is a AXp
- **Claim #03:** CXp-necessity is in P if CCDP is in P
  - I.e. this is the case for DTs, DGs, and monotonic classifiers, among others
- **Claim #04:** AXp-necessity of  $t \in \mathcal{F}$  is in P if  $t$  has a domain size which is polynomially-bounded on instance size
  - **This holds for any classifier!**

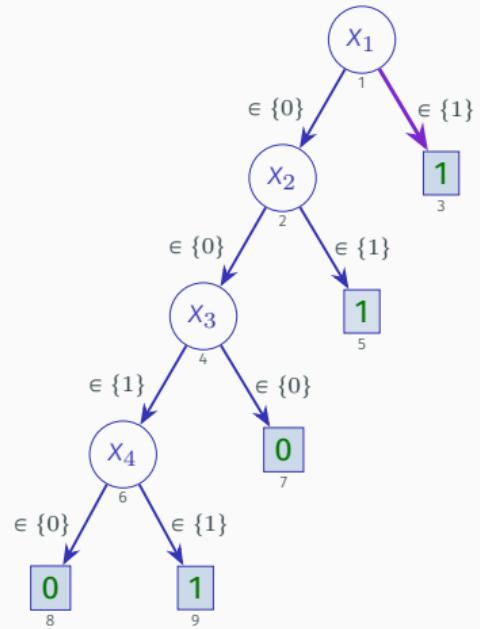
## More on feature necessity

[HCM<sup>+</sup> 23]

- **Claim #01:**  $t \in \mathcal{F}$  is AXp-necessary iff  $\{t\}$  is a CXp
- **Claim #02:**  $t \in \mathcal{F}$  is CXp-necessary iff  $\{t\}$  is a AXp
- **Claim #03:** CXp-necessity is in P if CCDP is in P
  - I.e. this is the case for DTs, DGs, and monotonic classifiers, among others
- **Claim #04:** AXp-necessity of  $t \in \mathcal{F}$  is in P if  $t$  has a domain size which is polynomially-bounded on instance size
  - **This holds for any classifier!**
  - Let  $\mathbf{u}$  be obtained from  $\mathbf{v}$  by replacing the constant  $v_t$  by some variable  $u_t \in \mathcal{D}_t$
  - Feature  $t$  is AXp-necessary if  $\kappa(\mathbf{u}) \neq \kappa(\mathbf{v})$  for some value  $u_t \in \mathcal{D}_t$

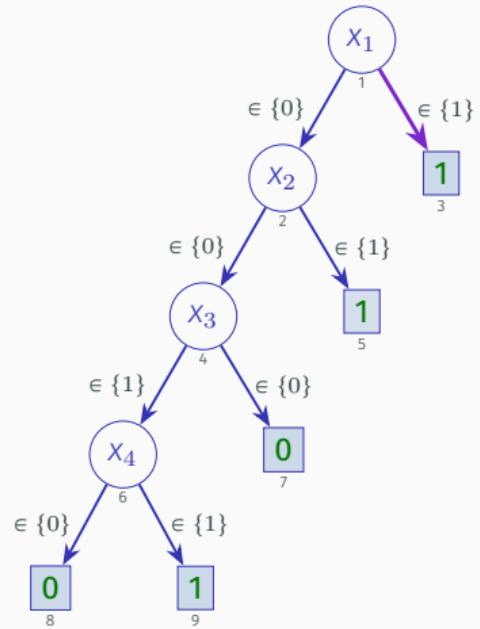
## An example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$



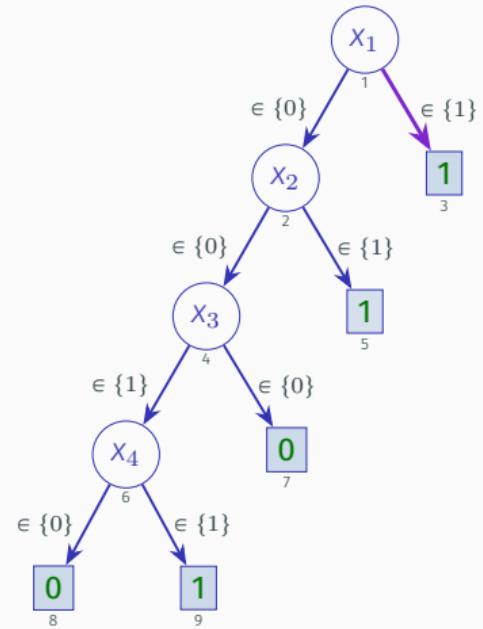
## An example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?



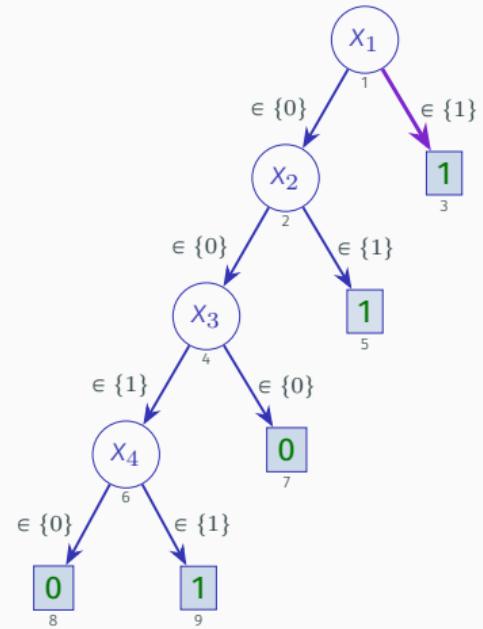
## An example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?



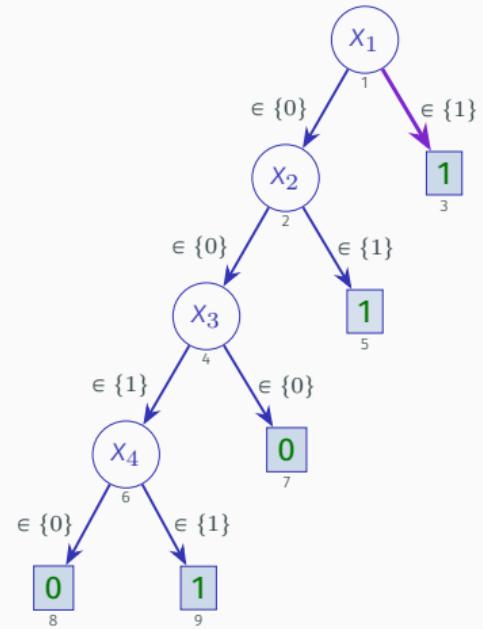
# An example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **Yes!** Thus, feature 1 is AXp-necessary



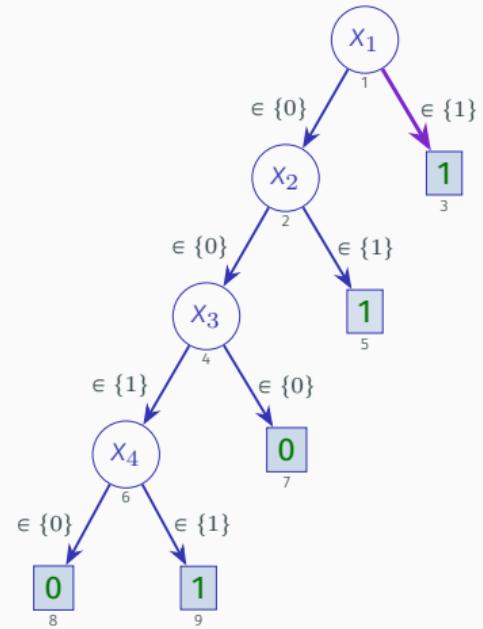
# An example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - Yes! Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?



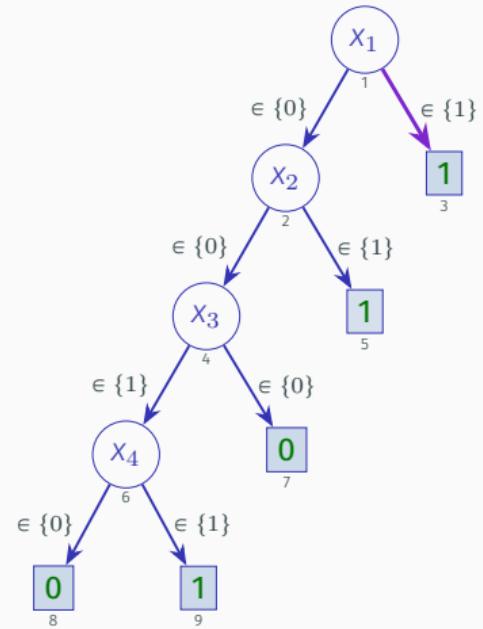
## An example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - Yes! Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?
  - Does there exist  $u_3$ , such that  $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$ ?



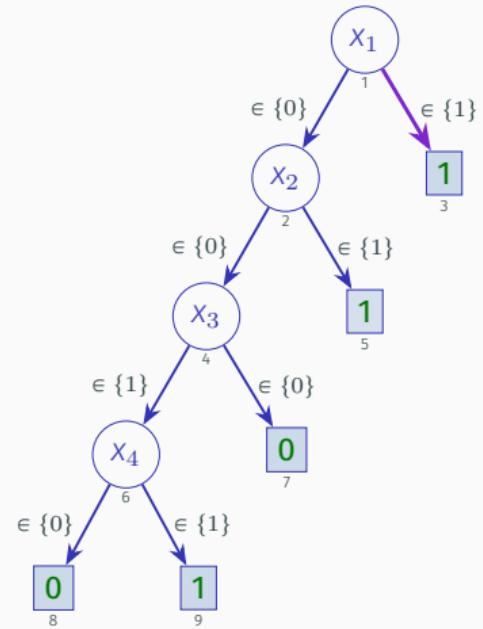
# An example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **Yes!** Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?
  - Does there exist  $u_3$ , such that  $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **No!** Thus, feature 3 is **not** AXp-necessary



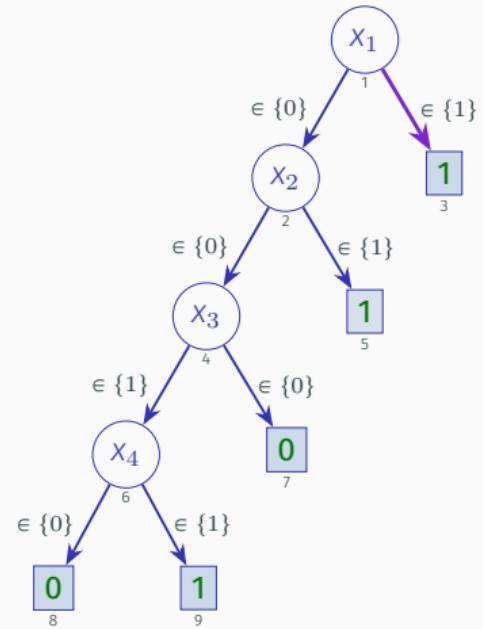
# An example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **Yes!** Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?
  - Does there exist  $u_3$ , such that  $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **No!** Thus, feature 3 is **not** AXp-necessary
- Confirmation:
  - CXps:
  - AXps:



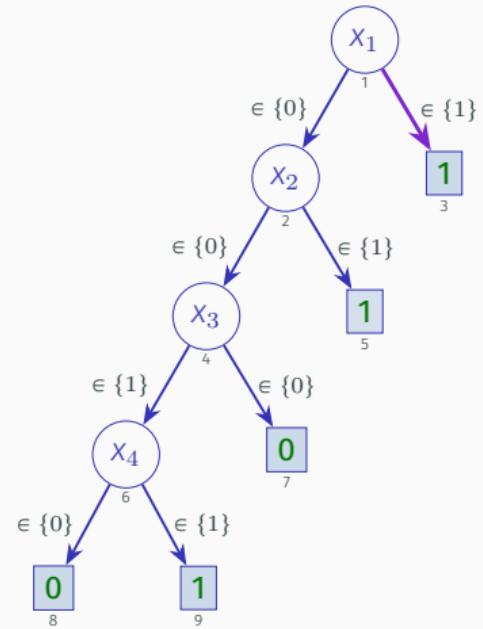
# An example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **Yes!** Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?
  - Does there exist  $u_3$ , such that  $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **No!** Thus, feature 3 is **not** AXp-necessary
- Confirmation:
  - CXps:  $\{\{1\}, \{2\}, \{3, 4\}\}$
  - AXps:



# An example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **Yes!** Thus, feature 1 is AXp-necessary
- Is feature 3 AXp-necessary?
  - Does there exist  $u_3$ , such that  $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **No!** Thus, feature 3 is **not** AXp-necessary
- Confirmation:
  - CXps:  $\{\{1\}, \{2\}, \{3, 4\}\}$
  - AXps:  $\{\{1, 2, 3\}, \{1, 2, 4\}\}$



## Feature relevancy

- Consider instance  $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{X})\}$$

## Feature relevancy

- Consider instance  $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{X})\}$$

- Features occurring in some AXp in  $\mathbb{A}$  and in some CXp in  $\mathbb{C}$ :

$$F_{\mathbb{A}} := \bigcup_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$F_{\mathbb{C}} := \bigcup_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

## Feature relevancy

- Consider instance  $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{X})\}$$

- Features occurring in some AXp in  $\mathbb{A}$  and in some CXp in  $\mathbb{C}$ :

$$F_{\mathbb{A}} := \bigcup_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$F_{\mathbb{C}} := \bigcup_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- **Claim:**  $F_{\mathbb{A}} = F_{\mathbb{C}}$ 
  - I.e. a feature exists in some AXp iff it exists in some CXp

## Feature relevancy

- Consider instance  $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{X})\}$$

- Features occurring in some AXp in  $\mathbb{A}$  and in some CXp in  $\mathbb{C}$ :

$$F_{\mathbb{A}} := \bigcup_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$F_{\mathbb{C}} := \bigcup_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- **Claim:**  $F_{\mathbb{A}} = F_{\mathbb{C}}$ 
  - I.e. a feature exists in some AXp iff it exists in some CXp
- A feature  $i \in \mathcal{F}$  is **relevant** if  $i \in F_{\mathbb{A}}$  (and so, if  $i \in F_{\mathbb{C}}$ )
  - A feature is **relevant** if it is included in some AXp (or CXp)

## Feature relevancy

- Consider instance  $(\mathbf{v}, c)$
- Sets of all AXp's & CXp's:

$$\mathbb{A} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{X})\}$$

$$\mathbb{C} := \{\mathcal{X} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{X})\}$$

- Features occurring in some AXp in  $\mathbb{A}$  and in some CXp in  $\mathbb{C}$ :

$$F_{\mathbb{A}} := \bigcup_{\mathcal{X} \in \mathbb{A}} \mathcal{X}$$

$$F_{\mathbb{C}} := \bigcup_{\mathcal{X} \in \mathbb{C}} \mathcal{X}$$

- **Claim:**  $F_{\mathbb{A}} = F_{\mathbb{C}}$ 
  - I.e. a feature exists in some AXp iff it exists in some CXp
- A feature  $i \in \mathcal{F}$  is **relevant** if  $i \in F_{\mathbb{A}}$  (and so, if  $i \in F_{\mathbb{C}}$ )
  - A feature is **relevant** if it is included in some AXp (or CXp)
- A feature  $i \in \mathcal{F}$  is **irrelevant** if  $i \notin F_{\mathbb{A}}$  (and so, if  $i \notin F_{\mathbb{C}}$ )
  - A feature is **irrelevant** if it is **not** included in **any** AXp (or CXp)

## An example

- Consider the classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- $(\mathbf{v}, c) = ((0, 0, 0, 1), 1)$

## An example

- Consider the classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- $(\mathbf{v}, c) = ((0, 0, 0, 1), 1)$
  - $\mathbb{A} = \{\{4\}\} = \mathbb{C}$
  - Why?
    - If 4 fixed, then prediction *must* be 1
    - If 4 is allowed to change, then prediction changes
    - Values of 1, 2, 3 not used to fix/change the prediction

## An example

- Consider the classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- ( $\mathbf{v}, c$ ) =  $((0, 0, 0, 1), 1)$
- $\mathbb{A} = \{\{4\}\} = \mathbb{C}$ 
  - Why?
    - If 4 fixed, then prediction *must* be 1
    - If 4 is allowed to change, then prediction changes
    - Values of 1, 2, 3 not used to fix/change the prediction
- Feature 4 is **relevant**, since it is included in one (and the only) AXp/CXp

## An example

- Consider the classifier:

$$\kappa(x_1, x_2, x_3, x_4) = \bigvee_{i=1}^4 x_i$$

- ( $\mathbf{v}, c$ ) =  $((0, 0, 0, 1), 1)$
- $\mathbb{A} = \{\{4\}\} = \mathbb{C}$
- Why?
  - If 4 fixed, then prediction *must* be 1
  - If 4 is allowed to change, then prediction changes
  - Values of 1, 2, 3 not used to fix/change the prediction
- Feature 4 is **relevant**, since it is included in one (and the only) AXp/CXp
- Features 1, 2, 3 are **irrelevant**, since there are not included in any AXp/CXp
- Obs: **irrelevant features are absolutely unimportant!**

We could propose some other explanation by adding features 1, 2 or 3 to AXp  $\{4\}$ , but prediction would remain unchanged for **any** value assigned to those features

- And we aim for **irreducibility** (**Occam's razor is a mainstay of AI/ML**)

## Deciding feature relevancy

## Deciding feature relevancy

- Deciding feature relevancy is in  $\Sigma_2^P$  – intuition:

## Deciding feature relevancy

- Deciding feature relevancy is in  $\Sigma_2^P$  – intuition:
  - Pick a set of features  $\mathcal{P}$  containing  $t$  (i.e. existential quantification), such that,

## Deciding feature relevancy

- Deciding feature relevancy is in  $\Sigma_2^P$  – intuition:
  - Pick a set of features  $\mathcal{P}$  containing  $t$  (i.e. existential quantification), such that,
    - $\mathcal{P}$  is a WAXp (i.e. universal quantification)

## Deciding feature relevancy

- Deciding feature relevancy is in  $\Sigma_2^P$  – intuition:
  - Pick a set of features  $\mathcal{P}$  containing  $t$  (i.e. existential quantification), such that,
    - $\mathcal{P}$  is a WAXp (i.e. universal quantification)
    - $\mathcal{P} \setminus \{t\}$  is a **not** a WAXp (i.e. universal quantification again)

## Deciding feature relevancy

- Deciding feature relevancy is in  $\Sigma_2^P$  – intuition:
  - Pick a set of features  $\mathcal{P}$  containing  $t$  (i.e. existential quantification), such that,
    - $\mathcal{P}$  is a WAXp (i.e. universal quantification)
    - $\mathcal{P} \setminus \{t\}$  is a **not** a WAXp (i.e. universal quantification again)
    - Thus, we can decide feature relevancy with  $\exists\forall$  alternation

## Deciding feature relevancy

- Deciding feature relevancy is in  $\Sigma_2^P$  – intuition:
  - Pick a set of features  $\mathcal{P}$  containing  $t$  (i.e. existential quantification), such that,
    - $\mathcal{P}$  is a WAXp (i.e. universal quantification)
    - $\mathcal{P} \setminus \{t\}$  is a **not** a WAXp (i.e. universal quantification again)
    - Thus, we can decide feature relevancy with  $\exists\forall$  alternation
  - For DTs, deciding feature relevancy is in P; **Why?**

## Deciding feature relevancy

- Deciding feature relevancy is in  $\Sigma_2^P$  – intuition:
  - Pick a set of features  $\mathcal{P}$  containing  $t$  (i.e. existential quantification), such that,
    - $\mathcal{P}$  is a WAXp (i.e. universal quantification)
    - $\mathcal{P} \setminus \{t\}$  is a **not** a WAXp (i.e. universal quantification again)
    - Thus, we can decide feature relevancy with  $\exists \forall$  alternation
  - For DTs, deciding feature relevancy is in P; **Why?**
    - **Obs:** We know that  $F_A = F_C$ ; thus

## Deciding feature relevancy

- Deciding feature relevancy is in  $\Sigma_2^P$  – intuition:
  - Pick a set of features  $\mathcal{P}$  containing  $t$  (i.e. existential quantification), such that,
    - $\mathcal{P}$  is a WAXp (i.e. universal quantification)
    - $\mathcal{P} \setminus \{t\}$  is a **not** a WAXp (i.e. universal quantification again)
    - Thus, we can decide feature relevancy with  $\exists\forall$  alternation
- For DTs, deciding feature relevancy is in P; **Why?**
  - **Obs:** We know that  $F_A = F_C$ ; thus
    - Computing all CXps in polynomial-time decides feature relevancy

# Deciding feature relevancy

- Deciding feature relevancy is in  $\Sigma_2^P$  – intuition:
  - Pick a set of features  $\mathcal{P}$  containing  $t$  (i.e. existential quantification), such that,
    - $\mathcal{P}$  is a WAXp (i.e. universal quantification)
    - $\mathcal{P} \setminus \{t\}$  is a **not** a WAXp (i.e. universal quantification again)
    - Thus, we can decide feature relevancy with  $\exists\forall$  alternation
- For DTs, deciding feature relevancy is in P; **Why?**
  - **Obs:** We know that  $F_A = F_C$ ; thus
    - Computing all CXps in polynomial-time decides feature relevancy
- General case: best solution is to exploit **abstraction refinement**

## Abstraction refinement for feature relevancy

- Claim:  $\mathcal{X} \subseteq \mathcal{F}$  and  $t \in \mathcal{X}$ . If  $\text{WAXp}(\mathcal{X})$  holds and  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  does not hold, then **any AXp**  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  must contain feature  $t$ .

## Abstraction refinement for feature relevancy

- Claim:  $\mathcal{X} \subseteq \mathcal{F}$  and  $t \in \mathcal{X}$ . If  $\text{WAXp}(\mathcal{X})$  holds and  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  does not hold, then **any**  $\text{AXp } \mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  must contain feature  $t$ .

Proof:

## Abstraction refinement for feature relevancy

- Claim:  $\mathcal{X} \subseteq \mathcal{F}$  and  $t \in \mathcal{X}$ . If WAXp( $\mathcal{X}$ ) holds and WAXp( $\mathcal{X} \setminus \{t\}$ ) does not hold, then any AXp  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  must contain feature  $t$ .

Proof:

- Let  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  be an AXp such that  $t \notin \mathcal{Z}$ .

## Abstraction refinement for feature relevancy

- Claim:  $\mathcal{X} \subseteq \mathcal{F}$  and  $t \in \mathcal{X}$ . If  $\text{WAXp}(\mathcal{X})$  holds and  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  does not hold, then **any**  $\text{AXp } \mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  must contain feature  $t$ .

Proof:

- Let  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  be an  $\text{AXp}$  such that  $t \notin \mathcal{Z}$ .
- Then  $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$ .

## Abstraction refinement for feature relevancy

- Claim:  $\mathcal{X} \subseteq \mathcal{F}$  and  $t \in \mathcal{X}$ . If  $\text{WAXp}(\mathcal{X})$  holds and  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  does not hold, then **any AXp**  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  must contain feature  $t$ .

Proof:

- Let  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  be an AXp such that  $t \notin \mathcal{Z}$ .
- Then  $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$ .
- But then, by monotonicity,  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  must hold (i.e. any superset of  $\mathcal{Z}$  is a weak AXp); hence a contradiction.

## Abstraction refinement for feature relevancy

- Claim:  $\mathcal{X} \subseteq \mathcal{F}$  and  $t \in \mathcal{X}$ . If  $\text{WAXp}(\mathcal{X})$  holds and  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  does not hold, then **any AXp**  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  must contain feature  $t$ .

Proof:

- Let  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  be an AXp such that  $t \notin \mathcal{Z}$ .
- Then  $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$ .
- But then, by monotonicity,  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  must hold (i.e. any superset of  $\mathcal{Z}$  is a weak AXp); hence a contradiction.

- Approach:

# Abstraction refinement for feature relevancy

- Claim:  $\mathcal{X} \subseteq \mathcal{F}$  and  $t \in \mathcal{X}$ . If  $\text{WAXp}(\mathcal{X})$  holds and  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  does not hold, then **any AXp**  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  must contain feature  $t$ .

Proof:

- Let  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  be an AXp such that  $t \notin \mathcal{Z}$ .
- Then  $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$ .
- But then, by monotonicity,  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  must hold (i.e. any superset of  $\mathcal{Z}$  is a weak AXp); hence a contradiction.

- Approach:

- Repeatedly guess weak WAXp candidates  $\mathcal{X}$ , with  $t \in \mathcal{X}$

[e.g. use SAT oracle]

## Abstraction refinement for feature relevancy

- Claim:  $\mathcal{X} \subseteq \mathcal{F}$  and  $t \in \mathcal{X}$ . If  $\text{WAXp}(\mathcal{X})$  holds and  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  does not hold, then any AXp  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  must contain feature  $t$ .

Proof:

- Let  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  be an AXp such that  $t \notin \mathcal{Z}$ .
- Then  $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$ .
- But then, by monotonicity,  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  must hold (i.e. any superset of  $\mathcal{Z}$  is a weak AXp); hence a contradiction.

- Approach:

- Repeatedly guess weak WAXp candidates  $\mathcal{X}$ , with  $t \in \mathcal{X}$  [e.g. use SAT oracle]
- Check that WAXp condition holds for  $\mathcal{X}$ :  $\text{WAXp}(\mathcal{X})$ ; and [e.g. use WAXp oracle]

# Abstraction refinement for feature relevancy

- Claim:  $\mathcal{X} \subseteq \mathcal{F}$  and  $t \in \mathcal{X}$ . If  $\text{WAXp}(\mathcal{X})$  holds and  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  does not hold, then any AXp  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  must contain feature  $t$ .

Proof:

- Let  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  be an AXp such that  $t \notin \mathcal{Z}$ .
- Then  $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$ .
- But then, by monotonicity,  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  must hold (i.e. any superset of  $\mathcal{Z}$  is a weak AXp); hence a contradiction.

- Approach:

- Repeatedly guess weak WAXp candidates  $\mathcal{X}$ , with  $t \in \mathcal{X}$
- Check that WAXp condition holds for  $\mathcal{X}$ :  $\text{WAXp}(\mathcal{X})$  ; and
- Check that WAXp condition fails for  $\mathcal{X} \setminus \{t\}$ :  $\neg \text{WAXp}(\mathcal{X} \setminus \{t\})$

[e.g. use SAT oracle]

[e.g. use WAXp oracle]

[e.g. use WAXp oracle]

# Abstraction refinement for feature relevancy

- Claim:  $\mathcal{X} \subseteq \mathcal{F}$  and  $t \in \mathcal{X}$ . If  $\text{WAXp}(\mathcal{X})$  holds and  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  does not hold, then any AXp  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  must contain feature  $t$ .

Proof:

- Let  $\mathcal{Z} \subseteq \mathcal{X} \subseteq \mathcal{F}$  be an AXp such that  $t \notin \mathcal{Z}$ .
- Then  $\mathcal{Z} \subseteq \mathcal{X} \setminus \{t\}$ .
- But then, by monotonicity,  $\text{WAXp}(\mathcal{X} \setminus \{t\})$  must hold (i.e. any superset of  $\mathcal{Z}$  is a weak AXp); hence a contradiction.

- Approach:

- Repeatedly guess weak WAXp candidates  $\mathcal{X}$ , with  $t \in \mathcal{X}$
- Check that WAXp condition holds for  $\mathcal{X}$ :  $\text{WAXp}(\mathcal{X})$  ; and
- Check that WAXp condition fails for  $\mathcal{X} \setminus \{t\}$ :  $\neg \text{WAXp}(\mathcal{X} \setminus \{t\})$
- Block counterexamples in both cases

[e.g. use SAT oracle]

[e.g. use WAXp oracle]

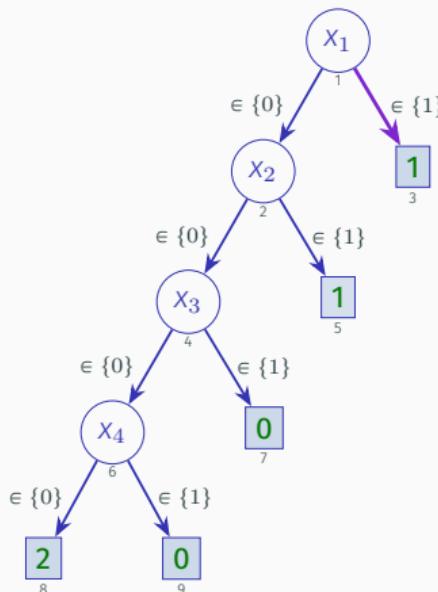
[e.g. use WAXp oracle]

# A general abstraction refinement algorithm

Input: Instance  $\mathbf{v}$ , Target Feature  $t$ ; Feature Set  $\mathcal{F}$ , Classifier  $\kappa$

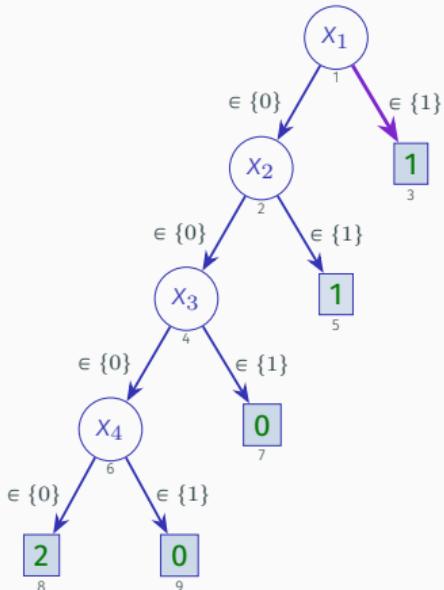
```
1: function FRPCGR( $\mathbf{v}, t; \mathcal{F}, \kappa$ )
2:    $\mathcal{H} \leftarrow \emptyset$                                  $\triangleright \mathcal{H}$  overapproximates the subsets of  $\mathcal{F}$  that do not contain an AXp containing  $t$ 
3:   repeat
4:     ( $\text{outc}, s$ )  $\leftarrow \text{SAT}(\mathcal{H}, s_t)$            $\triangleright$  Use SAT oracle to pick candidate WAXp containing  $t$ 
5:     if  $\text{outc} = \text{true}$  then
6:        $\mathcal{P} \leftarrow \{i \in \mathcal{F} \mid s_i = 1\}$             $\triangleright$  Set  $\mathcal{P}$  is the candidate WAXp, and  $t \in \mathcal{P}$ 
7:        $\mathcal{D} \leftarrow \{i \in \mathcal{F} \mid s_i = 0\}$             $\triangleright$  Set  $\mathcal{D}$  contains the features not included in  $\mathcal{P}$ 
8:       if  $\neg \text{WAXp}(\mathcal{P})$  then                   $\triangleright$  Is  $\mathcal{P}$  not a WAXp?
9:          $\mathcal{H} \leftarrow \mathcal{H} \cup \text{newPosCl}(\mathcal{D}; t, \kappa)$      $\triangleright \mathcal{P}$  is not a WAXp; must pick some non-picked feature
10:        else                                          $\triangleright \mathcal{P}$  is a WAXp
11:          if  $\neg \text{WAXp}(\mathcal{P} \setminus \{t\})$  then       $\triangleright \mathcal{P}$  without  $t$  not a WAXp?
12:            reportWeakAXp( $\mathcal{P}$ )                     $\triangleright$  Feature  $t$  is included in any AXp  $\mathcal{X} \subseteq \mathcal{P}$ 
13:            return true
14:           $\mathcal{H} \leftarrow \mathcal{H} \cup \text{newNegCl}(\mathcal{P}; t, \kappa)$      $\triangleright \text{WAXp}(\mathcal{P} \setminus \{t\})$  holds; some feature in  $\mathcal{P}$  must not be picked
15:        until  $\text{outc} = \text{false}$ 
16:      return false                                      $\triangleright$  If  $\mathcal{H}$  becomes inconsistent, then there is no AXp that contains  $t$ 
```

## An example: feature relevancy for DT, using abstraction refinement



- Instance:  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Is  $t = 1$  relevant?

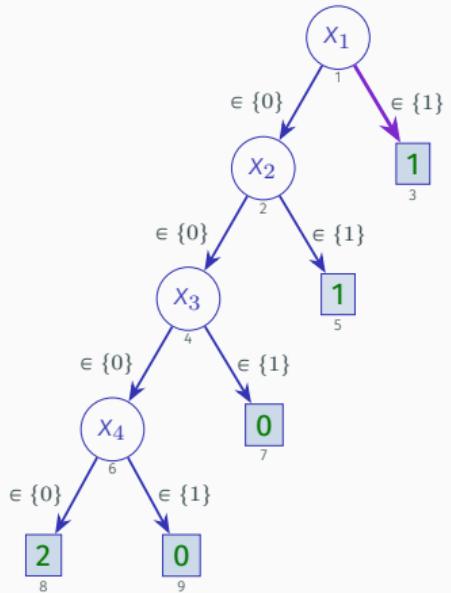
## An example: feature relevancy for DT, using abstraction refinement



- Instance:  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Is  $t = 1$  relevant?

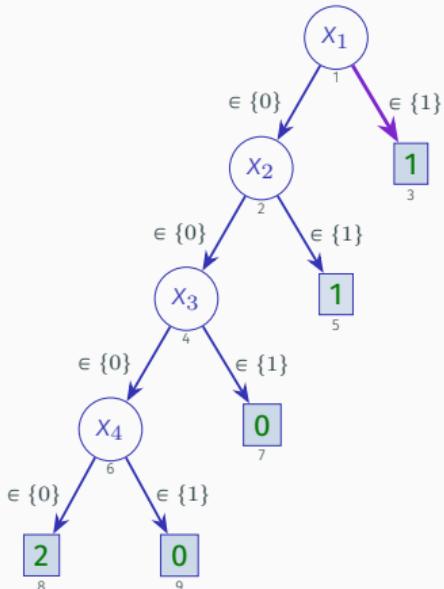
$t = 1$					
s	$\mathcal{P}$	WAXp( $\mathcal{P}$ )	WAXp( $\mathcal{P} \setminus \{t\}$ )	Return?	Clause
(1, 1, 1, 1)	{1, 2, 3, 4}	✓	✓	—	$(\neg u_2 \vee \neg u_3 \vee \neg u_4)$
(1, 1, 0, 1)	{1, 2, 4}	✓	✓	—	$(\neg u_2 \vee \neg u_4)$
(1, 1, 0, 0)	{1, 2}	✓	✓	—	$(\neg u_2)$
(1, 0, 0, 0)	{1}	✓	✗	true	—

## Another example



- Instance:  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Is  $t = 4$  relevant?

## Another example



- Instance:  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Is  $t = 4$  relevant?

$t = 4$					
$\mathbf{s}$	$\mathcal{P}$	WAXp( $\mathcal{P}$ )	WAXp( $\mathcal{P} \setminus \{t\}$ )	Return?	Clause
(1, 1, 1, 1)	{1, 2, 3, 4}	✓	✓	—	$(\neg u_1 \vee \neg u_2 \vee \neg u_3)$
(1, 1, 0, 1)	{1, 2, 4}	✓	✓	—	$(\neg u_1 \vee \neg u_2)$
(1, 0, 0, 1)	{1, 4}	✓	✓	—	$(\neg u_1)$
(0, 1, 0, 1)	{2, 4}	✓	✓	—	$(\neg u_2)$
(0, 0, 0, 1)	{4}	✗	—	—	$(u_1 \vee u_2 \vee u_3)$
(0, 0, 1, 1)	{3, 4}	✗	—	—	$(u_1 \vee u_2)$
[outc = false]	—	—	—	false	—

# Questions?

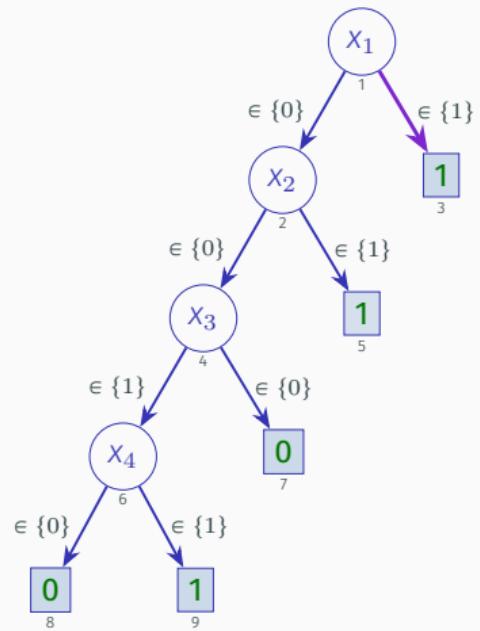
# Lecture 04

## Recapitulate third lecture

- Logic encoding for explaining DLs
  - And status of (in)tractability in logic-based XAI
- Query: enumeration of explanations
- Query: feature necessity, AXp & CXp
- Query: feature relevancy

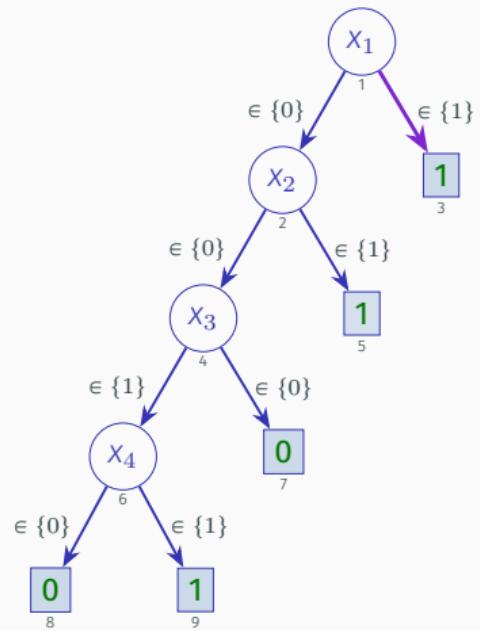
## Recap example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$



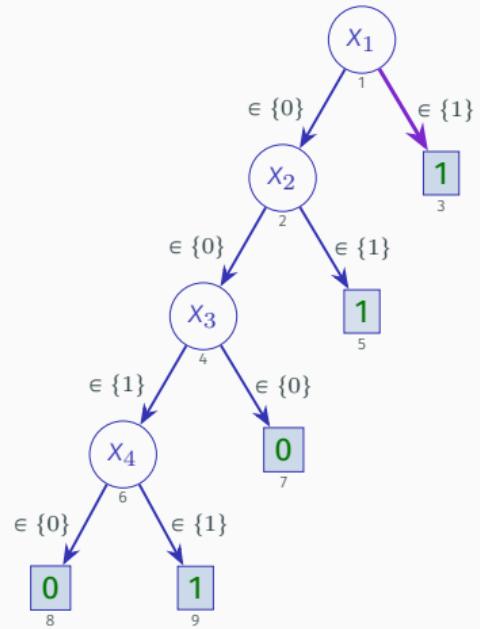
## Recap example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?



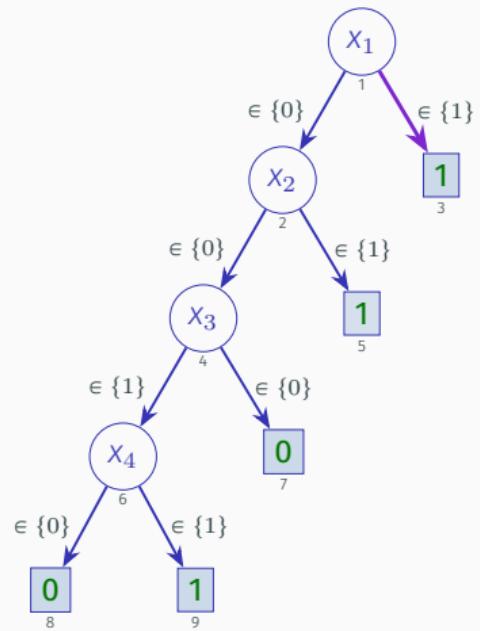
## Recap example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?



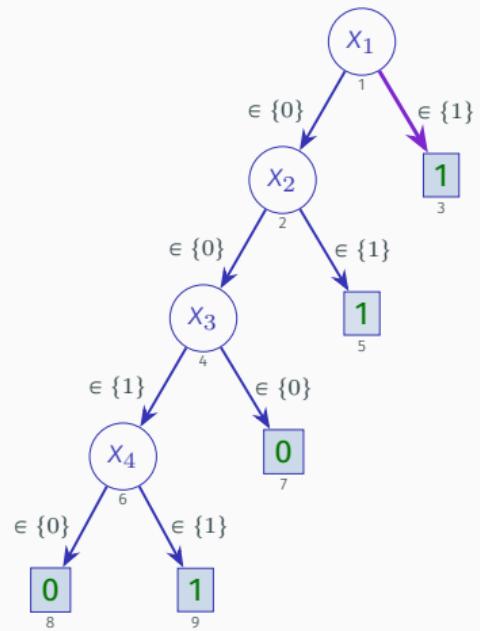
## Recap example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - Yes!** Thus, feature 1 is AXp-necessary (i.e. singleton CXp)



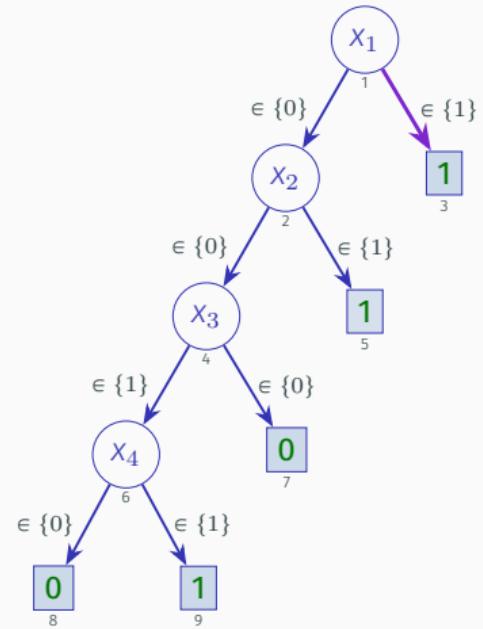
## Recap example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - Yes!** Thus, feature 1 is AXp-necessary (i.e. singleton CXp)
- Is feature 3 AXp-necessary?



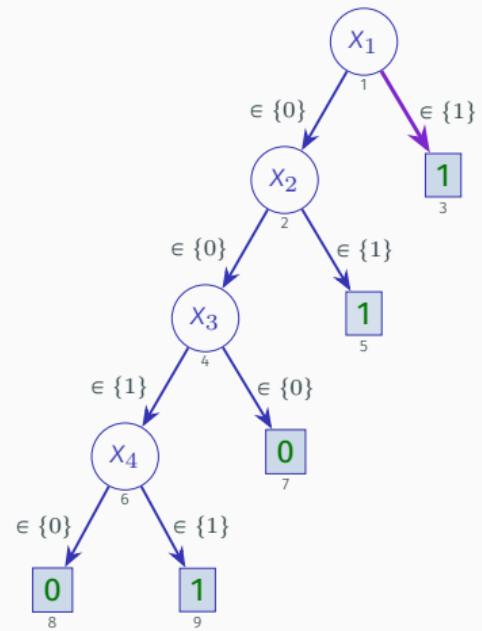
## Recap example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - Yes!** Thus, feature 1 is AXp-necessary (i.e. singleton CXp)
- Is feature 3 AXp-necessary?
  - Does there exist  $u_3$ , such that  $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$ ?



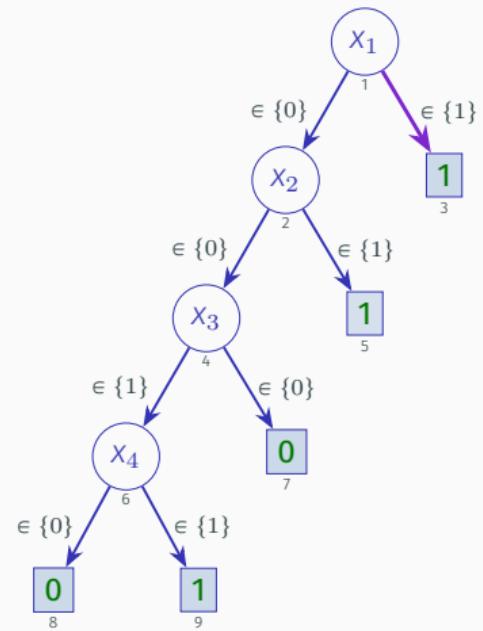
## Recap example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - Yes!** Thus, feature 1 is AXp-necessary (i.e. singleton CXp)
- Is feature 3 AXp-necessary?
  - Does there exist  $u_3$ , such that  $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - No!** Thus, feature 3 is **not** AXp-necessary



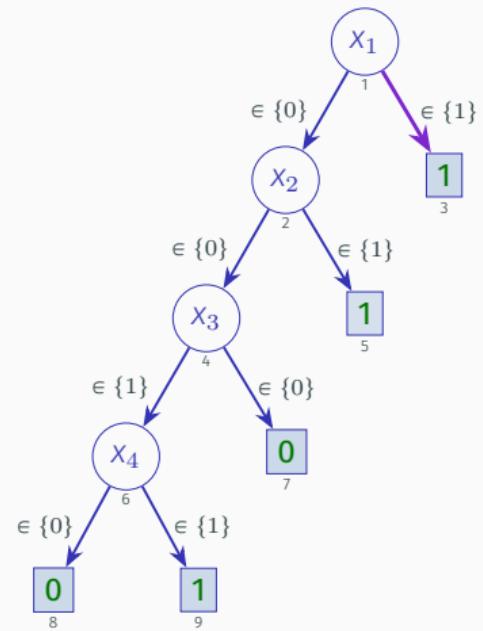
## Recap example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **Yes!** Thus, feature 1 is AXp-necessary (i.e. singleton CXp)
- Is feature 3 AXp-necessary?
  - Does there exist  $u_3$ , such that  $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **No!** Thus, feature 3 is **not** AXp-necessary
- Are there CXp-necessary features?
  - **No!** There are no singleton AXps



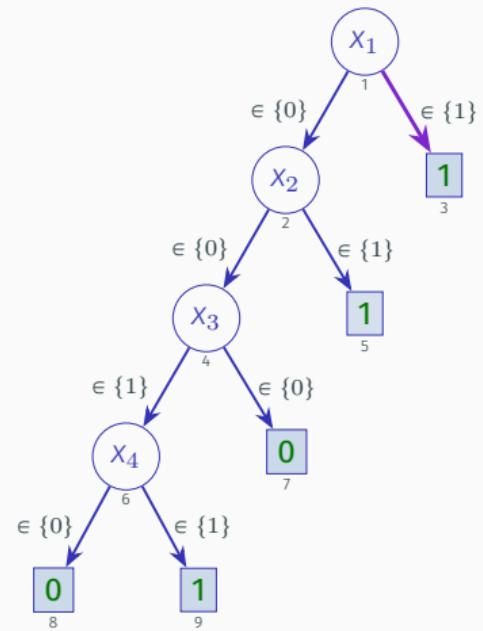
## Recap example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **Yes!** Thus, feature 1 is AXp-necessary (i.e. singleton CXp)
- Is feature 3 AXp-necessary?
  - Does there exist  $u_3$ , such that  $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **No!** Thus, feature 3 is **not** AXp-necessary
- Are there CXp-necessary features?
  - **No!** There are no singleton AXps
- Confirmation:



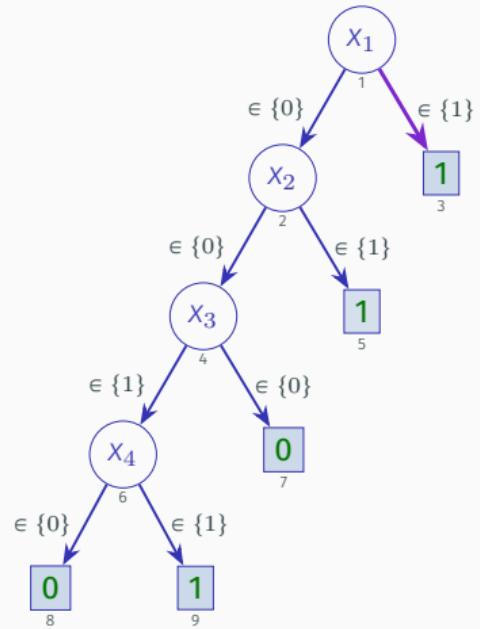
## Recap example

- Instance  $(\mathbf{v}, c) = ((0, 0, 0, 0), 0)$
- Is feature 1 AXp-necessary?
  - Does there exist  $u_1$ , such that  $\kappa(u_1, 0, 0, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **Yes!** Thus, feature 1 is AXp-necessary (i.e. singleton CXp)
- Is feature 3 AXp-necessary?
  - Does there exist  $u_3$ , such that  $\kappa(0, 0, u_3, 0) \neq \kappa(0, 0, 0, 0)$ ?
  - **No!** Thus, feature 3 is **not** AXp-necessary
- Are there CXp-necessary features?
  - **No!** There are no singleton AXps
- Confirmation:
  - CXps:  $\{\{1\}, \{2\}, \{3, 4\}\}$  (2 is also AXp-necessary)
  - AXps:  $\{\{1, 2, 3\}, \{1, 2, 4\}\}$



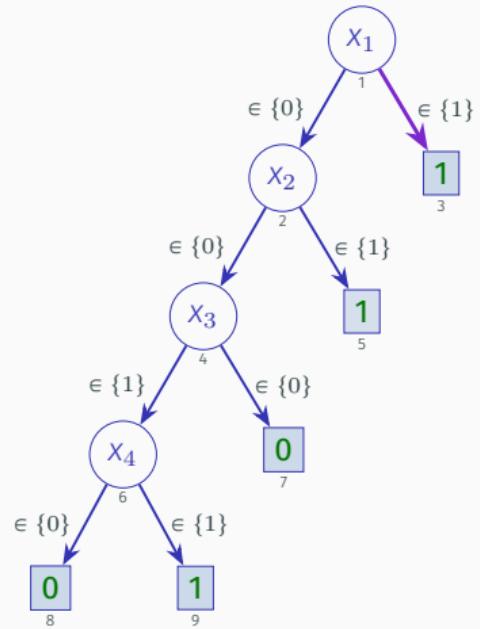
## Recap example – a different instance

- Instance  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$



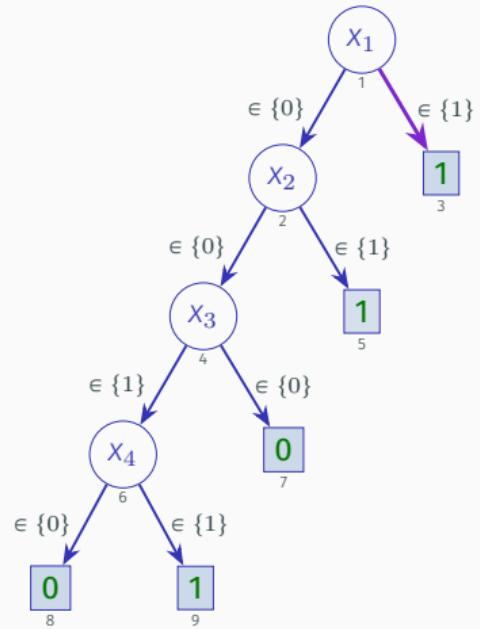
## Recap example – a different instance

- Instance  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Are there CXp-necessary features?



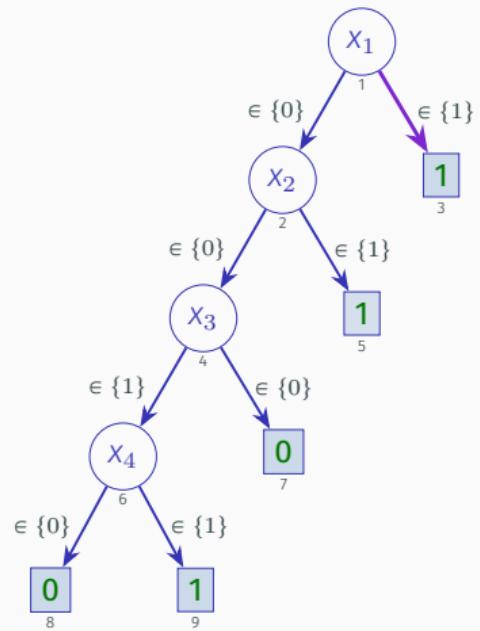
## Recap example – a different instance

- Instance  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Are there CXp-necessary features?
  - **Yes!** Features 1 and 2 (i.e. singleton AXps)



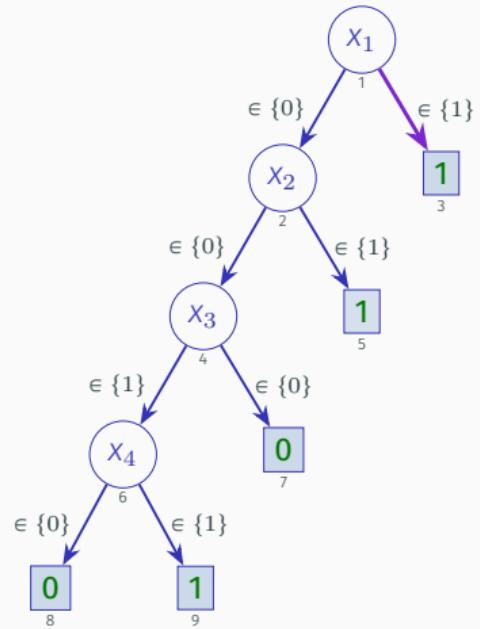
## Recap example – a different instance

- Instance  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Are there CXp-necessary features?
  - **Yes!** Features 1 and 2 (i.e. singleton AXps)
- Are there AXp-necessary features?



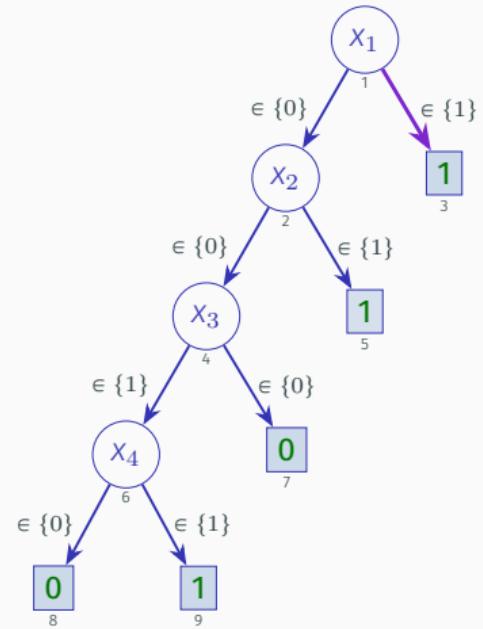
## Recap example – a different instance

- Instance  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Are there CXp-necessary features?
  - **Yes!** Features 1 and 2 (i.e. singleton AXps)
- Are there AXp-necessary features?
  - **No!** There are no singleton CXps



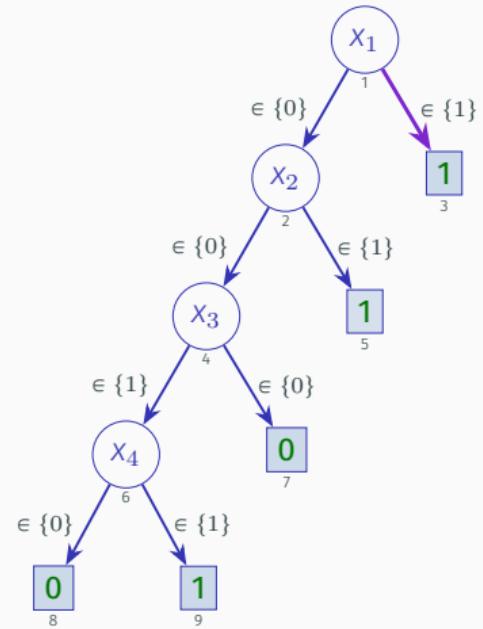
## Recap example – a different instance

- Instance  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Are there CXp-necessary features?
  - **Yes!** Features 1 and 2 (i.e. singleton AXps)
- Are there AXp-necessary features?
  - **No!** There are no singleton CXps
- Confirmation:



## Recap example – a different instance

- Instance  $(\mathbf{v}, c) = ((1, 1, 1, 1), 1)$
- Are there CXp-necessary features?
  - **Yes!** Features 1 and 2 (i.e. singleton AXps)
- Are there AXp-necessary features?
  - **No!** There are no singleton CXps
- Confirmation:
  - AXps:  $\{\{1\}, \{2\}, \{3, 4\}\}$
  - CXps:  $\{\{1, 2, 3\}, \{1, 2, 4\}\}$



## Another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}$ ;  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5$ ;  $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$

## Another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}$ ;  $\mathcal{D}_i = \{0, 1\}$ ,  $i = 1, \dots, 5$ ;  $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must consider only  $x_1 = 1$

## Another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}$ ;  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5$ ;  $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must consider only  $x_1 = 1$ 
  - Hint:** Can construct restricted truth-table

## Another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}$ ;  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5$ ;  $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must consider only  $x_1 = 1$ 
  - Hint:** Can construct restricted truth-table
- All AXps:

## Another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}$ ;  $\mathcal{D}_i = \{0, 1\}$ ,  $i = 1, \dots, 5$ ;  $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must consider only  $x_1 = 1$ 
  - Hint:** Can construct restricted truth-table
- All AXps:  $\{\{1, 2\}, \{1, 3\}\}$
- All CXps:

## Another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}$ ;  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5$ ;  $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must consider only  $x_1 = 1$ 
  - Hint:** Can construct restricted truth-table
- All AXps:  $\{\{1, 2\}, \{1, 3\}\}$
- All CXps:  $\{\{1\}, \{2, 3\}\}$
- AXp-necessary:

## Another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must consider only  $x_1 = 1$ 
  - Hint:** Can construct restricted truth-table
- All AXps:  $\{\{1, 2\}, \{1, 3\}\}$
- All CXps:  $\{\{1\}, \{2, 3\}\}$
- AXp-necessary:  $\{1\}$  (singleton CXp)
- CXp-necessary:

## Another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must consider only  $x_1 = 1$ 
  - Hint:** Can construct restricted truth-table
- All AXps:  $\{\{1, 2\}, \{1, 3\}\}$
- All CXps:  $\{\{1\}, \{2, 3\}\}$
- AXp-necessary:  $\{1\}$  (singleton CXp)
- CXp-necessary:  $\emptyset$
- Relevant:

## Another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must consider only  $x_1 = 1$

· **Hint:** Can construct restricted truth-table

- All AXps:  $\{\{1, 2\}, \{1, 3\}\}$
- All CXps:  $\{\{1\}, \{2, 3\}\}$
- AXp-necessary:  $\{1\}$  (singleton CXp)
- CXp-necessary:  $\emptyset$
- Relevant:  $\{1, 2, 3\}$
- Irrelevant:

## Another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 15) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must consider only  $x_1 = 1$ 
  - Hint:** Can construct restricted truth-table
- All AXps:  $\{\{1, 2\}, \{1, 3\}\}$
- All CXps:  $\{\{1\}, \{2, 3\}\}$
- AXp-necessary:  $\{1\}$  (singleton CXp)
- CXp-necessary:  $\emptyset$
- Relevant:  $\{1, 2, 3\}$
- Irrelevant:  $\{4, 5\}$

## Yet another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}$ ;  $\mathcal{D}_i = \{0, 1\}$ ,  $i = 1, \dots, 5$ ;  $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$

## Yet another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}$ ;  $\mathcal{D}_i = \{0, 1\}$ ,  $i = 1, \dots, 5$ ;  $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$  and  $x_2 = x_3 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must either set  $x_1 = 1$  or  $x_2 = x_3 = 1$

## Yet another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$  and  $x_2 = x_3 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must either set  $x_1 = 1$  or  $x_2 = x_3 = 1$ 
  - Hint:** Can construct restricted truth-tables

## Yet another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$  and  $x_2 = x_3 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must either set  $x_1 = 1$  or  $x_2 = x_3 = 1$ 
  - Hint:** Can construct restricted truth-tables
- All AXps:

## Yet another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$  and  $x_2 = x_3 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must either set  $x_1 = 1$  or  $x_2 = x_3 = 1$ 
  - Hint:** Can construct restricted truth-tables
- All AXps:  $\{\{1\}, \{2, 3\}\}$
- All CXps:

## Yet another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$  and  $x_2 = x_3 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must either set  $x_1 = 1$  or  $x_2 = x_3 = 1$ 
  - Hint:** Can construct restricted truth-tables
- All AXps:  $\{\{1\}, \{2, 3\}\}$
- All CXps:  $\{\{1, 2\}, \{1, 3\}\}$
- AXp-necessary:

## Yet another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}$ ;  $\mathcal{D}_i = \{0, 1\}$ ,  $i = 1, \dots, 5$ ;  $\mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$  and  $x_2 = x_3 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must either set  $x_1 = 1$  or  $x_2 = x_3 = 1$ 
  - Hint:** Can construct restricted truth-tables
- All AXps:  $\{\{1\}, \{2, 3\}\}$
- All CXps:  $\{\{1, 2\}, \{1, 3\}\}$
- AXp-necessary:  $\emptyset$
- CXp-necessary:

## Yet another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$  and  $x_2 = x_3 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must either set  $x_1 = 1$  or  $x_2 = x_3 = 1$ 
  - Hint:** Can construct restricted truth-tables
- All AXps:  $\{\{1\}, \{2, 3\}\}$
- All CXps:  $\{\{1, 2\}, \{1, 3\}\}$
- AXp-necessary:  $\emptyset$
- CXp-necessary:  $\{1\}$  (singleton AXp)
- Relevant:

## Yet another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$  and  $x_2 = x_3 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must either set  $x_1 = 1$  or  $x_2 = x_3 = 1$ 
  - Hint:** Can construct restricted truth-tables
- All AXps:  $\{\{1\}, \{2, 3\}\}$
- All CXps:  $\{\{1, 2\}, \{1, 3\}\}$
- AXp-necessary:  $\emptyset$
- CXp-necessary:  $\{1\}$  (singleton AXp)
- Relevant:  $\{1, 2, 3\}$
- Irrelevant:

## Yet another example – feature necessity & relevancy

- Classifier:  $\mathcal{F} = \{1, 2, 3, 4, 5\}; \mathcal{D}_i = \{0, 1\}, i = 1, \dots, 5; \mathcal{K} = \{0, 1\}$

$$\kappa(x_1, x_2, x_3, x_4, x_5) := \begin{cases} 1 & \text{IF } (10x_1 + 5x_2 + 5x_3 + 2x_4 + x_5 \geq 10) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $((1, 1, 1, 1, 1), 1)$
- Obs:** If  $x_1 = 0$  and  $x_2 = x_3 = 0$ , then  $\kappa(\mathbf{x}) = 0$ ; i.e. must either set  $x_1 = 1$  or  $x_2 = x_3 = 1$ 
  - Hint:** Can construct restricted truth-tables
- All AXps:  $\{\{1\}, \{2, 3\}\}$
- All CXps:  $\{\{1, 2\}, \{1, 3\}\}$
- AXp-necessary:  $\emptyset$
- CXp-necessary:  $\{1\}$  (singleton AXp)
- Relevant:  $\{1, 2, 3\}$
- Irrelevant:  $\{4, 5\}$

## Some use cases

## Some use cases

Q: How to decide whether some **protected** feature occurs in **some** explanation?

## Some use cases

**Q:** How to decide whether some **protected** feature occurs in **some** explanation?

- Decide feature relevancy

## Some use cases

**Q:** How to decide whether some **protected** feature occurs in **some** explanation?

- Decide feature relevancy

**Q:** How to decide whether some **protected** feature occurs in **all** explanations?

## Some use cases

**Q:** How to decide whether some **protected** feature occurs in **some** explanation?

- Decide feature relevancy

**Q:** How to decide whether some **protected** feature occurs in **all** explanations?

- Decide feature necessity

## Some use cases

**Q:** How to decide whether some **protected** feature occurs in **some** explanation?

- Decide feature relevancy

**Q:** How to decide whether some **protected** feature occurs in **all** explanations?

- Decide feature necessity

**Q:** What can we do if human decision maker finds computed AXp/CXp to be unsatisfactory?

## Some use cases

**Q:** How to decide whether some **protected** feature occurs in **some** explanation?

- Decide feature relevancy

**Q:** How to decide whether some **protected** feature occurs in **all** explanations?

- Decide feature necessity

**Q:** What can we do if human decision maker finds computed AXp/CXp to be unsatisfactory?

- Partially enumerate AXps/CXps, exploiting bias in enumeration

# Plan for this course

- Lecture 01 – unit(s):
  - #01: Foundations
- Lecture 02 – unit(s):
  - #02: Principles of symbolic XAI – feature selection
  - #03: Tractability in symbolic XAI (& myth of interpretability)
- Lecture 03 – unit(s):
  - #04: Intractability in symbolic XAI (& myth of model-agnostic XAI)
  - #05: Explainability queries
- Lecture 04 – unit(s):
  - #06: Recent, emerging & advanced topics
- Lecture 05 – unit(s):
  - #07: Principles of symbolic XAI – feature attribution (& myth of Shapley values in XAI)
  - #08: Corrected feature attribution – nuSHAP
  - #09: Conclusions & research directions

## Detour: Monotonic Classification & Voting Power

## Monotonically increasing boolean classifiers

## Monotonically increasing boolean classifiers

- Monotonic classifier  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ , such that each  $\mathcal{D}_i = \{0, 1\}$  and  $\mathcal{K} = \{0, 1\}$  are ordered (i.e.  $0 < 1$ ), and
  - $\kappa(\mathbf{1}) = 1$  ;
  - Non-constant classifier, i.e.  $\kappa(\mathbf{0}) = 0$  ; and
  - $\kappa(\mathbf{x}_1) \leq \kappa(\mathbf{x}_2)$  when  $\mathbf{x}_1 \leq \mathbf{x}_2$

## Monotonically increasing boolean classifiers

- Monotonic classifier  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ , such that each  $\mathcal{D}_i = \{0, 1\}$  and  $\mathcal{K} = \{0, 1\}$  are ordered (i.e.  $0 < 1$ ), and
  - $\kappa(\mathbf{1}) = 1$  ;
  - Non-constant classifier, i.e.  $\kappa(\mathbf{0}) = 0$  ; and
  - $\kappa(\mathbf{x}_1) \leq \kappa(\mathbf{x}_2)$  when  $\mathbf{x}_1 \leq \mathbf{x}_2$
- Let  $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}$  be such that  $\kappa(\mathbf{v}_1) = \kappa(\mathbf{v}_2) = 1$ , and  $\mathbf{v}_1 \leq \mathbf{v}_2$   
Define the explanation problems:
  - $\mathcal{E}_1 = (\mathcal{M}, (\mathbf{v}_1, 1))$
  - $\mathcal{E}_2 = (\mathcal{M}, (\mathbf{v}_2, 1))$
  - $\mathcal{E}_{\mathbb{1}} = (\mathcal{M}, ((1, \dots, 1), 1)) = (\mathcal{M}, (\mathbb{1}, 1))$

## Monotonically increasing boolean classifiers

- Monotonic classifier  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ , such that each  $\mathcal{D}_i = \{0, 1\}$  and  $\mathcal{K} = \{0, 1\}$  are ordered (i.e.  $0 < 1$ ), and
  - $\kappa(\mathbf{1}) = 1$  ;
  - Non-constant classifier, i.e.  $\kappa(\mathbf{0}) = 0$  ; and
  - $\kappa(\mathbf{x}_1) \leq \kappa(\mathbf{x}_2)$  when  $\mathbf{x}_1 \leq \mathbf{x}_2$
- Let  $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{F}$  be such that  $\kappa(\mathbf{v}_1) = \kappa(\mathbf{v}_2) = 1$ , and  $\mathbf{v}_1 \leq \mathbf{v}_2$   
Define the explanation problems:
  - $\mathcal{E}_1 = (\mathcal{M}, (\mathbf{v}_1, 1))$
  - $\mathcal{E}_2 = (\mathcal{M}, (\mathbf{v}_2, 1))$
  - $\mathcal{E}_{\mathbf{1}} = (\mathcal{M}, ((1, \dots, 1), 1)) = (\mathcal{M}, (\mathbf{1}, 1))$
- Then,
  - If WAXp( $\mathcal{S}; \mathcal{E}_1$ ) holds, then WAXp( $\mathcal{S}; \mathcal{E}_2$ ) holds; in particular:
  - $\mathbb{A}(\mathcal{E}_{\mathbf{1}})$  contains all the AXps of any instance of the form  $(\mathbf{v}_r, 1)$ 
    - Why?
      - Pick any explanation problem  $\mathcal{E}_r$  with instance  $(\mathbf{v}_r, 1)$
      - Start from  $\mathbf{1} = (1, 1, \dots, 1)$
      - Remove features that take value 0 in  $\mathbf{v}_r$ ; we still have an WAXp
      - Then compute any AXp starting from features taking value 1 in  $\mathbf{v}_r$
    - ∴ Suffices to find explanations for  $\mathcal{E}_{\mathbf{1}}$  (or alternatively, the global explanations for prediction 1)

## An example

- ML model  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ :
  - Boolean classifier:  $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features:  $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$ 
    - I.e.  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- $\kappa$  is a monotonically increasing boolean function

## An example

- ML model  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ :
  - Boolean classifier:  $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features:  $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$ 
    - I.e.  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- $\kappa$  is a monotonically increasing boolean function
- We are interested in identifying the AXps of  $\mathcal{M}$ , given the instance  $((1, 1, 1, 1, 1, 1), 1)$ 
  - Or alternatively, the global AXps for prediction 1
  - For example, with order  $\langle 1, 2, 3, 4, 5, 6 \rangle$ :

## An example

- ML model  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ :
  - Boolean classifier:  $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features:  $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$ 
    - I.e.  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- $\kappa$  is a monotonically increasing boolean function
- We are interested in identifying the AXps of  $\mathcal{M}$ , given the instance  $((1, 1, 1, 1, 1, 1), 1)$ 
  - Or alternatively, the global AXps for prediction 1
  - For example, with order  $\langle 1, 2, 3, 4, 5, 6 \rangle$ :
    - Feature 1: can be dropped

## An example

- ML model  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ :
  - Boolean classifier:  $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features:  $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$ 
    - I.e.  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- $\kappa$  is a monotonically increasing boolean function
- We are interested in identifying the AXps of  $\mathcal{M}$ , given the instance  $((1, 1, 1, 1, 1, 1), 1)$ 
  - Or alternatively, the global AXps for prediction 1
  - For example, with order  $\langle 1, 2, 3, 4, 5, 6 \rangle$ :
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep

## An example

- ML model  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ :
  - Boolean classifier:  $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features:  $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$ 
    - I.e.  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- $\kappa$  is a monotonically increasing boolean function
- We are interested in identifying the AXps of  $\mathcal{M}$ , given the instance  $((1, 1, 1, 1, 1, 1), 1)$ 
  - Or alternatively, the global AXps for prediction 1
  - For example, with order  $\langle 1, 2, 3, 4, 5, 6 \rangle$ :
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep

## An example

- ML model  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ :
  - Boolean classifier:  $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features:  $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$ 
    - I.e.  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- $\kappa$  is a monotonically increasing boolean function
- We are interested in identifying the AXps of  $\mathcal{M}$ , given the instance  $((1, 1, 1, 1, 1, 1), 1)$ 
  - Or alternatively, the global AXps for prediction 1
  - For example, with order  $\langle 1, 2, 3, 4, 5, 6 \rangle$ :
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep
    - Feature 4: can no longer be dropped; keep

## An example

- ML model  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ :
  - Boolean classifier:  $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features:  $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$ 
    - I.e.  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- $\kappa$  is a monotonically increasing boolean function
- We are interested in identifying the AXps of  $\mathcal{M}$ , given the instance  $((1, 1, 1, 1, 1, 1), 1)$ 
  - Or alternatively, the global AXps for prediction 1
  - For example, with order  $\langle 1, 2, 3, 4, 5, 6 \rangle$ :
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep
    - Feature 4: can no longer be dropped; keep
    - Feature 5: can no longer be dropped; keep

## An example

- ML model  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ :
  - Boolean classifier:  $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features:  $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$ 
    - I.e.  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- $\kappa$  is a monotonically increasing boolean function
- We are interested in identifying the AXps of  $\mathcal{M}$ , given the instance  $((1, 1, 1, 1, 1, 1), 1)$ 
  - Or alternatively, the global AXps for prediction 1
  - For example, with order  $\langle 1, 2, 3, 4, 5, 6 \rangle$ :
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep
    - Feature 4: can no longer be dropped; keep
    - Feature 5: can no longer be dropped; keep
    - Feature 6: can be dropped

## An example

- ML model  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ :
  - Boolean classifier:  $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features:  $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$ 
    - I.e.  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- $\kappa$  is a monotonically increasing boolean function
- We are interested in identifying the AXps of  $\mathcal{M}$ , given the instance  $((1, 1, 1, 1, 1, 1), 1)$ 
  - Or alternatively, the global AXps for prediction 1
  - For example, with order  $\langle 1, 2, 3, 4, 5, 6 \rangle$ :
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep
    - Feature 4: can no longer be dropped; keep
    - Feature 5: can no longer be dropped; keep
    - Feature 6: can be dropped
    - **AXp:**  $\{2, 3, 4, 5\}$

## An example

- ML model  $\mathcal{M} = (\mathcal{F}, \mathbb{F}, \mathcal{K}, \kappa)$ :
  - Boolean classifier:  $\mathcal{K} = \{0, 1\}$
  - Defined on 6 boolean features:  $\mathcal{F} = \{1, 2, 3, 4, 5, 6\}$ 
    - I.e.  $\mathcal{D}_i = \{0, 1\}, i = 1, \dots, 6$
  - With classification function:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- $\kappa$  is a monotonically increasing boolean function
- We are interested in identifying the AXps of  $\mathcal{M}$ , given the instance  $((1, 1, 1, 1, 1, 1), 1)$ 
  - Or alternatively, the global AXps for prediction 1
  - For example, with order  $\langle 1, 2, 3, 4, 5, 6 \rangle$ :
    - Feature 1: can be dropped
    - Feature 2: can no longer be dropped; keep
    - Feature 3: can no longer be dropped; keep
    - Feature 4: can no longer be dropped; keep
    - Feature 5: can no longer be dropped; keep
    - Feature 6: can be dropped
    - **AXp:**  $\{2, 3, 4, 5\}$  ; **Q:** Is feature 6 relevant?

## All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $(1, 1)$

## All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $(1, 1)$

- Computing the AXps:

- Must pick 2 out of features  $\{1, 2, 3\}$
- If only 2 out of features  $\{1, 2, 3\}$  picked, then we must pick both features 4 and 5
- Feature 6 is never matters, i.e. it is irrelevant...

## All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $(1, 1)$

- Computing the AXps:

- Must pick 2 out of features  $\{1, 2, 3\}$
- If only 2 out of features  $\{1, 2, 3\}$  picked, then we must pick both features 4 and 5
- Feature 6 is never matters, i.e. it is irrelevant...

- AXps:

## All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $(1, 1)$

- Computing the AXps:

- Must pick 2 out of features  $\{1, 2, 3\}$
- If only 2 out of features  $\{1, 2, 3\}$  picked, then we must pick both features 4 and 5
- Feature 6 is never matters, i.e. it is irrelevant...

- AXps:

$$\mathbb{A} = \{\{1, 2, 3\}, \{1, 2, 4, 5\}, \{1, 3, 4, 5\}, \{2, 3, 4, 5\}\}$$

## All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $(1, 1)$

- Computing the AXps:

- Must pick 2 out of features  $\{1, 2, 3\}$
- If only 2 out of features  $\{1, 2, 3\}$  picked, then we must pick both features 4 and 5
- Feature 6 is never matters, i.e. it is irrelevant...

- AXps:

$$\mathbb{A} = \{\{1, 2, 3\}, \{1, 2, 4, 5\}, \{1, 3, 4, 5\}, \{2, 3, 4, 5\}\}$$

- CXps:

## All AXps & all CXps...

- Classifier:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

- Instance:  $(1, 1)$

- Computing the AXps:

- Must pick 2 out of features  $\{1, 2, 3\}$
- If only 2 out of features  $\{1, 2, 3\}$  picked, then we must pick both features 4 and 5
- Feature 6 is never matters, i.e. it is irrelevant...

- AXps:

$$\mathbb{A} = \{\{1, 2, 3\}, \{1, 2, 4, 5\}, \{1, 3, 4, 5\}, \{2, 3, 4, 5\}\}$$

- CXps:

$$\mathbb{C} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 4\}, \{1, 5\}, \{2, 4\}, \{2, 5\}, \{3, 4\}, \{3, 5\}\}$$

## What is a priori voting power?

- General set-up of weighted voting games:

## What is a priori voting power?

- General set-up of weighted voting games:
  - Assembly  $\mathcal{A}$  of voters, with  $m = |\mathcal{A}|$
  - Each voter  $i \in \mathcal{A}$  votes **Yes** with  $n_i$  votes; otherwise **no** votes are counted (and he/she votes **No**)

## What is a priori voting power?

- General set-up of weighted voting games:
  - Assembly  $\mathcal{A}$  of voters, with  $m = |\mathcal{A}|$
  - Each voter  $i \in \mathcal{A}$  votes **Yes** with  $n_i$  votes; otherwise **no** votes are counted (and he/she votes **No**)
  - A coalition is a subset of voters,  $\mathcal{C} \subseteq \mathcal{A}$
  - Quota  $q$  is the sum of votes required for a proposal to be approved
    - Coalitions leading to sums not less than  $q$  are **winning** coalitions

## What is a priori voting power?

- General set-up of weighted voting games:
  - Assembly  $\mathcal{A}$  of voters, with  $m = |\mathcal{A}|$
  - Each voter  $i \in \mathcal{A}$  votes **Yes** with  $n_i$  votes; otherwise **no** votes are counted (and he/she votes **No**)
  - A coalition is a subset of voters,  $\mathcal{C} \subseteq \mathcal{A}$
  - Quota  $q$  is the sum of votes required for a proposal to be approved
    - Coalitions leading to sums not less than  $q$  are **winning** coalitions
  - A **weighted voting game (WVG)** is a tuple  $[q; n_1, \dots, n_m]$ 
    - Example:  $[12; 4, 4, 4, 2, 2, 1]$

## What is a priori voting power?

- General set-up of weighted voting games:
  - Assembly  $\mathcal{A}$  of voters, with  $m = |\mathcal{A}|$
  - Each voter  $i \in \mathcal{A}$  votes **Yes** with  $n_i$  votes; otherwise **no** votes are counted (and he/she votes **No**)
  - A coalition is a subset of voters,  $\mathcal{C} \subseteq \mathcal{A}$
  - Quota  $q$  is the sum of votes required for a proposal to be approved
    - Coalitions leading to sums not less than  $q$  are **winning** coalitions
  - A **weighted voting game (WVG)** is a tuple  $[q; n_1, \dots, n_m]$ 
    - Example:  $[12; 4, 4, 4, 2, 2, 1]$
  - Problem: find a measure of importance of each voter !
    - I.e. measure the **a priori voting power** of each voter

## An example – EEC (EU) members voting power in 1958

Country	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:		12

## An example – EEC (EU) members voting power in 1958

Country	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:		12

• WVG: [12; 4, 4, 4, 2, 2, 1]

## An example – EEC (EU) members voting power in 1958

Country	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:		12

- WVG: [12; 4, 4, 4, 2, 2, 1]
- Q: What should be the voting power of Luxembourg?

## An example – EEC (EU) members voting power in 1958

Country	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:		12

- WVG: [12; 4, 4, 4, 2, 2, 1]
- Q: What should be the voting power of Luxembourg?
- Can Luxembourg (L) **matter** for some winning coalition?

## An example – EEC (EU) members voting power in 1958

Country	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:		12

- WVG: [12; 4, 4, 4, 2, 2, 1]
- Q: What should be the voting power of Luxembourg?
- Can Luxembourg (L) **matter** for some winning coalition?
- Perhaps surprisingly, answer is **No!**
  - In 1958, Luxembourg was a **dummy** voter/player

## Understanding weighted voting games

- Obs: A WVG is a monotonically increasing boolean classifier
- Each subset-minimal winning coalition is an AXp of the instance  $(\mathbb{1}, \mathbb{1})$

# Understanding weighted voting games

- Obs: A WVG is a monotonically increasing boolean classifier
- Each subset-minimal winning coalition is an AXp of the instance  $(1, 1)$
- Recall EEC voting example:

Country	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:		12

# Understanding weighted voting games

- Obs: A WVG is a monotonically increasing boolean classifier
- Each subset-minimal winning coalition is an AXp of the instance  $(1, 1)$
- Recall EEC voting example:

Coutry	Acronym	# Votes
France	F	4
Germany	D	4
Italy	I	4
Belgium	B	2
Netherlands	N	2
Luxembourg	L	1
Quota:		12

- The corresponding classifier is:

$$\kappa(x_1, x_2, x_3, x_4, x_5, x_6) := \begin{cases} 1 & \text{IF } (4x_1 + 4x_2 + 4x_3 + 2x_4 + 2x_5 + x_6 \geq 12) \\ 0 & \text{otherwise} \end{cases}$$

which we have seen before! E.g.  $\{2, 3, 4, 5\}$  is an AXp & feature 6 (L) is **irrelevant**

## Another example

- WVG: [ 21; 12, 9, 4, 4, 1, 1, 1 ]

## Another example

- WVG: [ 21; 12, 9, 4, 4, 1, 1, 1 ]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7

## Another example

- WVG: [ 21; 12, 9, 4, 4, 1, 1, 1 ]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7
- AXps:

## Another example

- WVG: [ 21; 12, 9, 4, 4, 1, 1, 1 ]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7
- AXps:
$$\mathbb{A} = \{\{1, 2\}, \{1, 3, 4, 5\}, \{1, 3, 4, 6\}, \{1, 3, 4, 7\}\}$$

## Another example

- WVG: [ 21; 12, 9, 4, 4, 1, 1, 1 ]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7
- AXps:

$$\mathbb{A} = \{\{1, 2\}, \{1, 3, 4, 5\}, \{1, 3, 4, 6\}, \{1, 3, 4, 7\}\}$$

- CXps:

## Another example

- WVG: [ 21; 12, 9, 4, 4, 1, 1, 1 ]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7
- AXps:

$$\mathbb{A} = \{\{1, 2\}, \{1, 3, 4, 5\}, \{1, 3, 4, 6\}, \{1, 3, 4, 7\}\}$$

- CXps:

$$\mathbb{C} = \{\{1\}, \{2, 3\}, \{2, 4\}, \{2, 5, 6, 7\}\}$$

## Another example

- WVG: [ 21; 12, 9, 4, 4, 1, 1, 1 ]
- Computing the AXps:
  - Must include feature 1; sum of weights of others equals 20...
  - Either include feature 2, or features 3 and 4, plus any one of features 5, 6, 7
- AXps:

$$\mathbb{A} = \{\{1, 2\}, \{1, 3, 4, 5\}, \{1, 3, 4, 6\}, \{1, 3, 4, 7\}\}$$

- CXps:
- $\mathbb{C} = \{\{1\}, \{2, 3\}, \{2, 4\}, \{2, 5, 6, 7\}\}$
- Q: How should features be ranked in terms of importance?

## Yet another example

- WVG: [ 16; 9, 9, 7, 3, 1, 1 ]

## Yet another example

- WVG: [ 16; 9, 9, 7, 3, 1, 1 ]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter

## Yet another example

- WVG: [ 16; 9, 9, 7, 3, 1, 1 ]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:

## Yet another example

- WVG: [ 16; 9, 9, 7, 3, 1, 1 ]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:

$$\mathbb{A} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$

## Yet another example

- WVG: [ 16; 9, 9, 7, 3, 1, 1 ]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:
$$\mathbb{A} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$
- CXps:

## Yet another example

- WVG: [ 16; 9, 9, 7, 3, 1, 1 ]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:

$$\mathbb{A} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$

- CXps:
- $$\mathbb{C} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$

## Yet another example

- WVG: [ 16; 9, 9, 7, 3, 1, 1 ]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:
$$\mathbb{A} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$
- CXps:
$$\mathbb{C} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$
- **Obs:** features (resp. voters) 4, 5 and 6 are irrelevant (resp. dummy)

## Yet another example

- WVG: [ 16; 9, 9, 7, 3, 1, 1 ]
- Computing the AXps:
  - Sum of any pair of the first three features (i.e. voters) exceeds/matches the quota
  - The other features never matter
- AXps:

$$\mathbb{A} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$

- CXps:
$$\mathbb{C} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}\}$$

- **Obs:** features (resp. voters) 4, 5 and 6 are irrelevant (resp. dummy)
- **Q:** How should features be ranked in terms of importance?

## Why should we care about voting power?

- SHAP scores, i.e. the use of Shapley values for XAI, exhibit critical theoretical flaws  
(more tomorrow)

[MSH24, HMS24, HM23c]

# Why should we care about voting power?

- SHAP scores, i.e. the use of Shapley values for XAI, exhibit critical theoretical flaws  
(more tomorrow)

[MSH24, HMS24, HM23c]

- Recently, we have devised ways of **correcting** SHAP scores

[LHMS24]

# Why should we care about voting power?

- SHAP scores, i.e. the use of Shapley values for XAI, exhibit critical theoretical flaws  
(more tomorrow)

[MSH24, HMS24, HM23c]

- Recently, we have devised ways of **correcting** SHAP scores

[LHMS24]

- In turn, this revealed novel connections between logic-based XAI and a priori voting power

[LHAMS24]

# Why should we care about voting power?

- SHAP scores, i.e. the use of Shapley values for XAI, exhibit critical theoretical flaws  
(more tomorrow)

[MSH24, HMS24, HM23c]

- Recently, we have devised ways of **correcting** SHAP scores

[LHMS24]

- In turn, this revealed novel connections between logic-based XAI and a priori voting power

[LHAMS24]

- Homework:

- Create your own weighted voting games;
- Compute the sets of AXps and CXps; and
- Assess the importance of features and how they compare to each other

Unit #06

Advanced Topics

# Outline – Unit #06

Sample-Based Explanations

Changing Assumptions (in Plain Logic-Based XAI)

Inflated Explanations

Constrained Explanations

Distance-Restricted Explanations

Probabilistic Explanations

Additional Topics

## Recurring challenge: how to explain highly-complex ML models?

- **Not** with non-symbolic XAI

## Recurring challenge: how to explain highly-complex ML models?

- **Not** with non-symbolic XAI
  - Existing solutions are flawed...
    - SHAP, LIME, Anchors, etc. cannot be trusted for rigor

## Recurring challenge: how to explain highly-complex ML models?

- **Not** with non-symbolic XAI
  - Existing solutions are flawed...
    - SHAP, LIME, Anchors, etc. cannot be trusted for rigor
- Likely **not** with symbolic XAI
  - Logic-based explainability faces the challenge of scalability
    - Unlikely standard logic reasoning will scale for extremely complex models...

## Recurring challenge: how to explain highly-complex ML models?

- **Not** with non-symbolic XAI
  - Existing solutions are flawed...
    - SHAP, LIME, Anchors, etc. cannot be trusted for rigor
- Likely **not** with symbolic XAI
  - Logic-based explainability faces the challenge of scalability
    - Unlikely standard logic reasoning will scale for extremely complex models...
- How can this conundrum be solved?

# Recurring challenge: how to explain highly-complex ML models?

- **Not** with non-symbolic XAI
  - Existing solutions are flawed...
    - SHAP, LIME, Anchors, etc. cannot be trusted for rigor
- Likely **not** with symbolic XAI
  - Logic-based explainability faces the challenge of scalability
    - Unlikely standard logic reasoning will scale for extremely complex models...
- How can this conundrum be solved?
  - The use of **sampling** is ubiquitous in non-symbolic XAI
    - Many examples: LIME, SHAP, Anchors, etc.

# Recurring challenge: how to explain highly-complex ML models?

- **Not** with non-symbolic XAI
  - Existing solutions are flawed...
    - SHAP, LIME, Anchors, etc. cannot be trusted for rigor
- Likely **not** with symbolic XAI
  - Logic-based explainability faces the challenge of scalability
    - Unlikely standard logic reasoning will scale for extremely complex models...
- How can this conundrum be solved?
  - The use of **sampling** is ubiquitous in non-symbolic XAI
    - Many examples: LIME, SHAP, Anchors, etc.
  - **And training data is nothing but a sample**
    - From which ML models are learned!

# Recurring challenge: how to explain highly-complex ML models?

- **Not** with non-symbolic XAI
  - Existing solutions are flawed...
    - SHAP, LIME, Anchors, etc. cannot be trusted for rigor
- Likely **not** with symbolic XAI
  - Logic-based explainability faces the challenge of scalability
    - Unlikely standard logic reasoning will scale for extremely complex models...
- How can this conundrum be solved?
  - The use of **sampling** is ubiquitous in non-symbolic XAI
    - Many examples: LIME, SHAP, Anchors, etc.
  - **And training data is nothing but a sample**
    - From which ML models are learned!
  - Here is an idea:

# Recurring challenge: how to explain highly-complex ML models?

- **Not** with non-symbolic XAI
  - Existing solutions are flawed...
    - SHAP, LIME, Anchors, etc. cannot be trusted for rigor
- Likely **not** with symbolic XAI
  - Logic-based explainability faces the challenge of scalability
    - Unlikely standard logic reasoning will scale for extremely complex models...
- How can this conundrum be solved?
  - The use of **sampling** is ubiquitous in non-symbolic XAI
    - Many examples: LIME, SHAP, Anchors, etc.
  - **And training data is nothing but a sample**
    - From which ML models are learned!
  - Here is an idea:
    - Adopt **symbolic** (and so, rigorous) sample-based XAI

[Amg23, CA23, ACD24]

## An example...

Sample:

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\cdot)$
0	0	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

Instance:  $((1, 1, 1, 1), 1)$

## An example...

Sample:

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\cdot)$
0	0	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

- How to explain prediction given only the sample

Instance:  $((1, 1, 1, 1), 1)$

## An example...

Sample:

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\cdot)$
0	0	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

Instance:  $((1, 1, 1, 1), 1)$

- How to explain prediction given only the sample
- If  $x_1 = 1$ , then prediction is 1 (given the sample)

## An example...

Sample:

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\cdot)$
0	0	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

Instance:  $((1, 1, 1, 1), 1)$

- How to explain prediction given only the sample
- If  $x_1 = 1$ , then prediction is 1 (given the sample)
- Sample-based AXp (sbAXp):  $\{1\}$

## Definitions in sample-based XAI – replace feature space with sample...

[MSLLM25]

- Let  $\mathbb{S} \subseteq \mathbb{F}$  denote a **sample**, and let instance be  $(\mathbf{v}, c)$
- Then, for  $\mathcal{X} \subseteq \mathcal{F}$ ,

$$\text{sbWAXp}(\mathcal{X}) \coloneqq \forall (\mathbf{x} \in \mathbb{S}). \left( \bigwedge_{i \in \mathcal{X}} x_i = v_i \right) \rightarrow (\kappa(\mathbf{x}) = c)$$

- And, for  $\mathcal{Y} \subseteq \mathcal{F}$ ,

$$\text{sbWCXp}(\mathcal{Y}) \coloneqq \exists (\mathbf{x} \in \mathbb{S}). \left( \bigwedge_{i \in \mathcal{F} \setminus \mathcal{Y}} x_i = v_i \right) \wedge (\kappa(\mathbf{x}) \neq c)$$

- sbAXps (resp. sbCXps) are the subset-minimal sets that respect the above definition for sbWAXp (resp. sbWCXp)

## Definitions in sample-based XAI – replace feature space with sample...

[MSLLM25]

- Let  $\mathbb{S} \subseteq \mathbb{F}$  denote a **sample**, and let instance be  $(\mathbf{v}, c)$
- Then, for  $\mathcal{X} \subseteq \mathcal{F}$ ,

$$\text{sbWAXp}(\mathcal{X}) \coloneqq \forall (\mathbf{x} \in \mathbb{S}). \left( \bigwedge_{i \in \mathcal{X}} x_i = v_i \right) \rightarrow (\kappa(\mathbf{x}) = c)$$

- And, for  $\mathcal{Y} \subseteq \mathcal{F}$ ,

$$\text{sbWCXp}(\mathcal{Y}) \coloneqq \exists (\mathbf{x} \in \mathbb{S}). \left( \bigwedge_{i \in \mathcal{F} \setminus \mathcal{Y}} x_i = v_i \right) \wedge (\kappa(\mathbf{x}) \neq c)$$

- sbAXps (resp. sbCXps) are the subset-minimal sets that respect the above definition for sbWAXp (resp. sbWCXp)
  - Rigorous alternative to Anchor & variants

## Approach for computing sbCXps & sbAXps

Sample:

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\cdot)$
0	0	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

Instance:  $((1, 1, 1, 1), 1)$

## Approach for computing sbCXps & sbAXps

Sample:

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\cdot)$
0	0	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Instance:  $((1, 1, 1, 1), 1)$

## Approach for computing sbCXps & sbAXps

Sample:

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\cdot)$
0	0	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Instance:  $((1, 1, 1, 1), 1)$

## Approach for computing sbCXps & sbAXps

Sample:

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\cdot)$
0	0	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

Instance:  $((1, 1, 1, 1), 1)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

sbCXps:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

## Approach for computing sbCXps & sbAXps

Sample:

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\cdot)$
0	0	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

Instance:  $((1, 1, 1, 1), 1)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

sbCXps:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

- Set of sbCXps:  $\mathbb{C} = \{\{1, 2\}, \{1, 4\}\}$

## Approach for computing sbCXps & sbAXps

Sample:

$x_1$	$x_2$	$x_3$	$x_4$	$\kappa(\cdot)$
0	0	0	0	0
0	0	1	0	0
0	0	1	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	0	1
1	1	1	1	1

Instance:  $((1, 1, 1, 1), 1)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

sbCXps:

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

- Set of sbCXps:  $\mathbb{C} = \{\{1, 2\}, \{1, 4\}\}$
- Set of sbAXps (by MHS duality):  $\mathbb{A} = \{\{1\}, \{2, 4\}\}$

- MHS duality holds for sample-based explanations:
  - $\mathcal{Y} \subseteq \mathcal{F}$  is sbCXp iff it is a MHS of set of sbAXps
  - $\mathcal{X} \subseteq \mathcal{F}$  is sbAXp iff it is a MHS of set of sbCXps
- Number of sb(W)CXps is linear on  $|\mathbb{S}|$
- Number of sb(W)AXps can be exponentially large on  $|\mathbb{S}|$
- Additional results:

Problem	Complexity	
	Total	Given sbCXps
All sbCXps	$\mathcal{O}(mn^2)$	—
One sbCXp	$\mathcal{O}(mn)$	$\mathcal{O}(1)$
One (smallest) sbCXp	$\mathcal{O}(mn^2)$	$\mathcal{O}(n)$
One sbAXp	$\mathcal{O}(mn)$	$\mathcal{O}(mn)$
Feature relevancy	$\mathcal{O}(mn^2)$	$\mathcal{O}(n)$
sbAXp-necessity	$\mathcal{O}(mn^2)$	$\mathcal{O}(mn)$
sbCXp-necessity	$\mathcal{O}(mn^2)$	$\mathcal{O}(n)$

- Complexity-wise:
  - Deciding the existence of an sbAXp of size no larger than  $k$  is NP-complete.
  - sbAXp enumeration corresponds to hypergraph transversal

# Does sample-based XAI suffice?

- Sample-based explanations lack **coherency**:
  - There exist two instances with different predictions with AXps that cover at least one common point

[ACD24]

# Does sample-based XAI suffice?

- Sample-based explanations lack **coherency**:
  - There exist two instances with different predictions with AXps that cover at least one common point
- An example:

- Sample:

Entry	$x_1$	$x_2$	$\kappa(\cdot)$
1	0	1	0
2	1	0	1
3	0	0	2

- Instance 1:  $((0, 1), 0)$
- Instance 2:  $((1, 0), 1)$

[ACD24]

$\text{AXp}(s)$  for  $((0, 1), 0)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((0, 1), 0)$

## $\text{AXp}(s)$ for $((0, 1), 0)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Instance:  $((0, 1), 0)$

## $\text{AXp}(s)$ for $((0, 1), 0)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((0, 1), 0)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

## $\text{AXp}(s)$ for $((0, 1), 0)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((0, 1), 0)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

sbCXps:

$$\begin{bmatrix} 0 & 1 \end{bmatrix}$$

## $\text{AXp}(s)$ for $((0, 1), 0)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((0, 1), 0)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

sbCXps:

$$\begin{bmatrix} 0 & 1 \end{bmatrix}$$

- Set of sbCXps:  $\mathbb{C} = \{\{2\}\}$

## $\text{AXp}(s)$ for $((0, 1), 0)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((0, 1), 0)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

sbCXps:

$$\begin{bmatrix} 0 & 1 \end{bmatrix}$$

- Set of sbCXps:  $\mathbb{C} = \{\{2\}\}$
- Set of sbAXps (by MHS duality):  $\mathbb{A} = \{\{2\}\}$

## $\text{AXp}(s)$ for $((0, 1), 0)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((0, 1), 0)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

sbCXps:

$$\begin{bmatrix} 0 & 1 \end{bmatrix}$$

- Set of sbCXps:  $\mathbb{C} = \{\{2\}\}$
- Set of sbAXps (by MHS duality):  $\mathbb{A} = \{\{2\}\}$ 
  - Meaning: IF  $(x_2 = 1)$  THEN  $\kappa(\mathbf{x}) = 0$

$\text{AXp}(s)$  for  $((1, 0), 1)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((1, 0), 1)$

## $\text{AXp}(s)$ for $((1, 0), 1)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Instance:  $((1, 0), 1)$

## $\text{AXp}(s)$ for $((1, 0), 1)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((1, 0), 1)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

## $\text{AXp}(s)$ for $((1, 0), 1)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((1, 0), 1)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

sbCXps:

$$\begin{bmatrix} 1 & 0 \end{bmatrix}$$

## $\text{AXp}(s)$ for $((1, 0), 1)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((1, 0), 1)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

sbCXps:

$$\begin{bmatrix} 1 & 0 \end{bmatrix}$$

- Set of sbCXps:  $\mathbb{C} = \{\{1\}\}$

## $\text{AXp}(s)$ for $((1, 0), 1)$

Sample:

$x_1$	$x_2$	$\kappa(\cdot)$
0	1	0
1	0	1
0	0	2

Instance:  $((1, 0), 1)$

Points  $\mathbf{x}$  w/  $\kappa(\mathbf{x}) \neq c$ :

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$$

Weak sbCXps:

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

sbCXps:

$$\begin{bmatrix} 1 & 0 \end{bmatrix}$$

- Set of sbCXps:  $\mathbb{C} = \{\{1\}\}$
- Set of sbAXps (by MHS duality):  $\mathbb{A} = \{\{1\}\}$ 
  - Meaning: IF  $(x_1 = 1)$  THEN  $\kappa(\mathbf{x}) = 1$

## The problem of lack of coherency

- Instance  $((0, 1), 0)$  with  $\text{AXp } \{2\}$ , for  $x_2 = 1$ :
  - Points consistent with  $\text{AXp}$ :  $\{(0, 1), (1, 1)\}$
  - I.e. prediction is 0 for the points  $\{(0, 1), (1, 1)\}$

## The problem of lack of coherency

- Instance  $((0, 1), 0)$  with  $\text{AXp } \{2\}$ , for  $x_2 = 1$ :
  - Points consistent with  $\text{AXp}$ :  $\{(0, 1), (1, 1)\}$
  - I.e. prediction is 0 for the points  $\{(0, 1), (1, 1)\}$
- Instance  $((1, 0), 1)$  with  $\text{AXp } \{1\}$ , for  $x_1 = 1$ :
  - Points consistent with  $\text{AXp}$ :  $\{(1, 0), (1, 1)\}$
  - I.e. prediction is 1 for the points  $\{(1, 0), (1, 1)\}$

## The problem of lack of coherency

- Instance  $((0, 1), 0)$  with  $\text{AXp } \{2\}$ , for  $x_2 = 1$ :
    - Points consistent with  $\text{AXp}$ :  $\{(0, 1), (1, 1)\}$
    - I.e. prediction is 0 for the points  $\{(0, 1), (\mathbf{1}, \mathbf{1})\}$
  - Instance  $((1, 0), 1)$  with  $\text{AXp } \{1\}$ , for  $x_1 = 1$ :
    - Points consistent with  $\text{AXp}$ :  $\{(1, 0), (1, 1)\}$
    - I.e. prediction is 1 for the points  $\{(1, 0), (\mathbf{1}, \mathbf{1})\}$
- $\therefore (1, 1)$  assumed to have different predictions!

## The problem of lack of coherency

- Instance  $((0, 1), 0)$  with  $\text{AXp } \{2\}$ , for  $x_2 = 1$ :
    - Points consistent with  $\text{AXp}$ :  $\{(0, 1), (1, 1)\}$
    - I.e. prediction is 0 for the points  $\{(0, 1), (1, 1)\}$
  - Instance  $((1, 0), 1)$  with  $\text{AXp } \{1\}$ , for  $x_1 = 1$ :
    - Points consistent with  $\text{AXp}$ :  $\{(1, 0), (1, 1)\}$
    - I.e. prediction is 1 for the points  $\{(1, 0), (1, 1)\}$
- $\therefore (1, 1)$  assumed to have different predictions!
- Open topic of research...

# Outline – Unit #06

Sample-Based Explanations

Changing Assumptions (in Plain Logic-Based XAI)

Inflated Explanations

Constrained Explanations

Distance-Restricted Explanations

Probabilistic Explanations

Additional Topics

## General definition of prediction sufficiency

- Instance  $(\mathbf{v}, c)$
- Let  $\mathcal{S} \subseteq \mathcal{F}$ :
  - Recall,

$$\Upsilon(\mathcal{S}; \mathbf{v}) = \{\mathbf{x} \in \mathbb{F} \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}\}$$

- $\mathcal{S} \subseteq \mathcal{F}$  suffices for prediction  $c$  if:

$$\forall (\mathbf{x} \in \mathbb{F}). (\mathbf{x} \in \Upsilon(\mathcal{S}; \mathbf{v})) \rightarrow (\sigma(\mathbf{x}))$$

- **Obs:** a WAXp is just one possible example
- But there are other ways to study prediction sufficiency:
  - One can envision defining other sets of points  $\Gamma$ , parameterized by  $\mathcal{E} = (\mathcal{M}, (\mathbf{v}, c))$ ;  
 $\mathcal{S} \subseteq \mathcal{F}$  suffices for prediction  $c$  if:

$$\forall (\mathbf{x} \in \mathbb{F}). (\mathbf{x} \in \Gamma(\mathcal{S}; \mathcal{E})) \rightarrow (\sigma(\mathbf{x}))$$

- And one can also envision generalizations of  $\sigma$ !

# Outline – Unit #06

Sample-Based Explanations

Changing Assumptions (in Plain Logic-Based XAI)

Inflated Explanations

Constrained Explanations

Distance-Restricted Explanations

Probabilistic Explanations

Additional Topics

## Towards more expressive explanations – inflated explanations

[IISM24]

- Recall:

$$\text{WAXp}(\mathcal{X}) \quad := \quad \forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

- For non-boolean features, use of  $=$  may convey little information, e.g. with real-valued features, having  $x_1 = 1.157$  does not help in understanding what values of feature 1 are also acceptable

## Towards more expressive explanations – inflated explanations

[IISM24]

- Recall:

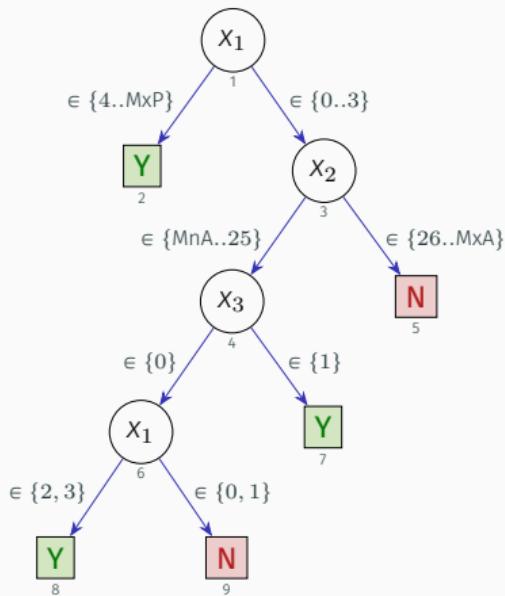
$$\text{WAXp}(\mathcal{X}) \quad := \quad \forall (\mathbf{x} \in \mathbb{F}). \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \rightarrow (\kappa(\mathbf{x}) = c)$$

- For non-boolean features, use of  $=$  may convey little information, e.g. with real-valued features, having  $x_1 = 1.157$  does not help in understanding what values of feature 1 are also acceptable
- **Inflated explanations** allow for more expressive literals, i.e.  $=$  replaced with  $\in$ , and individual values replaced by ranges of values
  - Operational definition: Given an AXp, expand set of values of each feature, in some chosen order, such that the set of picked features remains unchanged

# Inflated explanations – an example

[IIM22]

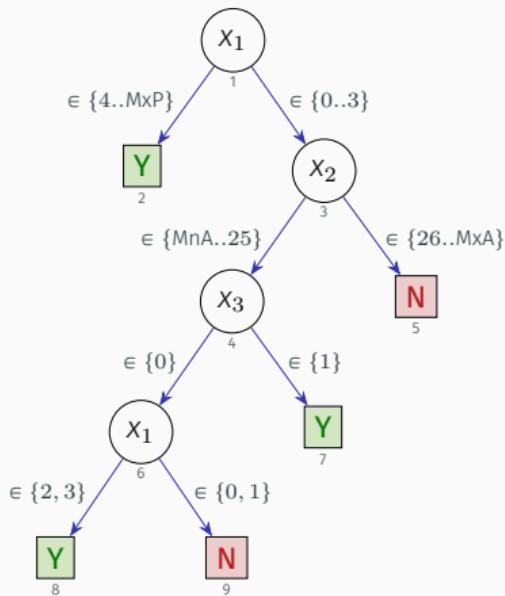
- Explanation for  $((2, 20, 0), Y)?$  (Obs: MnA = 18; MxP > 4)



# Inflated explanations – an example

[IIM22]

- Explanation for  $((2, 20, 0), Y)?$  (Obs: MnA = 18; MxP > 4)
  - AXp: {1, 2}



# Inflated explanations – an example

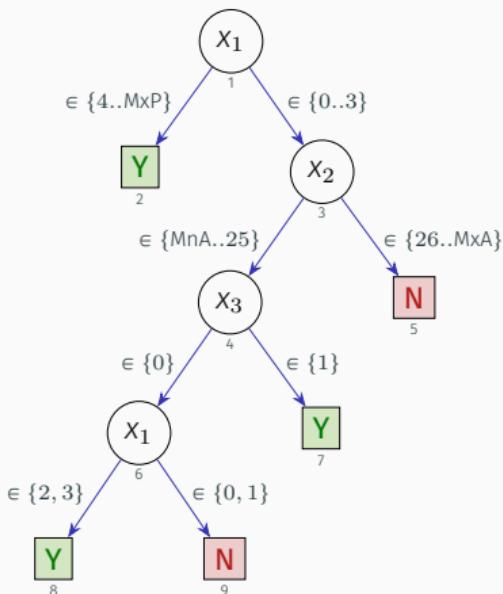
[IIM22]

- Explanation for  $((2, 20, 0), Y)$ ? (Obs: MnA = 18; MxP > 4)

- AXp: {1, 2}

- Default interpretation:

$$\forall (\mathbf{x} \in \mathbb{F}).(x_1 = 2 \wedge x_2 = 20) \rightarrow (\kappa(\mathbf{x}) = Y)$$



# Inflated explanations – an example

[IIM22]

- Explanation for  $((2, 20, 0), Y)$ ? (Obs:  $MnA = 18; MxP > 4$ )

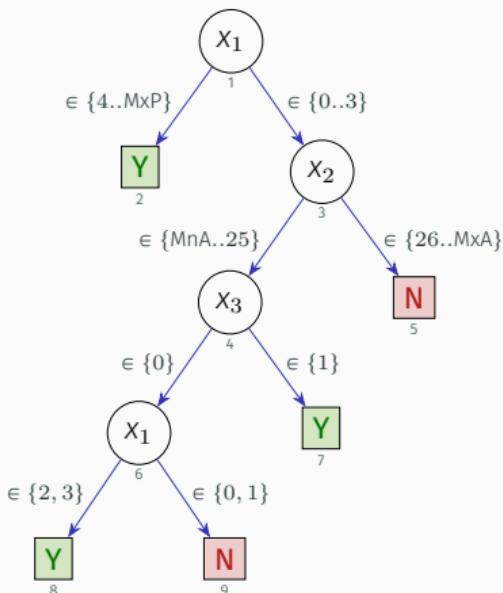
- AXp: {1, 2}

- Default interpretation:

$$\forall (\mathbf{x} \in \mathbb{F}).(x_1 = 2 \wedge x_2 = 20) \rightarrow (\kappa(\mathbf{x}) = Y)$$

- Corresponding rule:

$$\text{IF } (x_1 = 2 \wedge x_2 = 20) \text{ THEN } (\kappa(\mathbf{x}) = Y)$$



# Inflated explanations – an example

[IIM22]

- Explanation for  $((2, 20, 0), Y)$ ? (Obs: MnA = 18; MxP > 4)

- AxP: {1, 2}

- Default interpretation:

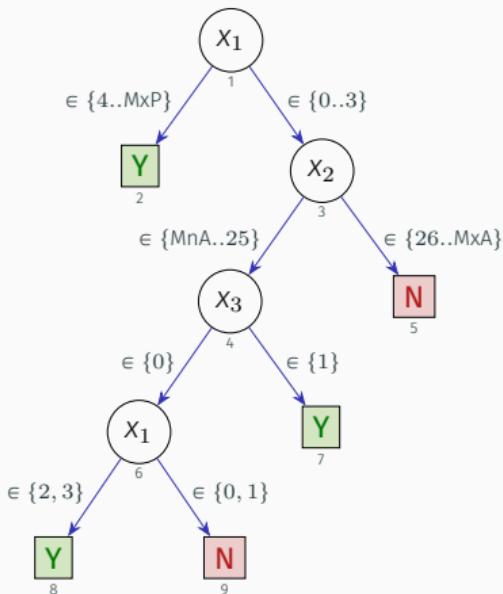
$$\forall (\mathbf{x} \in \mathbb{F}).(x_1 = 2 \wedge x_2 = 20) \rightarrow (\kappa(\mathbf{x}) = Y)$$

- Corresponding rule:

$$\text{IF } (x_1 = 2 \wedge x_2 = 20) \text{ THEN } (\kappa(\mathbf{x}) = Y)$$

- With inflated explanations:

$$\forall (\mathbf{x} \in \mathbb{F}).(x_1 \in \{2..MxP\} \wedge x_2 \in \{MnA..25\}) \rightarrow (\kappa(\mathbf{x}) = Y)$$



# Inflated explanations – an example

[IIM22]

- Explanation for  $((2, 20, 0), Y)$ ? (Obs:  $MnA = 18; MxP > 4$ )

- A $Xp$ : {1, 2}

- Default interpretation:

$$\forall (\mathbf{x} \in \mathbb{F}).(x_1 = 2 \wedge x_2 = 20) \rightarrow (\kappa(\mathbf{x}) = Y)$$

- Corresponding rule:

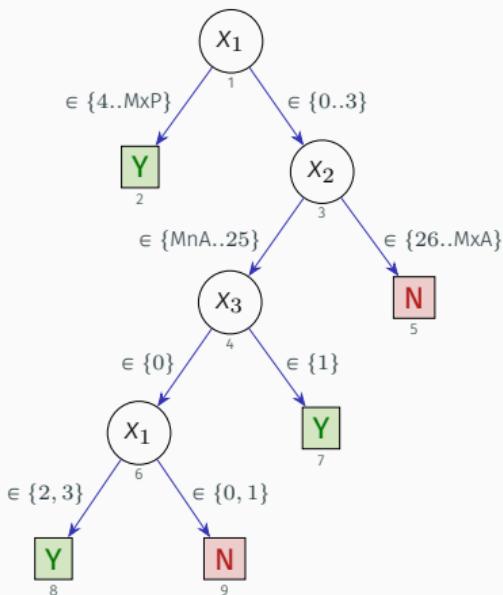
$$\text{IF } (x_1 = 2 \wedge x_2 = 20) \text{ THEN } (\kappa(\mathbf{x}) = Y)$$

- With inflated explanations:

$$\forall (\mathbf{x} \in \mathbb{F}).(x_1 \in \{2..MxP\} \wedge x_2 \in \{MnA..25\}) \rightarrow (\kappa(\mathbf{x}) = Y)$$

- Corresponding rule:

$$\text{IF } (x_1 \in \{2..MxP\} \wedge x_2 \in \{MnA..25\}) \text{ THEN } (\kappa(\mathbf{x}) = Y)$$



# Approach

- Compute  $Ax$
- For each feature:
  - Categorical: iteratively add elements to literal
  - Ordinal:
    - Expand literal for larger values;
    - Expand literal for smaller values

# Approach

- Compute  $\text{AXp } \mathcal{X}$
- For each feature:
  - Categorical: iteratively add elements to literal
  - Ordinal:
    - Expand literal for larger values;
    - Expand literal for smaller values
- **Obs:** More complex alternative is to find  $\text{AXp}$  and expand domains simultaneously
  - This is conjectured to change the complexity class of finding one explanation

# Another fun example (from 2019) – (inflated) AXp?

[CdPA<sup>+</sup>19]

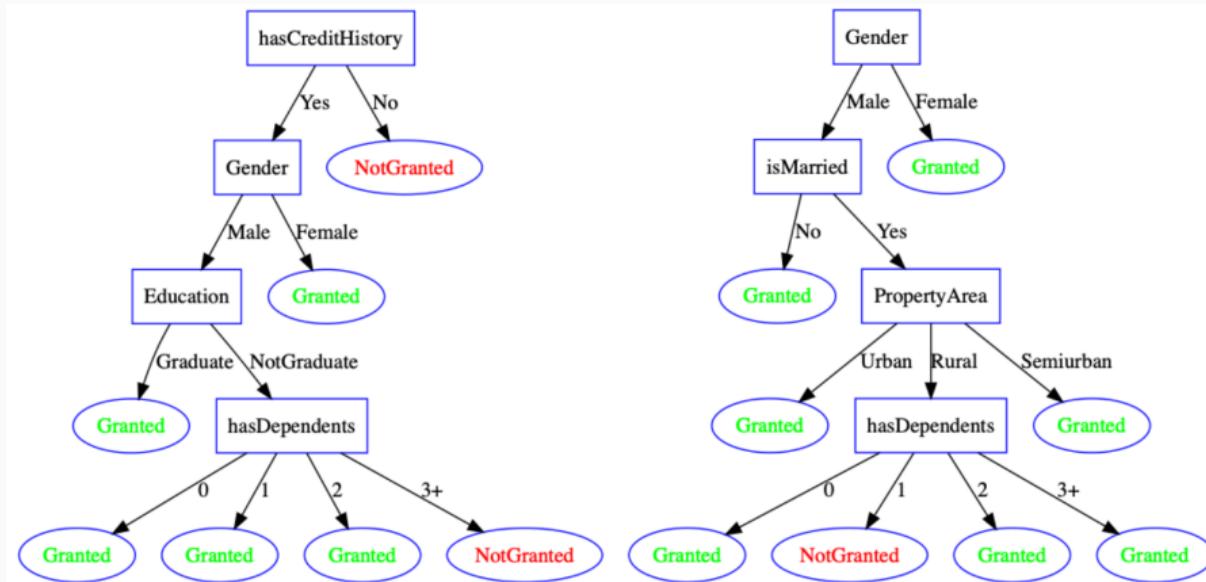
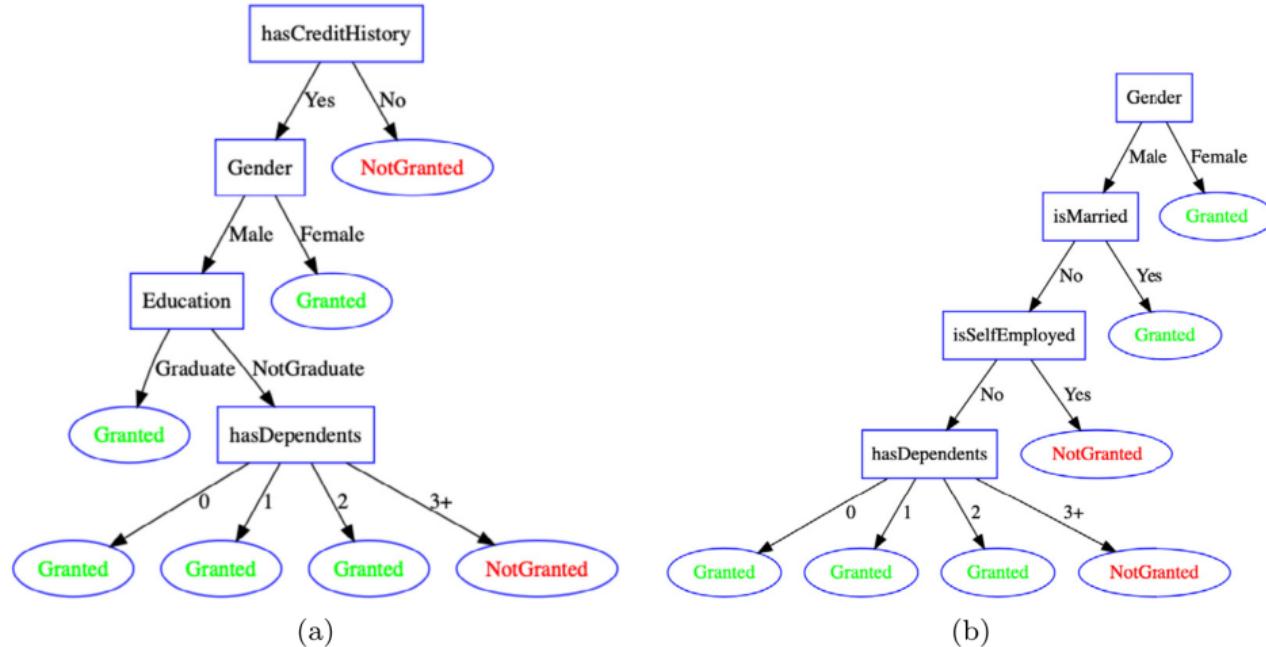


Fig. 2: Decision trees of size ‘small’ in the loan domain, extracted without (left) and with (right) a domain ontology. As it can be seen the features used in the creation of the conditions in the split nodes are different.

Instance: Gender=Male, Education=NotGraduate, hasCreditHistory=Yes, isMarried=Yes, PropertyArea=Rural, isSelfEmployed>No, hasDependents=2

# Another fun example (now from 2020/21) – (inflated) AXp?

[CWBdPM21]



**Fig. 2.** Decision trees of size ‘small’ in the loan domain, extracted without (a) and with (b) a domain ontology. It can be seen that the use of an ontology leads to different features appearing in the split nodes. For instance, in the ontology used to build tree (b) the concept `Gender` is more abstract than `isMarried` and `isSelfEmployed` has thus a lower information content according to Definition 3.1. Concepts with lower information content are favoured as conditions for split nodes in the tree according to Definition 3.2, which leads to `Gender` being used first by TREPAN-Reloaded when it generated the split nodes of tree (b). Furthermore, the ontology does not include concepts associated to `hasCreditHistory` and `Education`, which are therefore not considered in the construction of tree (b).

# Outline – Unit #06

Sample-Based Explanations

Changing Assumptions (in Plain Logic-Based XAI)

Inflated Explanations

Constrained Explanations

Distance-Restricted Explanations

Probabilistic Explanations

Additional Topics

## Not all inputs may be possible – input constraints

[GR22, YIS<sup>+</sup>23]

- The (implicit) assumption that all inputs are possible is often unrealistic
  - I.e. it may be impossible for some points in feature space to be observed

## Not all inputs may be possible – input constraints

[GR22, YIS<sup>+</sup>23]

- The (implicit) assumption that all inputs are possible is often unrealistic
  - I.e. it may be impossible for some points in feature space to be observed
- Infer constraints on the inputs
  - Learn simple rules relating inputs
  - Represent rules as a constraint set, e.g.  $\mathcal{C}(\mathbf{x})$

# Not all inputs may be possible – input constraints

[GR22, YIS<sup>+</sup>23]

- The (implicit) assumption that all inputs are possible is often unrealistic
  - I.e. it may be impossible for some points in feature space to be observed
- Infer constraints on the inputs
  - Learn simple rules relating inputs
  - Represent rules as a constraint set, e.g.  $\mathcal{C}(\mathbf{x})$
- Redefine WAXps/WCXps to account for input constraints:

$$\forall (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge \mathcal{C}(\mathbf{x}) \right] \rightarrow (\kappa(\mathbf{x}) = c)$$

$$\exists (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge \mathcal{C}(\mathbf{x}) \right] \wedge (\kappa(\mathbf{x}) \neq c)$$

- Compute AXps/CXps given new definitions

# Not all inputs may be possible – input constraints

[GR22, YIS<sup>+</sup>23]

- The (implicit) assumption that all inputs are possible is often unrealistic
  - I.e. it may be impossible for some points in feature space to be observed
- Infer constraints on the inputs
  - Learn simple rules relating inputs
  - Represent rules as a constraint set, e.g.  $\mathcal{C}(\mathbf{x})$
- Redefine WAXps/WCXps to account for input constraints:

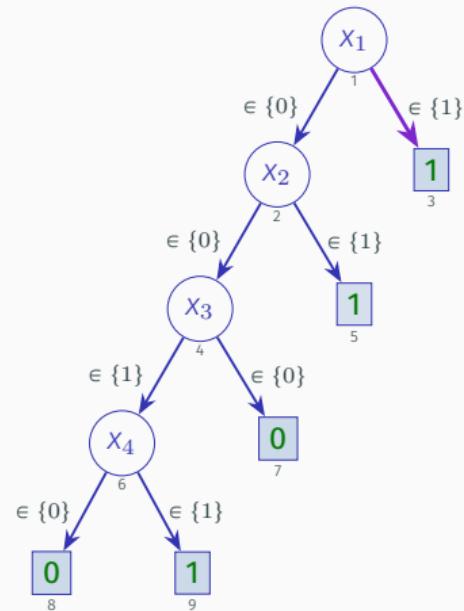
$$\forall (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge \mathcal{C}(\mathbf{x}) \right] \rightarrow (\kappa(\mathbf{x}) = c)$$

$$\exists (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge \mathcal{C}(\mathbf{x}) \right] \wedge (\kappa(\mathbf{x}) \neq c)$$

- Compute AXps/CXps given new definitions
- Constrained AXps/CXps find other applications!

## An example

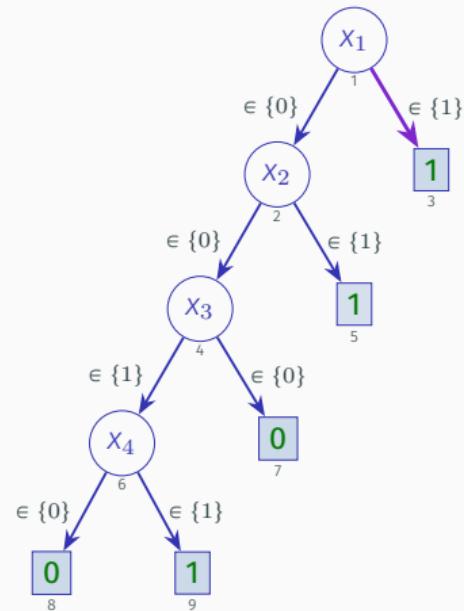
- Instance:  $((1, 1, 1, 1), 1)$
- Unconstrained AXps:



- Constraint:  $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

# An example

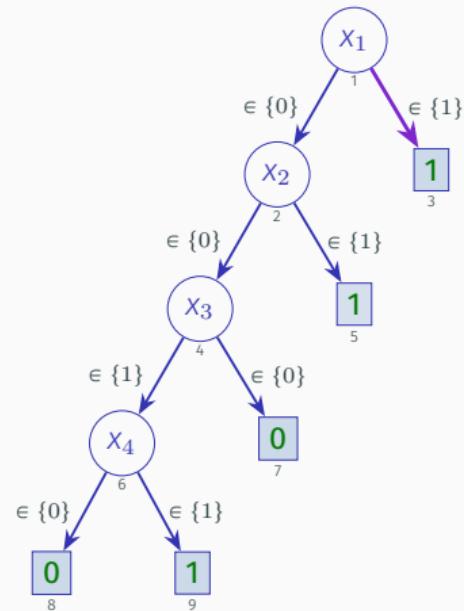
- Instance:  $((1, 1, 1, 1), 1)$
- Unconstrained AXps:
  - AXps:



- Constraint:  $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

# An example

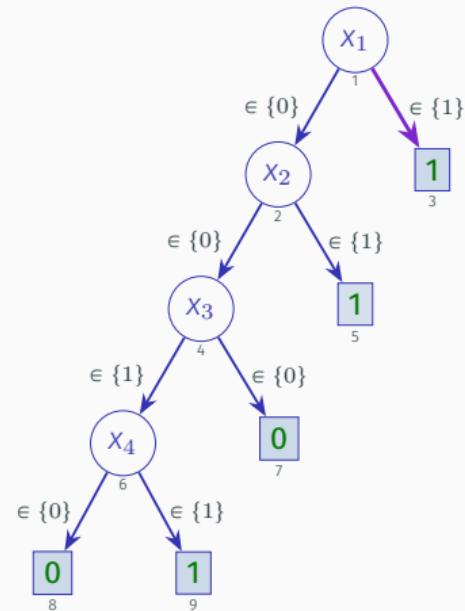
- Instance:  $((1, 1, 1, 1), 1)$
- Unconstrained AXps:
  - AXps:  $\{\{1\}, \{2\}, \{3, 4\}\}$



- Constraint:  $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

# An example

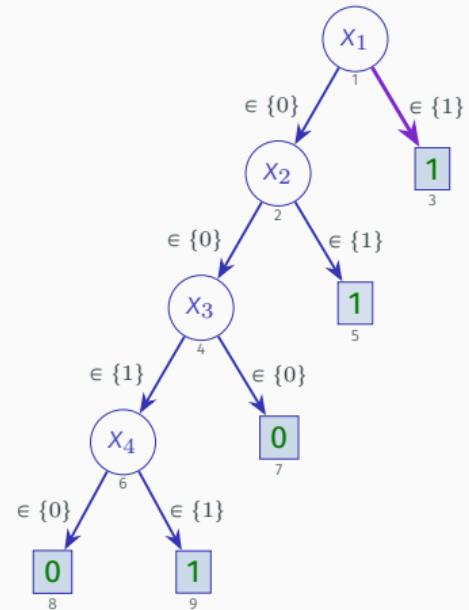
- Instance:  $((1, 1, 1, 1), 1)$
- Unconstrained AXps:
  - AXps:  $\{\{1\}, \{2\}, \{3, 4\}\}$
- Constrained AXps:



- Constraint:  $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

# An example

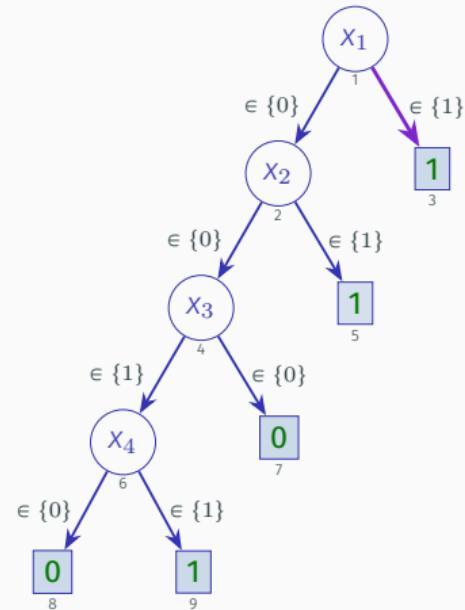
- Instance:  $((1, 1, 1, 1), 1)$
- Unconstrained AXps:
  - AXps:  $\{\{1\}, \{2\}, \{3, 4\}\}$
- Constrained AXps:
  - If feature 3 is fixed (with value 1), then feature 4 must be assigned value 1



- Constraint:  $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

# An example

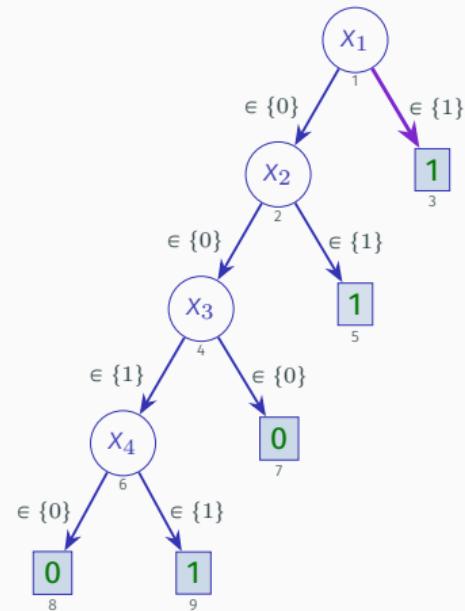
- Instance:  $((1, 1, 1, 1), 1)$
- Unconstrained AXps:
  - AXps:  $\{\{1\}, \{2\}, \{3, 4\}\}$
- Constrained AXps:
  - If feature 3 is fixed (with value 1), then feature 4 must be assigned value 1
  - If feature 4 is fixed (with value 1), then feature 3 must be assigned value 1



- Constraint:  $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

# An example

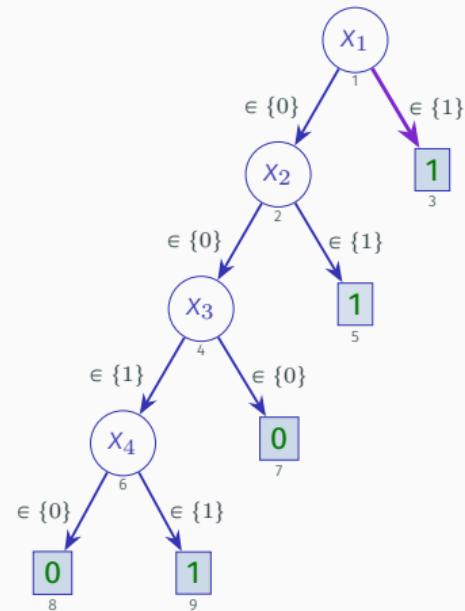
- Instance:  $((1, 1, 1, 1), 1)$
- Unconstrained AXps:
  - AXps:  $\{\{1\}, \{2\}, \{3, 4\}\}$
- Constrained AXps:
  - If feature 3 is fixed (with value 1), then feature 4 must be assigned value 1
  - If feature 4 is fixed (with value 1), then feature 3 must be assigned value 1
  - AXps:



• Constraint:  $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

# An example

- Instance:  $((1, 1, 1, 1), 1)$
- Unconstrained AXps:
  - AXps:  $\{\{1\}, \{2\}, \{3, 4\}\}$
- Constrained AXps:
  - If feature 3 is fixed (with value 1), then feature 4 must be assigned value 1
  - If feature 4 is fixed (with value 1), then feature 3 must be assigned value 1
  - AXps:  $\{\{1\}, \{2\}, \{3\}, \{4\}\}$



- Constraint:  $\{(x_3 \rightarrow x_4), (x_4 \rightarrow x_3)\}$

# Outline – Unit #06

Sample-Based Explanations

Changing Assumptions (in Plain Logic-Based XAI)

Inflated Explanations

Constrained Explanations

Distance-Restricted Explanations

Probabilistic Explanations

Additional Topics

## How to tackle poor performance on NNs?

- For NNs, computation of plain AXps scales to a few tens of neurons

[INM19a]

## How to tackle poor performance on NNs?

- For NNs, computation of plain AXps scales to a few tens of neurons
- But, robustness tools scale for much larger NNs

[INM19a]

## How to tackle poor performance on NNs?

- For NNs, computation of plain AXps scales to a few tens of neurons
- But, robustness tools scale for much larger NNs
  - Q: can we relate AXps with adversarial examples?

[INM19a]

## How to tackle poor performance on NNs?

- For NNs, computation of plain AXps scales to a few tens of neurons [INM19a]
- But, robustness tools scale for much larger NNs
  - Q: can we relate AXps with adversarial examples?
  - Obs: we already proved some basic (duality) properties for **global** explanations [INM19b]

# How to tackle poor performance on NNs?

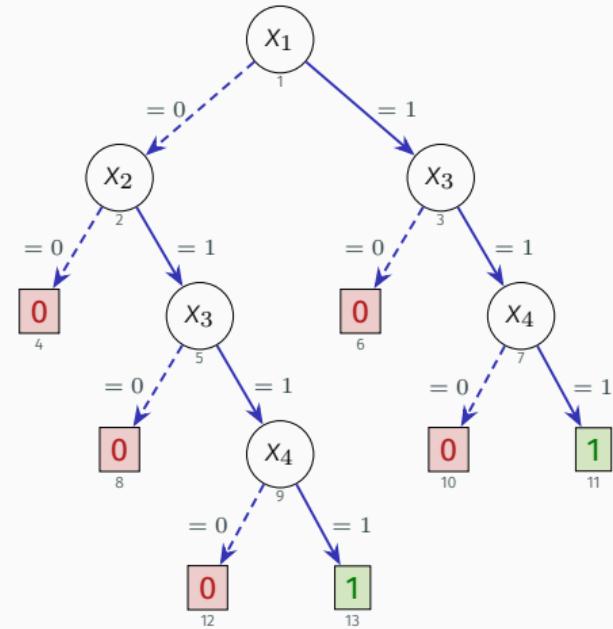
- For NNs, computation of plain AXps scales to a few tens of neurons [INM19a]
- But, robustness tools scale for much larger NNs
  - Q: can we relate AXps with adversarial examples?
  - Obs: we already proved some basic (duality) properties for **global** explanations [INM19b]
- Change definition of WAXp/WCXp to account for  $l_p$  distance to  $\mathbf{v}$ :

$$\forall (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\|\mathbf{x} - \mathbf{v}\|_{l_p} \leq \epsilon) \right] \rightarrow (\sigma(\mathbf{x}))$$

$$\exists (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\|\mathbf{x} - \mathbf{v}\|_{l_p} \leq \epsilon) \right] \wedge (\neg \sigma(\mathbf{x}))$$

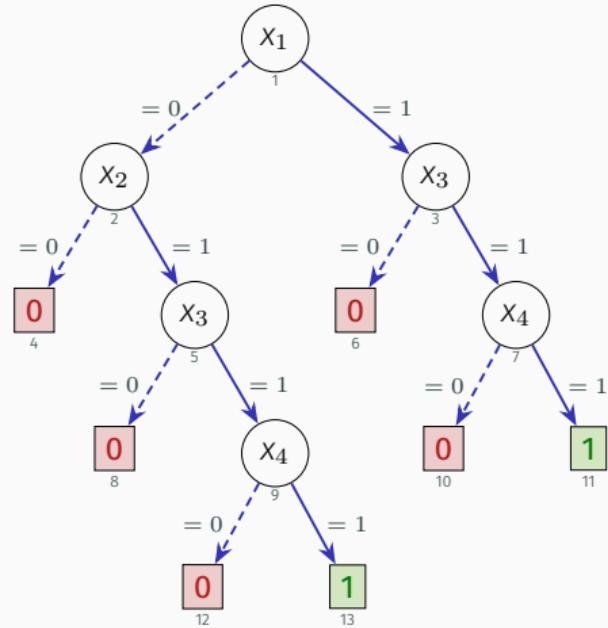
- Norm  $l_p$  is arbitrary, e.g. Hamming, Manhattan, Euclidean, etc.
- **Distance-restricted explanations:**  $\partial \text{AXp} / \partial \text{CXp}$

## An example – DT & instance $((1, 1, 1, 1), 1)$



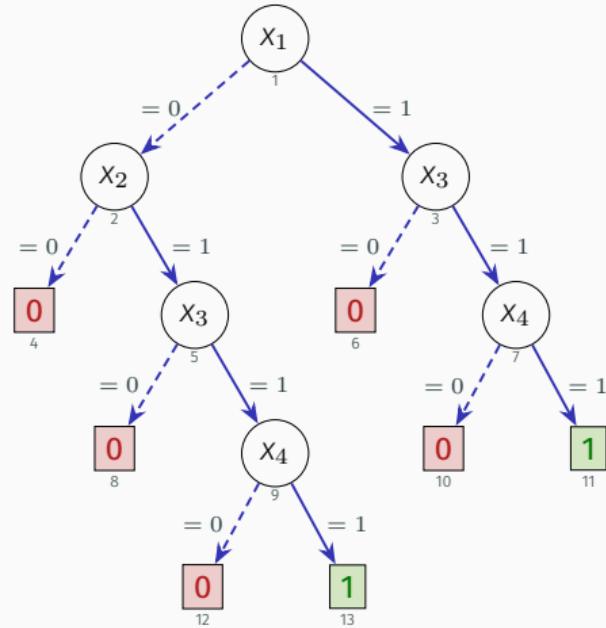
## An example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:



## An example – DT & instance $((1, 1, 1, 1), 1)$

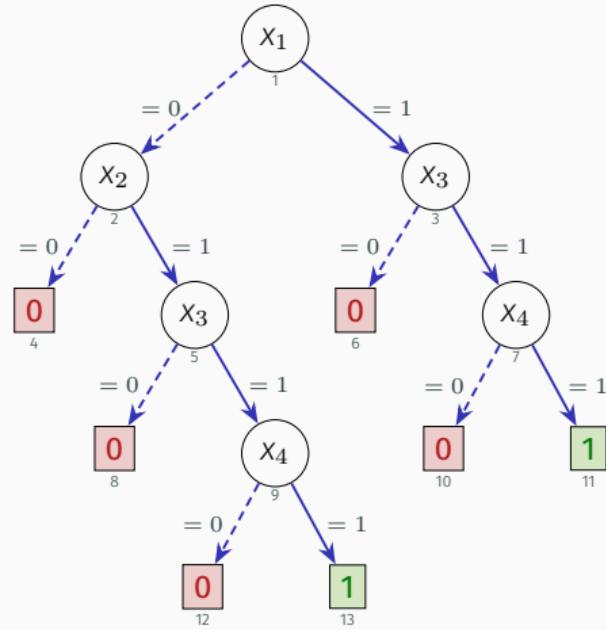
- Plain AXps/CXps:
  - AXps?



## An example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:

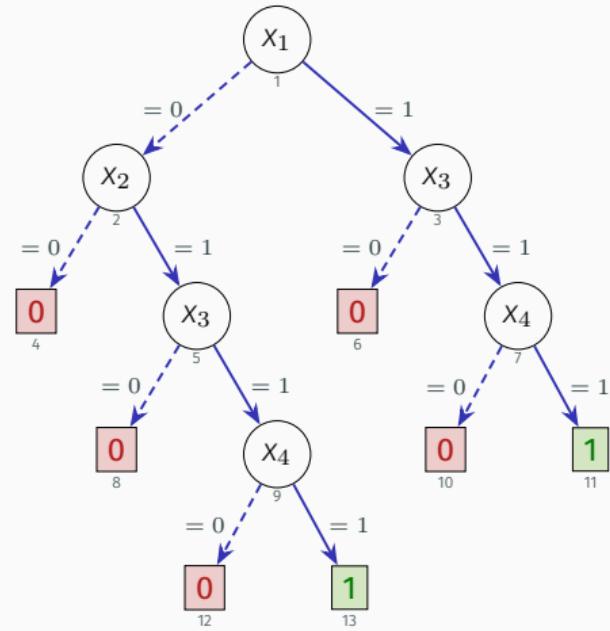
- AXps?  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
- CXps?



## An example – DT & instance $((1, 1, 1, 1), 1)$

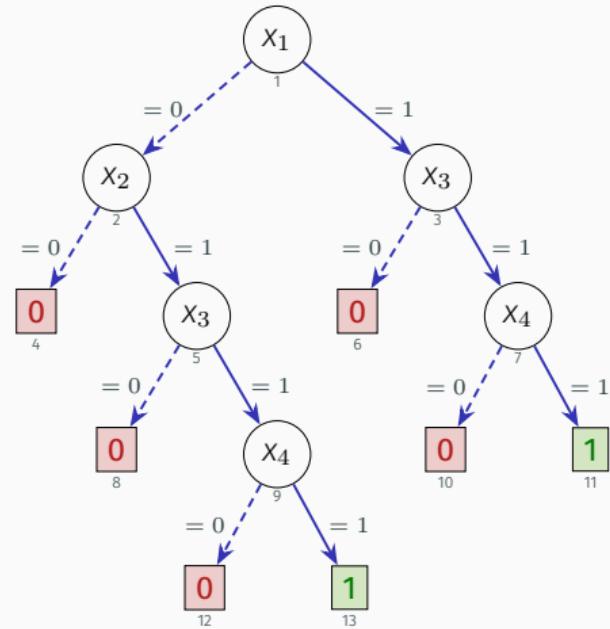
- Plain AXps/CXps:

- AXps?  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
- CXps?  $\{\{1, 2\}, \{3\}, \{4\}\}$



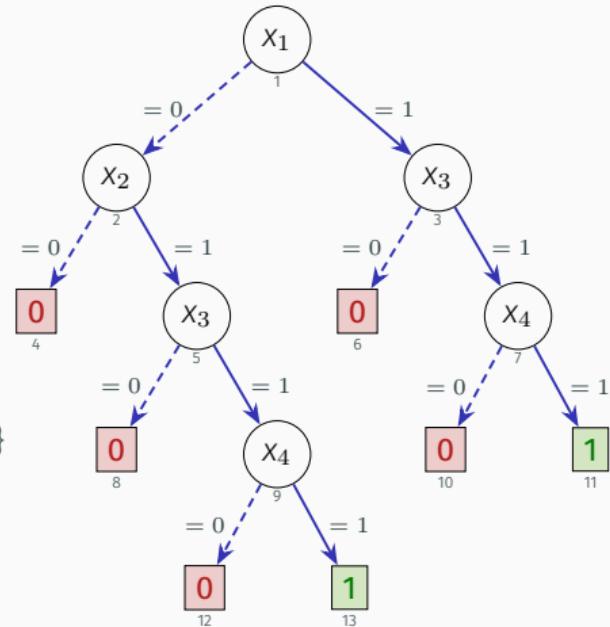
## An example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
  - CXps?  $\{\{1, 2\}, \{3\}, \{4\}\}$
- Distance-restricted AXps/CXps,  $\partial$ AXp/ $\partial$ CXp, with Hamming distance ( $l_0$ ) and  $\epsilon = 1$ :



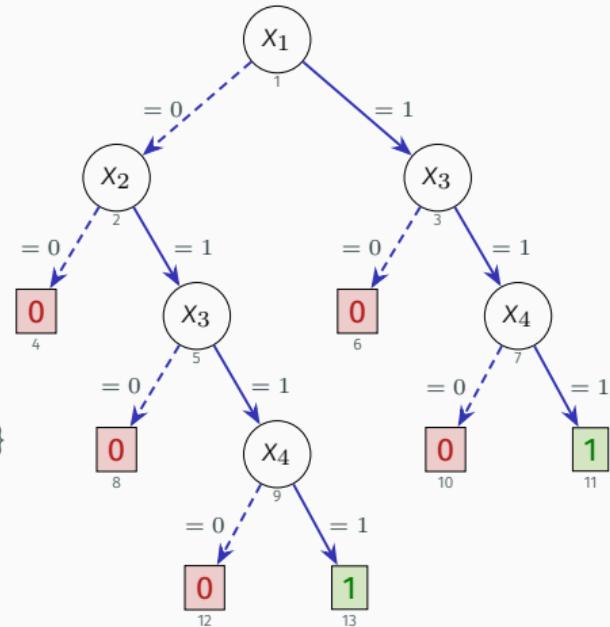
## An example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
  - CXps?  $\{\{1, 2\}, \{3\}, \{4\}\}$
- Distance-restricted AXps/CXps,  $\partial$ AXp/ $\partial$ CXp, with Hamming distance ( $l_0$ ) and  $\epsilon = 1$ :
  - Points of interest:  
 $\{(1, 1, 1, 1), (0, 1, 1, 1), (1, 0, 1, 1), (1, 1, 0, 1), (1, 1, 1, 0)\}$



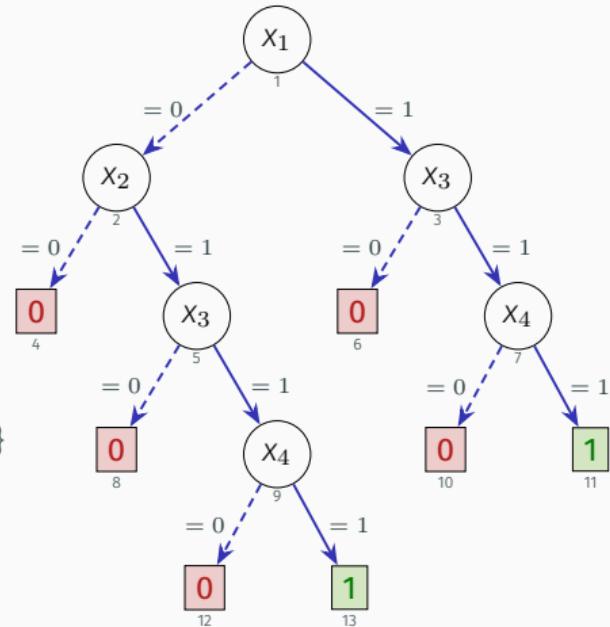
## An example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
  - CXps?  $\{\{1, 2\}, \{3\}, \{4\}\}$
- Distance-restricted AXps/CXps,  $\partial$ AXp/ $\partial$ CXp, with Hamming distance ( $l_0$ ) and  $\epsilon = 1$ :
  - Points of interest:  
 $\{(1, 1, 1, 1), (0, 1, 1, 1), (1, 0, 1, 1), (1, 1, 0, 1), (1, 1, 1, 0)\}$
  - $\partial$ AXps?



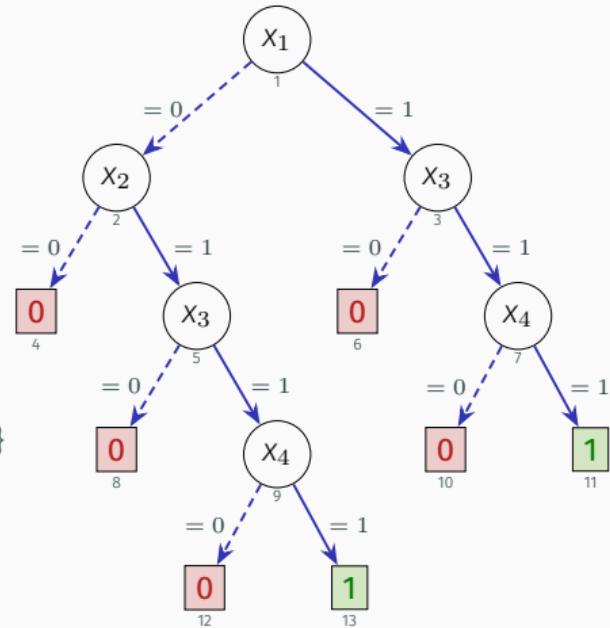
## An example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
  - CXps?  $\{\{1, 2\}, \{3\}, \{4\}\}$
- Distance-restricted AXps/CXps,  $\partial$ AXp/ $\partial$ CXp, with Hamming distance ( $l_0$ ) and  $\epsilon = 1$ :
  - Points of interest:  
 $\{(1, 1, 1, 1), (0, 1, 1, 1), (1, 0, 1, 1), (1, 1, 0, 1), (1, 1, 1, 0)\}$
  - $\partial$ AXps?  $\{\{3, 4\}\}$
  - $\partial$ CXps?



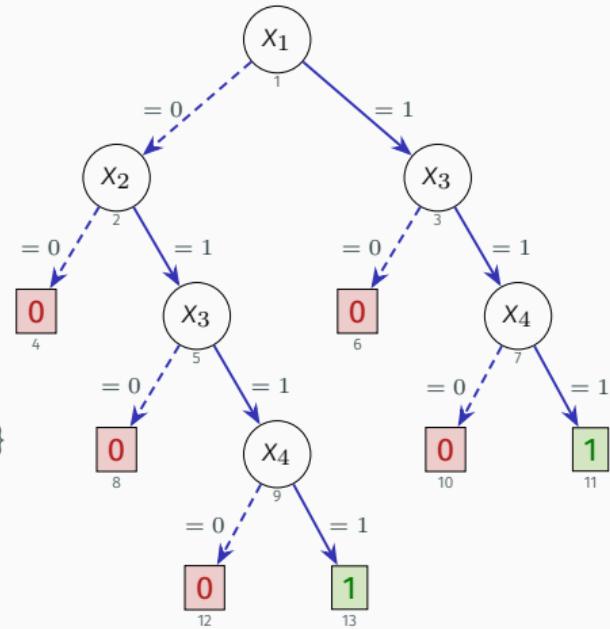
## An example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
  - CXps?  $\{\{1, 2\}, \{3\}, \{4\}\}$
- Distance-restricted AXps/CXps,  $\partial$ AXp/ $\partial$ CXp, with Hamming distance ( $l_0$ ) and  $\epsilon = 1$ :
  - Points of interest:  
 $\{(1, 1, 1, 1), (0, 1, 1, 1), (1, 0, 1, 1), (1, 1, 0, 1), (1, 1, 1, 0)\}$
  - $\partial$ AXps?  $\{\{3, 4\}\}$
  - $\partial$ CXps?  $\{\{3\}, \{4\}\}$

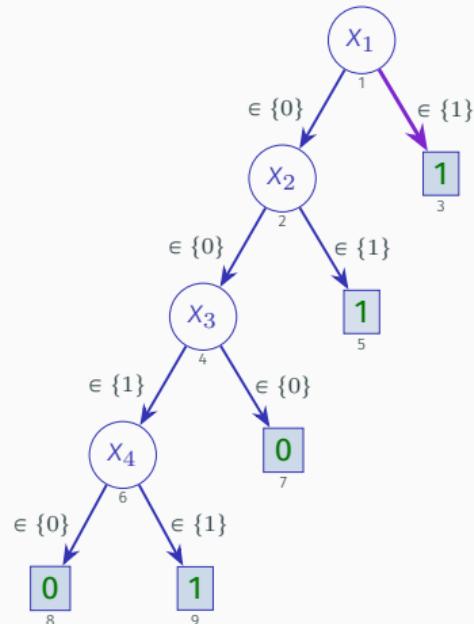


## An example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
  - CXps?  $\{\{1, 2\}, \{3\}, \{4\}\}$
- Distance-restricted AXps/CXps,  $\partial$ AXp/ $\partial$ CXp, with Hamming distance ( $l_0$ ) and  $\epsilon = 1$ :
  - Points of interest:  
 $\{(1, 1, 1, 1), (0, 1, 1, 1), (1, 0, 1, 1), (1, 1, 0, 1), (1, 1, 1, 0)\}$
  - $\partial$ AXps?  $\{\{3, 4\}\}$
  - $\partial$ CXps?  $\{\{3\}, \{4\}\}$
- Given  $\epsilon$ , larger adversarial examples are excluded

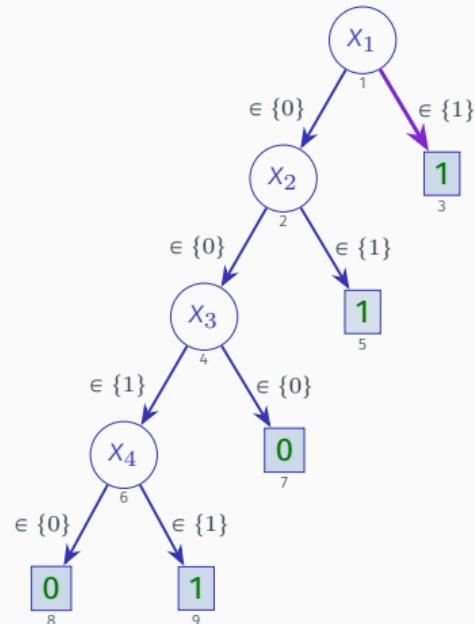


## Another example – DT & instance $((1, 1, 1, 1), 1)$



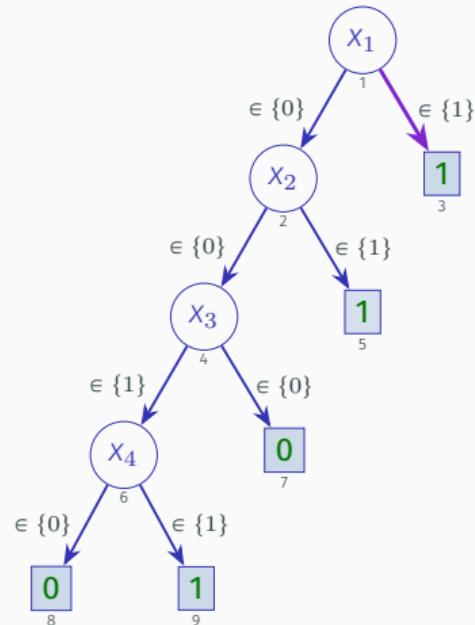
## Another example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:



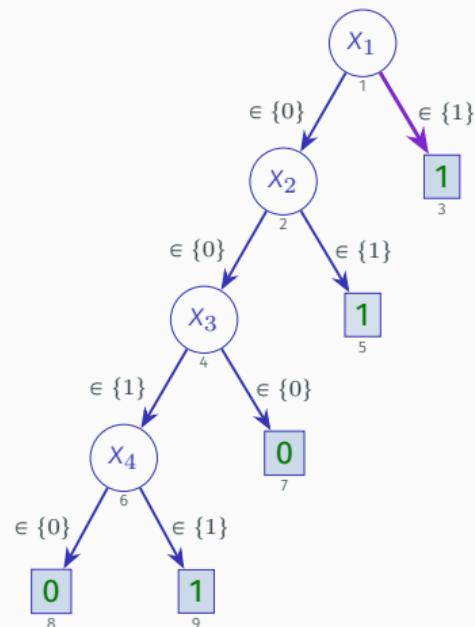
## Another example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?



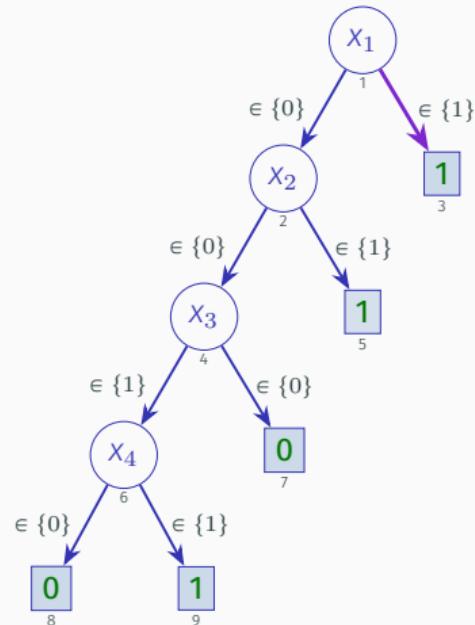
## Another example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1\}, \{2\}, \{3, 4\}\}$
  - CXps?



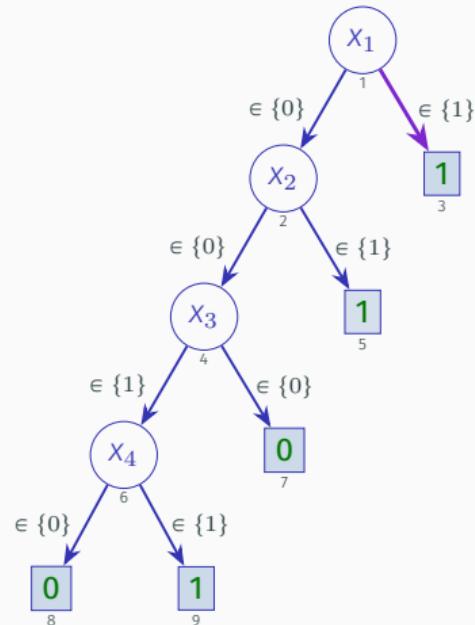
## Another example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1\}, \{2\}, \{3, 4\}\}$
  - CXps?  $\{\{1, 2, 3\}, \{1, 2, 4\}\}$



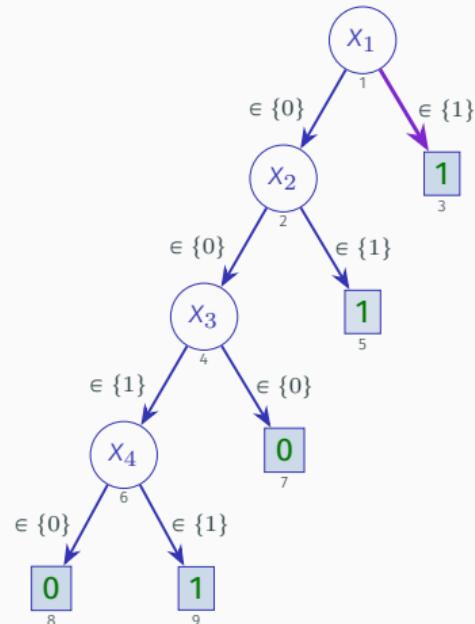
## Another example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1\}, \{2\}, \{3, 4\}\}$
  - CXps?  $\{\{1, 2, 3\}, \{1, 2, 4\}\}$
- Distance-restricted AXps/CXps,  $\partial$ AXp/ $\partial$ CXp, with Hamming distance ( $l_0$ ) and  $\epsilon = 1$ :
  - Points of interest:  
 $\{(1, 1, 1, 1), (0, 1, 1, 1), (1, 0, 1, 1), (1, 1, 0, 1), (1, 1, 1, 0)\}$



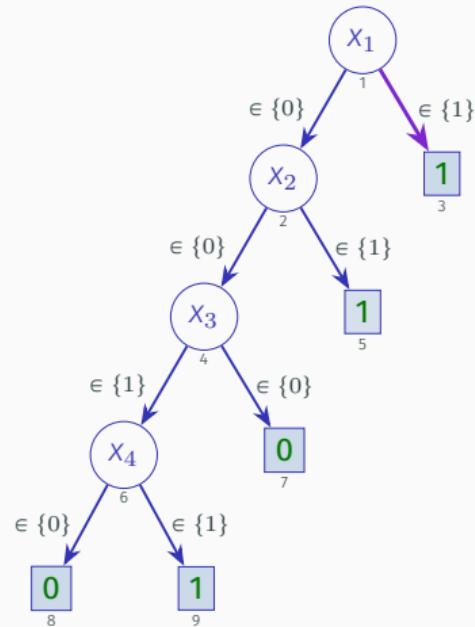
## Another example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1\}, \{2\}, \{3, 4\}\}$
  - CXps?  $\{\{1, 2, 3\}, \{1, 2, 4\}\}$
- Distance-restricted AXps/CXps,  $\partial$ AXp/ $\partial$ CXp, with Hamming distance ( $l_0$ ) and  $\epsilon = 1$ :
  - Points of interest:  
 $\{(1, 1, 1, 1), (0, 1, 1, 1), (1, 0, 1, 1), (1, 1, 0, 1), (1, 1, 1, 0)\}$
  - Constant function...



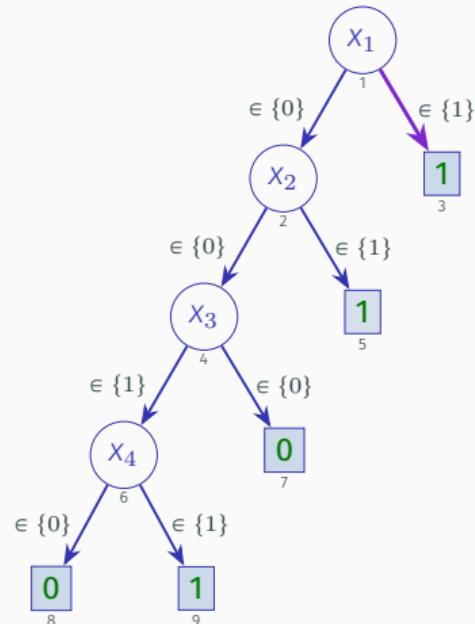
## Another example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1\}, \{2\}, \{3, 4\}\}$
  - CXps?  $\{\{1, 2, 3\}, \{1, 2, 4\}\}$
- Distance-restricted AXps/CXps,  $\partial$ AXp/ $\partial$ CXp, with Hamming distance ( $l_0$ ) and  $\epsilon = 1$ :
  - Points of interest:  
 $\{(1, 1, 1, 1), (0, 1, 1, 1), (1, 0, 1, 1), (1, 1, 0, 1), (1, 1, 1, 0)\}$
  - Constant function...
  - $\partial$ AXps?



## Another example – DT & instance $((1, 1, 1, 1), 1)$

- Plain AXps/CXps:
  - AXps?  $\{\{1\}, \{2\}, \{3, 4\}\}$
  - CXps?  $\{\{1, 2, 3\}, \{1, 2, 4\}\}$
- Distance-restricted AXps/CXps,  $\partial$ AXp/ $\partial$ CXp, with Hamming distance ( $l_0$ ) and  $\epsilon = 1$ :
  - Points of interest:  
 $\{(1, 1, 1, 1), (0, 1, 1, 1), (1, 0, 1, 1), (1, 1, 0, 1), (1, 1, 1, 0)\}$
  - Constant function...
  - $\partial$ AXps?  $\{\emptyset\}$



## Relating explanations with adversarial examples

- Distance-restricted WAXps/WCXps:

$$\forall (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\|\mathbf{x} - \mathbf{v}\|_{l_p} \leq \epsilon) \right] \rightarrow (\sigma(\mathbf{x}))$$

$$\exists (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\|\mathbf{x} - \mathbf{v}\|_{l_p} \leq \epsilon) \right] \wedge (\neg\sigma(\mathbf{x}))$$

## Relating explanations with adversarial examples

- Distance-restricted WAXps/WCXps:

$$\forall(\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\|\mathbf{x} - \mathbf{v}\|_{l_p} \leq \epsilon) \right] \rightarrow (\sigma(\mathbf{x}))$$

$$\exists(\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\|\mathbf{x} - \mathbf{v}\|_{l_p} \leq \epsilon) \right] \wedge (\neg\sigma(\mathbf{x}))$$

- Given norm  $l_p$  and distance  $\epsilon$ , there exists a (distance-restricted) WCXp iff there exists an adversarial example
  - Use robustness tool to decide existence of WCXp
  - But, WAXp decided given non existence of CXp!

# Relating explanations with adversarial examples

- Distance-restricted WAXps/WCXps:

$$\forall (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\|\mathbf{x} - \mathbf{v}\|_{l_p} \leq \epsilon) \right] \rightarrow (\sigma(\mathbf{x}))$$

$$\exists (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\|\mathbf{x} - \mathbf{v}\|_{l_p} \leq \epsilon) \right] \wedge (\neg\sigma(\mathbf{x}))$$

- Given norm  $l_p$  and distance  $\epsilon$ , there exists a (distance-restricted) WCXp iff there exists an adversarial example
  - Use robustness tool to decide existence of WCXp
  - But, WAXp decided given non existence of CXp!
- Efficiency of distance-restricted explanations correlates with efficiency of finding adversarial examples
  - One can use most complete robustness tools, e.g. VNN-COMP

[BMB<sup>+</sup>23]

# Relating explanations with adversarial examples

- Distance-restricted WAXps/WCXps:

$$\forall (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\|\mathbf{x} - \mathbf{v}\|_{l_p} \leq \epsilon) \right] \rightarrow (\sigma(\mathbf{x}))$$

$$\exists (\mathbf{x} \in \mathbb{F}). \left[ \bigwedge_{j \in \mathcal{X}} (x_j = v_j) \wedge (\|\mathbf{x} - \mathbf{v}\|_{l_p} \leq \epsilon) \right] \wedge (\neg\sigma(\mathbf{x}))$$

- Given norm  $l_p$  and distance  $\epsilon$ , there exists a (distance-restricted) WCXP iff there exists an adversarial example
  - Use robustness tool to decide existence of WCXP
  - But, WAXp decided given non existence of CXp!
- Efficiency of distance-restricted explanations correlates with efficiency of finding adversarial examples
  - One can use most complete robustness tools, e.g. VNN-COMP [BMB<sup>+</sup>23]
- Clear scalability improvements for explaining NNs (see next) [HM23b, WWB23, IHM<sup>+</sup>24a, IHM<sup>+</sup>24b]

Input: Arguments:  $\epsilon$ ; Parameters:  $\mathcal{E}, p$

Output: One  $\text{dAXp } \mathcal{S}$

```
1: function FindAXpDel( $\epsilon; \mathcal{E}, p$ )
2:    $\mathcal{S} \leftarrow \mathcal{F}$ 
3:   for  $i \in \mathcal{F}$  do
4:      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
5:     outc  $\leftarrow$  FindAdvEx( $\epsilon, \mathcal{S}; \mathcal{E}, p$ )
6:     if outc then
7:        $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$ 
8:   return  $\mathcal{S}$ 
```

$\triangleright$  Initially, no feature is allowed to change

$\triangleright$  Invariant:  $\text{dWAXp}(\mathcal{S})$

$\triangleright \text{dWAXp}(\mathcal{S}) \wedge \text{minimal}(\mathcal{S}) \rightarrow \text{dAXp}(\mathcal{S})$

Input: Arguments:  $\epsilon$ ; Parameters:  $\mathcal{E}, p$

Output: One  $\text{dAXp } \mathcal{S}$

```
1: function FindAXpDel( $\epsilon; \mathcal{E}, p$ )
2:    $\mathcal{S} \leftarrow \mathcal{F}$                                  $\triangleright$  Initially, no feature is allowed to change
3:   for  $i \in \mathcal{F}$  do                          $\triangleright$  Invariant:  $\text{dWAXp}(\mathcal{S})$ 
4:      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
5:     outc  $\leftarrow$  FindAdvEx( $\epsilon, \mathcal{S}; \mathcal{E}, p$ )
6:     if outc then
7:        $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$ 
8:   return  $\mathcal{S}$                                  $\triangleright \text{dWAXp}(\mathcal{S}) \wedge \text{minimal}(\mathcal{S}) \rightarrow \text{dAXp}(\mathcal{S})$ 
```

- **Obs:** Efficiency of logic-based XAI tracks efficiency of robustness tools

Input: Arguments:  $\epsilon$ ; Parameters:  $\mathcal{E}, p$

Output: One  $\text{dAXp } \mathcal{S}$

```
1: function FindAXpDel( $\epsilon; \mathcal{E}, p$ )
2:    $\mathcal{S} \leftarrow \mathcal{F}$                                  $\triangleright$  Initially, no feature is allowed to change
3:   for  $i \in \mathcal{F}$  do                          $\triangleright$  Invariant:  $\text{dWAXp}(\mathcal{S})$ 
4:      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
5:     outc  $\leftarrow$  FindAdvEx( $\epsilon, \mathcal{S}; \mathcal{E}, p$ )
6:     if outc then
7:        $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$ 
8:   return  $\mathcal{S}$                                  $\triangleright \text{dWAXp}(\mathcal{S}) \wedge \text{minimal}(\mathcal{S}) \rightarrow \text{dAXp}(\mathcal{S})$ 
```

- **Obs:** Efficiency of logic-based XAI tracks efficiency of robustness tools
- **Limitation:** Running time grows with number of features

# Results for NNs in 2023 (using Marabou [KHI<sup>+</sup>19])

[HM23b]

DNN	points	AXp	#Calls	Time	#TO	AXp	#Calls	Time	#TO
$\epsilon = 0.1$					$\epsilon = 0.05$				
ACASXU_1_5	#1	3	5	185.9	0	2	5	113.8	0
	#2	2	5	273.8	0	1	5	33.2	0
	#3	0	5	714.2	0	0	5	4.3	0
ACASXU_3_1	#1	0	5	2219.3	0	0	5	14.2	0
	#2	2	5	4263.5	1	0	5	1853.1	0
	#3	1	5	581.8	0	0	5	355.9	0
ACASXU_3_2	#1	3	5	13739.3	2	1	5	6890.1	1
	#2	3	5	226.4	0	2	5	125.1	0
	#3	2	5	1740.6	0	2	5	173.6	0
ACASXU_3_5	#1	4	5	43.6	0	2	5	59.4	0
	#2	3	5	5039.4	0	2	5	4303.8	1
	#3	2	5	5574.9	1	2	5	2660.3	0
ACASXU_3_6	#1	1	5	6225.0	1	0	5	51.0	0
	#2	3	5	4957.2	1	2	5	1897.3	0
	#3	1	5	196.1	0	1	5	919.2	0
ACASXU_3_7	#1	3	5	6256.2	0	4	5	26.9	0
	#2	4	5	311.3	0	1	5	6958.6	1
	#3	2	5	7756.5	1	1	5	7807.6	1
ACASXU_4_1	#1	2	5	12413.0	2	1	5	5090.5	1
	#2	1	5	5035.1	1	0	5	2335.6	0
	#3	4	5	1237.3	0	4	5	1143.4	0
ACASXU_4_2	#1	4	5	15.9	0	4	5	12.1	0
	#2	3	5	1507.6	0	1	5	111.3	0
	#3	2	5	5641.6	2	0	5	1639.1	0

# Results for NNs in 2023 (using Marabou [KHI<sup>+</sup>19])

[HM23b]

DNN	points	AXp	#Calls	Time	#TO	AXp	#Calls	Time	#TO
$\epsilon = 0.1$					$\epsilon = 0.05$				
ACASXU_1_5	#1	3	5	185.9	0	2	5	113.8	0
	#2	2	5	273.8	0	1	5	33.2	0
	#3	0	5	714.2	0	0	5	4.3	0
ACASXU_3_1	#1	0	5	2219.3	0	0	5	14.2	0
	#2	2	5	4263.5	1	0	5	1853.1	0
	#3	1	5	581.8	0	0	5	355.9	0
ACASXU_3_2	#1	3	5	13739.3	2	1	5	6890.1	1
	#2	3	5	226.4	0	2	5	125.1	0
	#3	2	5	1740.6	0	2	5	173.6	0
ACASXU_3_5	#1	4	5	43.6	0	2	5	59.4	0
	#2	3	5	5039.4	0	2	5	4303.8	1
	#3	2	5	5574.9	1	2	5	2660.3	0
ACASXU_3_6	#1	1	5	6225.0	1	0	5	51.0	0
	#2	3	5	4957.2	1	2	5	1897.3	0
	#3	1	5	196.1	0	1	5	919.2	0
ACASXU_3_7	#1	3	5	6256.2	0	4	5	26.9	0
	#2	4	5	311.3	0	1	5	6958.6	1
	#3	2	5	7756.5	1	1	5	7807.6	1
ACASXU_4_1	#1	2	5	12413.0	2	1	5	5090.5	1
	#2	1	5	5035.1	1	0	5	2335.6	0
	#3	4	5	1237.3	0	4	5	1143.4	0
ACASXU_4_2	#1	4	5	15.9	0	4	5	12.1	0
	#2	3	5	1507.6	0	1	5	111.3	0
	#3	2	5	5641.6	2	0	5	1639.1	0

Scales to a few hundred neurons

# Recent improvements

Input: Arguments:  $\epsilon$ ; Parameters:  $\mathcal{E}, p$

Output: One  $\text{dAXp } \mathcal{S}$

```
1: function FindAXpDel( $\epsilon; \mathcal{E}, p$ )
2:    $\mathcal{S} \leftarrow \mathcal{F}$                                  $\triangleright$  Initially, no feature is allowed to change
3:   for  $i \in \mathcal{F}$  do                          $\triangleright$  Invariant:  $\text{dWAXp}(\mathcal{S})$ 
4:      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
5:     outc  $\leftarrow$  FindAdvEx( $\epsilon, \mathcal{S}; \mathcal{E}, p$ )
6:     if outc then
7:        $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$ 
8:   return  $\mathcal{S}$                                  $\triangleright \text{dWAXp}(\mathcal{S}) \wedge \text{minimal}(\mathcal{S}) \rightarrow \text{dAXp}(\mathcal{S})$ 
```

# Recent improvements

Input: Arguments:  $\epsilon$ ; Parameters:  $\mathcal{E}, p$

Output: One  $\text{dAXp } \mathcal{S}$

```
1: function FindAXpDel( $\epsilon; \mathcal{E}, p$ )
2:    $\mathcal{S} \leftarrow \mathcal{F}$                                  $\triangleright$  Initially, no feature is allowed to change
3:   for  $i \in \mathcal{F}$  do                          $\triangleright$  Invariant:  $\text{dWAXp}(\mathcal{S})$ 
4:      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
5:     outc  $\leftarrow$  FindAdvEx( $\epsilon, \mathcal{S}; \mathcal{E}, p$ )
6:     if outc then
7:        $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$ 
8:   return  $\mathcal{S}$                                  $\triangleright \text{dWAXp}(\mathcal{S}) \wedge \text{minimal}(\mathcal{S}) \rightarrow \text{dAXp}(\mathcal{S})$ 
```

- To drop features from  $\mathcal{S} \subseteq \mathcal{F}$ , it is open whether parallelization might be applicable
  - Algorithm FindAXpDel is mostly sequential (see above)
  - Exploit parallelization for other algorithms, e.g. [dichotomic search](#)

[IHM<sup>+</sup> 24b]

# Recent improvements

Input: Arguments:  $\epsilon$ ; Parameters:  $\mathcal{E}, p$

Output: One  $\text{dAXp } \mathcal{S}$

```
1: function FindAXpDel( $\epsilon; \mathcal{E}, p$ )
2:    $\mathcal{S} \leftarrow \mathcal{F}$                                  $\triangleright$  Initially, no feature is allowed to change
3:   for  $i \in \mathcal{F}$  do                          $\triangleright$  Invariant:  $\text{dWAXp}(\mathcal{S})$ 
4:      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{i\}$ 
5:     outc  $\leftarrow$  FindAdvEx( $\epsilon, \mathcal{S}; \mathcal{E}, p$ )
6:     if outc then
7:        $\mathcal{S} \leftarrow \mathcal{S} \cup \{i\}$ 
8:   return  $\mathcal{S}$                                  $\triangleright \text{dWAXp}(\mathcal{S}) \wedge \text{minimal}(\mathcal{S}) \rightarrow \text{dAXp}(\mathcal{S})$ 
```

- To drop features from  $\mathcal{S} \subseteq \mathcal{F}$ , it is open whether parallelization might be applicable
  - Algorithm FindAXpDel is mostly sequential (see above)
  - Exploit parallelization for other algorithms, e.g. [dichotomic search](#) [IHM+ 24b]
- However, to decide whether  $\mathcal{S}$  is an AXp, we can exploit parallelization:
  - Recall:  $\text{AXp}(\mathcal{X}) := \text{WAXp}(\mathcal{X}) \wedge \forall(t \in \mathcal{X}). \neg \text{WAXp}(\mathcal{X} \setminus \{t\})$
  - Each  $\neg \text{WAXp}(\cdot)$  (and also  $\text{WAXp}(\cdot)$ ) check can be run in parallel!
  - Do this opportunistically, i.e. when set  $\mathcal{S}$  is expected to be AXp [IHM+ 24b]

# More recent results (from 2024)...

[IHM<sup>+</sup> 24a, IHM<sup>+</sup> 24b]

Model	Deletion							SwiftXplain						
	avgC	nCalls	Len	Mn	Mx	avg	TO	avgC	nCalls	Len	FD%	Mn	Mx	avg
gtsrb-dense	0.06	1024	448	52.0	76.3	63.1	0	0.23	54	447	77.4	10.8	14.0	12.2
gtsrb-convSmall	0.06	1024	309	59.2	82.6	65.1	0	0.22	74	313	39.7	15.1	19.5	16.2
gtsrb-conv	—	—	—	—	—	—	100	96.49	45	174	33.2	3858.7	6427.7	4449.4
mnist-denseSmall	0.28	784	177	190.9	420.3	220.4	0	0.77	111	180	15.5	77.6	104.4	85.1
mnist-dense	0.19	784	231	138.1	179.9	150.6	0	0.75	183	229	11.5	130.1	145.5	136.8
mnist-convSmall	—	—	—	—	—	—	100	98.56	52	116	21.3	4115.2	6858.3	5132.8

## More recent results (from 2024)...

[IHM<sup>+</sup> 24a, IHM<sup>+</sup> 24b]

Model	Deletion							SwiftXplain						
	avgC	nCalls	Len	Mn	Mx	avg	TO	avgC	nCalls	Len	FD%	Mn	Mx	avg
gtsrb-dense	0.06	1024	448	52.0	76.3	63.1	0	0.23	54	447	77.4	10.8	14.0	12.2
gtsrb-convSmall	0.06	1024	309	59.2	82.6	65.1	0	0.22	74	313	39.7	15.1	19.5	16.2
gtsrb-conv	—	—	—	—	—	—	100	96.49	45	174	33.2	3858.7	6427.7	4449.4
mnist-denseSmall	0.28	784	177	190.9	420.3	220.4	0	0.77	111	180	15.5	77.6	104.4	85.1
mnist-dense	0.19	784	231	138.1	179.9	150.6	0	0.75	183	229	11.5	130.1	145.5	136.8
mnist-convSmall	—	—	—	—	—	—	100	98.56	52	116	21.3	4115.2	6858.3	5132.8

Scales to **tens of thousands** of neurons!

## More recent results (from 2024)...

[IHM+ 24a, IHM+ 24b]

Model	Deletion								SwiftXplain							
	avgC	nCalls	Len	Mn	Mx	avg	TO	avgC	nCalls	Len	FD%	Mn	Mx	avg		
gtsrb-dense	0.06	1024	448	52.0	76.3	63.1	0	0.23	54	447	77.4	10.8	14.0	12.2		
gtsrb-convSmall	0.06	1024	309	59.2	82.6	65.1	0	0.22	74	313	39.7	15.1	19.5	16.2		
gtsrb-conv	—	—	—	—	—	—	100	96.49	45	174	33.2	3858.7	6427.7	4449.4		
mnist-denseSmall	0.28	784	177	190.9	420.3	220.4	0	0.77	111	180	15.5	77.6	104.4	85.1		
mnist-dense	0.19	784	231	138.1	179.9	150.6	0	0.75	183	229	11.5	130.1	145.5	136.8		
mnist-convSmall	—	—	—	—	—	—	100	98.56	52	116	21.3	4115.2	6858.3	5132.8		

Scales to **tens of thousands** of neurons!

Largest for MNIST: 10142 neurons  
Largest for GSTRB: 94308 neurons

# Outline – Unit #06

Sample-Based Explanations

Changing Assumptions (in Plain Logic-Based XAI)

Inflated Explanations

Constrained Explanations

Distance-Restricted Explanations

Probabilistic Explanations

Additional Topics

# Probabilistic (formal) explanations

[WMHK21, IIN<sup>+</sup>22, IH<sup>+</sup>22, ABOS22, IH<sup>+</sup>23, IMM24]

- Explanation size is critical for human understanding [Mil56]
- Probabilistic explanations provide smaller explanations, by trading off rigor of explanation by explanation size

# Probabilistic (formal) explanations

[WMHK21, IIN<sup>+</sup>22, IH<sup>+</sup>22, ABOS22, IH<sup>+</sup>23, IMM24]

- Explanation size is critical for human understanding [Mil56]
- Probabilistic explanations provide smaller explanations, by trading off rigor of explanation by explanation size
- Definition of weak probabilistic AXp  $\mathcal{X} \subseteq \mathcal{F}$ :

$$\text{WPAXp}(\mathcal{X}) \quad := \quad \Pr(\kappa(\mathbf{x}) = c \mid \mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}) \geq \delta$$

- Obs:  $\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}$  requires points  $\mathbf{x} \in \mathbb{F}$  to match the values of  $\mathbf{v}$  for the features dictated by  $\mathcal{X}$
- Obs: for  $\delta = 1$  we obtain a WAXp

# Definitions

## Definitions

- Weak probabilistic AXp (WPAXp):

$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) :=$

$$\Pr_{\mathbf{x}}(\kappa(\mathbf{x}) = c \mid \mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}) \geq \delta := \frac{|\{\mathbf{x} \in \mathbb{F} : \kappa(\mathbf{x}) = c \wedge (\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}})\}|}{|\{\mathbf{x} \in \mathbb{F} : (\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}})\}|} \geq \delta$$

## Definitions

- Weak probabilistic AXp (WPAXp):

$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) :=$

$$\Pr_{\mathbf{x}}(\kappa(\mathbf{x}) = c \mid \mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}) \geq \delta := \frac{|\{\mathbf{x} \in \mathbb{F} : \kappa(\mathbf{x}) = c \wedge (\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}})\}|}{|\{\mathbf{x} \in \mathbb{F} : (\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}})\}|} \geq \delta$$

- Probabilistic AXp (PAXp):

$\text{PAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) :=$

$$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) \wedge \forall (\mathcal{X}' \subsetneq \mathcal{X}). \neg \text{WeakPAXp}(\mathcal{X}'; \mathbb{F}, \kappa, \mathbf{v}, c, \delta)$$

## Definitions

- Weak probabilistic AXp (WPAXp):

$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) :=$

$$\Pr_{\mathbf{x}}(\kappa(\mathbf{x}) = c \mid \mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}) \geq \delta := \frac{|\{\mathbf{x} \in \mathbb{F} : \kappa(\mathbf{x}) = c \wedge (\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}})\}|}{|\{\mathbf{x} \in \mathbb{F} : (\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}})\}|} \geq \delta$$

- Probabilistic AXp (PAXp):

$\text{PAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) :=$

$$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) \wedge \forall(\mathcal{X}' \subsetneq \mathcal{X}). \neg \text{WeakPAXp}(\mathcal{X}'; \mathbb{F}, \kappa, \mathbf{v}, c, \delta)$$

- Locally-minimal PAXp (LmPAXp):

$\text{LmPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) :=$

$$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) \wedge \forall(j \in \mathcal{X}). \neg \text{WeakPAXp}(\mathcal{X} \setminus \{j\}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta)$$

# Definitions

- Weak probabilistic AXp (WPAXp):

– definition is non-monotonic

$$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) :=$$

$$\Pr_{\mathbf{x}}(\kappa(\mathbf{x}) = c \mid \mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}}) \geq \delta := \frac{|\{\mathbf{x} \in \mathbb{F} : \kappa(\mathbf{x}) = c \wedge (\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}})\}|}{|\{\mathbf{x} \in \mathbb{F} : (\mathbf{x}_{\mathcal{X}} = \mathbf{v}_{\mathcal{X}})\}|} \geq \delta$$

- Probabilistic AXp (PAXp):

$$\text{PAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) :=$$

$$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) \wedge \forall(\mathcal{X}' \subsetneq \mathcal{X}). \neg \text{WeakPAXp}(\mathcal{X}'; \mathbb{F}, \kappa, \mathbf{v}, c, \delta)$$

- Locally-minimal PAXp (LmPAXp):

– may differ from PAXp due to non-monotonicity

$$\text{LmPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) :=$$

$$\text{WeakPAXp}(\mathcal{X}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta) \wedge \forall(j \in \mathcal{X}). \neg \text{WeakPAXp}(\mathcal{X} \setminus \{j\}; \mathbb{F}, \kappa, \mathbf{v}, c, \delta)$$

## What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)

## What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)
  - Standard algorithms for finding one AXp **cannot** be used

## What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)
  - Standard algorithms for finding one AXp **cannot** be used
  - For DTs, finding on PAXp is computationally hard

[ABOS22]

## What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)
  - Standard algorithms for finding one AXp **cannot** be used
  - For DTs, finding on PAXp is computationally hard [ABOS22]
  - In general, complexity is unwieldy [WMHK21]

# What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)

- Standard algorithms for finding one AXp **cannot** be used

- For DTs, finding on PAXp is computationally hard

[ABOS22]

- In general, complexity is unwieldy

[WMHK21]

- Recent dedicated algorithms for simple ML models

[IHJ<sup>+</sup>23]

# What is known about PAXps?

- **Obs:** Definition of WPAXp is **non-monotonic** (from previous slide)

- Standard algorithms for finding one AXp **cannot** be used

- For DTs, finding on PAXp is computationally hard

[ABOS22]

- In general, complexity is unwieldy

[WMHK21]

- Recent dedicated algorithms for simple ML models

[IHI<sup>+</sup>23]

- Recent approximate algorithms for complex ML models

[IMM24]

# Results for decision trees

Dataset	MinPAXp										LmPAXp							Anchor												
	DT		Path			$\delta$	Length			Prec		Time		Length			Prec		$m_{\leq}$		Time		D	Length			Prec		Time	
	N	A	M	m	avg		M	m	avg	avg	avg	avg	avg	M	m	avg	avg	avg	avg	avg	avg	M	m	avg	$F_{\#P}$	avg	avg			
adult	1241 89 14 3 10.7						100	11	3	6.8	100	2.34	11	3	6.9	100	100	0.00	d	12	2	7.0	26.8	76.8	0.96					
	95	11	3	6.2	98.4		5.36	11	3	6.3	98.6	99.0	0.01	u	12	3	10.0	29.4	93.7	2.20										
	90	11	2	5.6	94.6		4.64	11	2	5.8	95.2	96.4	0.01																	
dermatology	71 100 13 1 5.1						100	12	1	4.4	100	0.35	12	1	4.4	100	100	0.00	d	31	1	4.8	58.1	32.9	3.10					
	95	12	1	4.1	99.7		0.37	12	1	4.1	99.7	99.3	0.00	u	34	1	13.1	43.2	87.2	25.13										
	90	11	1	4.0	98.8		0.35	11	1	4.0	98.8	100	0.00																	
kr-vs-kp	231 100 14 3 6.6						100	12	2	4.8	100	0.93	12	2	4.9	100	100	0.00	d	36	2	7.9	44.8	69.4	1.94					
	95	11	2	3.9	98.1		0.97	11	2	4.0	98.1	100	0.00	u	12	2	3.6	16.6	97.3	1.81										
	90	10	2	3.2	95.4		0.92	10	2	3.3	95.4	99.0	0.00																	
letter	3261 93 14 4 11.8						100	12	4	8.2	100	16.06	11	4	8.2	100	100	0.00	d	16	3	13.2	43.1	71.3	12.22					
	95	12	4	8.0	99.6		18.28	11	4	8.0	99.5	100	0.00	u	16	3	13.7	47.3	66.3	10.15										
	90	12	4	7.7	97.7		16.35	10	4	7.8	97.8	100	0.00																	
soybean	219 100 16 3 7.3						100	14	3	6.4	100	0.92	14	3	6.5	100	100	0.00	d	35	2	8.6	55.4	33.6	5.43					
	95	14	3	6.4	99.8		0.95	14	3	6.4	99.8	100	0.00	u	35	3	19.2	66.0	75.0	38.96										
	90	14	3	6.1	98.1		0.94	14	3	6.1	98.2	98.5	0.00																	
spambase	141 99 14 3 8.5						0	12	3	7.4	100	1.23	12	3	7.5	100	100	0.01	d	38	2	6.3	65.3	63.3	24.12					
	95	9	1	3.7	96.1		2.16	9	1	3.8	96.5	100	0.01	u	57	3	28.0	86.2	65.3	834.70										
	90	6	1	2.4	92.4		2.15	8	1	2.4	92.2	100	0.01																	

# Results for naive Bayes classifiers

Dataset	(#F #I)	NBC	AXp	LmPAXp $\leq 9$						LmPAXp $\leq 7$						LmPAXp $\leq 4$					
		A%	Length	$\delta$	Length	Precision	W%	Time	Length	Precision	W%	Time	Length	Precision	W%	Time	Length	Precision	W%	Time	
adult	(13 200)	81.37	6.8 $\pm$ 1.2	98	6.8 $\pm$ 1.1	100 $\pm$ 0.0	100	0.003	6.3 $\pm$ 0.9	99.61 $\pm$ 0.6	96	0.023	4.8 $\pm$ 1.3	98.73 $\pm$ 0.5	48	0.059					
				95	6.8 $\pm$ 1.1	99.99 $\pm$ 0.2	100	0.074	5.9 $\pm$ 1.0	98.87 $\pm$ 1.8	99	0.058	3.9 $\pm$ 1.0	96.93 $\pm$ 1.1	80	0.071					
				93	6.8 $\pm$ 1.1	99.97 $\pm$ 0.4	100	0.104	5.7 $\pm$ 1.3	98.34 $\pm$ 2.6	100	0.086	3.4 $\pm$ 0.9	95.21 $\pm$ 1.6	90	0.093					
				90	6.8 $\pm$ 1.1	99.95 $\pm$ 0.6	100	0.164	5.5 $\pm$ 1.4	97.86 $\pm$ 3.4	100	0.100	3.0 $\pm$ 0.8	93.46 $\pm$ 1.5	94	0.103					
agaricus	(23 200)	95.41	10.3 $\pm$ 2.5	98	7.7 $\pm$ 2.7	99.12 $\pm$ 0.8	92	0.593	6.4 $\pm$ 3.0	98.75 $\pm$ 0.6	87	0.763	6.0 $\pm$ 3.1	98.67 $\pm$ 0.5	29	0.870					
				95	6.9 $\pm$ 3.1	97.62 $\pm$ 2.1	95	0.954	5.3 $\pm$ 3.2	96.59 $\pm$ 1.6	92	1.273	4.8 $\pm$ 3.3	96.24 $\pm$ 1.2	55	1.217					
				93	6.5 $\pm$ 3.1	96.65 $\pm$ 2.8	95	1.112	4.8 $\pm$ 3.1	95.38 $\pm$ 1.9	93	1.309	4.3 $\pm$ 3.1	94.92 $\pm$ 1.3	64	1.390					
				90	5.9 $\pm$ 3.3	94.95 $\pm$ 4.1	96	1.332	4.0 $\pm$ 3.0	92.60 $\pm$ 2.8	95	1.598	3.6 $\pm$ 2.8	92.08 $\pm$ 1.7	76	1.830					
chess	(37 200)	88.34	12.1 $\pm$ 3.7	98	8.1 $\pm$ 4.1	99.27 $\pm$ 0.6	64	0.383	5.9 $\pm$ 4.9	98.70 $\pm$ 0.4	64	0.454	5.7 $\pm$ 5.0	98.65 $\pm$ 0.4	46	0.457					
				95	7.7 $\pm$ 3.8	98.51 $\pm$ 1.4	68	0.404	5.5 $\pm$ 4.4	97.90 $\pm$ 0.9	64	0.483	5.3 $\pm$ 4.5	97.85 $\pm$ 0.8	46	0.478					
				93	7.3 $\pm$ 3.5	97.56 $\pm$ 2.4	68	0.419	5.0 $\pm$ 4.1	96.26 $\pm$ 2.2	64	0.485	4.8 $\pm$ 4.1	96.21 $\pm$ 2.1	64	0.493					
				90	7.3 $\pm$ 3.5	97.29 $\pm$ 2.9	70	0.413	4.9 $\pm$ 4.0	95.99 $\pm$ 2.6	64	0.483	4.8 $\pm$ 4.0	95.93 $\pm$ 2.5	64	0.543					
vote	(17 81)	89.66	5.3 $\pm$ 1.4	98	5.3 $\pm$ 1.4	100 $\pm$ 0.0	100	0.000	5.3 $\pm$ 1.3	99.95 $\pm$ 0.2	100	0.007	4.6 $\pm$ 1.1	99.60 $\pm$ 0.4	64	0.014					
				95	5.3 $\pm$ 1.4	100 $\pm$ 0.0	100	0.000	5.3 $\pm$ 1.3	99.93 $\pm$ 0.3	100	0.008	4.1 $\pm$ 1.0	98.25 $\pm$ 1.7	64	0.018					
				93	5.3 $\pm$ 1.4	100 $\pm$ 0.0	100	0.000	5.2 $\pm$ 1.3	99.78 $\pm$ 1.1	100	0.012	4.1 $\pm$ 0.9	98.10 $\pm$ 1.9	64	0.018					
				90	5.3 $\pm$ 1.4	100 $\pm$ 0.0	100	0.000	5.2 $\pm$ 1.3	99.78 $\pm$ 1.1	100	0.012	4.0 $\pm$ 1.2	97.24 $\pm$ 3.1	64	0.022					
kr-vs-kp	(37 200)	88.07	12.2 $\pm$ 3.9	98	7.8 $\pm$ 4.2	99.19 $\pm$ 0.5	64	0.387	6.5 $\pm$ 4.7	98.99 $\pm$ 0.4	64	0.427	6.1 $\pm$ 4.9	98.88 $\pm$ 0.3	43	0.457					
				95	7.3 $\pm$ 3.9	98.29 $\pm$ 1.4	64	0.416	6.0 $\pm$ 4.3	97.89 $\pm$ 1.1	64	0.453	5.5 $\pm$ 4.5	97.79 $\pm$ 0.9	43	0.462					
				93	6.9 $\pm$ 3.5	97.21 $\pm$ 2.5	69	0.422	5.6 $\pm$ 3.8	96.82 $\pm$ 2.2	64	0.448	5.2 $\pm$ 4.0	96.71 $\pm$ 2.1	43	0.468					
				90	6.8 $\pm$ 3.5	96.65 $\pm$ 3.1	69	0.418	5.4 $\pm$ 3.8	95.69 $\pm$ 3.0	64	0.468	5.0 $\pm$ 4.0	95.59 $\pm$ 2.8	61	0.487					
mushroom	(23 200)	95.51	10.7 $\pm$ 2.3	98	7.5 $\pm$ 2.4	98.99 $\pm$ 0.7	90	0.641	6.5 $\pm$ 2.6	98.74 $\pm$ 0.5	83	0.751	6.3 $\pm$ 2.7	98.70 $\pm$ 0.4	18	0.828					
				95	6.5 $\pm$ 2.6	97.35 $\pm$ 1.8	96	1.011	5.1 $\pm$ 2.5	96.52 $\pm$ 1.0	90	1.130	5.0 $\pm$ 2.5	96.39 $\pm$ 0.8	54	1.113					
				93	5.8 $\pm$ 2.8	95.77 $\pm$ 2.7	96	1.257	4.4 $\pm$ 2.5	94.67 $\pm$ 1.6	94	1.297	4.2 $\pm$ 2.4	94.48 $\pm$ 1.3	65	1.324 <sub>186 / 238</sub>					

# Results for decision diagrams

Dataset	#I	#F	OMDD		$\delta$	MinPAXp						LmPAXp					
						Length			Prec	Time	Length			Prec	$m_{\subseteq}$	Time	
			#N	A%		M	m	avg	avg	avg	M	m	avg	avg	avg	avg	
lending	100	9	1103	81.7	100	9	6	8.0	100	24.24	9	6	7.9	100	100	1.57	
					95	9	5	7.8	99.7	21.48	9	6	7.8	99.8	100	100	1.49
					90	9	4	7.2	96	24.65	9	5	7.4	97.0	100	100	1.48
monk2	100	6	70	79.3	100	6	4	5.1	100	0.10	6	4	5.1	100	100	0.03	
					95	6	4	5.1	100	0.09	6	4	5.1	100	100	0.03	
					90	6	3	4.8	98.1	0.09	6	3	4.8	98.1	100	100	0.03
postoperative	74	8	109	80	100	8	4	6.1	100	0.26	8	4	6.2	100	100	0.04	
					95	8	2	6.0	99.3	0.25	8	2	6.0	99.3	100	100	0.04
					90	8	2	5.3	95.9	0.23	8	2	5.4	96.6	94.6	100	0.04
tic_tac_toe	100	9	424	70.3	100	9	5	7.7	100	3.60	9	5	7.8	100	100	0.38	
					95	9	5	7.5	99.5	3.24	9	5	7.7	99.6	99.0	100	0.38
					90	9	3	7.3	98.3	4.06	9	3	7.5	98.6	98.0	100	0.38
xd6	100	9	76	83.1	100	9	4	4.6	100	0.10	9	4	4.6	100	100	0.03	
					95	9	3	3.8	97	0.09	9	3	3.8	97.0	99.0	100	0.03
					90	9	3	3.3	94.8	0.10	9	3	3.4	94.6	100	100	0.03

## Remarks on LmPAXps

[IHI<sup>+</sup>23]

- LmPAXps ignore non-monotonicity, and so overapproximate PAXps
  - Theoretical guarantees, but may be reducible
- For DTs, computation of LmPAXps is in P
- Experimental results confirm LmPAXps match PAXps in most cases
- Recent results on approximating LmPAXps for RFs

[IMM24]

# Outline – Unit #06

Sample-Based Explanations

Changing Assumptions (in Plain Logic-Based XAI)

Inflated Explanations

Constrained Explanations

Distance-Restricted Explanations

Probabilistic Explanations

Additional Topics

# Surrogate models in logic-based XAI

[BAMT21]

- Motivation:
  - Logic-based XAI does not yet scale for highly complex ML models
  - Surrogate models find many uses in ML, for approximating complex models

# Surrogate models in logic-based XAI

[BAMT21]

- Motivation:
  - Logic-based XAI does not yet scale for highly complex ML models
  - Surrogate models find many uses in ML, for approximating complex models
- Approach:
  - Train a **surrogate** model, e.g. DT, RF/TE, small(er) NN, etc.
  - Target high accuracy of surrogate model

# Surrogate models in logic-based XAI

[BAMT21]

- Motivation:
  - Logic-based XAI does not yet scale for highly complex ML models
  - Surrogate models find many uses in ML, for approximating complex models
- Approach:
  - Train a **surrogate** model, e.g. DT, RF/TE, small(er) NN, etc.
  - Target high accuracy of surrogate model
- Explain the surrogate model
  - Compute rigorous explanation: plain AXp, probabilistic AXp,

# Surrogate models in logic-based XAI

[BAMT21]

- Motivation:
  - Logic-based XAI does not yet scale for highly complex ML models
  - Surrogate models find many uses in ML, for approximating complex models
- Approach:
  - Train a **surrogate** model, e.g. DT, RF/TE, small(er) NN, etc.
  - Target high accuracy of surrogate model
- Explain the surrogate model
  - Compute rigorous explanation: plain AXp, probabilistic AXp,
- Report computed explanation as explanation for the complex ML model

## Certified explainer (for monotonic classification)

[HM23f]

- The implementation of a correct algorithm may **not** be correct
- Even comprehensive testing of implemented algorithms does not guarantee correctness

# Certified explainer (for monotonic classification)

[HM23f]

- The implementation of a correct algorithm may **not** be correct
- Even comprehensive testing of implemented algorithms does not guarantee correctness
- Certification of implementations is one possible alternative
  - Formalize algorithm, e.g. explanations for monotonic classifiers, e.g. using Coq
  - Prove that formalized algorithm is correct
  - Extract certified algorithm from proof of correctness

# Certified explainer (for monotonic classification)

[HM23f]

- The implementation of a correct algorithm may **not** be correct
- Even comprehensive testing of implemented algorithms does not guarantee correctness
- Certification of implementations is one possible alternative
  - Formalize algorithm, e.g. explanations for monotonic classifiers, e.g. using Coq
  - Prove that formalized algorithm is correct
  - Extract certified algorithm from proof of correctness
- Downsides:
  - Efficiency of certified algorithm
  - Dedicated algorithm for each explainer

# Certified explainer (for monotonic classification)

[HM23f]

- The implementation of a correct algorithm may **not** be correct
- Even comprehensive testing of implemented algorithms does not guarantee correctness
- Certification of implementations is one possible alternative
  - Formalize algorithm, e.g. explanations for monotonic classifiers, e.g. using Coq
  - Prove that formalized algorithm is correct
  - Extract certified algorithm from proof of correctness
- Downsides:
  - Efficiency of certified algorithm
  - Dedicated algorithm for each explainer
- Certification envisioned for **any** explainability algorithm

# Plan for this course – light at the end of the tunnel...

- Lecture 01 – unit(s):
  - #01: Foundations
- Lecture 02 – unit(s):
  - #02: Principles of symbolic XAI – **feature selection**
  - #03: Tractability in symbolic XAI (& myth of interpretability)
- Lecture 03 – unit(s):
  - #04: Intractability in symbolic XAI (& myth of model-agnostic XAI)
  - #05: Explainability queries
- Lecture 04 – unit(s):
  - #06: Recent, emerging & advanced topics
- Lecture 05 – unit(s):
  - #07: Principles of symbolic XAI – **feature attribution** (& myth of Shapley values in XAI)
  - #08: Corrected feature attribution – nuSHAP
  - #09: Conclusions & research directions

# Questions?

# Lecture 05

## Recapitulate fourth lecture

- Monotonic classifiers vs. weighted voting games
- Advanced topics:
  - Sample-based explanations
  - Inflated explanations
  - Probabilistic explanations
  - Constrained explanations
  - Distance-restricted explanations
  - Explanations using surrogate models
  - Certified explainability

- Every WVG  $\mathcal{G}$ , described by  $[q; n_1, \dots, n_m]$ , can be represented as a monotonically increasing boolean classifier  $\mathcal{M} = (\mathcal{F}, \{0, 1\}^m, \{0, 1\}, \kappa)$ , such that:
  - Each voter  $i$  is mapped to a boolean feature  $i$ , such that feature  $i$  takes value 1 if voter  $i$  votes Yes; otherwise it takes value 0;
  - The classification function  $\kappa : \mathbb{F} \rightarrow \{0, 1\}$  is defined by:

$$\kappa(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^m n_i x_i \geq q \\ 0 & \text{otherwise} \end{cases}$$

- The target instance is  $(1, 1)$ ; and
- Each minimal winning coalition  $\mathcal{C}$  corresponds to an AXp of  $\mathcal{E} = (\mathcal{M}, (1, 1))$

- Every WVG  $\mathcal{G}$ , described by  $[q; n_1, \dots, n_m]$ , can be represented as a monotonically increasing boolean classifier  $\mathcal{M} = (\mathcal{F}, \{0, 1\}^m, \{0, 1\}, \kappa)$ , such that:
  - Each voter  $i$  is mapped to a boolean feature  $i$ , such that feature  $i$  takes value 1 if voter  $i$  votes Yes; otherwise it takes value 0;
  - The classification function  $\kappa : \mathbb{F} \rightarrow \{0, 1\}$  is defined by:

$$\kappa(\mathbf{x}) = \begin{cases} 1 & \text{if } \sum_{i=1}^m n_i x_i \geq q \\ 0 & \text{otherwise} \end{cases}$$

- The target instance is  $(1, 1)$ ; and
- Each minimal winning coalition  $\mathcal{C}$  corresponds to an AXp of  $\mathcal{E} = (\mathcal{M}, (1, 1))$

∴ WVGs can be analyzed by studying the AXps/CXps of monotonically increasing boolean classifiers

## Another WVG

- WVG: [25; 10, 9, 7, 1, 1, 1, 1, 1, 1]

## Another WVG

- WVG: [25; 10, 9, 7, 1, 1, 1, 1, 1, 1]
- Computing the AXps:
  - Winning coalitions must include both 1 and 2
  - We can pick 3 or, alternatively, all the other ones

## Another WVG

- WVG: [25; 10, 9, 7, 1, 1, 1, 1, 1, 1]
- Computing the AXps:
  - Winning coalitions must include both 1 and 2
  - We can pick 3 or, alternatively, all the other ones
- AXps:

## Another WVG

- WVG: [25; 10, 9, 7, 1, 1, 1, 1, 1, 1]
- Computing the AXps:
  - Winning coalitions must include both 1 and 2
  - We can pick 3 or, alternatively, all the other ones
- AXps:

$$\mathbb{A} = \{\{1, 2, 3\}, \{1, 2, 4, 5, 6, 7, 8, 9\}\}$$

## Another WVG

- WVG: [25; 10, 9, 7, 1, 1, 1, 1, 1, 1]
- Computing the AXps:
  - Winning coalitions must include both 1 and 2
  - We can pick 3 or, alternatively, all the other ones
- AXps:

$$\mathbb{A} = \{\{1, 2, 3\}, \{1, 2, 4, 5, 6, 7, 8, 9\}\}$$

- CXps:

## Another WVG

- WVG: [25; 10, 9, 7, 1, 1, 1, 1, 1, 1]
- Computing the AXps:
  - Winning coalitions must include both 1 and 2
  - We can pick 3 or, alternatively, all the other ones
- AXps:

$$\mathbb{A} = \{\{1, 2, 3\}, \{1, 2, 4, 5, 6, 7, 8, 9\}\}$$

- CXps:
- $$\mathbb{C} = \{\{1\}, \{2\}, \{3, 4\}, \{3, 5\}, \{3, 6\}, \{3, 7\}, \{3, 8\}, \{3, 9\}, \}$$

## Another WVG

- WVG: [25; 10, 9, 7, 1, 1, 1, 1, 1, 1]
- Computing the AXps:
  - Winning coalitions must include both 1 and 2
  - We can pick 3 or, alternatively, all the other ones

- AXps:

$$\mathbb{A} = \{\{1, 2, 3\}, \{1, 2, 4, 5, 6, 7, 8, 9\}\}$$

- CXps:

$$\mathbb{C} = \{\{1\}, \{2\}, \{3, 4\}, \{3, 5\}, \{3, 6\}, \{3, 7\}, \{3, 8\}, \{3, 9\}, \}$$

- Q: How should features be ranked in terms of importance?

# Plan for this course – light at the end of the tunnel...

- Lecture 01 – unit(s):
  - #01: Foundations
- Lecture 02 – unit(s):
  - #02: Principles of symbolic XAI – **feature selection**
  - #03: Tractability in symbolic XAI (& myth of interpretability)
- Lecture 03 – unit(s):
  - #04: Intractability in symbolic XAI (& myth of model-agnostic XAI)
  - #05: Explainability queries
- Lecture 04 – unit(s):
  - #06: Recent, emerging & advanced topics
- Lecture 05 – unit(s):
  - #07: Principles of symbolic XAI – **feature attribution** (& myth of Shapley values in XAI)
  - #08: Corrected feature attribution – nuSHAP
  - #09: Conclusions & research directions

Unit #07

# Principles of Symbolic XAI – Feature Attribution

# Outline – Unit #07

Exact Shapley Values for XAI

Myth #03: Shapley Values for XAI

Corrected SHAP Scores

A Correct New SHAP – nuSHAP

Voting Power & Power Indices

Feature Importance Scores

**Detour:** Standard SHAP Intro (from another course...)

## What are Shapley values?

- First proposed in game theory in the early 50s by L. S. Shapley
  - Measures the contribution of each player to a cooperative game

[Sha53]

# What are Shapley values?

- First proposed in game theory in the early 50s by L. S. Shapley
  - Measures the contribution of each player to a cooperative game
- Application in XAI since the 2000s
  - Popularized by SHAP
  - Used for feature attribution, i.e. [relative feature importance](#)

[Sha53]

[LC01, SK10, SK14, DSZ16, LL17, ABBM21, VLSS21, VLSS22, ABBM23]

[LL17]

# What are Shapley values?

- First proposed in game theory in the early 50s by L. S. Shapley
  - Measures the contribution of each player to a cooperative game
- Application in XAI since the 2000s
  - Popularized by SHAP
  - Used for feature attribution, i.e. **relative feature importance**
- Shapley values are becoming ubiquitous in XAI... – E.g. see slides from other XAI course...

[LC01, SK10, SK14, DSZ16, LL17, ABBM21, VLSS21, VLSS22, ABBM23]

[LL17]

🛡️ 🔍 [https://en.wikipedia.org/wiki/Shapley\\_value](https://en.wikipedia.org/wiki/Shapley_value)

📋 ⭐

Accessed 2023/06/14

## In machine learning [edit]

The Shapley value provides a principled way to explain the predictions of nonlinear models common in the field of [machine learning](#). By interpreting a model trained on a set of features as a value function on a coalition of players, Shapley values provide a natural way to compute which features contribute to a prediction.<sup>[17]</sup> This unifies several other methods including Locally Interpretable Model-Agnostic Explanations (LIME),<sup>[18]</sup> DeepLIFT,<sup>[19]</sup> and Layer-Wise Relevance Propagation.<sup>[20]</sup>

17. ^ Lundberg, Scott M.; Lee, Su-In (2017). "A Unified Approach to Interpreting Model Predictions". *Advances in Neural Information Processing Systems*. 30: 4765–4774. arXiv:1705.07874. Retrieved 2021-01-30.

# What are Shapley values?

- First proposed in game theory in the early 50s by L. S. Shapley
  - Measures the contribution of each player to a cooperative game
- Application in XAI since the 2000s
  - Popularized by SHAP
  - Used for feature attribution, i.e. **relative feature importance**
- Shapley values are becoming ubiquitous in XAI... – E.g. see slides from other XAI course...

[LC01, SK10, SK14, DSZ16, LL17, ABBM21, VLSS21, VLSS22, ABBM23]

[LL17]

🛡️ 🔍 [https://en.wikipedia.org/wiki/Shapley\\_value](https://en.wikipedia.org/wiki/Shapley_value)

📋 ⭐

Accessed 2023/06/14

## In machine learning [edit]

The Shapley value provides a principled way to explain the predictions of nonlinear models common in the field of [machine learning](#). By interpreting a model trained on a set of features as a value function on a coalition of players, Shapley values provide a natural way to compute which features contribute to a prediction.<sup>[17]</sup> This unifies several other methods including Locally Interpretable Model-Agnostic Explanations (LIME),<sup>[18]</sup> DeepLIFT,<sup>[19]</sup> and Layer-Wise Relevance Propagation.<sup>[20]</sup>

17. ^ Lundberg, Scott M.; Lee, Su-In (2017). "A Unified Approach to Interpreting Model Predictions". *Advances in Neural Information Processing Systems*. 30: 4765–4774. [arXiv:1705.07874](https://arxiv.org/abs/1705.07874). Retrieved 2021-01-30.

- **Q:** Do Shapley values for XAI **really** provide a rigorous measure of feature importance?

## How are Shapley values used in explainability?

- Instance:  $(\mathbf{v}, c)$

## How are Shapley values used in explainability?

- Instance:  $(\mathbf{v}, c)$
- $\Upsilon: 2^{\mathcal{F}} \rightarrow 2^{\mathbb{F}}$  defined by,

[ABBM21, ABBM23]

$$\Upsilon(\mathcal{S}) = \{\mathbf{x} \in \mathbb{F} \mid \wedge_{i \in \mathcal{S}} x_i = v_i\}$$

$\Upsilon(\mathcal{S})$  gives points in feature space having the features in  $\mathcal{S}$  fixed to their values in  $\mathbf{v}$

## How are Shapley values used in explainability?

- Instance:  $(\mathbf{v}, c)$
- $\Upsilon: 2^{\mathcal{F}} \rightarrow 2^{\mathbb{F}}$  defined by,

[ABBM21, ABBM23]

$$\Upsilon(\mathcal{S}) = \{\mathbf{x} \in \mathbb{F} \mid \wedge_{i \in \mathcal{S}} x_i = v_i\}$$

- $\Upsilon(\mathcal{S})$  gives points in feature space having the features in  $\mathcal{S}$  fixed to their values in  $\mathbf{v}$
- $\phi: 2^{\mathcal{F}} \rightarrow \mathbb{R}$  defined by,

$$\phi(\mathcal{S}) = 1/2^{|\mathcal{F} \setminus \mathcal{S}|} \sum_{\mathbf{x} \in \Upsilon(\mathcal{S})} \kappa(\mathbf{x}) = v_e(\mathcal{S})$$

$\phi(\mathcal{S})$  represents the expected value of the classifier on the points given by  $\Upsilon(\mathcal{S})$

## How are Shapley values used in explainability?

- Instance:  $(\mathbf{v}, c)$
- $\Upsilon: 2^{\mathcal{F}} \rightarrow 2^{\mathbb{F}}$  defined by,

[ABBM21, ABBM23]

$$\Upsilon(\mathcal{S}) = \{\mathbf{x} \in \mathbb{F} \mid \wedge_{i \in \mathcal{S}} x_i = v_i\}$$

- $\Upsilon(\mathcal{S})$  gives points in feature space having the features in  $\mathcal{S}$  fixed to their values in  $\mathbf{v}$
- $\phi: 2^{\mathcal{F}} \rightarrow \mathbb{R}$  defined by,

$$\phi(\mathcal{S}) = 1/2^{|\mathcal{F} \setminus \mathcal{S}|} \sum_{\mathbf{x} \in \Upsilon(\mathcal{S})} \kappa(\mathbf{x}) = v_e(\mathcal{S})$$

$\phi(\mathcal{S})$  represents the expected value of the classifier on the points given by  $\Upsilon(\mathcal{S})$

- $\text{Sc}: \mathcal{F} \rightarrow \mathbb{R}$  defined by,

$$\text{Sc}(i) = \sum_{\mathcal{S} \subseteq (\mathcal{F} \setminus \{i\})} \frac{|\mathcal{S}|!(|\mathcal{F}| - |\mathcal{S}| - 1)!}{|\mathcal{F}|!} \times (\phi(\mathcal{S} \cup \{i\}) - \phi(\mathcal{S}))$$

For all subsets of features, excluding  $i$ , compute the expected value of the classifier, with and without  $i$  fixed, weighted by  $\frac{1}{n} \binom{n}{|\mathcal{S}|}^{-1}$

- Obs:** Uniform distribution assumed; it suffices for our purposes

# How are Shapley values used in explainability?

- Instance:  $(\mathbf{v}, c)$
- $\Upsilon: 2^{\mathcal{F}} \rightarrow 2^{\mathbb{F}}$  defined by,

Marginal contribution  
(in SHAP lingo)!

[ABBM21, ABBM23]

$$\Upsilon(\mathcal{S}) = \{\mathbf{x} \in \mathbb{F} \mid \wedge_{i \in \mathcal{S}} x_i = v_i\}$$

- $\Upsilon(\mathcal{S})$  gives points in feature space having the features in  $\mathcal{S}$  fixed to their values in  $\mathbf{v}$
- $\phi: 2^{\mathcal{F}} \rightarrow \mathbb{R}$  defined by,

$$\phi(\mathcal{S}) = 1/2^{|\mathcal{F} \setminus \mathcal{S}|} \sum_{\mathbf{x} \in \Upsilon(\mathcal{S})} \kappa(\mathbf{x}) = v_e(\mathcal{S})$$

$\phi(\mathcal{S})$  represents the expected value of the classifier on the points given by  $\Upsilon(\mathcal{S})$

- $Sc: \mathcal{F} \rightarrow \mathbb{R}$  defined by,

$$Sc(i) = \sum_{\mathcal{S} \subseteq (\mathcal{F} \setminus \{i\})} \frac{|\mathcal{S}|!(|\mathcal{F}| - |\mathcal{S}| - 1)!}{|\mathcal{F}|!} \times (\phi(\mathcal{S} \cup \{i\}) - \phi(\mathcal{S}))$$

For all subsets of features, excluding  $i$ , compute the expected value of the classifier, with and without  $i$  fixed, weighted by  $\frac{1}{n} \binom{n}{|\mathcal{S}|}^{-1}$

- Obs:** Uniform distribution assumed; it suffices for our purposes

## How are Shapley values computed in practice?

- Exact evaluation is computationally (very) hard [VLSS21, ABBM21, VLSS22, ABBM23, HMS24]
- SHAP proposes a sample-based approach; with **no** guarantees of rigor [LL17]
  - Recent experiments revealed little to **no** correlation between Shapley values and SHAP's results [HM23c]

## How are Shapley values computed in practice?

- Exact evaluation is computationally (very) hard [VLSS21, ABBM21, VLSS22, ABBM23, HMS24]
- SHAP proposes a sample-based approach; with **no** guarantees of rigor [LL17]
  - Recent experiments revealed little to **no** correlation between Shapley values and SHAP's results [HM23c]
- **Polynomial-time** algorithm for deterministic decomposable boolean circuits [ABBM21]
- **Polynomial-time** algorithm for boolean functions represented with a truth-table [HM23c]

## What do Shapley values tell in terms of feature importance?

- [SK10] reads:

*“According to the 2nd axiom, if two features values have an identical influence on the prediction they are assigned contributions of equal size. The 3rd axiom says that if a feature has no influence on the prediction it is assigned a contribution of 0.”*

(Obs: the axioms refer to the axiomatic characterization of Shapley values.)

## What do Shapley values tell in terms of feature importance?

- [SK10] reads:

*“According to the 2nd axiom, if two features values have an identical influence on the prediction they are assigned contributions of equal size. The 3rd axiom says that if a feature has no influence on the prediction it is assigned a contribution of 0.”*

(Obs: the axioms refer to the axiomatic characterization of Shapley values.)

- And [SK10] also reads:

*“When viewed together, these properties ensure that any effect the features might have on the classifiers output will be reflected in the generated contributions, which effectively deals with the issues of previous general explanation methods.”*

## What do Shapley values tell in terms of feature importance?

- [SK10] reads:

*“According to the 2nd axiom, if two features values have an identical influence on the prediction they are assigned contributions of equal size. The 3rd axiom says that if a feature has no influence on the prediction it is assigned a contribution of 0.”*

(Obs: the axioms refer to the axiomatic characterization of Shapley values.)
- And [SK10] also reads:

*“When viewed together, these properties ensure that any effect the features might have on the classifiers output will be reflected in the generated contributions, which effectively deals with the issues of previous general explanation methods.”*
- **Obs:** Shapley values are defined **axiomatically**, i.e. **no** immediate relationship with AXp's/CXp's or with feature (ir)relevancy

## What do Shapley values tell in terms of feature importance?

- [SK10] reads:

*“According to the 2nd axiom, if two features values have an identical influence on the prediction they are assigned contributions of equal size. The 3rd axiom says that if a feature has no influence on the prediction it is assigned a contribution of 0.”*

(Obs: the axioms refer to the axiomatic characterization of Shapley values.)
- And [SK10] also reads:

*“When viewed together, these properties ensure that any effect the features might have on the classifiers output will be reflected in the generated contributions, which effectively deals with the issues of previous general explanation methods.”*
- Obs: Shapley values are defined **axiomatically**, i.e. **no** immediate relationship with AXp's/CXp's or with feature (ir)relevancy
  - Qs: can we have **irrelevant** features with a non-zero Shapley value, and/or **relevant** features with a Shapley of zero?
    - Recall: **relevant** features occur in some AXp/CXp; **irrelevant** features do **not** occur in **any** AXp/CXp

## Outline – Unit #07

Exact Shapley Values for XAI

Myth #03: Shapley Values for XAI

Corrected SHAP Scores

A Correct New SHAP – nuSHAP

Voting Power & Power Indices

Feature Importance Scores

- Boolean classifier, instance  $(\mathbf{v}, c)$ , and some  $i, i_1, i_2 \in \mathcal{F}$ :

- Boolean classifier, instance  $(\mathbf{v}, c)$ , and some  $i, i_1, i_2 \in \mathcal{F}$ :
  - Issue I1 occurs if,

$$\text{Irrelevant}(i) \wedge (\text{Sv}(i) \neq 0)$$

## Shapley values vs. feature (ir)relevancy – identified issues

[HM23c, HM23d, HM23e, MH23, HMS24, MSH24]

- Boolean classifier, instance  $(\mathbf{v}, c)$ , and some  $i, i_1, i_2 \in \mathcal{F}$ :

- Issue I1 occurs if,

$$\text{Irrelevant}(i) \wedge (\text{Sv}(i) \neq 0)$$

- Issue I2 occurs if,

$$\text{Irrelevant}(i_1) \wedge \text{Relevant}(i_2) \wedge (|\text{Sv}(i_1)| > |\text{Sv}(i_2)|)$$

## Shapley values vs. feature (ir)relevancy – identified issues

[HM23c, HM23d, HM23e, MH23, HMS24, MSH24]

- Boolean classifier, instance  $(\mathbf{v}, c)$ , and some  $i, i_1, i_2 \in \mathcal{F}$ :

- Issue I1 occurs if,

$$\text{Irrelevant}(i) \wedge (\text{Sv}(i) \neq 0)$$

- Issue I2 occurs if,

$$\text{Irrelevant}(i_1) \wedge \text{Relevant}(i_2) \wedge (|\text{Sv}(i_1)| > |\text{Sv}(i_2)|)$$

- Issue I3 occurs if,

$$\text{Relevant}(i) \wedge (\text{Sv}(i) = 0)$$

# Shapley values vs. feature (ir)relevancy – identified issues

[HM23c, HM23d, HM23e, MH23, HMS24, MSH24]

- Boolean classifier, instance  $(\mathbf{v}, c)$ , and some  $i, i_1, i_2 \in \mathcal{F}$ :

- Issue I1 occurs if,

$$\text{Irrelevant}(i) \wedge (\text{Sv}(i) \neq 0)$$

- Issue I2 occurs if,

$$\text{Irrelevant}(i_1) \wedge \text{Relevant}(i_2) \wedge (|\text{Sv}(i_1)| > |\text{Sv}(i_2)|)$$

- Issue I3 occurs if,

$$\text{Relevant}(i) \wedge (\text{Sv}(i) = 0)$$

- Issue I4 occurs if,

$$[\text{Irrelevant}(i_1) \wedge (\text{Sv}(i_1) \neq 0)] \wedge [\text{Relevant}(i_2) \wedge (\text{Sv}(i_2) = 0)]$$

# Shapley values vs. feature (ir)relevancy – identified issues

[HM23c, HM23d, HM23e, MH23, HMS24, MSH24]

- Boolean classifier, instance  $(\mathbf{v}, c)$ , and some  $i, i_1, i_2 \in \mathcal{F}$ :

- Issue I1 occurs if,

$$\text{Irrelevant}(i) \wedge (\text{Sv}(i) \neq 0)$$

- Issue I2 occurs if,

$$\text{Irrelevant}(i_1) \wedge \text{Relevant}(i_2) \wedge (|\text{Sv}(i_1)| > |\text{Sv}(i_2)|)$$

- Issue I3 occurs if,

$$\text{Relevant}(i) \wedge (\text{Sv}(i) = 0)$$

- Issue I4 occurs if,

$$[\text{Irrelevant}(i_1) \wedge (\text{Sv}(i_1) \neq 0)] \wedge [\text{Relevant}(i_2) \wedge (\text{Sv}(i_2) = 0)]$$

- Issue I5 occurs if,

$$[\text{Irrelevant}(i) \wedge \forall_{1 \leq j \leq m, j \neq i} (|\text{Sv}(j)| < |\text{Sv}(i)|)]$$

# Shapley values vs. feature (ir)relevancy – identified issues

[HM23c, HM23d, HM23e, MH23, HMS24, MSH24]

- Boolean classifier, instance  $(\mathbf{v}, c)$ , and some  $i, i_1, i_2 \in \mathcal{F}$ :

- Issue I1 occurs if,

$$\text{Irrelevant}(i) \wedge (\text{Sv}(i) \neq 0)$$

- Issue I2 occurs if,

$$\text{Irrelevant}(i_1) \wedge \text{Relevant}(i_2) \wedge (|\text{Sv}(i_1)| > |\text{Sv}(i_2)|)$$

- Issue I3 occurs if,

$$\text{Relevant}(i) \wedge (\text{Sv}(i) = 0)$$

Any of these issues is a cause  
of (serious) concern per se!

- Issue I4 occurs if,

$$[\text{Irrelevant}(i_1) \wedge (\text{Sv}(i_1) \neq 0)] \wedge [\text{Relevant}(i_2) \wedge (\text{Sv}(i_2) = 0)]$$

- Issue I5 occurs if,

$$[\text{Irrelevant}(i) \wedge \forall_{1 \leq j \leq m, j \neq i} (|\text{Sv}(j)| < |\text{Sv}(i)|)]$$

# Some stats – all boolean functions with 4 variables

[HM23c, HM23d, HM23e, MH23, HMS24, MSH24]

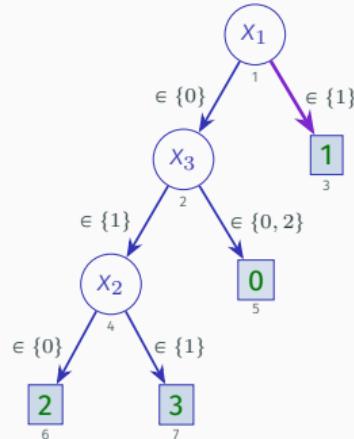
Issue-related metric	Value	Recap issue
# of functions	65536	
# number of instances	1048576	
# of I1 issues	781696	
# of functions with I1 issues	65320	
% I1 issues / function	99.67	$[\text{Irrelevant}(i) \wedge (\text{Sv}(i) \neq 0)]$
# of I2 issues	105184	
# of functions with I2 issues	40448	
% I2 issues / function	61.72	$[\text{Irrelevant}(i_1) \wedge \text{Relevant}(i_2) \wedge ( \text{Sv}(i_1)  >  \text{Sv}(i_2) )]$
# of I3 issues	43008	
# of functions with I3 issues	7800	
% I3 issues / function	11.90	$[\text{Relevant}(i) \wedge (\text{Sv}(i) = 0)]$
# of I4 issues	5728	
# of functions with I4 issues	2592	
% I4 issues / function	3.96	$[\text{Irrelevant}(i_1) \wedge (\text{Sv}(i_1) \neq 0)] \wedge [\text{Relevant}(i_2) \wedge (\text{Sv}(i_2) = 0)]$
# of I5 issues	1664	
# of functions with I5 issues	1248	
% I5 issues / function	1.90	$[\text{Irrelevant}(i) \wedge \forall_{1 \leq j \leq m, j \neq i} ( \text{Sv}(j)  <  \text{Sv}(i) )]$

Previous results do matter! Let's go non-boolean...

row #	$x_1$	$x_2$	$x_3$	$\kappa_1(\mathbf{x})$	$\kappa_2(\mathbf{x})$
1	0	0	0	0	0
2	0	0	1	4	2
3	0	0	2	0	0
4	0	1	0	0	0
5	0	1	1	7	3
6	0	1	2	0	0
7	1	0	0	1	1
8	1	0	1	1	1
9	1	0	2	1	1
10	1	1	0	1	1
11	1	1	1	1	1
12	1	1	2	1	1

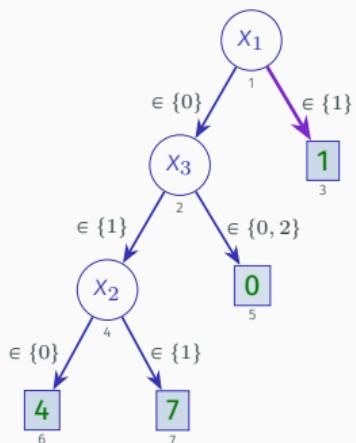
DT1

## Tabular representations



DT2

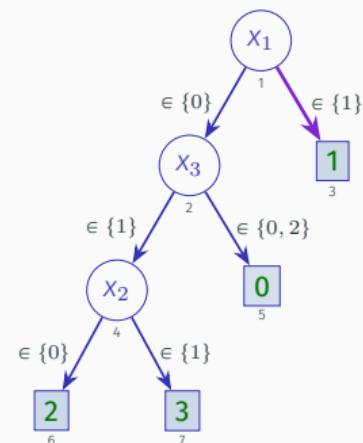
# Instance $((1, 1, 2), 1)$ – which feature matters the most for prediction 1?



DT1

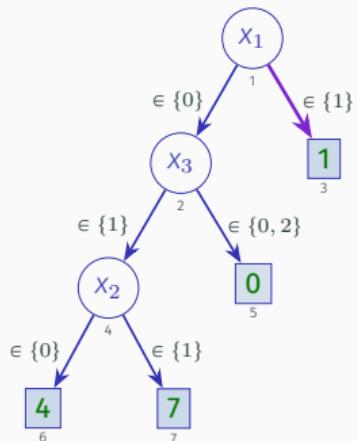
row #	$x_1$	$x_2$	$x_3$	$\kappa_1(\mathbf{x})$	$\kappa_2(\mathbf{x})$
1	0	0	0	0	0
2	0	0	1	4	2
3	0	0	2	0	0
4	0	1	0	0	0
5	0	1	1	7	3
6	0	1	2	0	0
7	1	0	0	1	1
8	1	0	1	1	1
9	1	0	2	1	1
10	1	1	0	1	1
11	1	1	1	1	1
12	1	1	2	1	1

Tabular representations



DT2

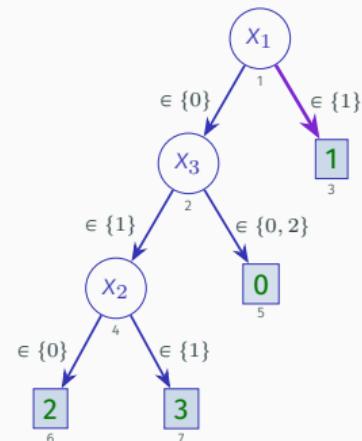
# Computing XPs – make sense...



DT1

row #	$x_1$	$x_2$	$x_3$	$\kappa_1(\mathbf{x})$	$\kappa_2(\mathbf{x})$
1	0	0	0	0	0
2	0	0	1	4	2
3	0	0	2	0	0
4	0	1	0	0	0
5	0	1	1	7	3
6	0	1	2	0	0
7	1	0	0	1	1
8	1	0	1	1	1
9	1	0	2	1	1
10	1	1	0	1	1
11	1	1	1	1	1
12	1	1	2	1	1

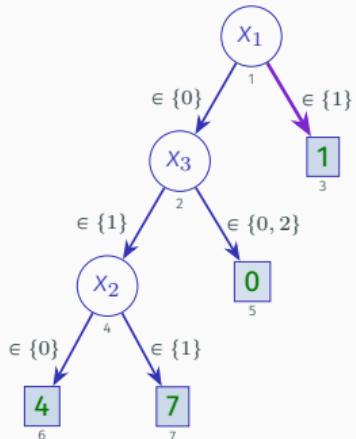
Tabular representations



DT2

XPs: AXps/CXps		
DT	AXps	CXps
DT1	{1}	{1}
DT2	{1}	{1}

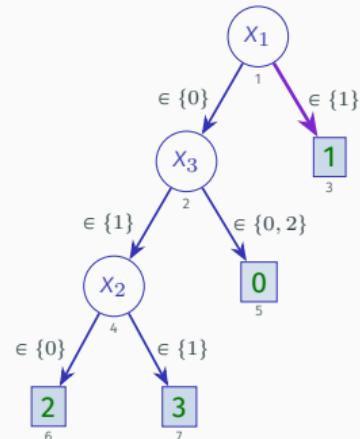
# Computing XPs, AEs – also make sense...



DT1

row #	$x_1$	$x_2$	$x_3$	$\kappa_1(\mathbf{x})$	$\kappa_2(\mathbf{x})$
1	0	0	0	0	0
2	0	0	1	4	2
3	0	0	2	0	0
4	0	1	0	0	0
5	0	1	1	7	3
6	0	1	2	0	0
7	1	0	0	1	1
8	1	0	1	1	1
9	1	0	2	1	1
10	1	1	0	1	1
11	1	1	1	1	1
12	1	1	2	1	1

Tabular representations

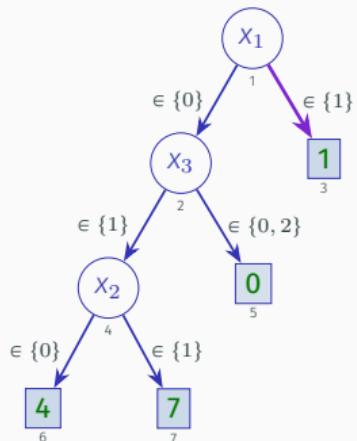


DT2

XPs: AXps/CXps		
DT	AXps	CXps
DT1	{1}	{1}
DT2	{1}	{1}

Adversarial Examples	
DT	$l_0$ -minimal AEs
DT1	{1}
DT2	{1}

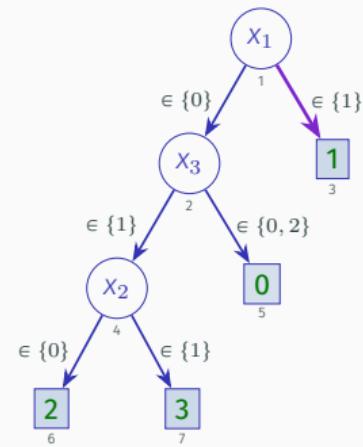
# Computing XPs, AEs & Svs



DT1

row #	$x_1$	$x_2$	$x_3$	$\kappa_1(\mathbf{x})$	$\kappa_2(\mathbf{x})$
1	0	0	0	0	0
2	0	0	1	4	2
3	0	0	2	0	0
4	0	1	0	0	0
5	0	1	1	7	3
6	0	1	2	0	0
7	1	0	0	1	1
8	1	0	1	1	1
9	1	0	2	1	1
10	1	1	0	1	1
11	1	1	1	1	1
12	1	1	2	1	1

Tabular representations



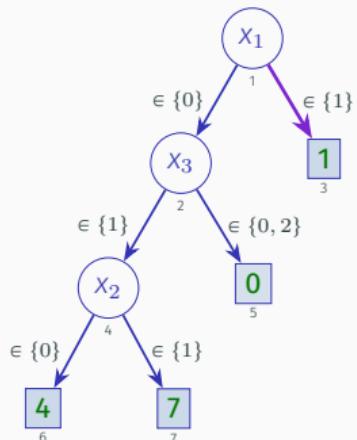
DT2

XPs: AXps/CXps		
DT	AXps	CXps
DT1	{1}	{1}
DT2	{1}	{1}

Adversarial Examples	
DT	$l_0$ -minimal AEs
DT1	{1}
DT2	{1}

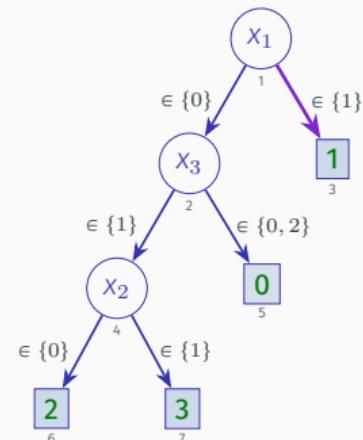
Shapley values			
DT	Sc(1)	Sc(2)	Sc(3)
DT1	0.000	0.083	-0.500
DT2	0.278	0.028	-0.222

# Computing XPs, AEs & Svs – what???



DT1

row #	$x_1$	$x_2$	$x_3$	$\kappa_1(\mathbf{x})$	$\kappa_2(\mathbf{x})$
1	0	0	0	0	0
2	0	0	1	4	2
3	0	0	2	0	0
4	0	1	0	0	0
5	0	1	1	7	3
6	0	1	2	0	0
7	1	0	0	1	1
8	1	0	1	1	1
9	1	0	2	1	1
10	1	1	0	1	1
11	1	1	1	1	1
12	1	1	2	1	1



DT2

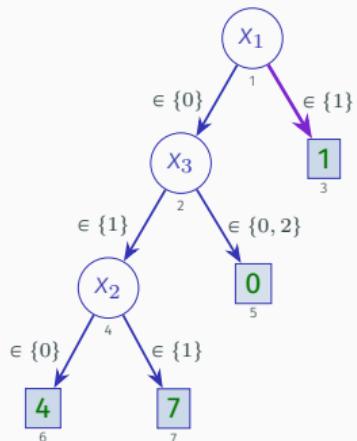
Tabular representations

XPs: AXps/CXps		
DT	AXps	CXps
DT1	{1}	{1}
DT2	{1}	{1}

Adversarial Examples	
DT	$l_0$ -minimal AEs
DT1	{1}
DT2	{1}

Shapley values				
DT	Sc(1)	Sc(2)	Sc(3)	
DT1	0.000	0.083	-0.500	!!!
DT2	0.278	0.028	-0.222	

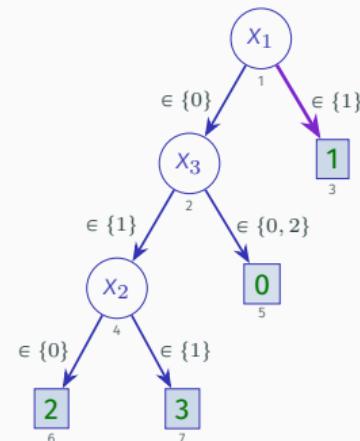
# Computing XPs, AEs & Svs – what???



DT1

row #	$x_1$	$x_2$	$x_3$	$\kappa_1(\mathbf{x})$	$\kappa_2(\mathbf{x})$
1	0	0	0	0	0
2	0	0	1	4	2
3	0	0	2	0	0
4	0	1	0	0	0
5	0	1	1	7	3
6	0	1	2	0	0
7	1	0	0	1	1
8	1	0	1	1	1
9	1	0	2	1	1
10	1	1	0	1	1
11	1	1	1	1	1
12	1	1	2	1	1

Tabular representations



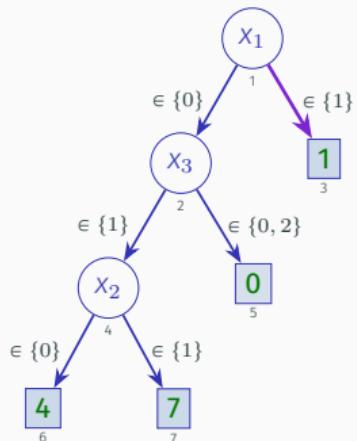
DT2

XPs: AXps/CXps		
DT	AXps	CXps
DT1	{1}	{1}
DT2	{1}	{1}

Adversarial Examples	
DT	$l_0$ -minimal AEs
DT1	{1}
DT2	{1}

Shapley values				
DT	Sc(1)	Sc(2)	Sc(3)	
DT1	0.000	0.083	-0.500	!!!
DT2	0.278	0.028	-0.222	!!

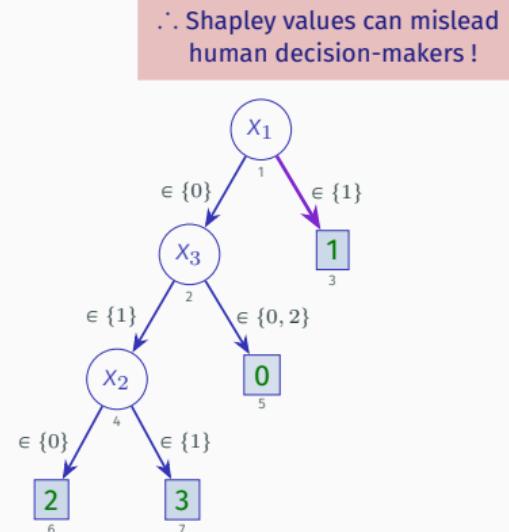
# Computing XPs, AEs & Svs – what???



DT1

row #	$x_1$	$x_2$	$x_3$	$\kappa_1(\mathbf{x})$	$\kappa_2(\mathbf{x})$
1	0	0	0	0	0
2	0	0	1	4	2
3	0	0	2	0	0
4	0	1	0	0	0
5	0	1	1	7	3
6	0	1	2	0	0
7	1	0	0	1	1
8	1	0	1	1	1
9	1	0	2	1	1
10	1	1	0	1	1
11	1	1	1	1	1
12	1	1	2	1	1

Tabular representations



DT2

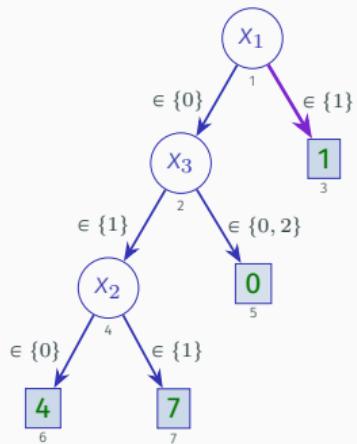
∴ Shapley values can mislead human decision-makers !

XPs: AXps/CXps		
DT	AXps	CXps
DT1	{1}	{1}
DT2	{1}	{1}

Adversarial Examples	
DT	$l_0$ -minimal AEs
DT1	{1}
DT2	{1}

Shapley values				
DT	Sc(1)	Sc(2)	Sc(3)	
DT1	0.000	0.083	-0.500	!!!
DT2	0.278	0.028	-0.222	!!

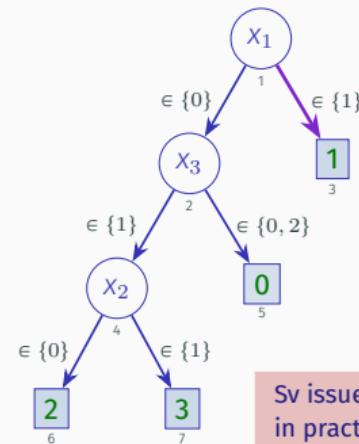
# Computing XPs, AEs & Svs – what???



DT1

row #	$x_1$	$x_2$	$x_3$	$\kappa_1(\mathbf{x})$	$\kappa_2(\mathbf{x})$
1	0	0	0	0	0
2	0	0	1	4	2
3	0	0	2	0	0
4	0	1	0	0	0
5	0	1	1	7	3
6	0	1	2	0	0
7	1	0	0	1	1
8	1	0	1	1	1
9	1	0	2	1	1
10	1	1	0	1	1
11	1	1	1	1	1
12	1	1	2	1	1

Tabular representations



DT2

. . . Shapley values can mislead human decision-makers !

Sv issues also occur in practice [HM23e]

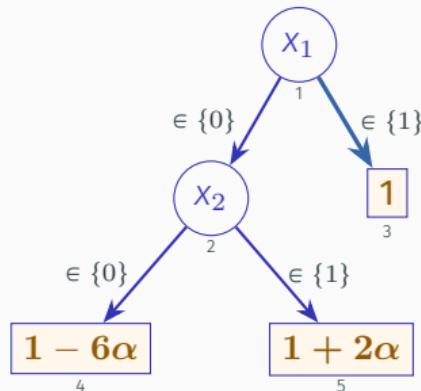
XPs: AXps/CXps		
DT	AXps	CXps
DT1	{1}	{1}
DT2	{1}	{1}

Adversarial Examples	
DT	$l_0$ -minimal AEs
DT1	{1}
DT2	{1}

Shapley values				
DT	Sc(1)	Sc(2)	Sc(3)	
DT1	0.000	0.083	-0.500	!!!
DT2	0.278	0.028	-0.222	!!

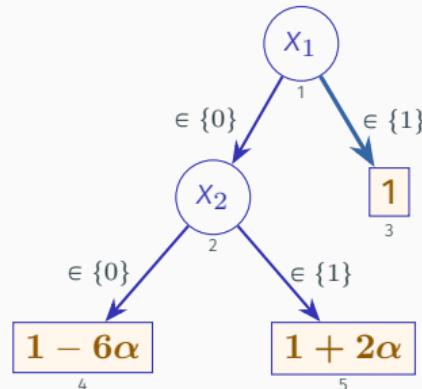
## Another example – arbitrary mistakes!

[LHAMS24]



## Another example – arbitrary mistakes!

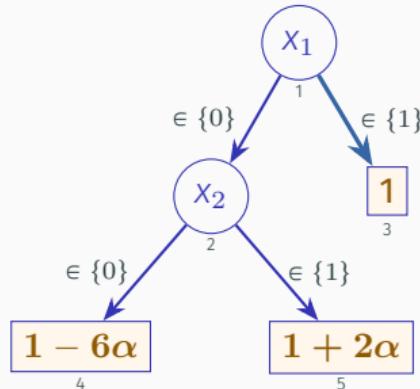
[LHAMS24]



- Instance:  $((1, 1), 1)$
- Obs:  $\alpha \neq 0$

## Another example – arbitrary mistakes!

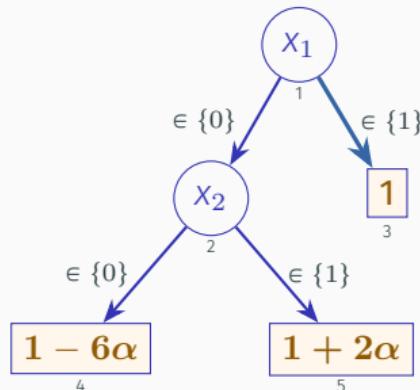
[LHAMS24]



- Instance:  $((1, 1), 1)$
- Obs:  $\alpha \neq 0$
- $\text{Sc}(1) = 0$
- $\text{Sc}(2) = \alpha$

## Another example – arbitrary mistakes!

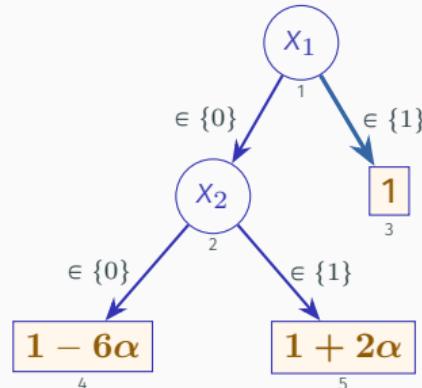
[LHAMS24]



- Instance:  $((1, 1), 1)$
- Obs:  $\alpha \neq 0$
- $\text{Sc}(1) = 0$
- $\text{Sc}(2) = \alpha$  (you can pick the  $\alpha$ ...)

## Another example – arbitrary mistakes!

[LHAMS24]

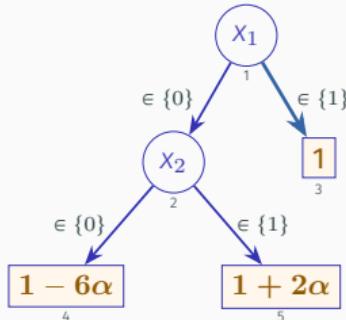


- Instance:  $((1, 1), 1)$
- Obs:  $\alpha \neq 0$
- $\text{Sc}(1) = 0$
- $\text{Sc}(2) = \alpha$  (you can pick the  $\alpha$ ...)

Example devised by O. Letoffe, PhD student at IRIT

## More detail

row	$x_1$	$x_2$	$\rho(\mathbf{x})$	$\rho_a(\mathbf{x})$ $\alpha = 1/2$	$\rho_b(\mathbf{x})$ $\alpha = 1/4$
1	0	0	$1 - 6\alpha$	-2	$-1/2$
2	0	1	$1 + 2\alpha$	2	$3/2$
3	1	0	1	1	1
4	1	1	1	1	1



$\mathcal{S}$	$\text{rows}(\mathcal{S})$	$v_e(\mathcal{S})$
$\emptyset$	1, 2, 3, 4	$1 - \alpha$
$\{x_1\}$	3, 4	1
$\{x_2\}$	2, 4	$1 + \alpha$
$\{x_1, x_2\}$	4	1

$i = 1$					
$\mathcal{S}$	$v_e(\mathcal{S})$	$v_e(\mathcal{S} \cup \{1\})$	$\Delta_1(\mathcal{S})$	$\varsigma(\mathcal{S})$	$\varsigma(\mathcal{S}) \times \Delta_1(\mathcal{S})$
$\emptyset$	$1 - \alpha$	1	$\alpha$	$1/2$	$\alpha/2$
$\{2\}$	$1 + \alpha$	1	$-\alpha$	$1/2$	$-\alpha/2$
$\text{Sc}_E(1) = 0$					
$i = 2$					
$\mathcal{S}$	$v_e(\mathcal{S})$	$v_e(\mathcal{S} \cup \{2\})$	$\Delta_2(\mathcal{S})$	$\varsigma(\mathcal{S})$	$\varsigma(\mathcal{S}) \times \Delta_2(\mathcal{S})$
$\emptyset$	$1 - \alpha$	$1 + \alpha$	$2\alpha$	$1/2$	$\alpha$
$\{1\}$	1	1	0	$1/2$	0
$\text{Sc}_E(2) = \alpha$					

## SHAP scores also fail with regression models

[LHM24]

- Let  $\mathcal{F} = \{1, 2\}$ ,  $\mathbb{D}_1 = \mathbb{D}_2 = \mathbb{D} = [-1/2, 3/2]$ ,  $\mathbb{F} = \mathbb{D} \times \mathbb{D}$ 
  - Also, let  $\mathbb{D}^+ = [1/2, 3/2]$  and  $\mathbb{D}^- = \mathbb{D} \setminus \mathbb{D}^+$
- Regression model maps to real values, i.e.  $\mathbb{K} = \mathbb{R}$ :

$$\rho_2(x_1, x_2) = \begin{cases} x_1 & \text{if } x_1 \in \mathbb{D}^+ \\ x_2 - 2 & \text{if } x_1 \notin \mathbb{D}^+ \wedge x_2 \notin \mathbb{D}^+ \\ x_2 + 1 & \text{if } x_1 \notin \mathbb{D}^+ \wedge x_2 \in \mathbb{D}^+ \end{cases}$$

- Average values unchanged (wrt previous example), and so  $Sc(1) = 0$  and  $Sc(2) = \alpha$

## SHAP scores even fail when Lipschitz continuity holds!

$$\rho_3(x_1, x_2) = \begin{cases} x_1 & \text{if } x_2 \leq 1 \wedge \alpha x_1 \leq \alpha \\ (1 + 4|\alpha|)x_1 - 4|\alpha| & \text{if } x_2 \leq 1 \wedge \alpha x_1 \geq \alpha \\ 28|\alpha|x_1x_2 + (1 - 28|\alpha|)x_1 - 28|\alpha|x_2 + 28|\alpha| & \text{if } x_2 \geq 1 \wedge \alpha x_1 \leq \alpha \\ -4|\alpha|x_1x_2 + (1 + 8|\alpha|)x_1 + 4|\alpha|x_2 - 8|\alpha| & \text{if } x_2 \geq 1 \wedge \alpha x_1 \geq \alpha \end{cases}$$

[LHM24]

- As before, average values unchanged, and so  $\text{Sc}(1) = 0$  and  $\text{Sc}(2) = \alpha$

# Outline – Unit #07

Exact Shapley Values for XAI

Myth #03: Shapley Values for XAI

Corrected SHAP Scores

A Correct New SHAP – nuSHAP

Voting Power & Power Indices

Feature Importance Scores

## Corrected SHAP scores & feature importance scores

[LHMS24, LHAMS24]

- Is the theory of Shapley values **incorrect?**

## Corrected SHAP scores & feature importance scores

[LHMS24, LHAMS24]

- Is the theory of Shapley values **incorrect?**    **No!**

# Corrected SHAP scores & feature importance scores

[LHMS24, LHAMS24]

- Is the theory of Shapley values **incorrect?** **No!**
- What is inadequate is the **characteristic function** used in XAI
  - In XAI: characteristic function uses the **expected value**
  - This defines the *marginal contribution* in SHAP lingo...

[SK10, SK14, LL17]

# Corrected SHAP scores & feature importance scores

[LHMS24, LHAMS24]

- Is the theory of Shapley values **incorrect?** **No!**
- What is inadequate is the **characteristic function** used in XAI
  - In XAI: characteristic function uses the **expected value**
  - This defines the *marginal contribution* in SHAP lingo...
- Replace characteristic function based on **expected values** by new characteristic function based on **AXps/WAXps**
  - Resulting scores are (**still**) Shapley values & identified issues no longer observed

[LHMS24]

# Corrected SHAP scores & feature importance scores

[LHMS24, LHAMS24]

- Is the theory of Shapley values **incorrect?** **No!**
- What is inadequate is the **characteristic function** used in XAI
  - In XAI: characteristic function uses the **expected value**
  - This defines the *marginal contribution* in SHAP lingo...
- Replace characteristic function based on **expected values** by new characteristic function based on **AXps/WAXps**
  - Resulting scores are (**still**) Shapley values & identified issues no longer observed
- Observed tight connection between feature attribution and power indices from a priori voting power

[SK10, SK14, LL17]

[LHMS24]

# Corrected SHAP scores & feature importance scores

[LHMS24, LHAMS24]

- Is the theory of Shapley values **incorrect?** **No!**
- What is inadequate is the **characteristic function** used in XAI
  - In XAI: characteristic function uses the **expected value**
  - This defines the *marginal contribution* in SHAP lingo...
- Replace characteristic function based on **expected values** by new characteristic function based on **AXps/WAXps**
  - Resulting scores are (**still**) Shapley values & identified issues no longer observed
- Observed tight connection between feature attribution and power indices from a priori voting power
  - **Feature importance scores** (more later):
    - Generalize recent axiomatic aggregations
    - Adapt best known power indices
    - Devise new scores for XAI

[SK10, SK14, LL17]

[LHMS24]

[LHAMS24]

[BIL<sup>+</sup>24]

## An initial compromise

[LHAMS24]

- Replace the characteristic function used for SHAP scores:

$$v_e(\mathcal{S}) := \mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}]$$

# An initial compromise

[LHAMS24]

- Replace the characteristic function used for SHAP scores:

$$v_e(\mathcal{S}) := \mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}]$$

- Recall the similarity predicate:

$$\sigma(\mathbf{x}) = \begin{cases} 1 & \text{if } (\kappa(\mathbf{x}) = \kappa(\mathbf{v})) \\ 0 & \text{otherwise} \end{cases}$$

# An initial compromise

[LHAMS24]

- Replace the characteristic function used for SHAP scores:

$$v_e(\mathcal{S}) := \mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}]$$

- Recall the similarity predicate:

$$\sigma(\mathbf{x}) = \begin{cases} 1 & \text{if } (\kappa(\mathbf{x}) = \kappa(\mathbf{v})) \\ 0 & \text{otherwise} \end{cases}$$

- The new characteristic function becomes:

$$v_s(\mathcal{S}) := \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}]$$

## An initial compromise

[LHAMS24]

- Replace the characteristic function used for SHAP scores:

$$v_e(\mathcal{S}) := \mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}]$$

- Recall the similarity predicate:

$$\sigma(\mathbf{x}) = \begin{cases} 1 & \text{if } (\kappa(\mathbf{x}) = \kappa(\mathbf{v})) \\ 0 & \text{otherwise} \end{cases}$$

- The new characteristic function becomes:

$$v_s(\mathcal{S}) := \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}]$$

- Issues with non-boolean classifiers **disappear**; issues with boolean classifiers **remain**

# An initial compromise

[LHAMS24]

- Replace the characteristic function used for SHAP scores:

$$v_e(\mathcal{S}) := \mathbf{E}[\tau(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}]$$

- Recall the similarity predicate:

$$\sigma(\mathbf{x}) = \begin{cases} 1 & \text{if } (\kappa(\mathbf{x}) = \kappa(\mathbf{v})) \\ 0 & \text{otherwise} \end{cases}$$

- The new characteristic function becomes:

$$v_s(\mathcal{S}) := \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}]$$

- Issues with non-boolean classifiers **disappear**; issues with boolean classifiers **remain**
- Developed SSHAP prototype using SHAP's code base

[LHMS24]

## Fixing the known issues of SHAP scores

## Fixing the known issues of SHAP scores

- New characteristic function (based on WAXps):

$$v_a(\mathcal{S}) := \begin{cases} 1 & \text{if } \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1 \\ 0 & \text{otherwise} \end{cases}$$

## Fixing the known issues of SHAP scores

- New characteristic function (based on WAXps):

$$v_a(\mathcal{S}) := \begin{cases} 1 & \text{if } \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1 \\ 0 & \text{otherwise} \end{cases}$$

- Recall:  $\mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1$  holds iff  $\mathcal{S}$  is a WAXp!

## Fixing the known issues of SHAP scores

- New characteristic function (based on WAXps):

$$v_a(\mathcal{S}) := \begin{cases} 1 & \text{if } \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1 \\ 0 & \text{otherwise} \end{cases}$$

- Recall:  $\mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1$  holds iff  $\mathcal{S}$  is a WAXp!
- Known issues of SHAP scores guaranteed **not** to occur

## Fixing the known issues of SHAP scores

- New characteristic function (based on WAXps):

$$v_a(\mathcal{S}) := \begin{cases} 1 & \text{if } \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1 \\ 0 & \text{otherwise} \end{cases}$$

- Recall:  $\mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1$  holds iff  $\mathcal{S}$  is a WAXp!
- Known issues of SHAP scores guaranteed **not** to occur
- **Corrected** SHAP scores reveal tight connection between XAI by feature selection (i.e. WAXps) and feature attribution

# Outline – Unit #07

Exact Shapley Values for XAI

Myth #03: Shapley Values for XAI

Corrected SHAP Scores

A Correct New SHAP – nuSHAP

Voting Power & Power Indices

Feature Importance Scores

# A glimpse of NuSHAP

- Shapley values estimated with well-known algorithm – CGT
  - Strong theoretical guarantees on approximation of Shapley values
  - With enough sampling, given by  $\epsilon$  and  $\alpha$ :

[CGT09]

$$\text{Prob}[|\hat{Sc}(i) - Sc(i)| \leq \epsilon] \geq 1 - \alpha$$

- E.g. we used  $\epsilon = 0.0015$  and  $\alpha = 0.015$

# A glimpse of NuSHAP

- Shapley values estimated with well-known algorithm – CGT
  - Strong theoretical guarantees on approximation of Shapley values
  - With enough sampling, given by  $\epsilon$  and  $\alpha$ :

[CGT09]

$$\text{Prob}[|\hat{Sc}(i) - Sc(i)| \leq \epsilon] \geq 1 - \alpha$$

- E.g. we used  $\epsilon = 0.0015$  and  $\alpha = 0.015$
- Replace expected value with novel characteristic function: test for WAXp
  - New approach for computing symbolic explanations, targeting scalability for complex models
    - Replace WAXp check with sbWAXp check, i.e. polynomial-time check
    - Flexibility in the sample chosen, e.g. training data

## NuSHAP vs. \*SHAP

- Compute SHAP scores with NuSHAP and SHAP
- Compare features' rankings using Rank-Biased Overlap (RBO) [WMZ10]
  - Give weight to differences in first  $k$  elements of ranking
  - E.g.  $\text{RBO}([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [0, 1, 2, 3, 4, 5, 6, 7, 8, 9], k = 5) = 1.0$
- Results for different datasets and ML models (using scikit, XGBoost):
  - LR: (adult, corral); DT: (iris, mux6); KNN: (connect\_4, spambase, spectf);  
BT: (clean1, coil2000, dna); CNN: (MNIST)

## NuSHAP vs. \*SHAP

- Compute SHAP scores with NuSHAP and SHAP
- Compare features' rankings using Rank-Biased Overlap (RBO) [WMZ10]
  - Give weight to differences in first  $k$  elements of ranking
  - E.g.  $\text{RBO}([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [0, 1, 2, 3, 4, 9, 8, 7, 6, 5], k = 5) = 1.0$
- Results for different datasets and ML models (using scikit, XGBoost):
  - LR: (adult, corral); DT: (iris, mux6); KNN: (connect\_4, spambase, spectf);  
BT: (clean1, coil2000, dna); CNN: (MNIST)

## NuSHAP vs. \*SHAP

- Compute SHAP scores with NuSHAP and SHAP
- Compare features' rankings using Rank-Biased Overlap (RBO)
  - Give weight to differences in first  $k$  elements of ranking
  - E.g.  $\text{RBO}([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [4, 3, 2, 1, 0, 5, 6, 7, 8, 9], k = 5) = 0.42$
- Results for different datasets and ML models (using scikit, XGBoost):
  - LR: (adult, corral); DT: (iris, mux6); KNN: (connect\_4, spambase, spectf);  
BT: (clean1, coil2000, dna); CNN: (MNIST)

[WMZ10]

## NuSHAP vs. \*SHAP

- Compute SHAP scores with NuSHAP and SHAP
- Compare features' rankings using Rank-Biased Overlap (RBO) [WMZ10]
  - Give weight to differences in first  $k$  elements of ranking
  - E.g.  $\text{RBO}([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [5, 6, 7, 8, 9, 0, 1, 2, 3, 4], k = 5) = 0.0$
- Results for different datasets and ML models (using scikit, XGBoost):
  - LR: (adult, corral); DT: (iris, mux6); KNN: (connect\_4, spambase, spectf);  
BT: (clean1, coil2000, dna); CNN: (MNIST)

## NuSHAP vs. \*SHAP

- Compute SHAP scores with NuSHAP and SHAP
- Compare features' rankings using Rank-Biased Overlap (RBO)
  - Give weight to differences in first  $k$  elements of ranking
  - E.g.  $\text{RBO}([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [5, 6, 7, 8, 9, 0, 1, 2, 3, 4], k = 5) = 0.0$
- Results for different datasets and ML models (using scikit, XGBoost):
  - LR: (adult, corral); DT: (iris, mux6); KNN: (connect\_4, spambase, spectf);  
BT: (clean1, coil2000, dna); CNN: (MNIST)

[WMZ10]

		adult	corral	iris	mux6	connect_4	spambase	spectf	clean1	coil2000	dna	MNIST
Min	nuSHAP vs. SHAP	0.08	0.17	0.31	0.32	0.0	0.01	0.0	0.0	0.0	0.0	0.0
	nuSHAP vs.  SHAP	0.05	0.12	0.27	0.32	0.0	0.05	0.0	0.0	0.0	0.03	0.0
Max	nuSHAP vs. SHAP	0.96	0.96	0.94	0.97	0.9	0.94	0.91	0.69	0.69	0.88	0.06
	nuSHAP vs.  SHAP	0.88	0.97	0.94	0.95	0.77	0.94	0.91	0.88	0.69	0.88	0.06
Mean	nuSHAP vs. SHAP	0.37	0.53	0.84	0.7	0.21	0.41	0.2	0.12	0.05	0.17	0.0
	nuSHAP vs.  SHAP	0.31	0.5	0.84	0.69	0.19	0.42	0.19	0.17	0.08	0.43	0.0

## NuSHAP vs. \*SHAP – run times

- Compute SHAP scores with NuSHAP and SHAP
- Compare features' rankings using Rank-Biased Overlap (RBO)
  - Give weight to differences in first  $k$  elements of ranking
  - E.g.  $\text{RBO}([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [5, 6, 7, 8, 9, 0, 1, 2, 3, 4], k = 5) = 0.0$
- Results for different datasets and ML models (using scikit, XGBoost):
  - LR: (adult, corral); DT: (iris, mux6); KNN: (connect\_4, spambase, spectf);  
BT: (clean1, coil2000, dna); CNN: (MNIST)

		adult	corral	iris	mux6	connect_4	spambase	spectf	clean1	coil2000	dna	MNIST
Min	nuSHAP vs. SHAP	0.08	0.17	0.31	0.32	0.0	0.01	0.0	0.0	0.0	0.0	0.0
	nuSHAP vs.  SHAP	0.05	0.12	0.27	0.32	0.0	0.05	0.0	0.0	0.0	0.03	0.0
Max	nuSHAP vs. SHAP	0.96	0.96	0.94	0.97	0.9	0.94	0.91	0.69	0.69	0.88	0.06
	nuSHAP vs.  SHAP	0.88	0.97	0.94	0.95	0.77	0.94	0.91	0.88	0.69	0.88	0.06
Mean	nuSHAP vs. SHAP	0.37	0.53	0.84	0.7	0.21	0.41	0.2	0.12	0.05	0.17	0.0
	nuSHAP vs.  SHAP	0.31	0.5	0.84	0.69	0.19	0.42	0.19	0.17	0.08	0.43	0.0

Tool SHAP produces results of very poor quality!

## NuSHAP vs. \*SHAP – run times

- Compute SHAP scores with NuSHAP and SHAP
- Compare features' rankings using Rank-Biased Overlap (**RBO**)
  - Give weight to differences in first  $k$  elements of ranking
  - E.g.  $\text{RBO}([0, 1, 2, 3, 4, 5, 6, 7, 8, 9], [5, 6, 7, 8, 9, 0, 1, 2, 3, 4], k = 5) = 0.0$
- Results for different datasets and ML models (using scikit, XGBoost):
  - LR: (adult, corral); DT: (iris, mux6); KNN: (connect\_4, spambase, spectf);  
BT: (clean1, coil2000, dna); CNN: (MNIST)

	adult	corral	iris	mux6	connect_4	spambase	spectf	clean1	coil2000	dna	MNIST
SHAP	3.4	<b>0.1</b>	0.0	0.0	21.7	0.5	0.7	6.8	28.2	23.0	281.3
nuSHAP	<b>1.9</b>	1.5	1.5	1.5	<b>4.5</b>	2.7	2.9	2.7	<b>1.7</b>	<b>4.5</b>	<b>48.9</b>
#Samples	68045.5	63.7	955.4	63.9	9202.5	13654.1	36459.5	3929.6	2960.8	2756.8	2929.9

# Outline – Unit #07

Exact Shapley Values for XAI

Myth #03: Shapley Values for XAI

Corrected SHAP Scores

A Correct New SHAP – nuSHAP

Voting Power & Power Indices

Feature Importance Scores

## Recap: weighted voting games

- General set up of weighted voting games:
  - Assembly  $\mathcal{A}$  of voters, with  $m = |\mathcal{A}|$
  - Each voter  $i \in \mathcal{A}$  votes **Yes** with  $n_i$  votes; otherwise no votes are counted (and he/she votes **No**)
  - A coalition is a subset of voters,  $\mathcal{C} \subseteq \mathcal{A}$
  - Quota  $q$  is the sum of votes required for a proposal to be approved
    - Coalitions leading to sums not less than  $q$  are **winning** coalitions
  - A **weighted voting game (WVG)** is a tuple  $[q; n_1, \dots, n_m]$ 
    - Example:  $[12; 4, 4, 4, 2, 2, 1]$
  - Problem: find a measure of importance of each voter !
    - I.e. measure the **a priori voting power** of each voter

## What are power indices?

- Power indices assign a measure of importance to each voter

## What are power indices?

- Power indices assign a measure of importance to each voter
- Many power indices proposed over the years:

- Penrose [Pen46]
- Shapley-Shubik [SS54]
- Banzhaf [Bil65]
- Coleman [Col71]
- Johnston [Joh78]
- Deegan-Packel [DP78]
- Holler-Packel [HP83]
- Andjiga [ACL03]
- Responsability\* [CH04, BIL<sup>+</sup>24]
- ...

# What are power indices?

- Power indices assign a measure of importance to each voter
- Many power indices proposed over the years:
  - Penrose [Pen46]
  - Shapley-Shubik [SS54]
  - Banzhaf [BI65]
  - Coleman [Col71]
  - Johnston [Joh78]
  - Deegan-Packel [DP78]
  - Holler-Packel [HP83]
  - Andjiga [ACLO3]
  - Responsability\* [CH04, BIL<sup>+</sup>24]
  - ...
- What characterizes power indices?
  - Account for the cases when voter is *critical* for a winning coalition
    - E.g. in previous example, Luxembourg is never critical for a winning coalition
  - Account for whether coalition is subset-minimal or cardinality-minimal

## Towards defining power indices

- Understanding **criticality** (used at least since 1954):

[SS54]

## Towards defining power indices

- Understanding **criticality** (used at least since 1954):  
[SS54]
  - Since the work of Shapley-Shubik [SS54], the **criticality** of a voter has been accounted for:  
*"Our definition of the power of an individual member depends on the chance he has of being critical to the success of a winning coalition."*

## Towards defining power indices

- Understanding **criticality** (used at least since 1954):  
[SS54]
  - Since the work of Shapley-Shubik [SS54], the **criticality** of a voter has been accounted for:  
*"Our definition of the power of an individual member depends on the chance he has of being critical to the success of a winning coalition."*
  - This means that a voter  $i$  is **critical** when:
    - If the voter votes Yes, then we have a winning coalition; and
    - If the voter votes No, then we have a losing coalition.

## Towards defining power indices

- Understanding **criticality** (used at least since 1954):  
[SS54]
  - Since the work of Shapley-Shubik [SS54], the **criticality** of a voter has been accounted for:  
*"Our definition of the power of an individual member depends on the chance he has of being critical to the success of a winning coalition."*
  - This means that a voter  $i$  is **critical** when:
    - If the voter votes Yes, then we have a winning coalition; and
    - If the voter votes No, then we have a losing coalition.
- Understanding (subset-)minimal winning coalitions:

- Understanding **criticality** (used at least since 1954):  
[SS54]
  - Since the work of Shapley-Shubik [SS54], the **criticality** of a voter has been accounted for:  
*"Our definition of the power of an individual member depends on the chance he has of being critical to the success of a winning coalition."*
  - This means that a voter  $i$  is **critical** when:
    - If the voter votes Yes, then we have a winning coalition; and
    - If the voter votes No, then we have a losing coalition.
- Understanding (subset-)minimal winning coalitions:
  - A winning coalition is subset-minimal if removing any single voter results in a losing coalition

## Towards defining power indices

- Understanding **criticality** (used at least since 1954):  
[SS54]
  - Since the work of Shapley-Shubik [SS54], the **criticality** of a voter has been accounted for:  
*"Our definition of the power of an individual member depends on the chance he has of being critical to the success of a winning coalition."*
  - This means that a voter  $i$  is **critical** when:
    - If the voter votes Yes, then we have a winning coalition; and
    - If the voter votes No, then we have a losing coalition.
- Understanding (subset-)minimal winning coalitions:
  - A winning coalition is subset-minimal if removing any single voter results in a losing coalition
  - A winning coalition is cardinality-minimal if it has the smallest cardinality among subset-minimal winning coalitions

## Towards defining power indices

- Understanding **criticality** (used at least since 1954):  
[SS54]
  - Since the work of Shapley-Shubik [SS54], the **criticality** of a voter has been accounted for:  
*"Our definition of the power of an individual member depends on the chance he has of being critical to the success of a winning coalition."*
  - This means that a voter  $i$  is **critical** when:
    - If the voter votes Yes, then we have a winning coalition; and
    - If the voter votes No, then we have a losing coalition.
- Understanding (subset-)minimal winning coalitions:
  - A winning coalition is subset-minimal if removing any single voter results in a losing coalition
  - A winning coalition is cardinality-minimal if it has the smallest cardinality among subset-minimal winning coalitions
  - Recall that minimal winning coalitions can be obtained by computing the AXps of a monotonically increasing boolean classifier

## Example power indices I

[LHAMS24]

- Necessary definitions (using formal XAI notation...):

$$\mathbb{W}\mathbb{A}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{WAXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

$$\mathbb{W}\mathbb{C}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{WCXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

$$\mathbb{A}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

$$\mathbb{C}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

- Definitions of  $\mathbb{W}\mathbb{A}$ ,  $\mathbb{W}\mathbb{C}$ ,  $\mathbb{A}$ , and  $\mathbb{C}$  mimic the ones above, but without specifying a voter

## Example power indices I

[LHAMS24]

- Necessary definitions (using formal XAI notation...):

$$\mathbb{W}\mathbb{A}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{WAXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

$$\mathbb{W}\mathbb{C}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{WCXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

$$\mathbb{A}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

$$\mathbb{C}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

- Definitions of  $\mathbb{W}\mathbb{A}$ ,  $\mathbb{W}\mathbb{C}$ ,  $\mathbb{A}$ , and  $\mathbb{C}$  mimic the ones above, but without specifying a voter
- Power indices of Holler-Packel and Deegan-Packel:

[HP83, DP78]

$$\text{Sc}_H(i; \mathcal{E}) = \sum_{\mathcal{S} \in \mathbb{A}_i(\mathcal{E})} (1/|\mathbb{A}(\mathcal{E})|)$$

$$\text{Sc}_D(i; \mathcal{E}) = \sum_{\mathcal{S} \in \mathbb{A}_i(\mathcal{E})} (1/(|\mathcal{S}| \times |\mathbb{A}(\mathcal{E})|))$$

# Example power indices I

[LHAMS24]

- Necessary definitions (using formal XAI notation...):

$$\text{WA}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{WAXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

$$\text{WC}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{WCXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

$$\text{A}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{AXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

$$\text{C}_i(\mathcal{E}) = \{\mathcal{S} \subseteq \mathcal{F} \mid \text{CXp}(\mathcal{S}; \mathcal{E}) \wedge i \in \mathcal{S}\}$$

- Definitions of  $\text{WA}$ ,  $\text{WC}$ ,  $\text{A}$ , and  $\text{C}$  mimic the ones above, but without specifying a voter
- Power indices of Holler-Packel and Deegan-Packel:

[HP83, DP78]

$$\text{Sc}_H(i; \mathcal{E}) = \sum_{\mathcal{S} \in \text{A}_i(\mathcal{E})} (1/|\text{A}(\mathcal{E})|)$$

$$\text{Sc}_D(i; \mathcal{E}) = \sum_{\mathcal{S} \in \text{A}_i(\mathcal{E})} (1/(|\mathcal{S}| \times |\text{A}(\mathcal{E})|))$$

- **Obs:** One *only* needs the **AXps**

## Example power indices II

- Additional definitions:

$$\text{Crit}(i, \mathcal{S}; \mathcal{E}) := \text{WAXp}(\mathcal{S}; \mathcal{E}) \wedge \neg \text{WAXp}(\mathcal{S} \setminus \{i\}; \mathcal{E})$$

## Example power indices II

- Additional definitions:

$$\text{Crit}(i, \mathcal{S}; \mathcal{E}) := \text{WAXp}(\mathcal{S}; \mathcal{E}) \wedge \neg \text{WAXp}(\mathcal{S} \setminus \{i\}; \mathcal{E})$$

- Power indices of Shapley-Shubik, Banzhaf and Johnston:

[SS54, BI65, Joh78]

$$Sc_S(i; \mathcal{E}) = \sum_{\mathcal{S} \subseteq \mathcal{F} \wedge \text{Crit}(i, \mathcal{S}; \mathcal{E})} \left( 1 / \binom{|\mathcal{F}| - 1}{|\mathcal{S}| - 1} \right)$$

$$Sc_B(i; \mathcal{E}) = \sum_{\mathcal{S} \subseteq \mathcal{F} \wedge \text{Crit}(i, \mathcal{S}; \mathcal{E})} \left( 1 / 2^{|\mathcal{F}| - 1} \right)$$

$$Sc_J(i; \mathcal{E}) = \sum_{\mathcal{S} \subseteq \mathcal{F} \wedge \text{Crit}(i, \mathcal{S}; \mathcal{E})} \left( 1 / \Delta(\mathcal{S}) \right)$$

## Example power indices II

- Additional definitions:

$$\text{Crit}(i, \mathcal{S}; \mathcal{E}) := \text{WAXp}(\mathcal{S}; \mathcal{E}) \wedge \neg \text{WAXp}(\mathcal{S} \setminus \{i\}; \mathcal{E})$$

- Power indices of Shapley-Shubik, Banzhaf and Johnston:

[SS54, BI65, Joh78]

$$Sc_S(i; \mathcal{E}) = \sum_{\mathcal{S} \subseteq \mathcal{F} \wedge \text{Crit}(i, \mathcal{S}; \mathcal{E})} \left( 1 / \binom{|\mathcal{F}| - 1}{|\mathcal{S}| - 1} \right)$$

$$Sc_B(i; \mathcal{E}) = \sum_{\mathcal{S} \subseteq \mathcal{F} \wedge \text{Crit}(i, \mathcal{S}; \mathcal{E})} \left( 1 / 2^{|\mathcal{F}| - 1} \right)$$

$$Sc_J(i; \mathcal{E}) = \sum_{\mathcal{S} \subseteq \mathcal{F} \wedge \text{Crit}(i, \mathcal{S}; \mathcal{E})} \left( 1 / \Delta(\mathcal{S}) \right)$$

- One needs the **WAXps** to find critical voters...

## Example #01

- WVG: [9; 9, 2, 2, 2, 2, 1, 1]

## Example #01

- WVG: [9; 9, 2, 2, 2, 2, 1, 1]
- AXps:

1					
2	3	4	5	6	
2	3	4	5	7	

## Example #01

- WVG: [9; 9, 2, 2, 2, 2, 1, 1]
- AXps:

1					
2	3	4	5	6	
2	3	4	5	7	

- Holler-Packel scores:  $\langle 0.333, 0.667, 0.667, 0.667, 0.667, 0.333, 0.333 \rangle$
- Banzhaf scores (normalized):  $\langle 0.813, 0.040, 0.040, 0.040, 0.040, 0.013, 0.013 \rangle$
- Shapley-Shubik scores:  $\langle 0.810, 0.043, 0.043, 0.043, 0.043, 0.010, 0.010 \rangle$
- Different relative orders of voter importance... which ones seem more realistic?

## Example #02

- WVG: [16; 10, 6, 4, 2, 2]

## Example #02

- WVG: [16; 10, 6, 4, 2, 2]
- AXps:

1	2	
1	3	4
1	3	5

## Example #02

- WVG: [16; 10, 6, 4, 2, 2]
- AXps:

1	2	
1	3	4
1	3	5

- Deegan-Packel scores:  $\langle 0.389, 0.167, 0.222, 0.111, 0.111 \rangle$
- Banzhaf scores (normalized):  $\langle 0.524, 0.238, 0.143, 0.048, 0.048 \rangle$
- Shapley-Shubik scores:  $\langle 0.617, 0.200, 0.117, 0.033, 0.033 \rangle$
- Different relative orders of voter importance... which ones seem more realistic?

## Example #03

- WVG: [6; 4, 2, 1, 1, 1, 1]

## Example #03

- WVG: [6; 4, 2, 1, 1, 1, 1]
- AXps:

2	3	4	5	6
1	3	4		
1	4	5		
1	4	6		
1	3	6		
1	5	6		
1	2			
1	3	5		

## Example #03

- WVG: [6; 4, 2, 1, 1, 1, 1]
- AXps:

2	3	4	5	6
1	3	4		
1	4	5		
1	4	6		
1	3	6		
1	5	6		
1	2			
1	3	5		

- Deegan-Packel scores:  $\langle 0.312, 0.087, 0.150, 0.150, 0.150, 0.150 \rangle$
- Banzhaf scores (normalized):  $\langle 0.542, 0.125, 0.083, 0.083, 0.083, 0.083 \rangle$
- Shapley-Shubik scores:  $\langle 0.533, 0.133, 0.083, 0.083, 0.083, 0.083 \rangle$
- Different relative orders of voter importance... which ones seem more realistic?

## Example #04

- WVG: [21; 12, 9, 4, 4, 1, 1, 1]

## Example #04

- WVG: [21; 12, 9, 4, 4, 1, 1, 1]
- AXps:

1	2		
1	3	4	5
1	3	4	6
1	3	4	7

## Example #04

- WVG: [21; 12, 9, 4, 4, 1, 1, 1]
- AXps:

1	2		
1	3	4	5
1	3	4	6
1	3	4	7

- Deegan-Packel scores:  $\langle 0.312, 0.125, 0.188, 0.188, 0.062, 0.062, 0.062 \rangle$
- Banzhaf scores (normalized):  $\langle 0.481, 0.309, 0.086, 0.086, 0.012, 0.012, 0.012 \rangle$
- Shapley-Shubik scores:  $\langle 0.574, 0.257, 0.074, 0.074, 0.007, 0.007, 0.007 \rangle$
- Different relative orders of voter importance... which ones seem more realistic?

# Outline – Unit #07

Exact Shapley Values for XAI

Myth #03: Shapley Values for XAI

Corrected SHAP Scores

A Correct New SHAP – nuSHAP

Voting Power & Power Indices

Feature Importance Scores

## From power indices to feature importance scores

- A **Feature Importance Score** (FIS) is a measure of feature importance in XAI, parameterizable on an **explanation problem** and a chosen **characteristic function**
  - Explanation problem:  $(\mathcal{M}, (\mathbf{v}, q))$
  - Define characteristic function using explanation problem (more next slide)
- Obs: Can adapt (generalized) power indices as templates for feature importance scores
- Obs: Can devise new templates and/or new FISs

## Some examples (1 of 2)

- More notation:

$$\Delta_i(\mathcal{S}; \mathcal{E}, v) = v(\mathcal{S}; \mathcal{E}) - v(\mathcal{S} \setminus \{i\}; \mathcal{E})$$

- Can use **any** characteristic function, including those presented earlier in this lecture

## Some examples (1 of 2)

- More notation:

$$\Delta_i(\mathcal{S}; \mathcal{E}, v) = v(\mathcal{S}; \mathcal{E}) - v(\mathcal{S} \setminus \{i\}; \mathcal{E})$$

- Can use **any** characteristic function, including those presented earlier in this lecture

- Some templates:

- Shapley-Shubik:

$$\text{TSc}_S(i; \mathcal{E}, v) := \sum_{\mathcal{S} \in \{\mathcal{T} \subseteq \mathcal{F} \mid i \in \mathcal{T}\}} \left( \frac{\Delta_i(\mathcal{S}; \mathcal{E}, v)}{|\mathcal{F}| \times \binom{|\mathcal{F}|-1}{|\mathcal{S}|-1}} \right)$$

- Banzhaf:

$$\text{TSc}_B(i; \mathcal{E}, v) := \sum_{\mathcal{S} \in \{\mathcal{T} \subseteq \mathcal{F} \mid i \in \mathcal{T}\}} \left( \frac{\Delta_i(\mathcal{S}; \mathcal{E}, v)}{2^{|\mathcal{F}|-1}} \right)$$

## Some examples (1 of 2)

- More notation:

$$\Delta_i(\mathcal{S}; \mathcal{E}, v) = v(\mathcal{S}; \mathcal{E}) - v(\mathcal{S} \setminus \{i\}; \mathcal{E})$$

- Can use **any** characteristic function, including those presented earlier in this lecture

- Some templates:

- Shapley-Shubik:

$$\text{TSc}_S(i; \mathcal{E}, v) := \sum_{\mathcal{S} \in \{\mathcal{T} \subseteq \mathcal{F} \mid i \in \mathcal{T}\}} \left( \frac{\Delta_i(\mathcal{S}; \mathcal{E}, v)}{|\mathcal{F}| \times \binom{|\mathcal{F}|-1}{|\mathcal{S}|-1}} \right)$$

- Banzhaf:

$$\text{TSc}_B(i; \mathcal{E}, v) := \sum_{\mathcal{S} \in \{\mathcal{T} \subseteq \mathcal{F} \mid i \in \mathcal{T}\}} \left( \frac{\Delta_i(\mathcal{S}; \mathcal{E}, v)}{2^{|\mathcal{F}|-1}} \right)$$

- Can use other templates

## Some examples (1 of 2)

- More notation:

$$\Delta_i(\mathcal{S}; \mathcal{E}, v) = v(\mathcal{S}; \mathcal{E}) - v(\mathcal{S} \setminus \{i\}; \mathcal{E})$$

- Can use **any** characteristic function, including those presented earlier in this lecture

- Some templates:

- Shapley-Shubik:

$$TSc_S(i; \mathcal{E}, v) := \sum_{\mathcal{S} \in \{\mathcal{T} \subseteq \mathcal{F} \mid i \in \mathcal{T}\}} \left( \frac{\Delta_i(\mathcal{S}; \mathcal{E}, v)}{|\mathcal{F}| \times \binom{|\mathcal{F}|-1}{|\mathcal{S}|-1}} \right)$$

- Banzhaf:

$$TSc_B(i; \mathcal{E}, v) := \sum_{\mathcal{S} \in \{\mathcal{T} \subseteq \mathcal{F} \mid i \in \mathcal{T}\}} \left( \frac{\Delta_i(\mathcal{S}; \mathcal{E}, v)}{2^{|\mathcal{F}|-1}} \right)$$

- Can use other templates
- Can devise FISs without exploiting existing templates

## Some examples (2 of 2)

- Recall WAXp-based characteristic function:

$$v_a(\mathcal{S}) := \begin{cases} 1 & \text{if } \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1 \\ 0 & \text{otherwise} \end{cases}$$

## Some examples (2 of 2)

- Recall WAXp-based characteristic function:

$$v_a(\mathcal{S}) := \begin{cases} 1 & \text{if } \mathbf{E}[\sigma(\mathbf{x}) \mid \mathbf{x}_{\mathcal{S}} = \mathbf{v}_{\mathcal{S}}] = 1 \\ 0 & \text{otherwise} \end{cases}$$

- Some FISs:

- Shapley-Shubik:

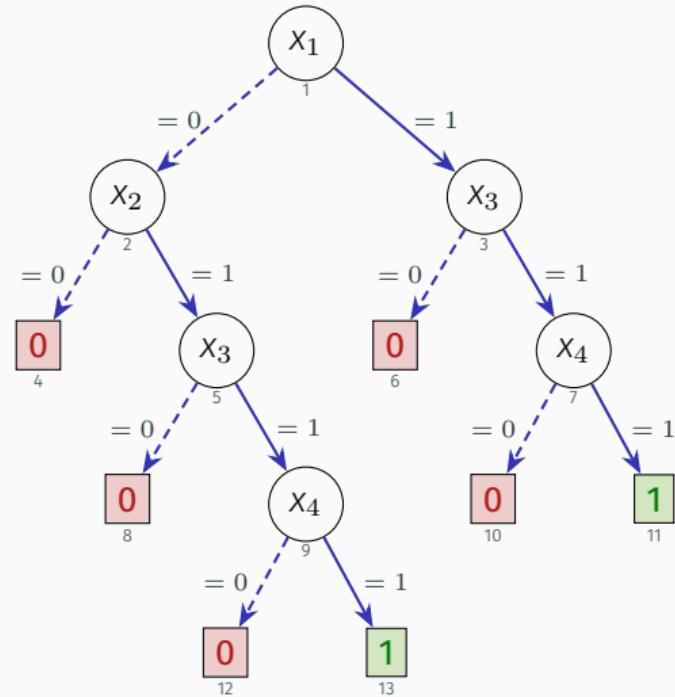
$$\text{Sc}_S(i; \mathcal{E}) := \text{TSc}_S(i; \mathcal{E}, v_a) := \sum_{\mathcal{S} \in \{\mathcal{T} \subseteq \mathcal{F} \mid i \in \mathcal{T}\}} \left( \frac{\Delta_i(\mathcal{S}; \mathcal{E}, v_a)}{|\mathcal{F}| \times \binom{|\mathcal{F}|-1}{|\mathcal{S}|-1}} \right)$$

- Banzhaf:

$$\text{Sc}_B(i; \mathcal{E}) := \text{TSc}_B(i; \mathcal{E}, v_a) := \sum_{\mathcal{S} \in \{\mathcal{T} \subseteq \mathcal{F} \mid i \in \mathcal{T}\}} \left( \frac{\Delta_i(\mathcal{S}; \mathcal{E}, v_a)}{2^{|\mathcal{F}|-1}} \right)$$

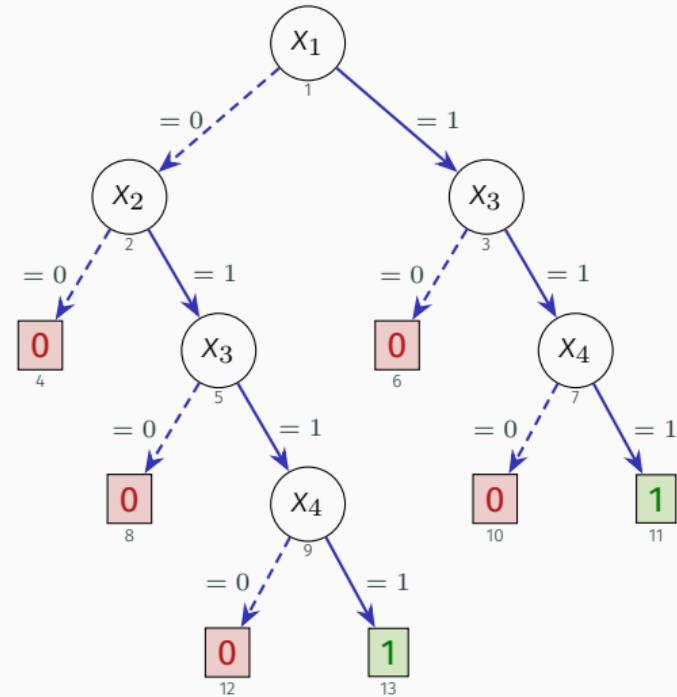
## A concrete example

- AXps:  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
- Feature attribution:



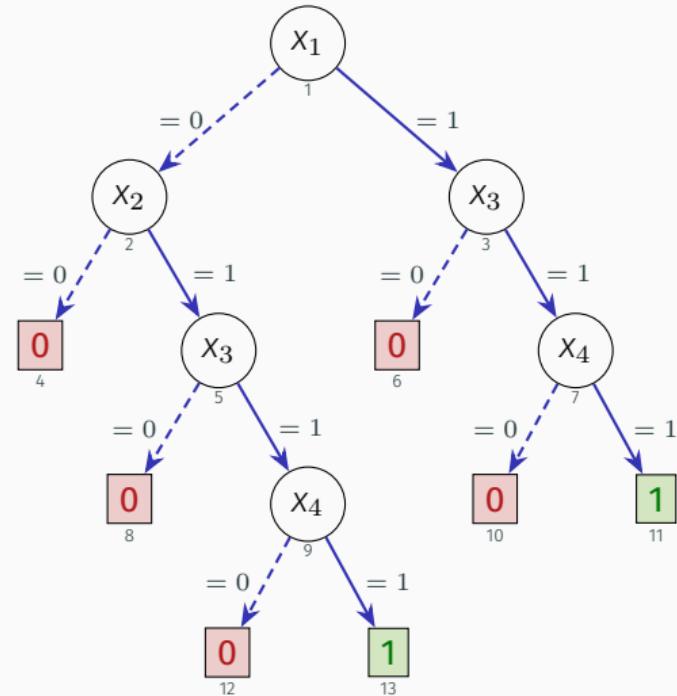
## A concrete example

- AXps:  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
- Feature attribution:
  - SS:  $\langle 0.083, 0.083, 0.417, 0.417 \rangle$



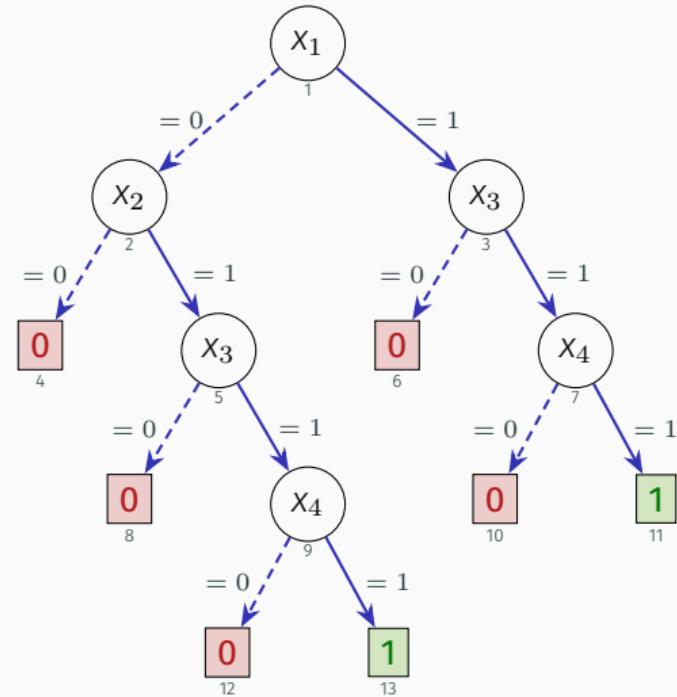
## A concrete example

- AXps:  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
- Feature attribution:
  - SS:  $\langle 0.083, 0.083, 0.417, 0.417 \rangle$
  - B (norm.):  $\langle 0.125, 0.125, 0.375, 0.375 \rangle$



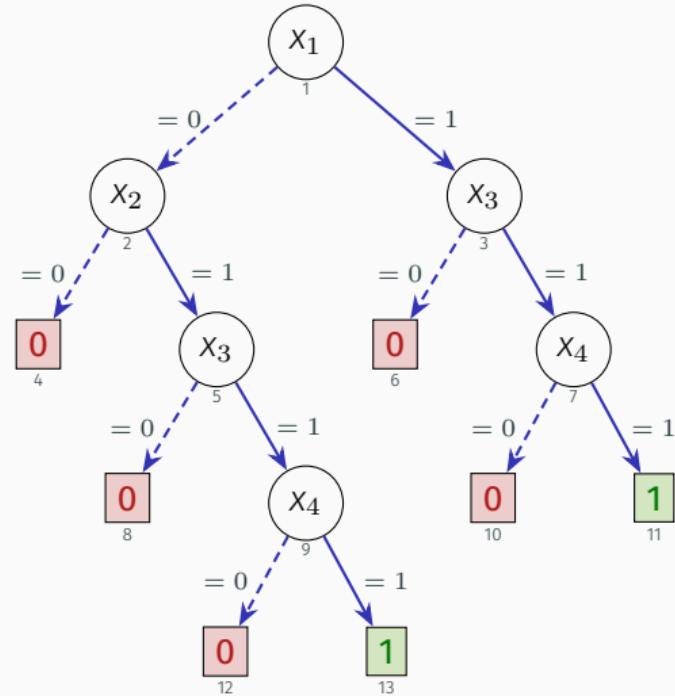
## A concrete example

- AXps:  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
- Feature attribution:
  - SS:  $\langle 0.083, 0.083, 0.417, 0.417 \rangle$
  - B (norm.):  $\langle 0.125, 0.125, 0.375, 0.375 \rangle$
  - J (norm.):  $\langle 0.111, 0.111, 0.389, 0.389 \rangle$



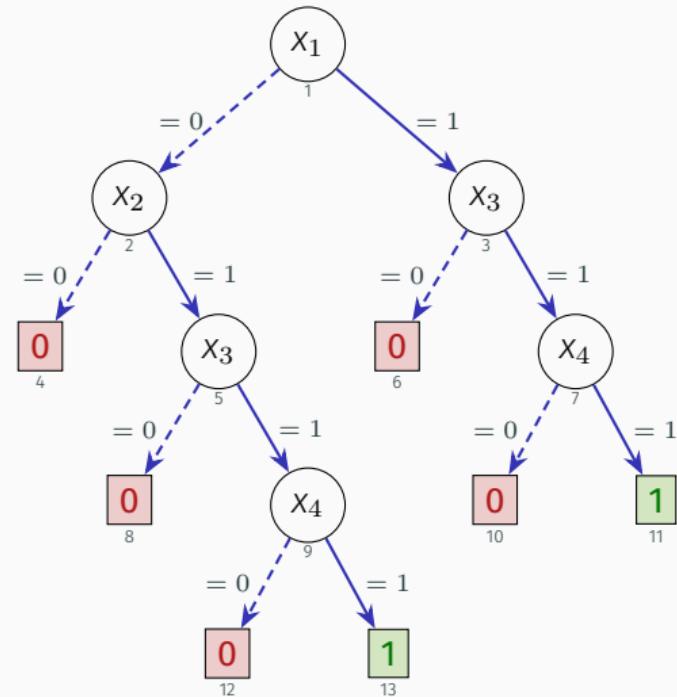
## A concrete example

- AXps:  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
- Feature attribution:
  - SS:  $\langle 0.083, 0.083, 0.417, 0.417 \rangle$
  - B (norm.):  $\langle 0.125, 0.125, 0.375, 0.375 \rangle$
  - J (norm.):  $\langle 0.111, 0.111, 0.389, 0.389 \rangle$
  - HP:  $\langle 0.167, 0.167, 0.333, 0.333 \rangle$



## A concrete example

- AXps:  $\{\{1, 3, 4\}, \{2, 3, 4\}\}$
- Feature attribution:
  - SS:  $\langle 0.083, 0.083, 0.417, 0.417 \rangle$
  - B (norm.):  $\langle 0.125, 0.125, 0.375, 0.375 \rangle$
  - J (norm.):  $\langle 0.111, 0.111, 0.389, 0.389 \rangle$
  - HP:  $\langle 0.167, 0.167, 0.333, 0.333 \rangle$
  - DP:  $\langle 0.167, 0.167, 0.333, 0.333 \rangle$



# Questions?

Unit #08

## Conclusions & Research Directions

## Outline – Unit #08

Some Words of Concern

Conclusions & Research Directions

# Can non-symbolic XAI's myths be stopped?

SHAP on 2023/05/31:

The screenshot shows a Google Scholar search results page. The search query is "A unified approach to interpreting model predictions". The top result is a paper by SM Lundberg and SI Lee, titled "A unified approach to interpreting model predictions". The abstract discusses the tension between accuracy and interpretability in complex models. The page includes filters for time (Any time, Since 2023, Since 2022, Since 2019, Custom range...), sorting options (Sort by relevance, Sort by date), and other search parameters (Any type, Review articles, include patents, include citations). A sidebar on the left shows a list of filters and sorting options.

A unified approach to interpreting model predictions

SM Lundberg, SI Lee - Advances in neural information ..., 2017 - proceedings.neurips.cc

Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications. However, the highest accuracy for large modern datasets is often achieved by complex models that even experts struggle to interpret, such as ensemble or deep learning models, creating a tension between accuracy and interpretability. In response, various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related and ...

☆ Save 99 Cite Cited by 13080 Related articles All 17 versions

Any time

Since 2023

Since 2022

Since 2019

Custom range...

Sort by relevance

Sort by date

Any type

Review articles

include patents

include citations

[PDF] neurips.cc

# Can non-symbolic XAI's myths be stopped?

SHAP on 2024/09/15:

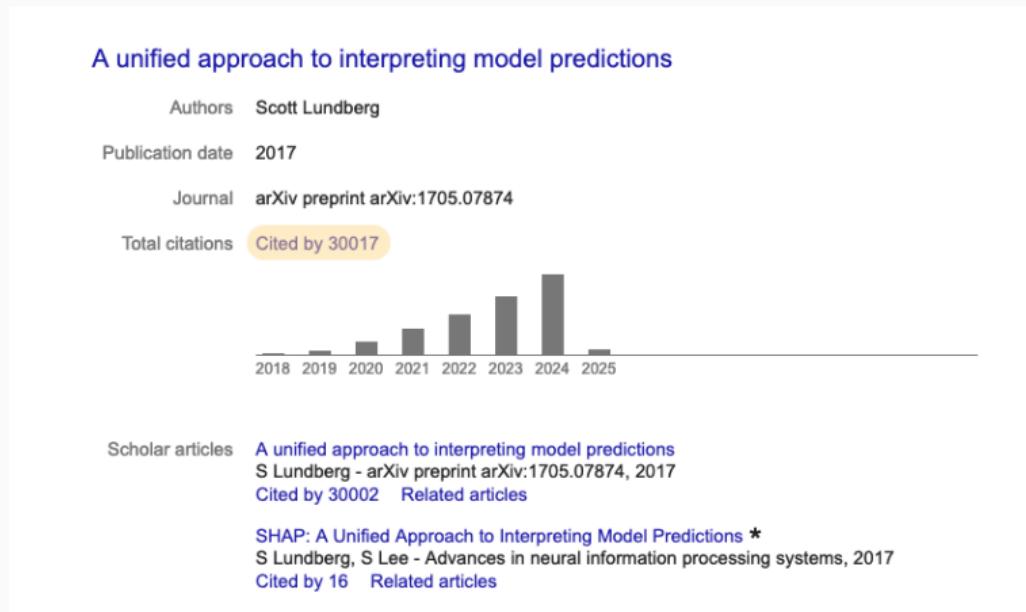
The screenshot shows a Google Scholar search results page. The search query is "A unified approach to interpreting model predictions". The results list two papers:

- [CITATION] A unified approach to interpreting model predictions**  
S Lundberg - arXiv preprint arXiv:1705.07874, 2017  
☆ Save ⚡ Cite Cited by 25778 Related articles
- [CITATION] A unified approach to interpreting model predictions**  
M Scott, L Su-In - Advances in neural information ..., 2017 - Curran Associates, Inc  
☆ Save ⚡ Cite Cited by 108 Related articles

On the left sidebar, there are filters for time (Any time, Since 2024, Since 2023, Since 2020, Custom range...), sorting options (Sort by relevance, Sort by date), and type (Any type, Review articles). At the bottom, there are checkboxes for "include patents" (unchecked) and "include citations" (checked).

# Can non-symbolic XAI's myths be stopped?

SHAP on 2025/01/12:



# Can non-symbolic XAI's myths be stopped?

SHAP on 2025/07/14:

## A unified approach to interpreting model predictions

Authors Scott M Lundberg, Su-In Lee

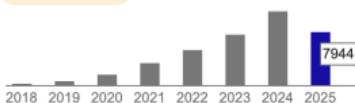
Publication date 2017

Journal Advances in neural information processing systems

Volume 30

Description Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications. However, the highest accuracy for large modern datasets is often achieved by complex models that even experts struggle to interpret, such as ensemble or deep learning models, creating a tension between accuracy and interpretability. In response, various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related and when one method is preferable over another. To address this problem, we present a unified framework for interpreting predictions, SHAP (SHapley Additive exPlanations). SHAP assigns each feature an importance value for a particular prediction. Its novel components include:(1) the identification of a new class of additive feature importance measures, and (2) theoretical results showing there is a unique solution in this class with a set of desirable properties. The new class unifies six existing methods, notable because several recent methods in the class lack the proposed desirable properties. Based on insights from this unification, we present new methods that show improved computational performance and/or better consistency with human intuition than previous approaches.

Total citations Cited by 37527



# Can non-symbolic XAI's myths be stopped?

SHAP on 2025/07/14:

## A unified approach to interpreting model predictions

Authors Scott M Lundberg, Su-In Lee

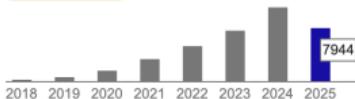
Publication date 2017

Journal Advances in neural information processing systems

Volume 30

Description Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications. However, the highest accuracy for large modern datasets is often achieved by complex models that even experts struggle to interpret, such as ensemble or deep learning models, creating a tension between accuracy and interpretability. In response, various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related and when one method is preferable over another. To address this problem, we present a unified framework for interpreting predictions, SHAP (SHapley Additive exPlanations). SHAP assigns each feature an importance value for a particular prediction. Its novel components include:(1) the identification of a new class of additive feature importance measures, and (2) theoretical results showing there is a unique solution in this class with a set of desirable properties. The new class unifies six existing methods, notable because several recent methods in the class lack the proposed desirable properties. Based on insights from this unification, we present new methods that show improved computational performance and/or better consistency with human intuition than previous approaches.

Total citations Cited by 37527



But,

Theoretical SHAP scores can mislead;  
& results of tool SHAP of poor quality!

# Many, many high-risk uses of SHAP & clones

He et al. *Journal of Translational Medicine* (2024) 22:686  
https://doi.org/10.1186/s12967-024-05417-y

Journal of Translational  
Medicine

RESEARCH

Open Access



Predictive models for personalized precision medical intervention in spontaneous regression stages of cervical precancerous lesions

XGBoost-SHAP-based interpretable diagnostic framework for alzheimer's disease

Hypothesis-free discovery of novel cancer predictors using machine learning

Explainable AI-based Deep-SHAP for mapping the multivariate relationships between regional neuroimaging biomarkers and cognition

npj Digital Medicine

Interpretable prediction of 30-day mortality in patients with acute pancreatitis based on machine learning and SHAP

scientific reports

scientific reports

OPEN

An explainable machine learning framework for lung cancer hospital length of stay prediction

© J. Marques-Silva

PLOS ONE

RESEARCH ARTICLE

Combining explainable machine learning, demographic and multi-omic data to inform precision medicine strategies for inflammatory bowel disease

scientific reports

OPEN

Explanatory predictive model for COVID-19 severity risk employing machine learning, shapley addition, and LIME



ARTICLE OPEN

Comparing machine learning algorithms for predicting ICU admission and mortality in COVID-19



scientific reports

OPEN

The importance of interpreting machine learning models for blood glucose prediction in diabetes: an analysis using SHAP



OPEN SHAP based predictive modeling for 1 year all-cause readmission risk in elderly heart failure patients: feature selection and model interpretation



230 / 238

# What next?

What next? Massive retraction (of tens of thousands) of papers??



thanks to ChatGPT...

What next? Massive retraction (of tens of thousands) of papers??



& beware of high-risk uses of SHAP! 😞

thanks to ChatGPT...

What's the bottom line?

# What's the bottom line?

- Non-symbolic XAI research experiences a persistent “*Don’t Look Up*” moment...



# What's the bottom line?

- Non-symbolic XAI research experiences a persistent “*Don’t Look Up*” moment...



BTW, there are a multitude  
of proposed uses of  
LIME/Shap in medicine... ⚠

## Some unsettling works...

- For DTs:
  - One AXp in polynomial-time [IIM20, HIIM21, IIM22]
  - All CXps in polynomial-time [HIIM21, IIM22]

# Some unsettling works...

- For DTs:
  - One AXp in polynomial-time [IIM20, HIIM21, IIM22]
  - All CXps in polynomial-time [HIIM21, IIM22]

## Declarative Reasoning on Explanations Using Constraint Logic Programming

**Abstract.** Explaining opaque Machine Learning (ML) models is an increasingly relevant problem. Current explanation in AI (XAI) methods suffer several shortcomings, among others an insufficient incorporation of background knowledge, and a lack of abstraction and interactivity with the user. We propose REASONX, an explanation method based on Constraint Logic Programming (CLP). REASONX can provide declarative, interactive explanations for decision trees, which can be the ML models under analysis or global/local surrogate models of any black-box model. Users can express background or common sense knowledge using linear constraints and MILP optimization over features of factual and contrastive instances, and interact with the answer constraints at different levels of abstraction through constraint projection. We present here the architecture of REASONX, which consists of a Python layer, closer to the user, and a CLP layer. REASONX's core execution engine is a Prolog meta-program with declarative semantics in terms of logic theories.

arXiv:2309.00422v1 [cs.AI] 1 Sep 2023

# Some unsettling works...

- For DTs:
  - One AXp in polynomial-time
  - All CXps in polynomial-time

[IIM20, HIIM21, IIM22]

[HIIM21, IIM22]

*HHAI 2024: Hybrid Human AI Systems for the Social Good*  
F. Lorig et al. (Eds.)  
© 2024 The Authors.

*This article is published online with Open Access by IOS Press and distributed under the terms  
of the Creative Commons Attribution Non-Commercial License 4.0 (CC BY-NC 4.0).  
doi:10.3233/FAIA240183*

## Exploring Large Language Models Capabilities to Explain Decision Trees

# Some unsettling works...

- For DTs:
  - One AXp in polynomial-time
  - All CXps in polynomial-time

[IIM20, HIIM21, IIM22]

[HIIM21, IIM22]

## **Explainable Artificial Intelligence for Academic Performance Prediction. An Experimental Study on the Impact of Accuracy and Simplicity of Decision Trees on Causability and Fairness Perceptions**

*FAccT '24, June 03–06, 2024, Rio de Janeiro, Brazil*  
© 2024 Copyright held by the owner/author(s).  
ACM ISBN 979-8-4007-0450-5/24/06  
<https://doi.org/10.1145/3630106.3658953>

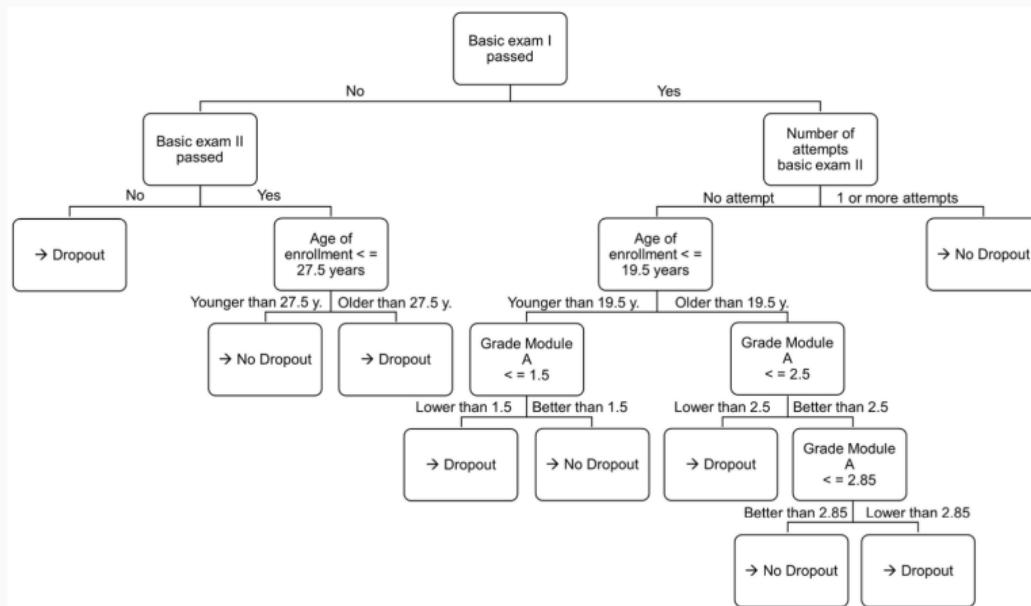
# Some unsettling works...

- For DTs:

- One AXp in polynomial-time
- All CXps in polynomial-time

[IIM20, HIIM21, IIM22]

[HIIM21, IIM22]



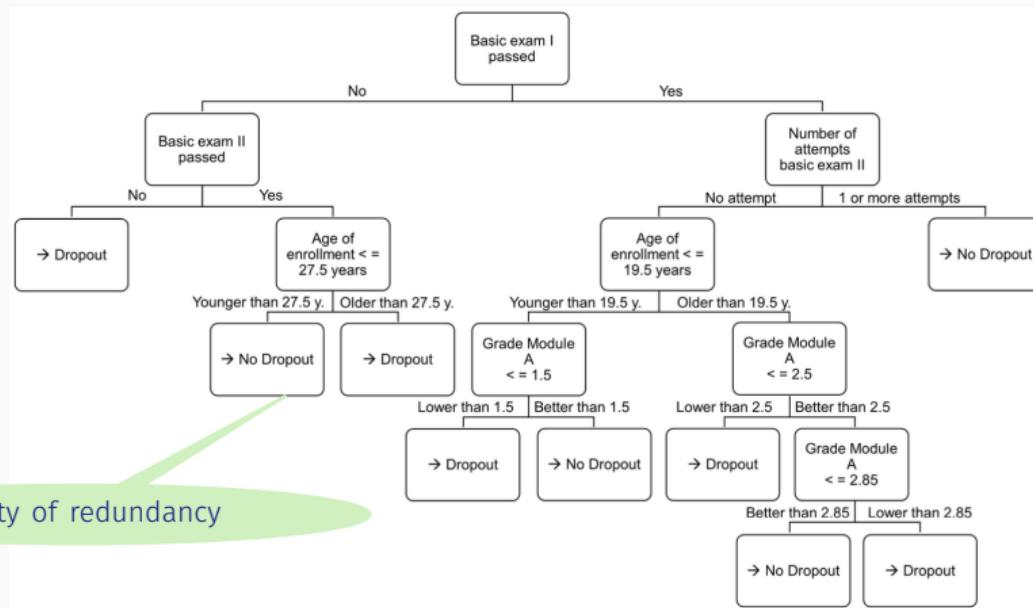
# Some unsettling works...

- For DTs:

- One AXp in polynomial-time
- All CXps in polynomial-time

[IIM20, HIIM21, IIM22]

[HIIM21, IIM22]



## Outline – Unit #08

Some Words of Concern

Conclusions & Research Directions

# Conclusions

- Covered logic-based (aka symbolic, aka formal) XAI & its recent progress:
  - Abductive & contrastive explanations
  - Reviewed their computation in practice
  - Duality & enumeration
  - Other explainability queries – feature necessity & relevancy

# Conclusions

- Covered logic-based (aka symbolic, aka formal) XAI & its recent progress:
  - Abductive & contrastive explanations
  - Reviewed their computation in practice
  - Duality & enumeration
  - Other explainability queries – feature necessity & relevancy
- Showed that formal XAI **disproves** some myths of (heuristic) XAI:
  - Explainability using intrinsic interpretability is a **myth**
  - The rigor of model-agnostic explanations is a **myth**
  - The rigor of SHAP scores as a measure of relative feature importance is a **myth**

# Conclusions

- Covered logic-based (aka symbolic, aka formal) XAI & its recent progress:
  - Abductive & contrastive explanations
  - Reviewed their computation in practice
  - Duality & enumeration
  - Other explainability queries – feature necessity & relevancy
- Showed that formal XAI **disproves** some myths of (heuristic) XAI:
  - Explainability using intrinsic interpretability is a **myth**
  - The rigor of model-agnostic explanations is a **myth**
  - The rigor of SHAP scores as a measure of relative feature importance is a **myth**
- Demonstrated tight connection between (rigorous) feature selection and (rigorous) feature attribution in XAI

# Conclusions

- Covered logic-based (aka symbolic, aka formal) XAI & its recent progress:
  - Abductive & contrastive explanations
  - Reviewed their computation in practice
  - Duality & enumeration
  - Other explainability queries – feature necessity & relevancy
- Showed that formal XAI **disproves** some myths of (heuristic) XAI:
  - Explainability using intrinsic interpretability is a **myth**
  - The rigor of model-agnostic explanations is a **myth**
  - The rigor of SHAP scores as a measure of relative feature importance is a **myth**
- Demonstrated tight connection between (rigorous) feature selection and (rigorous) feature attribution in XAI
- Symbolic XAI exhibits links with many fields of research:  
machine learning, artificial intelligence, formal methods, automated reasoning, optimization, computational social choice (& game theory), etc.

# Research directions

## Research directions

- Scalability, scalability, and scalability

## Research directions

- Scalability, scalability, and scalability
- Sample-based explanations & coherency

## Research directions

- Scalability, scalability, and scalability
- Sample-based explanations & coherency
- Rigorous feature attribution

## Research directions

- Scalability, scalability, and scalability
- Sample-based explanations & coherency
- Rigorous feature attribution
- Checking & certification of XAI tools

## Research directions

- Scalability, scalability, and scalability
- Sample-based explanations & coherency
- Rigorous feature attribution
- Checking & certification of XAI tools
- Open complexity results

## Research directions

- Scalability, scalability, and scalability
- Sample-based explanations & coherency
- Rigorous feature attribution
- Checking & certification of XAI tools
- Open complexity results
- Preferred explanations

## Research directions

- Scalability, scalability, and scalability
- Sample-based explanations & coherency
- Rigorous feature attribution
- Checking & certification of XAI tools
- Open complexity results
- Preferred explanations
- Distance-restricted explanations

## Research directions

- Scalability, scalability, and scalability
- Sample-based explanations & coherency
- Rigorous feature attribution
- Checking & certification of XAI tools
- Open complexity results
- Preferred explanations
- Distance-restricted explanations
- Probabilistic explanations

## Research directions

- Scalability, scalability, and scalability
- Sample-based explanations & coherency
- Rigorous feature attribution
- Checking & certification of XAI tools
- Open complexity results
- Preferred explanations
- Distance-restricted explanations
- Probabilistic explanations
- Any ideas from you?

## Research directions

- Scalability, scalability, and scalability
- Sample-based explanations & coherency
- Rigorous feature attribution
- Checking & certification of XAI tools
- Open complexity results
- Preferred explanations
- Distance-restricted explanations
- Probabilistic explanations
- Any ideas from you?
- ... And trying to curb the **massive** momentum of (heuristic) XAI **myths!**

# What this course covered

- Lecture 01 – unit(s):
  - #01: Foundations
- Lecture 02 – unit(s):
  - #02: Principles of symbolic XAI – feature selection
  - #03: Tractability in symbolic XAI (& myth of interpretability)
- Lecture 03 – unit(s):
  - #04: Intractability in symbolic XAI (& myth of model-agnostic XAI)
  - #05: Explainability queries
- Lecture 04 – unit(s):
  - #06: Recent, emerging & advanced topics
- Lecture 05 – unit(s):
  - #07: Principles of symbolic XAI – feature attribution (& myth of Shapley values in XAI)
  - #08: Corrected feature attribution – nuSHAP
  - #09: Conclusions & research directions

## Some food for thought...

*"All truths are easy to understand once they are discovered; the point is to discover them."*

(G. Galilei)

*"Beware of false knowledge; it is more dangerous than ignorance."*

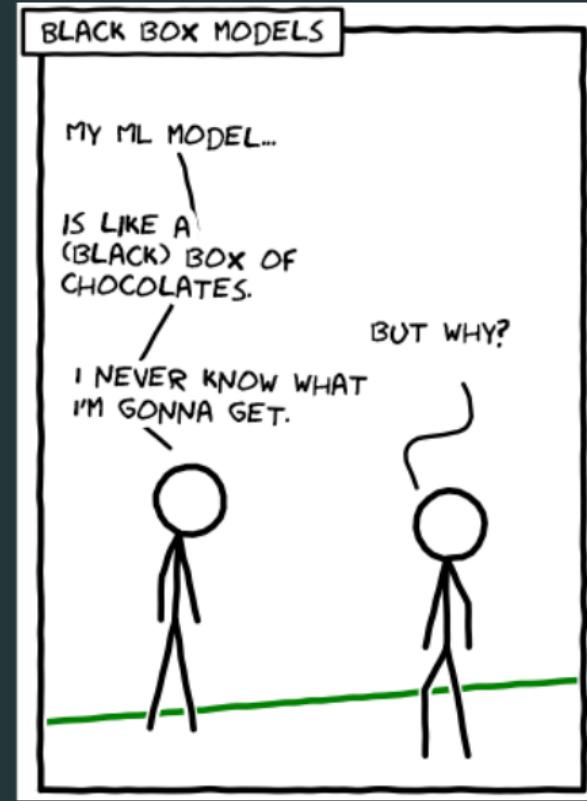
(G. B. Shaw)

*"He who loves practice without theory is like the sailor who boards ship without a rudder and compass and never knows where he may cast."*

(L. da Vinci)

## Q & A

Acknowledgment: joint work with N. Asher, R. Béjar, F. Chiariello, M. Cooper, H. Hu, X. Huang, A. Hurault, A. Ignatiev, Y. Izza, J. Lefebvre-Lobaina, O. Létoffé, V. Martinez, C. Mencía, A. Morgado, N. Narodytska, R. Passos, J. Planes, M. Siala, P. Stuckey et al.



<https://arxiv.org/abs/1901.01686> & <http://crmnx.io/edit/>

# References i

- [ABBM21] Marcelo Arenas, Pablo Barceló, Leopoldo E. Bertossi, and Mikaël Monet.  
**The tractability of SHAP-score-based explanations for classification over deterministic and decomposable boolean circuits.**  
In AAAI, pages 6670–6678, 2021.
- [ABBM23] Marcelo Arenas, Pablo Barceló, Leopoldo E. Bertossi, and Mikaël Monet.  
**On the complexity of SHAP-score-based explanations: Tractability via knowledge compilation and non-approximability results.**  
*J. Mach. Learn. Res.*, 24:63:1–63:58, 2023.
- [ABOS22] Marcelo Arenas, Pablo Barceló, Miguel A. Romero Orth, and Bernardo Subercaseaux.  
**On computing probabilistic explanations for decision trees.**  
In NeurIPS, 2022.
- [ACD24] Leila Amgoud, Martin C. Cooper, and Salim Debbaoui.  
**Axiomatic characterisations of sample-based explainers.**  
In ECAI, pages 770–777, 2024.
- [ACL03] Nicolas-Gabriel Andjiga, Frédéric Chantreuil, and Dominique Lepelley.  
**La mesure du pouvoir de vote.**  
*Mathématiques et sciences humaines. Mathematics and social sciences*, (163), 2003.

## References ii

- [Alp14] Ethem Alpaydin.  
*Introduction to machine learning.*  
MIT press, 2014.
- [Alp16] Ethem Alpaydin.  
*Machine Learning: The New AI.*  
MIT Press, 2016.
- [Amg23] Leila Amgoud.  
**Explaining black-box classifiers: Properties and functions.**  
*Int. J. Approx. Reason.*, 155:40–65, 2023.
- [BA97] Leonard A. Breslow and David W. Aha.  
**Simplifying decision trees: A survey.**  
*Knowledge Eng. Review*, 12(1):1–40, 1997.
- [BAMT21] Ryma Boumazouza, Fahima Cheikh Alili, Bertrand Mazure, and Karim Tabia.  
**ASTERYX: A model-agnostic sat-based approach for symbolic and score-based explanations.**  
In *CIKM*, pages 120–129, 2021.

## References iii

- [BBHK10] Michael R. Berthold, Christian Borgelt, Frank Höppner, and Frank Klawonn.  
*Guide to Intelligent Data Analysis - How to Intelligently Make Sense of Real Data*, volume 42 of *Texts in Computer Science*.  
Springer, 2010.
- [BBM<sup>+</sup>15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek.  
**On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation.**  
*PloS one*, 10(7):e0130140, 2015.
- [BFOS84] Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone.  
*Classification and Regression Trees*.  
Wadsworth, 1984.
- [BHO09] Christian Bessiere, Emmanuel Hebrard, and Barry O'Sullivan.  
**Minimising decision tree size as combinatorial optimisation.**  
In *CP*, pages 173–187, 2009.
- [BHvMW09] Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors.  
*Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009.

## References iv

- [Bil65] John F Banzhaf III.  
**Weighted voting doesn't work: A mathematical analysis.**  
*Rutgers L. Rev.*, 19:317, 1965.
- [BIL<sup>+</sup>24] Gagan Biradar, Yacine Izza, Elita Lobo, Vignesh Viswanathan, and Yair Zick.  
**Axiomatic aggregations of abductive explanations.**  
In *AAAI*, pages 11096–11104, 2024.
- [BMB<sup>+</sup>23] Christopher Brix, Mark Niklas Müller, Stanley Bak, Taylor T. Johnson, and Changliu Liu.  
**First three years of the international verification of neural networks competition (VNN-COMP).**  
*Int. J. Softw. Tools Technol. Transf.*, 25(3):329–339, 2023.
- [Bra20] Max Bramer.  
**Principles of Data Mining, 4th Edition.**  
Undergraduate Topics in Computer Science. Springer, 2020.
- [CA23] Martin C. Cooper and Leila Amgoud.  
**Abductive explanations of classifiers under constraints: Complexity and properties.**  
In *ECAI*, pages 469–476, 2023.

## References v

- [CdPA<sup>+</sup>19] Roberto Confalonieri, Fermín Moscoso del Prado, Sebastia Agramunt, Daniel Malagarriga, Daniele Faggion, Tillman Weyde, and Tarek R. Besold.  
**An ontology-based approach to explaining artificial neural networks.**  
*CoRR*, abs/1906.08362, 2019.
- [CG16] Tianqi Chen and Carlos Guestrin.  
**XGBoost: A scalable tree boosting system.**  
In *KDD*, pages 785–794, 2016.
- [CGT09] Javier Castro, Daniel Gómez, and Juan Tejada.  
**Polynomial calculation of the shapley value based on sampling.**  
*Comput. Oper. Res.*, 36(5):1726–1730, 2009.
- [CH04] Hana Chockler and Joseph Y Halpern.  
**Responsibility and blame: A structural-model approach.**  
*Journal of Artificial Intelligence Research*, 22:93–115, 2004.
- [CM21] Martin C. Cooper and Joao Marques-Silva.  
**On the tractability of explaining decisions of classifiers.**  
In *CP*, October 2021.

- [Col71] James S Coleman.  
**Control of collectivities and the power of a collectivity to act.**  
In Bernhardt Lieberman, editor, *Social choice*, chapter 2.10. Gordon and Breach, New York, 1971.
- [CWBdPM21] Roberto Confalonieri, Tillman Weyde, Tarek R. Besold, and Fermín Moscoso del Prado Martín.  
**Using ontologies to enhance human understandability of global post-hoc explanations of black-box models.**  
*Artif. Intell.*, 296:103471, 2021.
- [DL01] Sašo Džeroski and Nada Lavrač, editors.  
**Relational data mining.**  
Springer, 2001.
- [DP78] John Deegan and Edward W Packel.  
**A new index of power for simple  $n$ -person games.**  
*International Journal of Game Theory*, 7:113–123, 1978.
- [DSZ16] Anupam Datta, Shayak Sen, and Yair Zick.  
**Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems.**  
In *IEEE S&P*, pages 598–617, 2016.

## References vii

- [dud01] *Pattern classification.*  
John Wiley & Sons, 2001.
- [EG95] Thomas Eiter and Georg Gottlob.  
**Identifying the minimal transversals of a hypergraph and related problems.**  
*SIAM J. Comput.*, 24(6):1278–1304, 1995.
- [EU21a] EU.  
**European Artificial Intelligence Act.**  
<https://eur-lex.europa.eu/eli/reg/2024/1689/oj>, 2021.
- [EU21b] EU.  
**European Artificial Intelligence Act – Proposal.**  
<https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1623335154975&uri=CELEX%3A52021PC0206>, 2021.
- [FJ18] Matteo Fischetti and Jason Jo.  
**Deep neural networks and mixed integer linear optimization.**  
*Constraints*, 23(3):296–309, 2018.
- [FK96] Michael L. Fredman and Leonid Khachiyan.  
**On the complexity of dualization of monotone disjunctive normal forms.**  
*J. Algorithms*, 21(3):618–628, 1996.

## References viii

- [Fla12] Peter A. Flach.  
*Machine Learning - The Art and Science of Algorithms that Make Sense of Data.*  
Cambridge University Press, 2012.
- [GR22] Niku Gorji and Sasha Rubin.  
**Sufficient reasons for classifier decisions in the presence of domain constraints.**  
In AAAI, February 2022.
- [GZM20] Mohammad M. Ghiasi, Sohrab Zendehboudi, and Ali Asghar Mohsenipour.  
**Decision tree-based diagnosis of coronary artery disease: CART model.**  
*Comput. Methods Programs Biomed.*, 192:105400, 2020.
- [HCM<sup>+</sup>23] Xuanxiang Huang, Martin C. Cooper, António Morgado, Jordi Planes, and João Marques-Silva.  
**Feature necessity & relevancy in ML classifier explanations.**  
In TACAS, pages 167–186, 2023.
- [HII<sup>+</sup>22] Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, Martin Cooper, Nicholas Asher, and Joao Marques-Silva.  
**Tractable explanations for d-DNNF classifiers.**  
In AAAI, February 2022.

# References ix

- [HIIM21] Xuanxiang Huang, Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva.  
**On efficiently explaining graph-based classifiers.**  
In *KR*, November 2021.  
Preprint available from <https://arxiv.org/abs/2106.01350>.
- [HM23a] Xuanxiang Huang and João Marques-Silva.  
**From decision trees to explained decision sets.**  
In *ECAI*, pages 1100–1108, 2023.
- [HM23b] Xuanxiang Huang and João Marques-Silva.  
**From robustness to explainability and back again.**  
*CoRR*, abs/2306.03048, 2023.
- [HM23c] Xuanxiang Huang and João Marques-Silva.  
**The inadequacy of Shapley values for explainability.**  
*CoRR*, abs/2302.08160, 2023.
- [HM23d] Xuanxiang Huang and Joao Marques-Silva.  
**A refutation of shapley values for explainability.**  
*CoRR*, abs/2309.03041, 2023.

## References x

- [HM23e] Xuanxiang Huang and Joao Marques-Silva.  
**Refutation of shapley values for XAI – additional evidence.**  
*CoRR*, abs/2310.00416, 2023.
- [HM23f] Aurélie Hurault and João Marques-Silva.  
**Certified logic-based explainable AI - the case of monotonic classifiers.**  
In *TAP*, pages 51–67, 2023.
- [HMS24] Xuanxiang Huang and Joao Marques-Silva.  
**On the failings of Shapley values for explainability.**  
*International Journal of Approximate Reasoning*, page 109112, 2024.
- [HP83] Manfred J Holler and Edward W Packel.  
**Power, luck and the right index.**  
*Journal of Economics*, 43(1):21–29, 1983.
- [HRS19] Xiyang Hu, Cynthia Rudin, and Margo Seltzer.  
**Optimal sparse decision trees.**  
In *NeurIPS*, pages 7265–7273, 2019.
- [Ign20] Alexey Ignatiev.  
**Towards trustable explainable AI.**  
In *IJCAI*, pages 5154–5158, 2020.

- [IHI<sup>+</sup>22] Yacine Izza, Xuanxiang Huang, Alexey Ignatiev, Nina Narodytska, Martin C. Cooper, and João Marques-Silva.  
**On computing probabilistic abductive explanations.**  
*CoRR*, abs/2212.05990, 2022.
- [IHI<sup>+</sup>23] Yacine Izza, Xuanxiang Huang, Alexey Ignatiev, Nina Narodytska, Martin C. Cooper, and João Marques-Silva.  
**On computing probabilistic abductive explanations.**  
*Int. J. Approx. Reason.*, 159:108939, 2023.
- [IHM<sup>+</sup>24a] Yacine Izza, Xuanxiang Huang, Antonio Morgado, Jordi Planes, Alexey Ignatiev, and Joao Marques-Silva.  
**Distance-restricted explanations: Theoretical underpinnings & efficient implementation.**  
*CoRR*, abs/2405.08297, 2024.
- [IHM<sup>+</sup>24b] Yacine Izza, Xuanxiang Huang, Antonio Morgado, Jordi Planes, Alexey Ignatiev, and Joao Marques-Silva.  
**Distance-restricted explanations: Theoretical underpinnings & efficient implementation.**  
In *KR*, 2024.
- [IIM20] Yacine Izza, Alexey Ignatiev, and Joao Marques-Silva.  
**On explaining decision trees.**  
*CoRR*, abs/2010.11034, 2020.

## References xii

- [IIM22] Yacine Izza, Alexey Ignatiev, and João Marques-Silva.  
**On tackling explanation redundancy in decision trees.**  
*J. Artif. Intell. Res.*, 75:261–321, 2022.
- [IIN<sup>+</sup>22] Yacine Izza, Alexey Ignatiev, Nina Narodytska, Martin C. Cooper, and João Marques-Silva.  
**Provably precise, succinct and efficient explanations for decision trees.**  
*CoRR*, abs/2205.09569, 2022.
- [IISM24] Yacine Izza, Alexey Ignatiev, Peter J. Stuckey, and João Marques-Silva.  
**Delivering inflated explanations.**  
In *AAAI*, pages 12744–12753, 2024.
- [IISMS22] Alexey Ignatiev, Yacine Izza, Peter J. Stuckey, and Joao Marques-Silva.  
**Using MaxSAT for efficient explanations of tree ensembles.**  
In *AAAI*, February 2022.
- [IM21] Alexey Ignatiev and Joao Marques-Silva.  
**SAT-based rigorous explanations for decision lists.**  
In *SAT*, pages 251–269, July 2021.
- [IMM18] Alexey Ignatiev, António Morgado, and João Marques-Silva.  
**PySAT: A python toolkit for prototyping with SAT oracles.**  
In *SAT*, pages 428–437, 2018.

## References xiii

- [IMM24] Yacine Izza, Kuldeep Meel, and João Marques-Silva.  
**Locally-minimal probabilistic explanations.**  
In *ECAI*, 2024.
- [IMS21] Yacine Izza and Joao Marques-Silva.  
**On explaining random forests with SAT.**  
In *IJCAI*, pages 2584–2591, July 2021.
- [INAM20] Alexey Ignatiev, Nina Narodytska, Nicholas Asher, and João Marques-Silva.  
**From contrastive to abductive explanations and back again.**  
In *AlxIA*, pages 335–355, 2020.
- [INM19a] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.  
**Abduction-based explanations for machine learning models.**  
In *AAAI*, pages 1511–1519, 2019.
- [INM19b] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.  
**On relating explanations and adversarial examples.**  
In *NeurIPS*, pages 15857–15867, 2019.
- [INM19c] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva.  
**On validating, repairing and refining heuristic ML explanations.**  
*CoRR*, abs/1907.02509, 2019.

# References xiv

- [JKMC16] Mikolás Janota, William Klieber, Joao Marques-Silva, and Edmund M. Clarke.  
**Solving QBF with counterexample guided refinement.**  
*Artif. Intell.*, 234:1–25, 2016.
- [Joh78] Ronald John Johnston.  
**On the measurement of power: Some reactions to Laver.**  
*Environment and Planning A*, 10(8):907–914, 1978.
- [KBD<sup>+</sup>17] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer.  
**Reluplex: An efficient SMT solver for verifying deep neural networks.**  
In CAV, pages 97–117, 2017.
- [KHI<sup>+</sup>19] Guy Katz, Derek A. Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljic, David L. Dill, Mykel J. Kochenderfer, and Clark W. Barrett.  
**The marabou framework for verification and analysis of deep neural networks.**  
In CAV, pages 443–452, 2019.
- [KMND20] John D Kelleher, Brian Mac Namee, and Aoife D'arcy.  
**Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies.**  
MIT Press, 2020.

## References xv

- [Kot13] Sotiris B. Kotsiantis.  
**Decision trees: a recent overview.**  
*Artif. Intell. Rev.*, 39(4):261–283, 2013.
- [LC01] Stan Lipovetsky and Michael Conklin.  
**Analysis of regression in game theory approach.**  
*Applied Stochastic Models in Business and Industry*, 17(4):319–330, 2001.
- [LHAMS24] Olivier Létoffé, Xuanxiang Huang, Nicholas Asher, and Joao Marques-Silva.  
**From SHAP scores to feature importance scores.**  
*CoRR*, abs/2405.11766, 2024.
- [LHM24] Olivier Letoffe, Xuanxiang Huang, and João Marques-Silva.  
**SHAP scores fail pervasively even when Lipschitz succeeds.**  
Under review, July 2024.
- [LHMS24] Olivier Létoffé, Xuanxiang Huang, and Joao Marques-Silva.  
**On correcting SHAP scores.**  
*CoRR*, abs/2405.00076, 2024.
- [Lip18] Zachary C. Lipton.  
**The mythos of model interpretability.**  
*Commun. ACM*, 61(10):36–43, 2018.

## References xvi

- [LL17] Scott M. Lundberg and Su-In Lee.  
**A unified approach to interpreting model predictions.**  
In *NIPS*, pages 4765–4774, 2017.
- [LPMM16] Mark H. Liffiton, Alessandro Previti, Ammar Malik, and Joao Marques-Silva.  
**Fast, flexible MUS enumeration.**  
*Constraints*, 21(2):223–250, 2016.
- [LS08] Mark H. Liffiton and Karem A. Sakallah.  
**Algorithms for computing minimal unsatisfiable subsets of constraints.**  
*J. Autom. Reasoning*, 40(1):1–33, 2008.
- [Mar22] João Marques-Silva.  
**Logic-based explainability in machine learning.**  
In *Reasoning Web*, pages 24–104, 2022.
- [Mar24] João Marques-Silva.  
**Logic-based explainability: Past, present & future.**  
*CoRR*, abs/2406.11873, 2024.
- [MGC<sup>+</sup>20] João Marques-Silva, Thomas Gerspacher, Martin C. Cooper, Alexey Ignatiev, and Nina Narodytska.  
**Explaining naive bayes and other linear classifiers with polynomial time and delay.**  
In *NeurIPS*, 2020.

## References xvii

- [MGC<sup>+</sup>21] Joao Marques-Silva, Thomas Gerspacher, Martinc C. Cooper, Alexey Ignatiev, and Nina Narodytska.  
**Explanations for monotonic classifiers.**  
In *ICML*, pages 7469–7479, July 2021.
- [MH23] Joao Marques-Silva and Xuanxiang Huang.  
**Explainability is NOT a game.**  
*CoRR*, abs/2307.07514, 2023.
- [MHL<sup>+</sup>13] António Morgado, Federico Heras, Mark H. Liffiton, Jordi Planes, and Joao Marques-Silva.  
**Iterative and core-guided MaxSA solving: A survey and assessment.**  
*Constraints*, 18(4):478–534, 2013.
- [MI22] João Marques-Silva and Alexey Ignatiev.  
**Delivering trustworthy AI through formal XAI.**  
In *AAAI*, pages 12342–12350, 2022.
- [Mil56] George A Miller.  
**The magical number seven, plus or minus two: Some limits on our capacity for processing information.**  
*Psychological review*, 63(2):81–97, 1956.
- [Mil19] Tim Miller.  
**Explanation in artificial intelligence: Insights from the social sciences.**  
*Artif. Intell.*, 267:1–38, 2019.

## References xviii

- [MM20] João Marques-Silva and Carlos Mencía.  
**Reasoning about inconsistent formulas.**  
In *IJCAI*, pages 4899–4906, 2020.
- [Mol20] Christoph Molnar.  
***Interpretable machine learning.***  
Lulu.com, 2020.  
<https://christophm.github.io/interpretable-ml-book/>.
- [Mor82] Bernard M. E. Moret.  
**Decision trees and diagrams.**  
*ACM Comput. Surv.*, 14(4):593–623, 1982.
- [MS23] Joao Marques-Silva.  
**Disproving XAI myths with formal methods – initial results.**  
In *ICECCS*, 2023.
- [MSH24] Joao Marques-Silva and Xuanxiang Huang.  
**Explainability is *Not* a game.**  
*Commun. ACM*, 67(7):66–75, jul 2024.

## References xix

- [MSI23] Joao Marques-Silva and Alexey Ignatiev.  
**No silver bullet: interpretable ml models must be explained.**  
*Frontiers in Artificial Intelligence*, 6, 2023.
- [MSLLM25] Joao Marques-Silva, Jairo Lefebre-Lobaina, and Vanina Martinez.  
**Efficient and rigorous model-agnostic explanations.**  
In *IJCAI*, 2025.  
In press.
- [NH10] Vinod Nair and Geoffrey E. Hinton.  
**Rectified linear units improve restricted boltzmann machines.**  
In *ICML*, pages 807–814, 2010.
- [NSM<sup>+</sup>19] Nina Narodytska, Aditya A. Shrotri, Kuldeep S. Meel, Alexey Ignatiev, and Joao Marques-Silva.  
**Assessing heuristic machine learning explanations with model counting.**  
In *SAT*, pages 267–278, 2019.
- [Pen46] Lionel S Penrose.  
**The elementary statistics of majority voting.**  
*Journal of the Royal Statistical Society*, 109(1):53–57, 1946.

## References xx

- [PG86] David A. Plaisted and Steven Greenbaum.  
*A structure-preserving clause form translation.*  
*J. Symb. Comput.*, 2(3):293–304, 1986.
- [PM17] David Poole and Alan K. Mackworth.  
*Artificial Intelligence - Foundations of Computational Agents.*  
CUP, 2017.
- [Qui93] J Ross Quinlan.  
*C4.5: programs for machine learning.*  
Morgan-Kaufmann, 1993.
- [RCC<sup>+</sup>22] Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong.  
**Interpretable machine learning: Fundamental principles and 10 grand challenges.**  
*Statistics Surveys*, 16:1–85, 2022.
- [Rei87] Raymond Reiter.  
**A theory of diagnosis from first principles.**  
*Artif. Intell.*, 32(1):57–95, 1987.
- [RM08] Lior Rokach and Oded Z Maimon.  
*Data mining with decision trees: theory and applications.*  
World scientific, 2008.

## References xxi

- [RN10] Stuart J. Russell and Peter Norvig.  
*Artificial Intelligence - A Modern Approach.*  
Pearson Education, 2010.
- [RSG16] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin.  
"why should I trust you?": Explaining the predictions of any classifier.  
In KDD, pages 1135–1144, 2016.
- [RSG18] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin.  
Anchors: High-precision model-agnostic explanations.  
In AAAI, pages 1527–1535. AAAI Press, 2018.
- [Rud19] Cynthia Rudin.  
Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.  
*Nature Machine Intelligence*, 1(5):206–215, 2019.
- [Rud22] Cynthia Rudin.  
Why black box machine learning should be avoided for high-stakes decisions, in brief.  
*Nature Reviews Methods Primers*, 2(1):1–2, 2022.

## References xxii

- [SB14] Shai Shalev-Shwartz and Shai Ben-David.  
*Understanding Machine Learning - From Theory to Algorithms.*  
Cambridge University Press, 2014.
- [SCD18] Andy Shih, Arthur Choi, and Adnan Darwiche.  
**A symbolic approach to explaining bayesian network classifiers.**  
In *IJCAI*, pages 5103–5111, 2018.
- [Sha53] Lloyd S. Shapley.  
**A value for  $n$ -person games.**  
*Contributions to the Theory of Games*, 2(28):307–317, 1953.
- [SK10] Erik Strumbelj and Igor Kononenko.  
**An efficient explanation of individual classifications using game theory.**  
*J. Mach. Learn. Res.*, 11:1–18, 2010.
- [SK14] Erik Strumbelj and Igor Kononenko.  
**Explaining prediction models and individual predictions with feature contributions.**  
*Knowl. Inf. Syst.*, 41(3):647–665, 2014.
- [SS54] Lloyd S Shapley and Martin Shubik.  
**A method for evaluating the distribution of power in a committee system.**  
*American political science review*, 48(3):787–792, 1954.

## References xxiii

- [Tse68] G.S. Tseitin.  
**On the complexity of derivations in the propositional calculus.**  
In H.A.O. Slesenko, editor, *Structures in Constructive Mathematics and Mathematical Logic, Part II*, pages 115–125, 1968.
- [VLE+16] Gilmer Valdes, José Marcio Luna, Eric Eaton, Charles B Simone, Lyle H Ungar, and Timothy D Solberg.  
**MediBoost: a patient stratification tool for interpretable decision making in the era of precision medicine.**  
*Scientific reports*, 6(1):1–8, 2016.
- [VLSS21] Guy Van den Broeck, Anton Lykov, Maximilian Schleich, and Dan Suciu.  
**On the tractability of SHAP explanations.**  
In AAAI, pages 6505–6513, 2021.
- [VLSS22] Guy Van den Broeck, Anton Lykov, Maximilian Schleich, and Dan Suciu.  
**On the tractability of SHAP explanations.**  
*J. Artif. Intell. Res.*, 74:851–886, 2022.
- [WFHP17] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal.  
**Data Mining.**  
Morgan Kaufmann, 2017.

## References xxiv

- [WMHK21] Stephan Wäldchen, Jan MacDonald, Sascha Hauch, and Gitta Kutyniok.  
**The computational complexity of understanding binary classifier decisions.**  
*J. Artif. Intell. Res.*, 70:351–387, 2021.
- [WMZ10] William Webber, Alistair Moffat, and Justin Zobel.  
**A similarity measure for indefinite rankings.**  
*ACM Trans. Inf. Syst.*, 28(4):20:1–20:38, 2010.
- [WWB23] Min Wu, Haoze Wu, and Clark W. Barrett.  
**VeriX: Towards verified explainability of deep neural networks.**  
In *NeurIPS*, 2023.
- [YIS<sup>+</sup>23] Jinqiang Yu, Alexey Ignatiev, Peter J. Stuckey, Nina Narodytska, and Joao Marques-Silva.  
**Eliminating the impossible, whatever remains must be true: On extracting and applying background knowledge in the context of formal explanations.**  
In *AAAI*, 2023.
- [Zho12] Zhi-Hua Zhou.  
***Ensemble methods: foundations and algorithms.***  
CRC press, 2012.

## References xxv

- [Zho21] Zhi-Hua Zhou.  
*Machine Learning*.  
Springer, 2021.