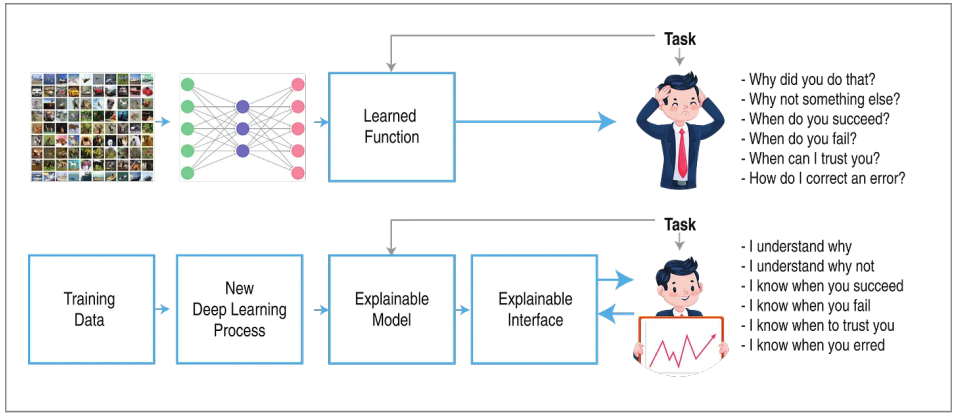
PyXAI: Computing Explanations for Tree-Based Classifiers

Gilles Audemard
(P. Marquis, JM. Lagniez, N. Szczepanski)

Nagoya University - April 2026

eXplainable Artificial Intelligence



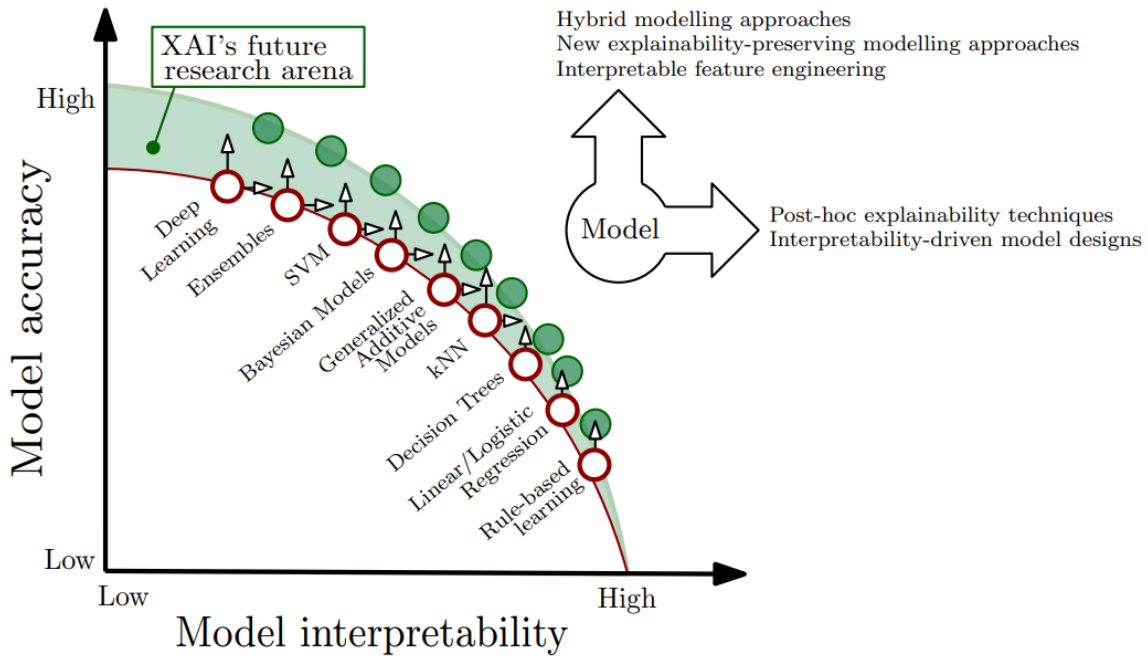
- Machine Learning models are commonly used. They are present in plenty of domains.
- The opacity of the most accurate ML models.
- XAI: make ML models more transparent and more trustable.

DARPA (Defense Advanced Research Projects Agency)

"to provide users with explanations that allow them to understand the forces and the overall weaknesses of the system in question, which allow them **to understand how it will behave in the future**, *or* **even to correct the system's errors**"

Problem 1: Interpretability

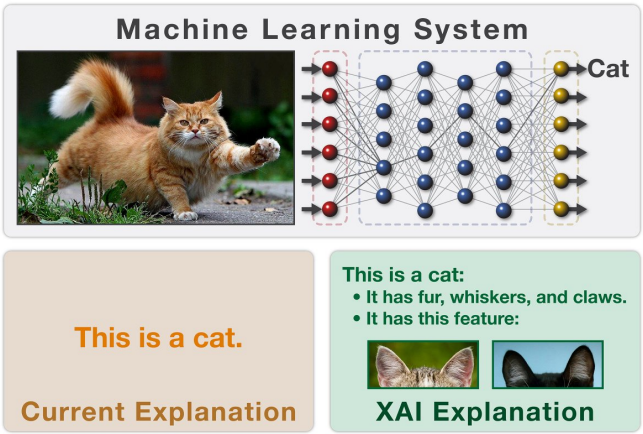
The more accurate, the less interpretable



Problem 2: What is an explanation?

There is no general definition of what is a good explanation

- Explaining is basically a multi-faceted reasoning activity.
- Explaining is a social process, a model of the explainee (the concepts he/she knows, the beliefs he/she has, etc.) must be taken into account.
  - An explanation for a doctor my differ than an explanation for the patient.
  - An explanation acceptable for me can be unacceptable for you.
- Explanations can have different types.



"Useful" explanation depend on the explainee (Humain-in-the-loop!)

What is a ”Good Classifier”?

- Several evaluation criteria can be considered to assess the quality of a classifier.

- Accuracy:
  - The probability of correctly labelling any instance x.
  - Minimizing the classification error.
  - In practice, estimated on the test set.

What is a ”Good Classifier”?

- Several evaluation criteria can be considered to assess the quality of a classifier.

- Accuracy:
  - The probability of correctly labelling any instance x.
  - Minimizing the classification error.
  - In practice, estimated on the test set.

- Intelligibility
  - Of the utmost importance when dealing with critical systems.
  - No agreement on the meaning of ”*intelligible*”.
  - Informally, the capacity to derive useful information from the predictor.
  - ”Useful” depends on the explainee.

Many XAI Queries can be Defined

- One can divide them into two categories.

- Explanation queries: related to instances. They are local explanations.
  - Enumerating Minimum-Cardinality explanations (EMC).
  - Deriving one Prime Implicant explanation (DPI).
  - Enumerating COunterfactual explanations (ECO).
  - ...

- Verification queries: global interpretability issue. They are independent of any instance.
  - Counting the INstances associated with a given class (CIN).
  - Enumerating the INstances associated with a given class (EIN).
  - Identifying MAndatory features or forbidden features in a given class (IMA).
  - Identifying IRrelevant features in a given class (IIR).
  - Identifying MOnotone (or anti-monotone) features in a given class (IMO).
  - ...

Computational intelligibility is the subset of XAI queries that are tractable*.

(*) Tractable: can be computed in polynomial time (delay)

Decision Trees are Computationally intelligible Models

- Because an abductive explanation (the direct reason) can be associated with each prediction made, that explains it somehow (local interpretability).
- Because each path of the tree can be read as a decision rule (global interpretability).

- But there exists other reasons.

Decision Trees are Computationally intelligible Models

- Because an abductive explanation (the direct reason) can be associated with each prediction made, that explains it somehow (local interpretability).
- Because each path of the tree can be read as a decision rule (global interpretability).
- But there exists other reasons.
- For each enumeration problem among EMC, ECO, EIN, there exists an enumeration algorithm **with polynomial delay** when the Boolean classifier is a Decision Tree.
- Furthermore, each problem among DPI, CIN, IMA, IIR, IMO, MCP is **in** P when the Boolean classifier is a Decision Tree.

Enumerating Minimum-Cardinality explanations (EMC).
Deriving one Prime Implicant explanation (DPI).
Enumerating COunterfactual explanations (ECO).
Counting the INstances associated with a given class (CIN).
Enumerating the INstances associated with a given class (EIN).
Identifying MAndatory features or forbidden features in a given class (IMA).
Identifying IRrelevant features in a given class (IIR).

Identifying MOnotone (or anti-monotone) features in a given class (IMO).

What about Other ML Models?

- They appear as **far less computationally intelligible** than Decision Trees!
- Each problem among EMC, DPI, ECO, CIN, EIN, IMA, IIR, IMO, MCP is NP-**hard** in the broad sense when the Boolean classifier is
  - a DNF formula
  - a decision list
  - a random forest
  - a boosted tree
  - a Boolean multilayer perceptron
  - a binarized neural network
- Somewhat surprising: at first sight, decision lists look closer to decision trees than to multilayer perceptrons, but that is not the case from the perspective of computational intelligibility

The price to pay to have a better accuracy is to be less intelligible

Our Contribution

PyXAI

1. The family of ML models that are handled: **tree-based** models
   - Decision trees (DT), random forests (RF), boosted trees (BT) for classification and regression
   - DT are intelligible by design and computationally interpretable
   - RF, BT are neither interpretable by design nor computationally interpretable
   - RF, BT are more accurate than DT in general
   - BT are state-of-the-art models when dealing with tabular datasets

2. The nature of explanations that are generated: **local, post-hoc**, consideration of **user preferences**

3. The other functionalities that are offered:
   - A **correction** method
   - **Visualization** facilities

4. The audience that is targeted: users with some domain knowledge (not ML specialists)

Abductive Explanations

Why this instance has been classified like that?

- Extract a subset of information of the input such that any instance that covers this subset is classified in the same way.
- Different types of abductive explanations:
  - Direct reason.
  - Sufficient reasons.
  - Approximation of sufficient reasons.
- Different sizes of explanations.
- But different complexities (depending on the model used).

Contrastive Explanations

Why this instance has not been classified differently?

- I am applying for a loan.
- The loan is not granted. Why?
- What can I change to get the loan granted?

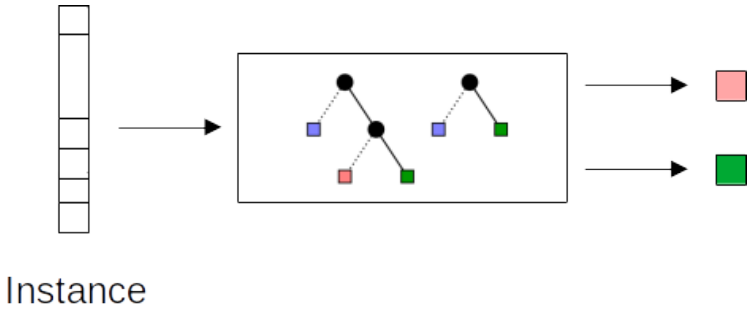
- Try to extract a most general explanation.
- Try to compute an explanation that is as small as possible.

Agnostic Approaches

- Many popular XAI approaches are agnostic ones.
- The predictor is a black box. No knowledge about it.
- Examples: LIME [Ribeiro et al., ACM SIGKDD'16], Anchor [Ribeiro et al., AAAI'18]...
- Useful when dealing with images (CNN for example)
- Local explanations are computed heuristically.
- Those approaches are scalable but they ensure no guarantee w.r.t. the underlying ML model [Garreau and von Luxburg, AISTATS'20] [Narodytska et al., SAT'19].
- Especially, two instances associated with distinct classes can share the same explanation! [Ignatiev, IJCAI'20].
- Cannot be used with critical systems.

Model-Based Approaches

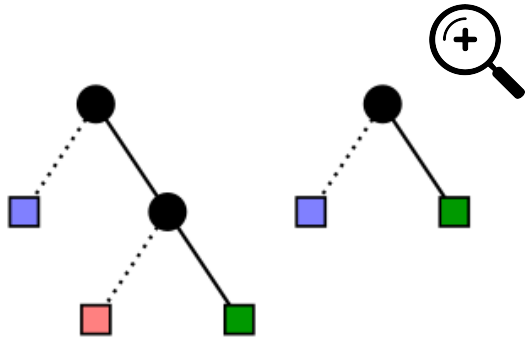
Model-Based Approaches



- Model Specific: associating with the predictor a circuit that has the same input-output behavior.
- Explanation and verification tasks about the predictor can be addressed by considering the circuit.
- Machine Learning classifiers based on trees: Decision Tree, Random Forest, Boosted Tree.
- Leveraging formal methods and automated reasoning techniques for XAI purposes.
- Guarantees are offered, but scalability can be an obstacle.

Formal XAI: one provides theoretical guarantees about the explanations

Inside Trees

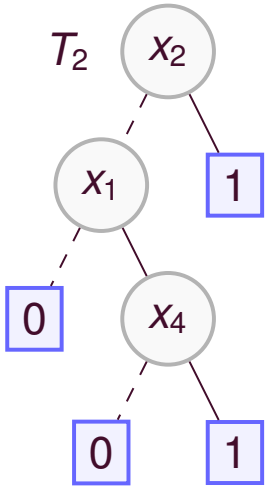


- Each node has the form $f \leq v$ where f is a feature and v a threshold.
- Depending on the value of the feature f in the instance, one goes to the left (if condition is false) or to the right (if condition is true).
- One considers such conditions as Boolean variables.

Explanations are based on such conditions

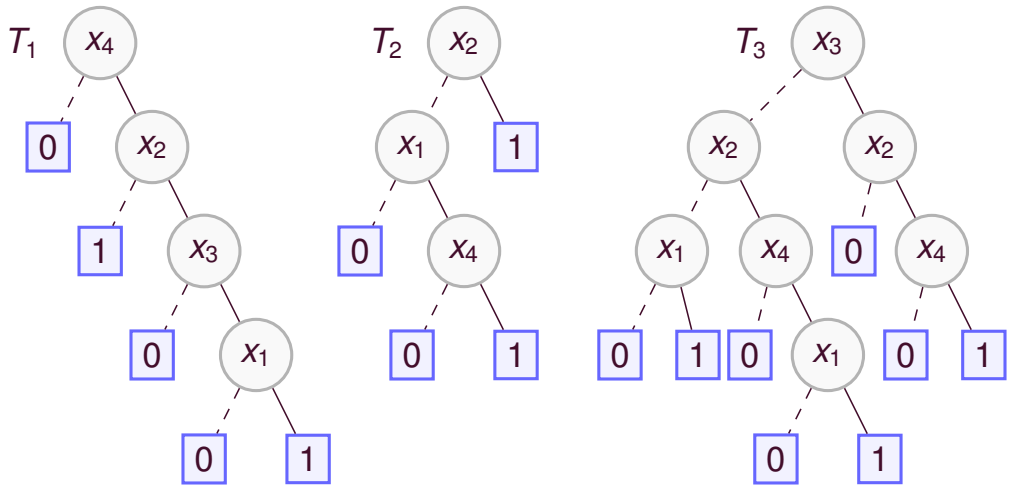
- We can consider nodes as boolean variables. They abstract some conditions over features.
- We can consider datasets with only boolean features.

Decision Trees



- Given an instance, one follows the path from the root to leave.
- Go right if the attribute is positive, left otherwise.
- The leaf gives the classification. The followed path corresponds to direct reason.
- Examples:
  - The instance $\mathbf{x} = (1, 1, 0, 0)$ is classified positively (direct reason is x_2).
  - The instance $\mathbf{x} = (1, 0, 0, 0)$ is classified negatively (direct reason is $x_1 \wedge \neg x_2 \wedge \neg x_4$).
  - The instance $\mathbf{x} = (1, 0, 0, 1)$ is classified positively (direct reason is $x_1, \wedge \neg x_2 \wedge x_4$).

Random Forests



- The classification is given by the majority of trees.
- Some examples:
  - The instance $\mathbf{x} = (1, 1, 0, 0)$ is classified negatively. (0 for T_1, 1 for T_2, 0 for T_3).
  - The instance $\mathbf{x} = (1, 0, 0, 0)$ is classified negatively. (0 for T_1, 0 for T_2, 1 for T_3).
  - The instance $\mathbf{x} = (1, 0, 0, 1)$ is classified positively. (1 for T_1, 1 for T_2, 1 for T_3).

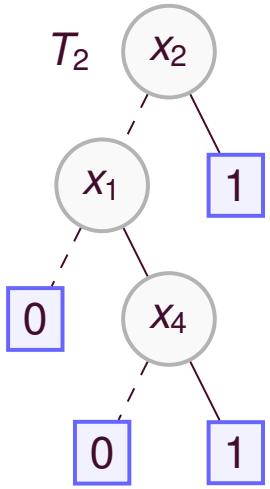
SAT Solving in Practice

- Before 2000, reduce a problem to SAT to show that it is intractable.
- After 2000, reduce a problem to SAT to solve it efficiently.
- SAT is a success story in computer science.
- Many practical applications.

Source J. Marques-Silva

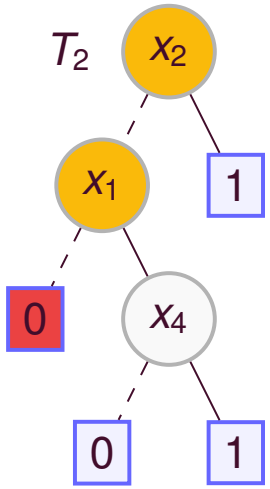
Use SAT Technology to find Explanations

From Trees to CNF



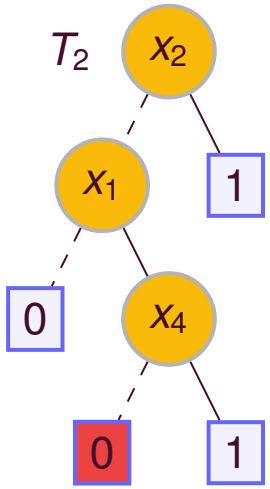
$$\texttt{CNF}(T_2) =$$

From Trees to CNF



$$\texttt{CNF}(T_2) = (x_2 \vee x_1) \wedge$$

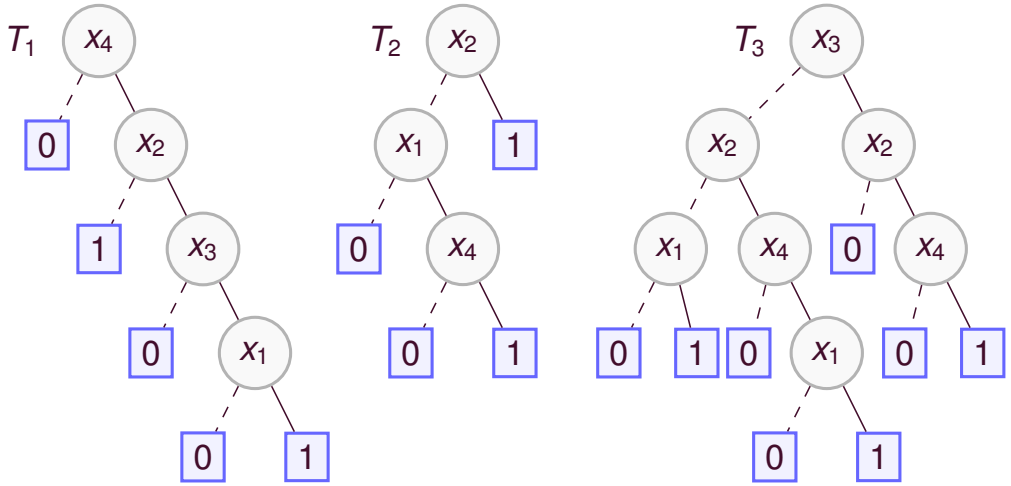
From Trees to CNF



$$\mathrm{CNF}(T_2) = (x_2 \vee x_1) \wedge (x_2 \vee \neg x_1 \vee x_4)$$

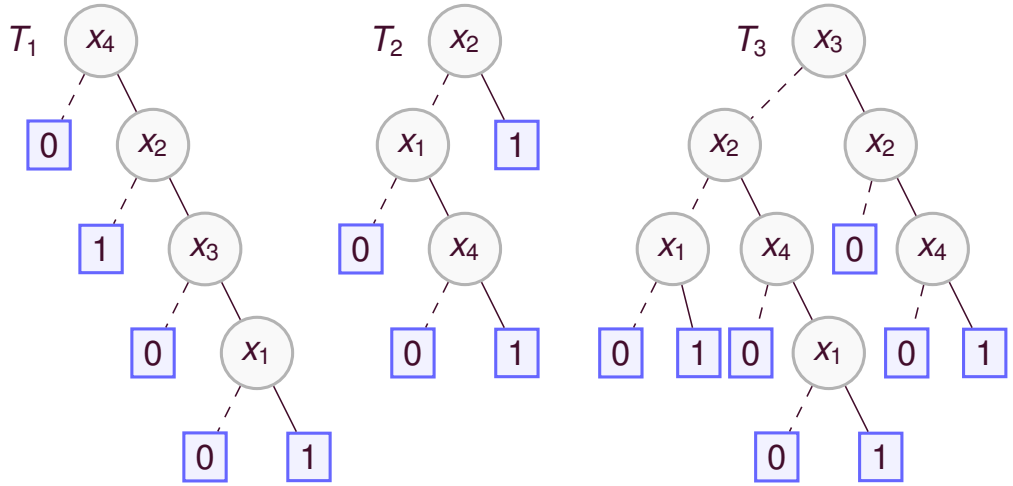
- An instance is classified positively if it is a model of the CNF.
- $\mathrm{CNF}(T2)$ is true for the instance $x = (1, 1, 0, 0)$.
- $\mathrm{CNF}(T2)$ is false for the instance $x = (1, 0, 0, 0)$ (second clause is false).
- $\mathrm{CNF}(T2)$ is true for the instance $x = (1, 0, 0, 1)$.

From Random Forests to CNF



- m trees.
- One fresh variable y_i per tree.
- $\neg y_i \vee CNF(T_i)$. If y_i is true then the tree T_i votes positively.
- $\Sigma_{i=1}^{m} y_i > \frac{m}{2}$
- An instance is classified positively if it is a model of the CNF.

From Random Forests to CNF



$\neg y_1 \vee x_4$
$\neg y_1 \vee \neg x_4 \vee \neg x_2 \vee x_3$
$\neg y_1 \vee \neg x_4 \vee \neg x_2 \vee \neg x_3 \vee x_1$

$\neg y_2 \vee x_2 \vee x_1$
$\neg y_2 \vee x_2 \vee \neg x_1 \vee x_4$

$\neg y_3 \vee x_3 \vee x_2 \vee x_1$
$\neg y_3 \vee x_3 \vee \neg x_2 \vee x_4$
$\neg y_3 \vee x_3 \vee \neg x_2 \vee \neg x_4 \vee x_1$
$\neg y_3 \vee \neg x_3 \vee x_2$
$\neg y_3 \vee \neg x_3 \vee \neg x_2 \vee x_4$

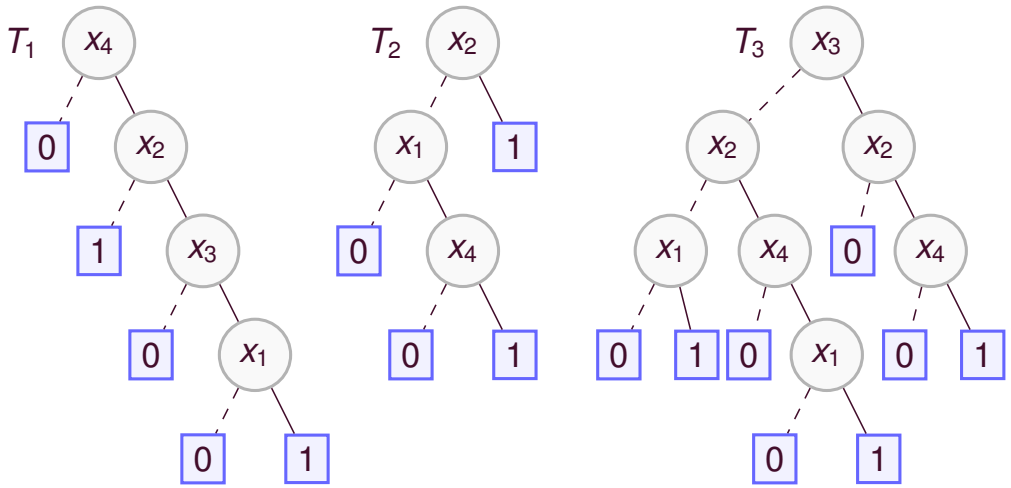
$y_1 \vee y_2$
$y_1 \vee y_3$
$y_2 \vee y_3$

- $\texttt{CNF}(F)$ is true for the instance $x = (1, 0, 0, 1)$.
- $\texttt{CNF}(F)$ is false for the instance $x = (1, 0, 0, 0)$.
- $\texttt{CNF}(F)$ is false for the instance $x = (1, 1, 0, 0)$.

A Focus on Random Forest

Direct Reason

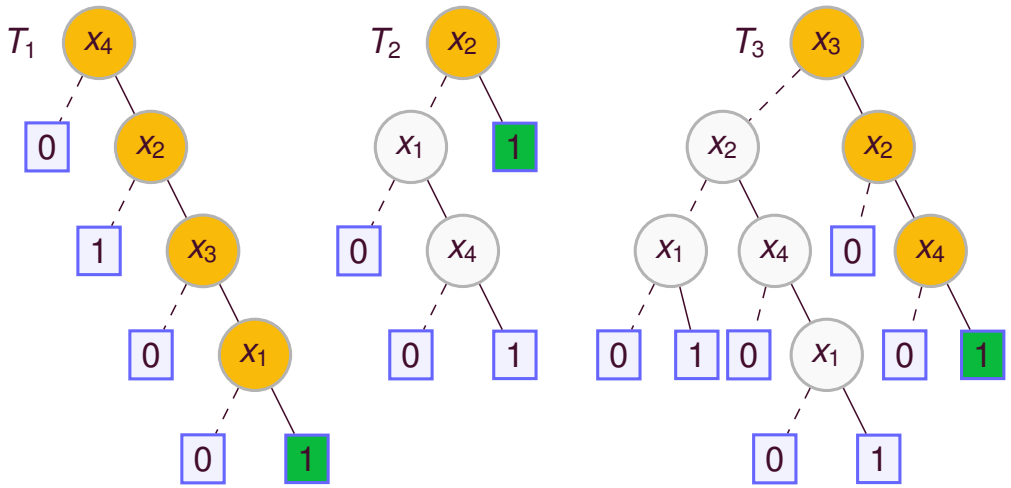
The **direct reason** for **x** corresponds to the conjunction of direct reasons for x given all trees in the forest F.



- The direct reason for $\mathbf{x} = (1, 1, 1, 1)$ is :

Direct Reason

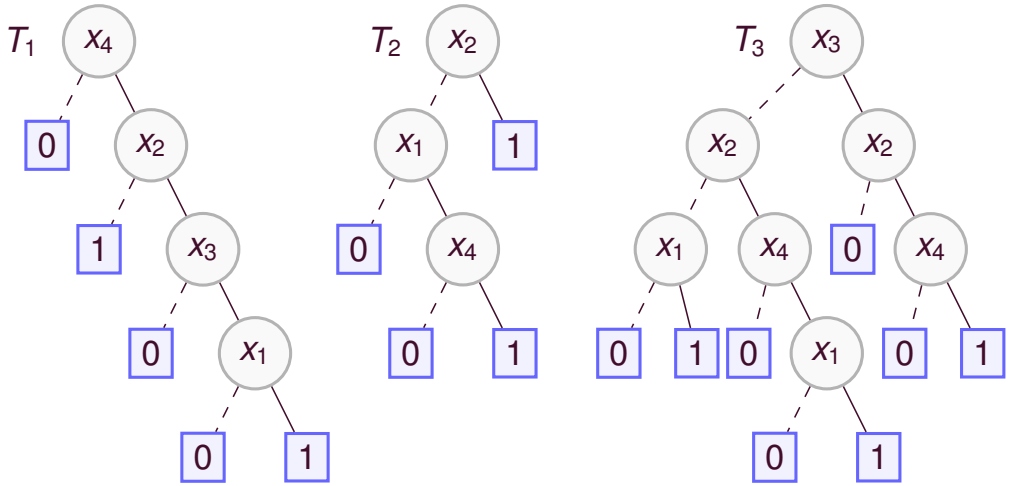
The **direct reason** for **x** corresponds to the conjunction of direct reasons for x given all trees in the forest F.



- The direct reason for $\mathbf{x} = (1, 1, 1, 1)$ is : $x_1 \wedge x_2 \wedge x_3 \wedge x_4$
- For a given instance, the direct reason is unique.
- Computing the direct reason is feasible in polynomial time.

Sufficient Reasons

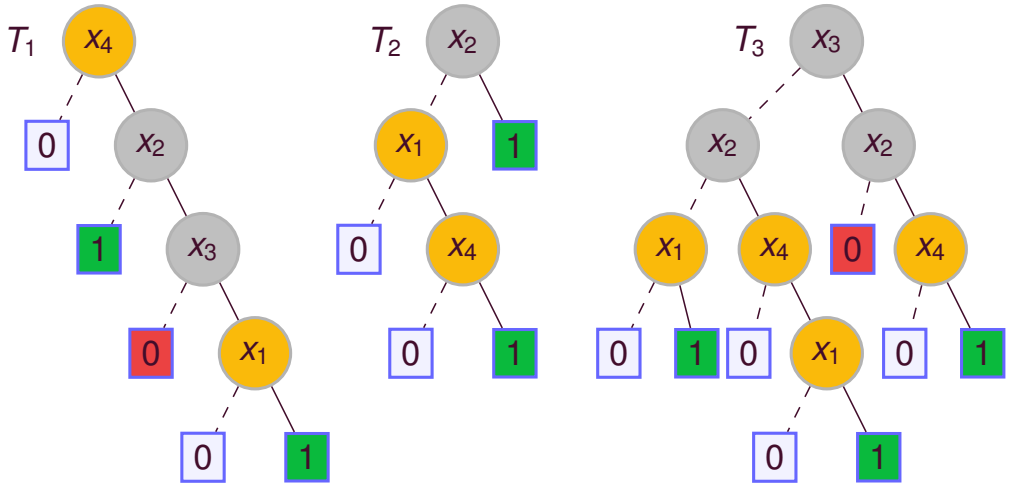
A **sufficient reason** for **x** given F is a subset-minimal subset $\mathbf{x}'$ of **x** such that any instance $\mathbf{x}''$ sharing the same characteristics as $\mathbf{x}'$ is classified in the same way.



- A sufficient reason for **x** is :

Sufficient Reasons

A **sufficient reason** for **x** given F is a subset-minimal subset $\mathbf{x}'$ of **x** such that any instance $\mathbf{x}''$ sharing the same characteristics as $\mathbf{x}'$ is classified in the same way.

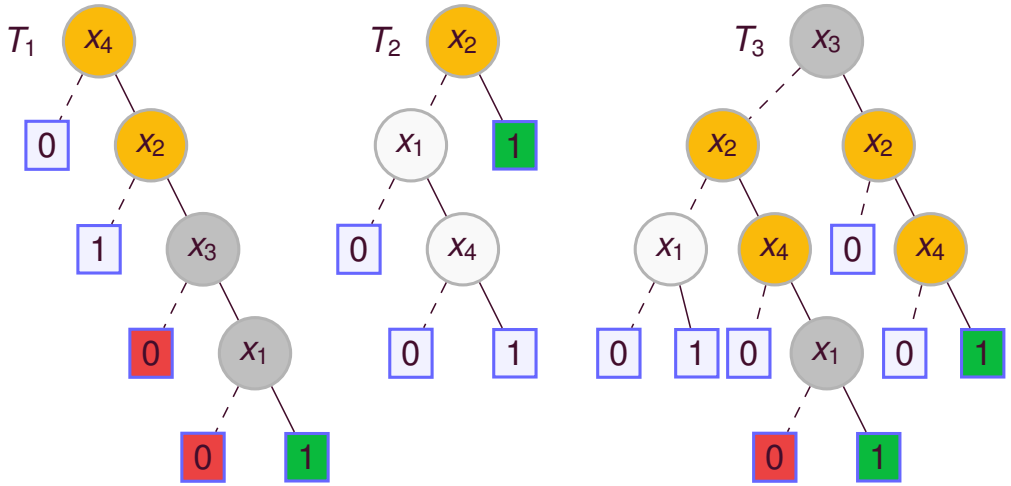


- A sufficient reason for $\mathbf{x} = (1, 1, 1, 1)$ is : $\mathbf{x}_1 \wedge \mathbf{x}_4$

x_1	x_2	x_3	x_4	T_1	T_2	T_3	F
1	**0**	**0**	1	1	1	1	1
1	**0**	**1**	1	1	1	0	1
1	**1**	**0**	1	0	1	1	1
1	**1**	**1**	1	1	1	1	1

Sufficient Reasons

A **sufficient reason** for **x** given F is a subset-minimal subset $\mathbf{x}'$ of **x** such that any instance $\mathbf{x}''$ sharing the same characteristics as $\mathbf{x}'$ is classified in the same way.



- A sufficient reason for $\mathbf{x} = (1, 1, 1, 1)$ is **NOT** $\mathbf{x}_2 \wedge \mathbf{x}_4$.

x_1	x_2	x_3	x_4	T_1	T_2	T_3	F
0	1	**0**	1	0	1	0	0
0	1	**1**	1	0	1	1	1
1	1	**0**	1	0	1	1	1
1	1	**1**	1	1	1	1	1

Computing Sufficient Reasons

- Identifying Sufficient Reasons is DP-Complete [Izza and Marques-Silva, IJCAI'21]

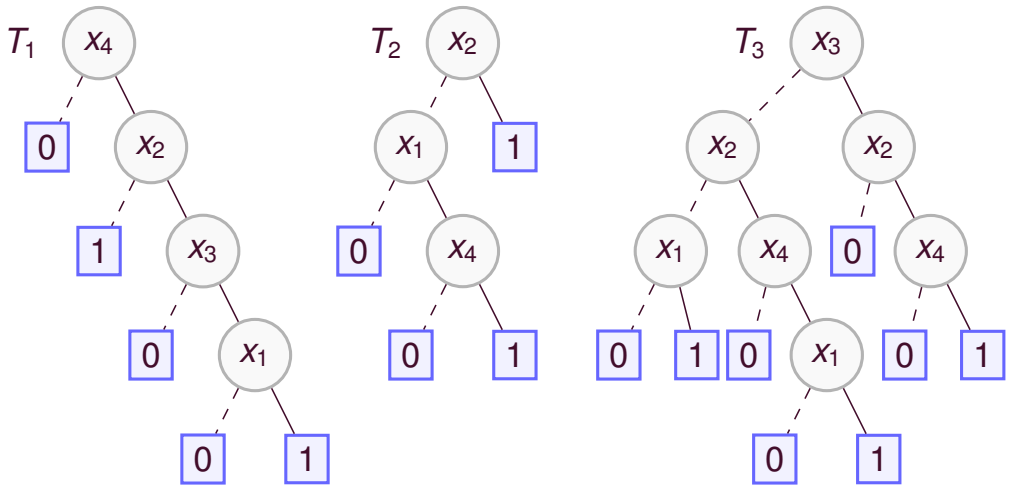
- One fresh variable y_i per tree
- $\neg y_i \vee CNF(\neg T_i)$. If y_i is true then the tree votes for the class 0.
- $\Sigma_{i=1}^{m} y_i > \frac{m}{2}$
- An instance is classified negatively if it is a model of the $CNF(\neg F)$.
- Thus, $x \wedge CNF(\neg F)$ is unsatisfiable. (x is a positive instance).
- x' is an implicant of F if $x' \wedge CNF(\neg F)$ is unsatisfiable. (x is a positive instance).
- x' is a prime implicant of F if $x' \wedge CNF(\neg F)$ is a Minimum Unsatisfiable Sub-formula (MUS).

- Use a MUS solver (Muser for example) to compute sufficient reasons.

Scalability can be an issue

Majoritary Reasons

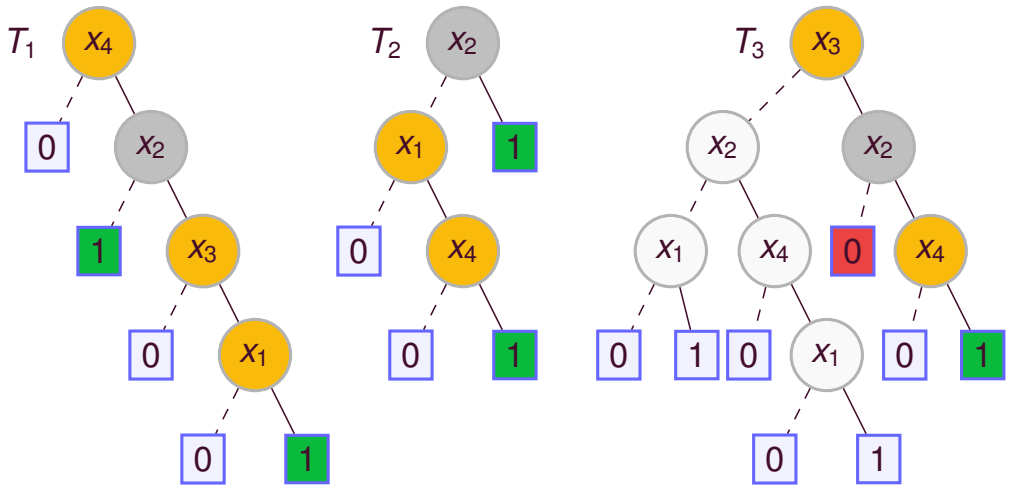
A **majoritary reason** for **x** given F is a term t covering **x**, such that t is an implicant of at least $\lfloor \frac{m}{2} \rfloor + 1$ decision trees, and for every $l \in t$, $t \setminus \{l\}$ does not satisfy this last condition



- Trees are considered independently.
- A majoritary reason for $x = (1, 1, 1, 1)$ is :

Majoritary Reasons

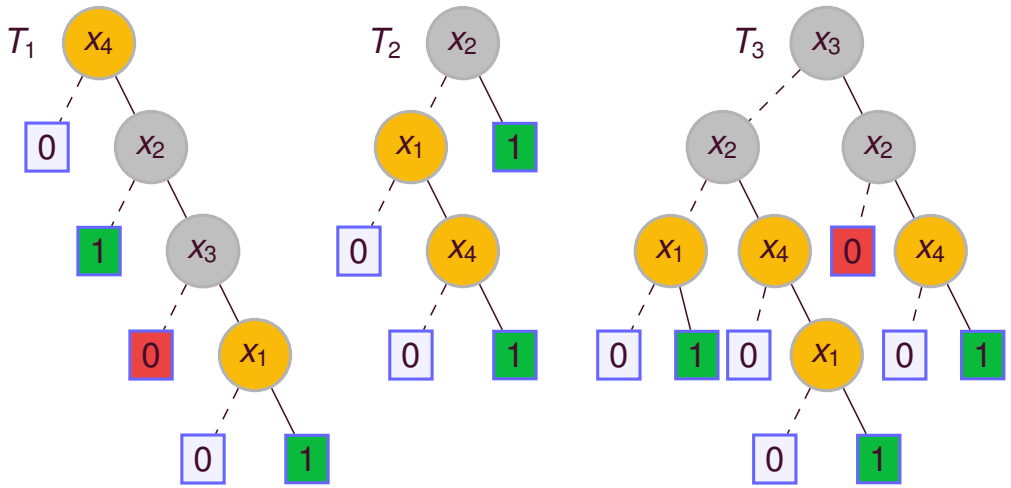
A **majoritary reason** for **x** given F is a term t covering **x**, such that t is an implicant of at least $\lfloor \frac{m}{2} \rfloor + 1$ decision trees, and for every $l \in t$, $t \setminus \{l\}$ does not satisfy this last condition



- Trees are considered independently.
- A majoritary reason for $x = (1, 1, 1, 1)$ is : $x_1 \wedge x_3 \wedge x_4$.

Majoritary Reasons

A **majoritary reason** for **x** given F is a term t covering **x**, such that t is an implicant of at least $\lfloor \frac{m}{2} \rfloor + 1$ decision trees, and for every $l \in t$, $t \setminus \{l\}$ does not satisfy this last condition



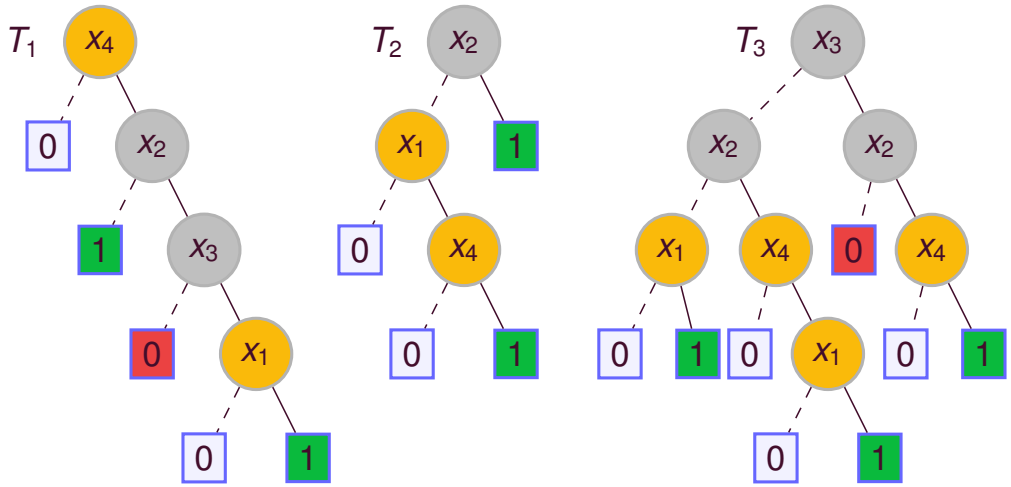
- Trees are considered independently.
- The sufficient reason $x_1 \wedge x_4$ for $x = (1, 1, 1, 1)$ is **NOT** a majoritary reason. Indeed, T_1 and T_3 can vote for 0.

Computing Majoritary Reasons

- One removes one attribute of the instance after the other in a greedy way.
- One computes the classification for each tree. If one reachable leaf is 0, the vote for this tree is 0.
- If the majoritary of trees votes for 1, one can definitely remove the attribute, otherwise, it is kept.
- Of course, order matters. Different orders usually produce different majoritary reasons.
- Since this algorithm runs in polynomial time, one can apply it several times using different orders and return the smallest majoritary reason found.

Majoritary Reasons VS Sufficient Reasons

- majoritary reasons and sufficient reasons do not coincide in general.
- In majoritary reasons, trees are considered independently.
- Some votes of different trees can be mutually incompatible. It is the case for example for the class 0 for the first tree where x_3 is false and the vote 0 for the last tree where x_3 is true.



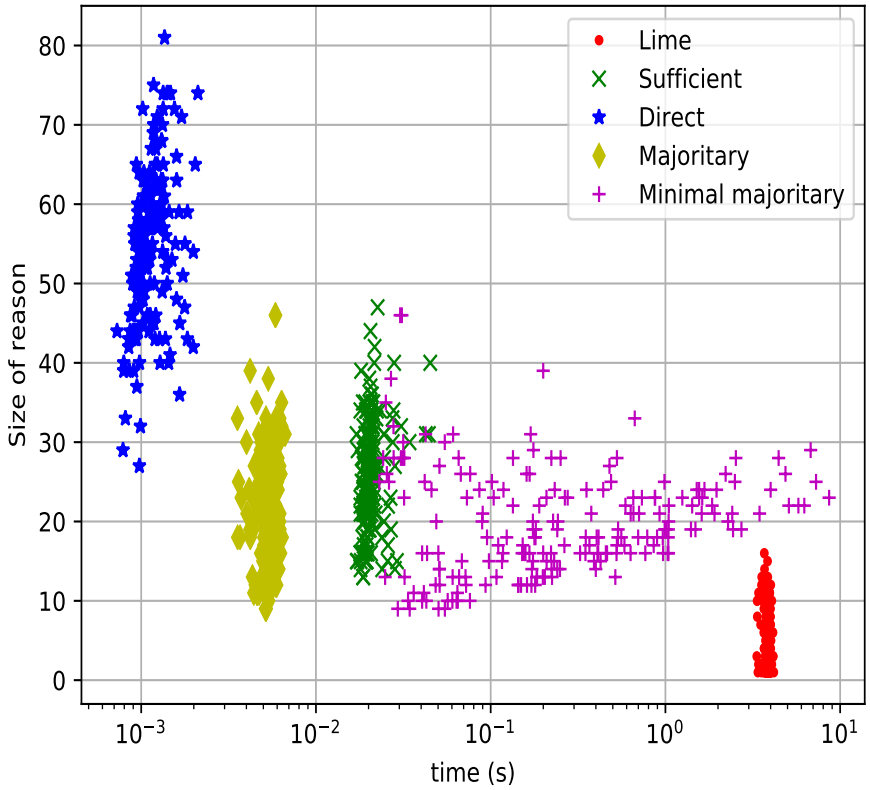
- majoritary reasons may contain irrelevant characteristics.
- From a majoritary reason, a sufficient reason can be extracted.
- One can compute a majoritary reason as a preprocessing step to compute a sufficient reason.

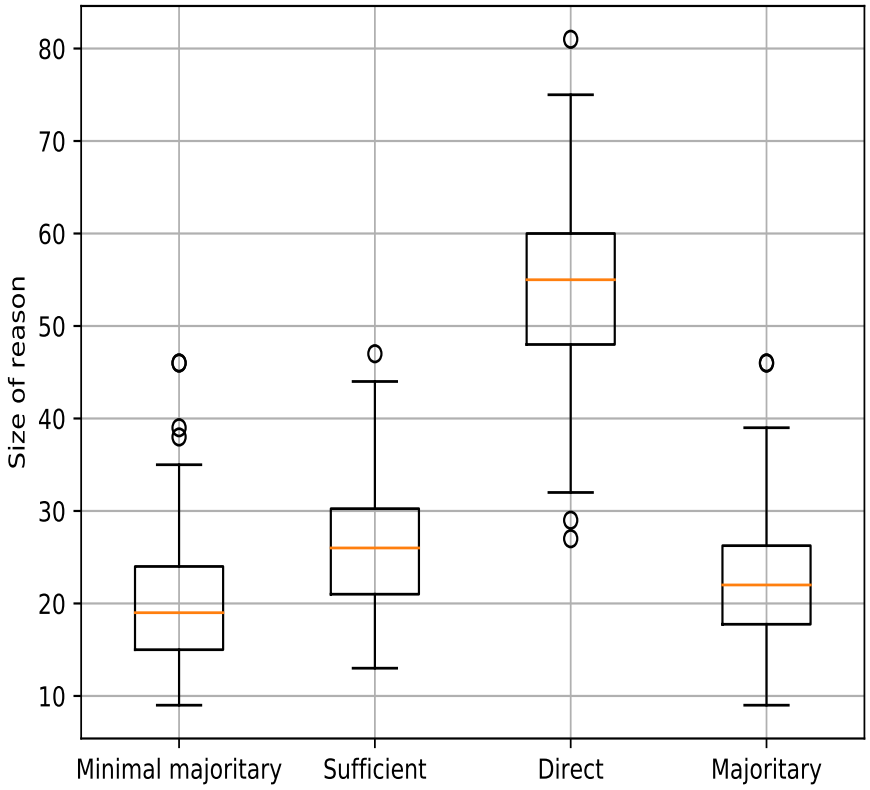
Computing Minimum-Size Majoritary Reasons

- As for contrastive explanations, one can use a MaxSAT solver to extract minimum-size majoritary reasons.
- One can also give weights to soft clauses (the instance) in order to keep preferred features (Weighted Partial MaxSAT).
- MaxSAT solvers are anytime solvers. They can provide best solution found when they are stopped.
- Interestingly, adding preferences reduce drastically the number of majoritary reasons.

Some Experiments: dataset *placement*

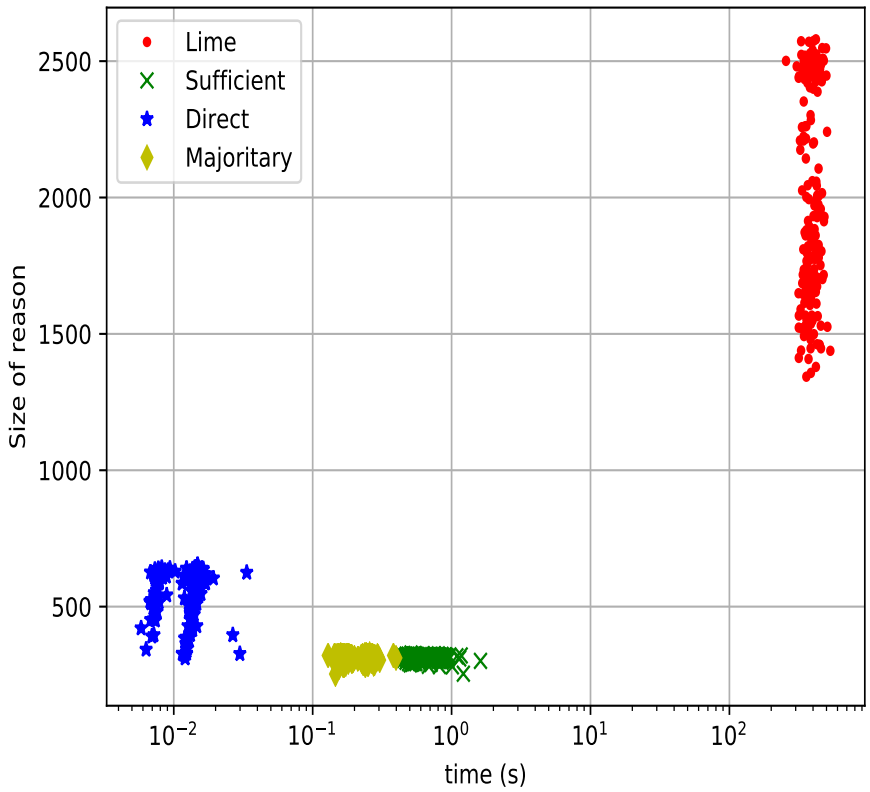
- 13 features, 230 Boolean attributes in average.
- Left plot: each dots represent an instance. x-axis represents the time needed to compute an explanation, y-axis, the size of the explanation.

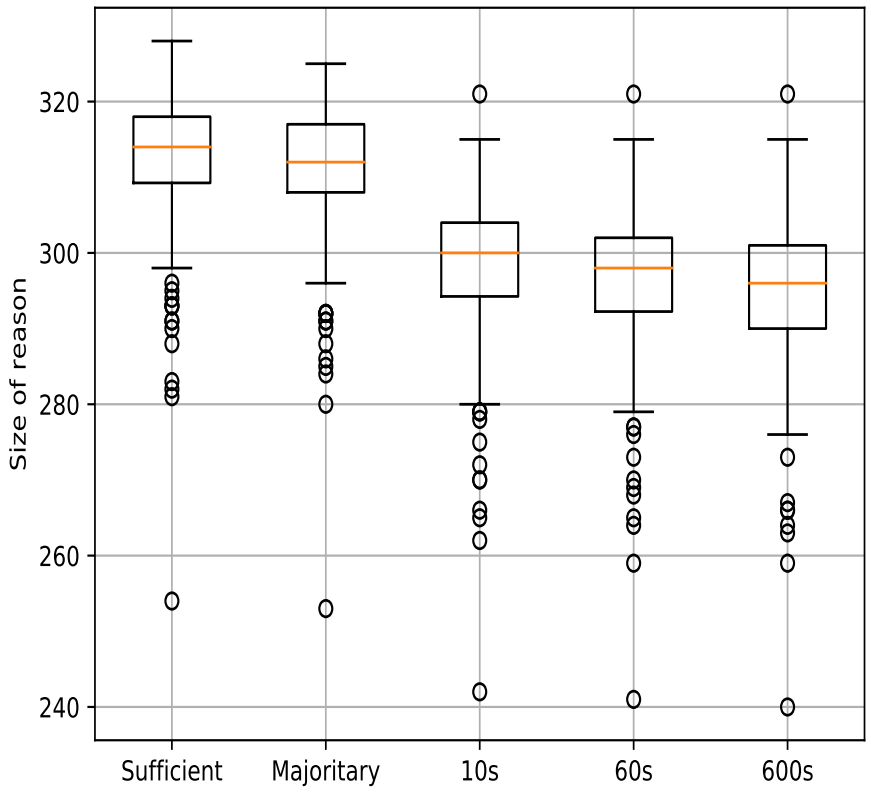




Some Experiments: dataset *gisette*

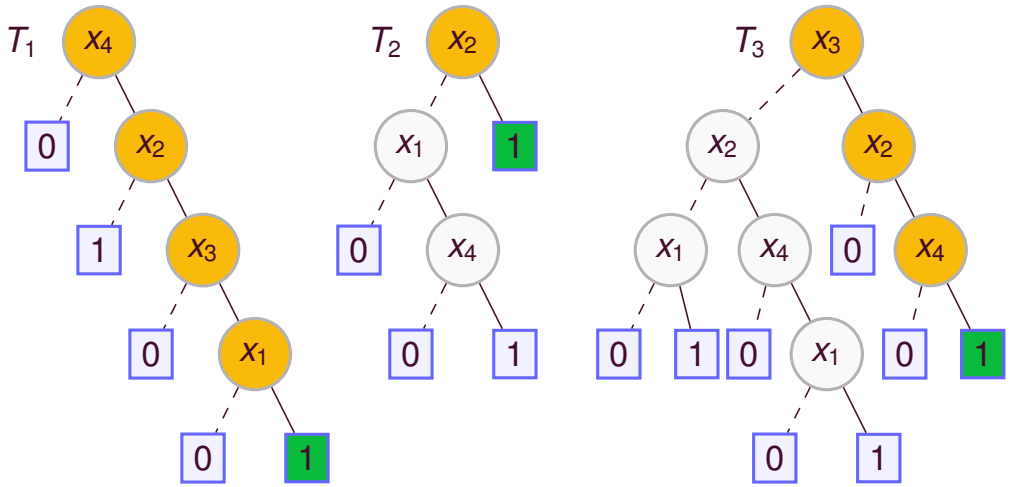
- 5000 features, 8000 Boolean attributes in average.
- Computing minimal Majoritary Reasons is an anytime algorithm: One can have an approximation.





Contrastive Explanations

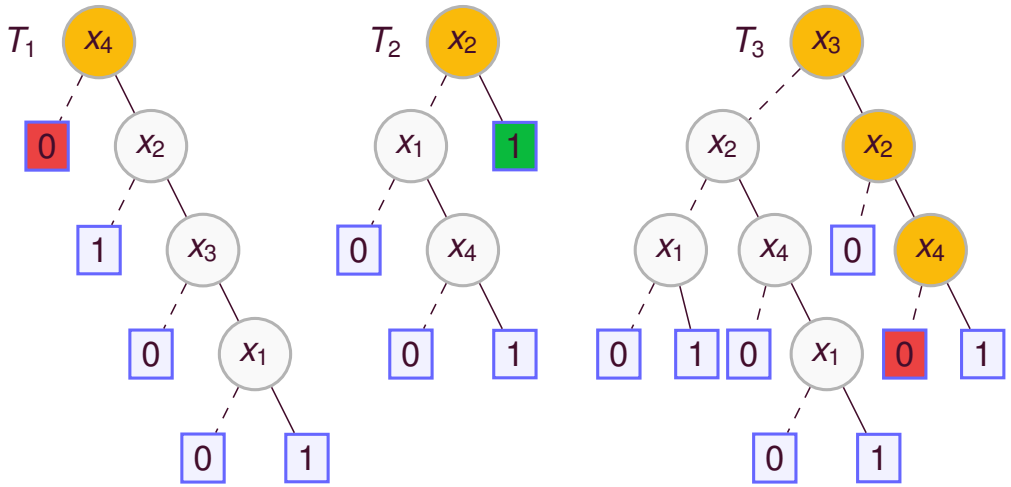
Minimal changes to be achieved to an instance x for changing classification.



- A constrastive explanation for $x = (1, 1, 1, 1)$ given F is

Contrastive Explanations

Minimal changes to be achieved to an instance x for changing classification.



- A constrastive explanation for $x = (1, 1, 1, 1)$ given F is x_4. (T_1 and T_3 vote for 0). The instance $(1, 1, 1, 0)$ is classified negatively.

Computing a Contrastive Explanation

- Identifying contrastive explanations is NP-Complete.

- One fresh variable y_i per tree
- $\neg y_i \vee CNF(\neg T_i)$. If y_i is true then the tree votes for the class 0.
- $\Sigma_{i=1}^{m} y_i > \frac{m}{2}$
- An instance is classified negatively if it is a model of the $CNF(\neg F)$.
- Thus, $x \wedge CNF(\neg F)$ is unsatisfiable (x is a positive instance).
- Partial MaxSAT problem:
  - Hard clauses are $CNF(\neg F)$.
  - Soft clauses are x.
  - Extract a maximum subset of soft clauses (called x') such that the resulting problem ($CNF(\neg F) \wedge x'$) is satisfiable.
  - x' is the longer instance classified negatively.
- Then, $x \setminus x'$ is a minimum contrastive explanation.

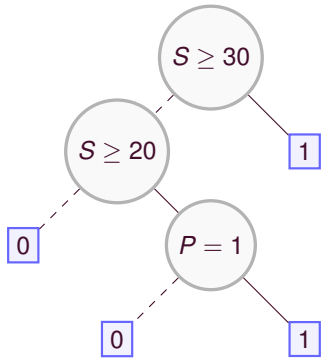
Contrastive Explanations and Domain Theory

- Boolean attributes are not independent from each other.
- It is the case for attributes representing numerical features:
  $x_{10} = (f_1 \leq 10)$ and $x_{15} = (f_1 \leq 15)$.
- It is the case for one-hot encoded attributes (for categorical features):
  $x_a = (f_2 = a)$ and $x_b = (f_2 = b)$.
- Exhibiting a contrastive explanation where x_{10} is true and x_{15} is false must be prohibited.
- Exhibiting a contrastive explanation where x_a and x_b is true must be prohibited.

Contrastive Explanations and Domain Theory

- In the same spirit, some definitions of contrastive explanation rely on the minimum number of features to be changed.
- This is relevant only if all features are Boolean.
- Let us consider the following loan approval scenario.

- S is the annual income of the applicant.
- P is a Boolean attribute that indicates whether or not the applicant has already reimbursed a previous loan.



- Suppose an instance $x = (18, 0)$. the loan is not granted.
- The applicant wants to know the minimal changes to be achieved in order to get the loan.
- "*Change your annual income*" or "*Change your annual income to 35k$*" are not satisfactory.
- "*Change your annual income to at least 30k$*" or "*Change your annual income to at least 20k$ and reimburse your previous loan*" are better explanations.

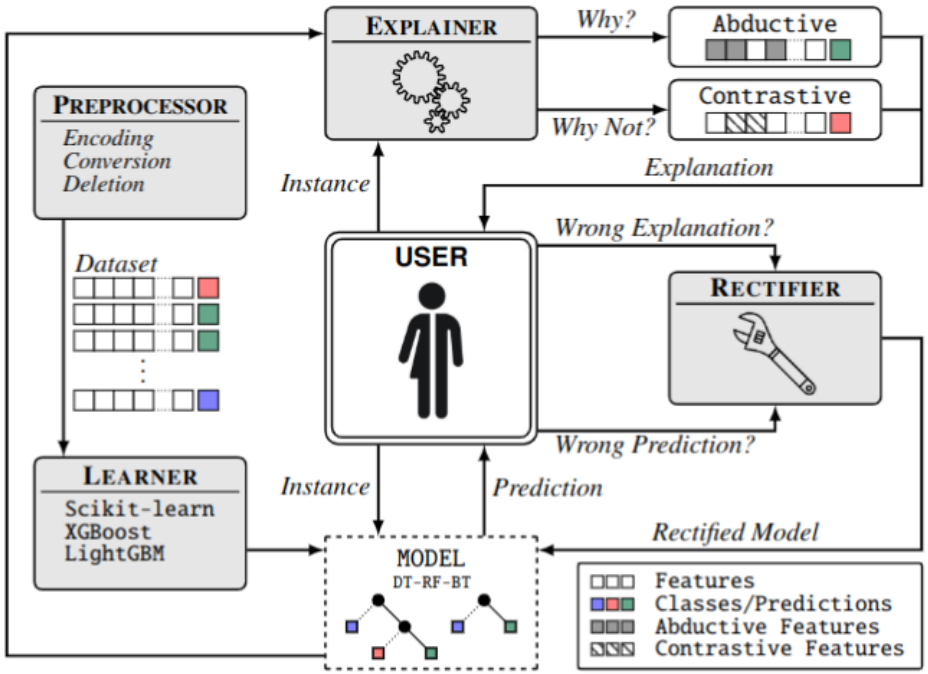
User Preferences

- An exponential number of explanations for x may exist.
- Providing only one explanation is not necessarily enough.
- Different types of preferences.
- Excluding explanations containing some features.
  - non actionable features (the age of a person).
  - unintelligible features.
- Extracting minimum-size explanations (smaller is better).
- Extracting explanations with respect to a preference order of features.

Human-In-The-Loop approach

PyXAI:
A Python Library for Computing Explanations

A General Overview of PyXAI



- **Learner/Preprocessor**:
  - Prepare the dataset (remove useless features, deal with unknown values...)
  - Help the user to deal with ML libraries
  - Extract specific instances
  - Save/load models
- **Explainer**:
  - Compute abductive/contrastive explanations
- **Rectifier**:
  - Modify the model with respect to user knowledge

Rectification: How to Correct the Model when it goes Wrong?

- The prediction made and the explanations found may **contradict** the user's knowledge
- PyXAI provides a tool for **correcting** the model in that case
- Each prediction for an instance that matches the classification rules given by the user is corrected accordingly
- The predictions associated with the other instances are unchanged

A demonstration

Conclusion

- PyXAI: a library to compute explanations for tree-based models
- User in the loop approach
- Already tested
  - electricity price forecasting
  - Care in retirement homes
  - Anomaly in hydroelectric generator

https://www.cril.univ-artois.fr/pyxai

- Implement new algorithms for user preferences (example based, coverage based).
- New version is coming soon (with improved documentation (API)).
- ...