

# One Tree to Explain Them All: Importance-Aware Rule Models for Black-Box Systems

Szymon Bobek

RuleML Webinar, November 2025



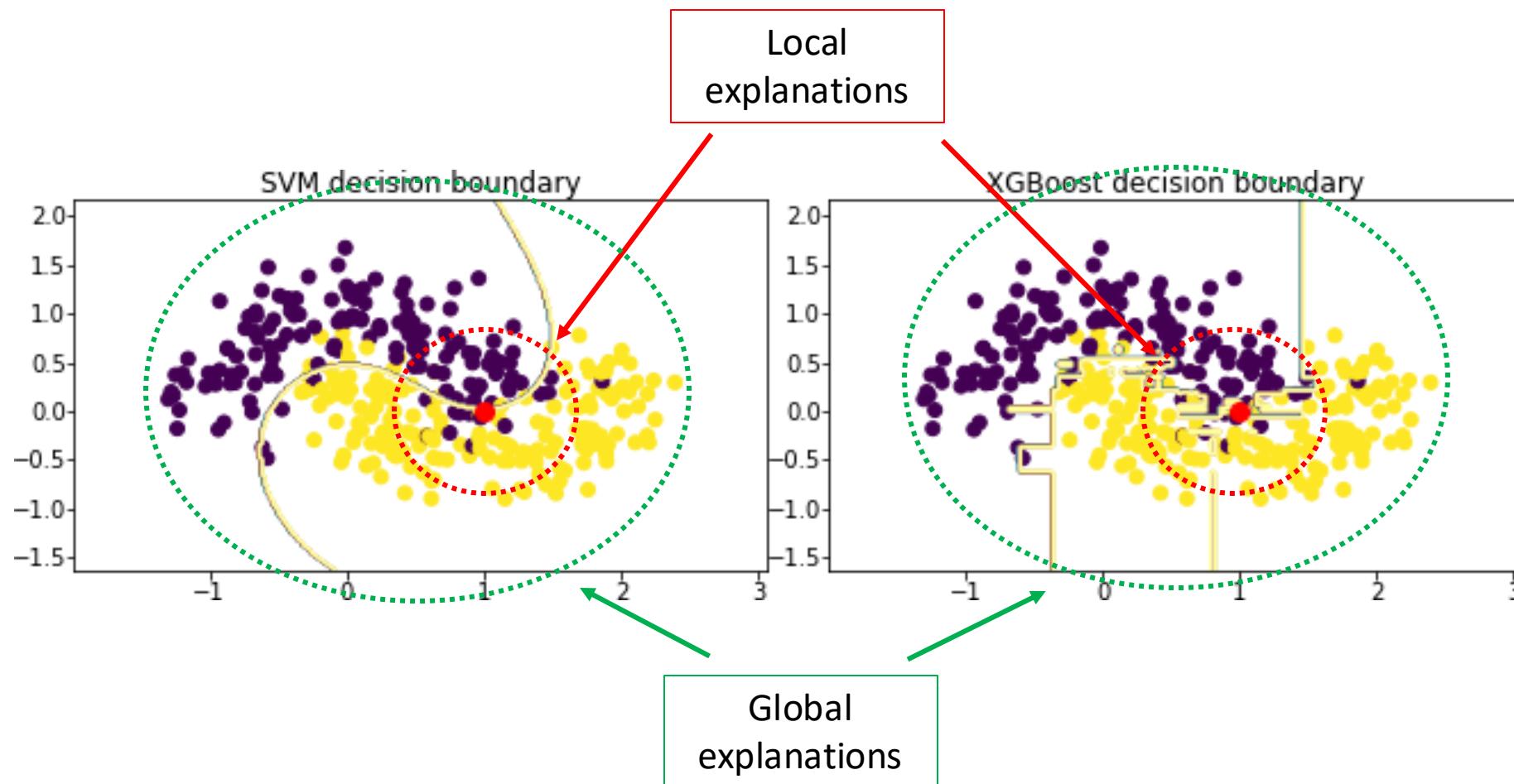
JAGIELLONIAN UNIVERSITY  
IN KRAKÓW



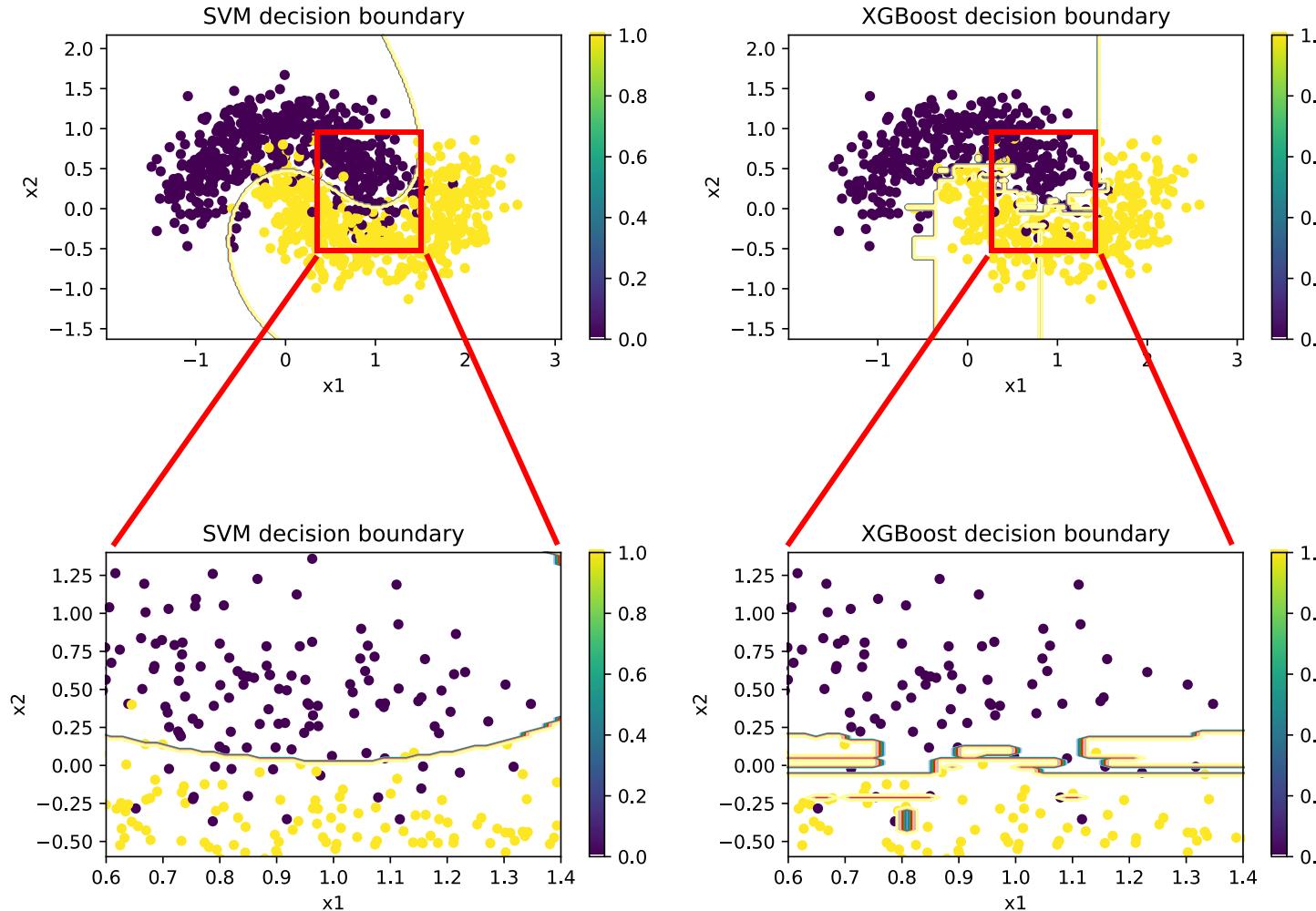
<https://geist.re>

The presentation covers several different works founded from the XPM (NCN UMO-2020/02/Y/ST6/00070) project  
funded by the National Science Centre, Poland under CHIST-ERA program .

# Local vs Global explanations



# Locally, the decision boundary is simpler



- In this approach we focus on explaining an instance
- "Zooming in" we can fit inherently interpretable model that will approximate the decision of the blackbox one
- The assumption is not always valid. There are models which have complex decision boundary even locally
- Term "Locally" is vague. The locality is subjective
- When zooming in, we are limiting the number of samples that can be used for training
- **Every ML model can be local surrogate**

# XAI-FUNGI: Dataset from the user study on comprehensibility of XAI algorithms

zenodo

Communities My dashboard

Published April 15, 2025 | Version 1.0.2

**XAI-FUNGI: Dataset from the user study on comprehensibility of XAI algorithms**

Bobek, Szymon (Researcher)<sup>1</sup> ; Korycińska, Paloma (Researcher)<sup>2</sup> ; Krakowska, Monika (Researcher)<sup>2</sup> ;  
Mozolewski, Maciej (Researcher)<sup>1</sup> ; Rak, Dorota (Researcher)<sup>3</sup> ; Zych, Magdalena (Researcher)<sup>2, 1</sup> ;  
Wójcik, Magdalena (Supervisor)<sup>1</sup> ; Nalepa, Grzegorz J. (Supervisor)<sup>1</sup>

**Contributors**

**Data collectors:** Korycińska, Paloma<sup>1</sup> ; Zych, Magdalena<sup>1, 2</sup> ; Rak, Dorota<sup>3</sup> ; Wójcik, Magdalena<sup>2</sup> ; Krakowska, Monika<sup>1</sup>   
**Data curators:** Mozolewski, Maciej<sup>2</sup> ; do Valle Miranda, Luiz<sup>2</sup> ; Zych, Honorata<sup>2</sup>   
**Editor:** Bobek, Szymon<sup>2</sup>

**XAI-FUNGI: Dataset from the user study on comprehensibility of XAI algorithms**

We present the dataset which was created during a user study on evaluation of explainability of artificial intelligence (AI) at the Jagiellonian University as a collaborative work of computer science (GEIST team) and information sciences research groups. The main goal of the research was to explore effective explanations of AI model patterns to diverse audiences.

The dataset contains material collected from 39 participants during the interviews conducted by the Information Sciences research group. The participants were recruited from 149 candidates to form three groups that represented domain experts in the field of mycology (DE), students with data science and visualization background (IT) and students from social sciences and humanities (SSH). Each group was given an explanation of a machine learning model trained to predict edible and non-edible mushrooms and asked

**786 VIEWS 2K DOWNLOADS**

**Versions**

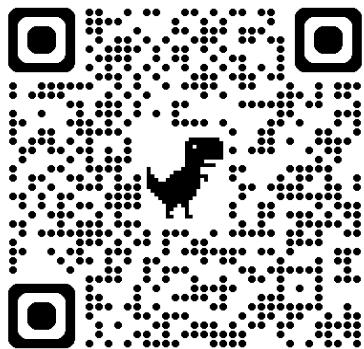
Version 1.0.2 10.5281/zenodo.15222484	Apr 15, 2025
Version 1.0.1 10.5281/zenodo.14980793	Jul 3, 2024
Version 1.0.0 10.5281/zenodo.11448395	Jun 3, 2024

[View all 3 versions](#)

**Cite all versions?** You can cite all versions by using the DOI 10.5281/zenodo.11448394. This DOI represents all versions, and will always resolve to the latest one. [Read more](#).

<https://zenodo.org/records/15222484>

# Rules to rule them all



DOI | [10.5281/zenodo.15222484](https://doi.org/10.5281/zenodo.15222484)

Example
<pre> cap_diameter_cm &gt; 8.54 cap_shape = convex cap_surface = smooth cap_color = white does_bruise_or_bleed = no_bruises_or_bleeding gills_attachment_to_the_stem = sinuate gills_spacing = close gills_color = white 5.96 &lt; stem_height_cm &lt;= 7.74 stem_width_mm &gt; 16.56 stem_root = no_data stem_surface = no_data stem_color = white veil_type = no_data veil_color = no_data spore_print_color = no_data habitat = meadows season = spring </pre>

A.I. prediction

● Edible

ANCHOR - method for explaining AI model

At the top right: anchor, which is a set of features whose combined presence (conjunction) determines how the AI model classifies a given mushroom.

The anchor does not have to reflect a real example from the data.

Below the anchor: the influence of the combined occurrence of the feature set (anchor) on the percentage of cases in which the model predicts a given class (i.e., the so-called classification certainty). When calculating this percentage, the model takes into account the features highlighted in blue.

Explanation of A.I. prediction

If ALL of these are true:

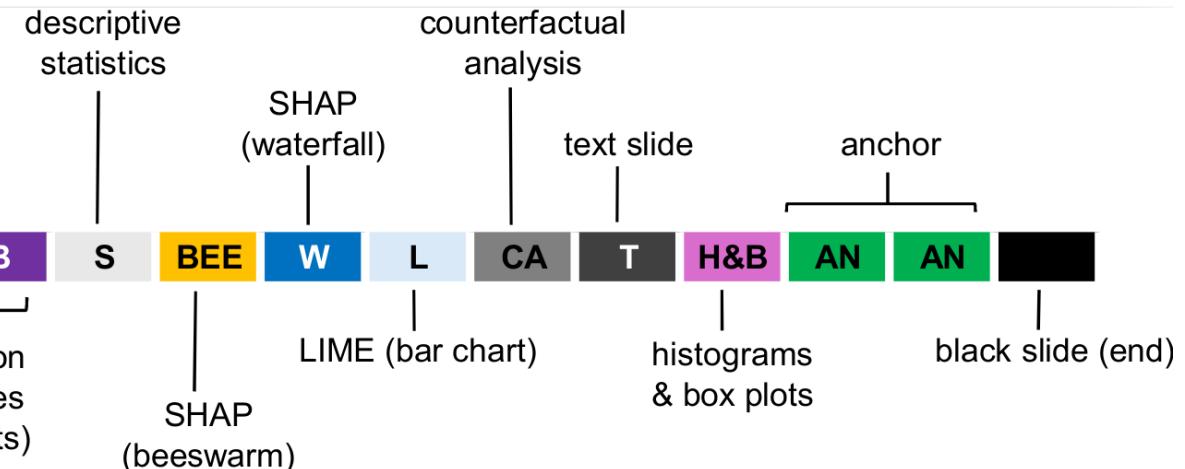
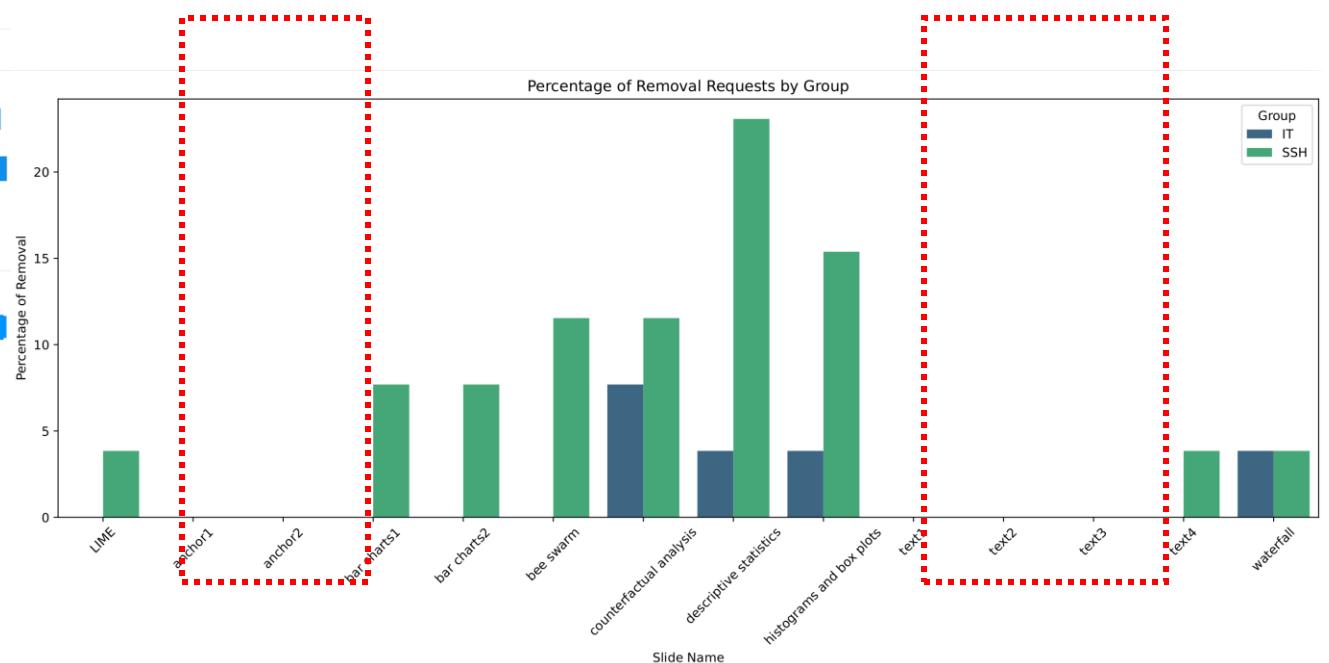
- stem\_surface = no\_data       stem\_width\_mm > 10.17
- stem\_root = no\_data       cap\_surface = smooth
- stem\_color = white       gills\_attachment\_to\_the\_stem = sinuate
- cap\_diameter\_cm > 5.87       stem\_height\_cm <= 7.74

The A.I. will predict edible 97.2% of the time

If ALL of these are true:

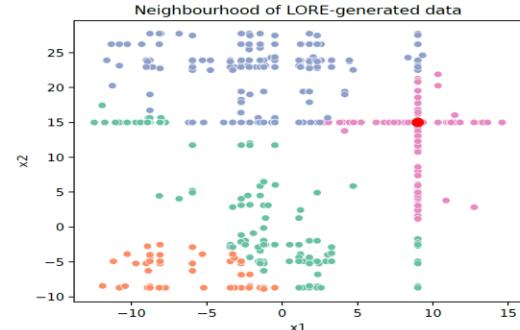
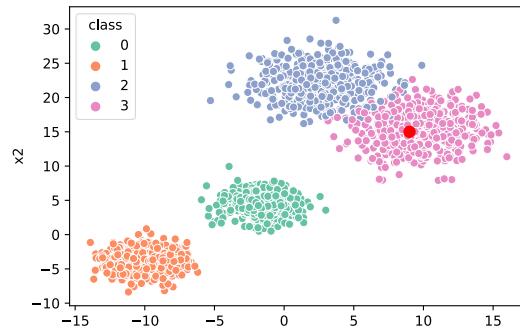
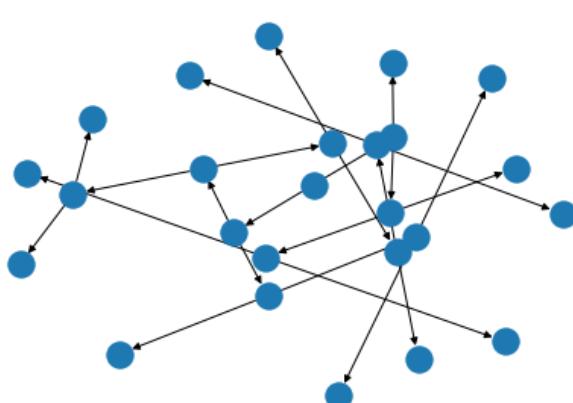
- stem\_surface = no\_data       stem\_width\_mm > 10.17
- stem\_root = no\_data       cap\_surface = smooth
- stem\_color = white       gills\_attachment\_to\_the\_stem = sinuate
- cap\_diameter\_cm > 5.87       stem\_height\_cm <= 7.74

The A.I. will predict edible 76.2% of the time



# LORE: Dataset generation and explanation creation

```
( [  
    {'class': 3},  
    { 'x1': '>6.532643',  
      'x2': '-0.82784 < x2 <=20.426761'}  
],  
[  
    { 'x2': '<=-0.82784' }  
])
```



- Generate samples by maximizing following fitness functions:

$$\text{fitness}_=(z) = I_{b(x)=b(z)} + (1 - d(x, z)) - I_{x=z}$$

$$\text{fitness}_{\neq}(z) = I_{b(x)\neq b(z)} + (1 - d(x, z)) - I_{x=z}$$

- Fitness functions determine survivors and generation performs (2-point) crossover and mutation scheme:

parent 1	25	clerk	10k	yes
parent 2	30	other	5k	no

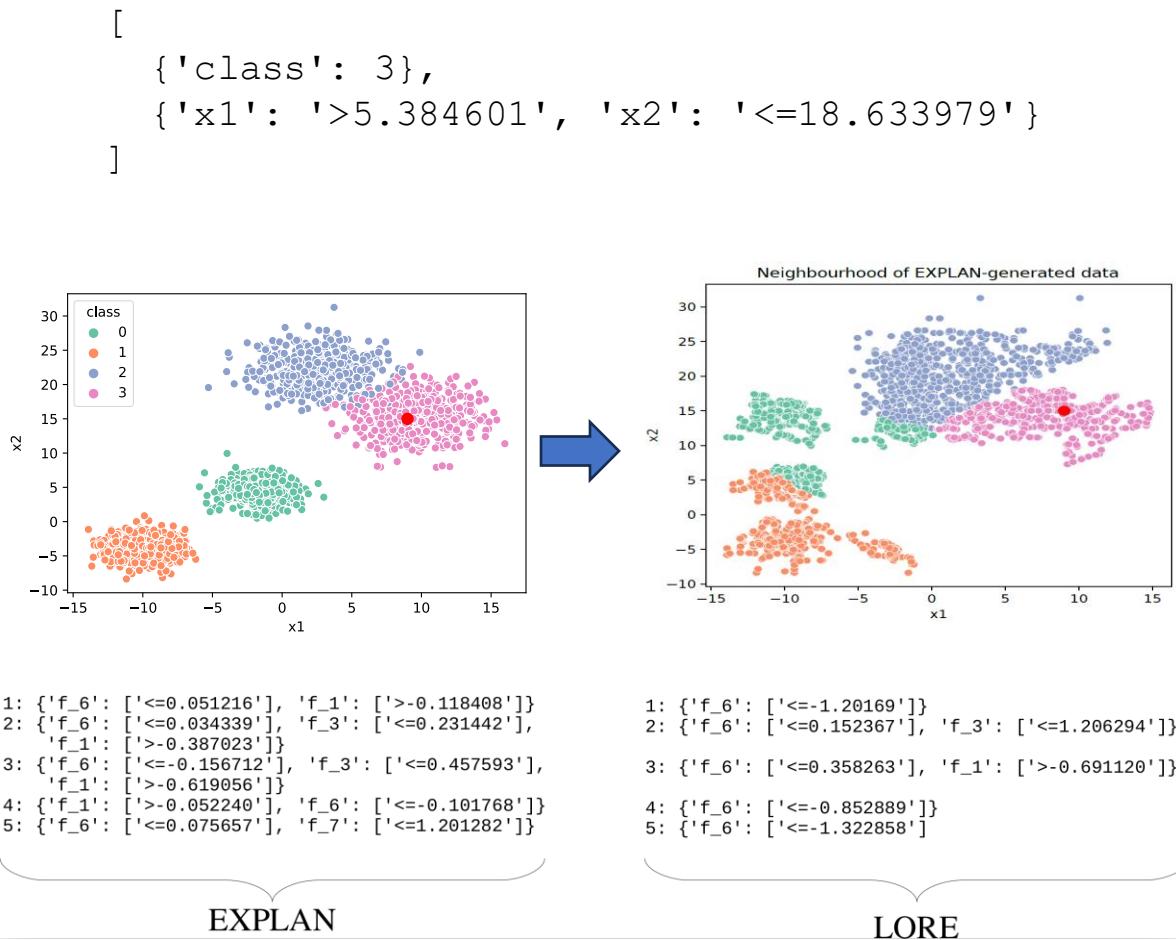
children 1	25	other	5k	yes
children 2	30	clerk	10k	no

parent	25	clerk	10k	yes
children	27	clerk	7k	yes

Replacing (mutation)  
according to empirical  
distribution of a  
feature

- The resulting dataset is balanced and completely artificial

# EXPLAN: Explanation creation



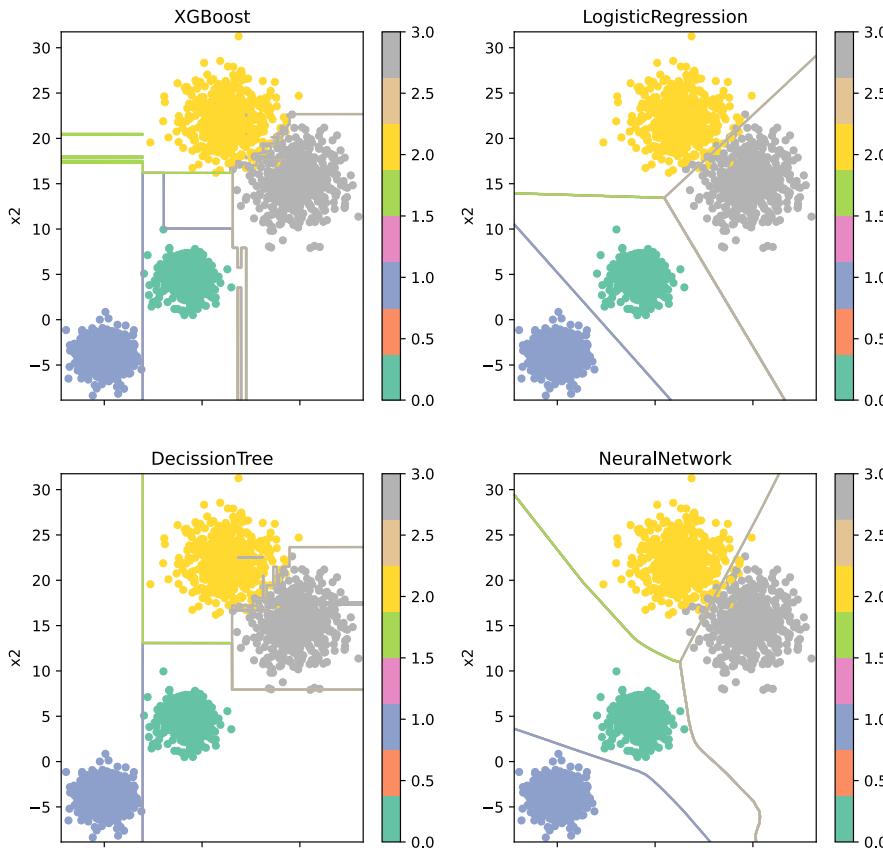
- The explanation is created using the same decision tree algorithm as LORE
- Generation with Importance-similarity perturbation, agglomerative clustering and SMOTE rebalancing
- Surprisingly the implementaiton of EXPLAN does not provide counterfactuals, which could be extracted
- The generation process may result in different explanations for the same instance

# State of the art methods

	Factual	Counterfactual	Visual	Example-based	Deterministic	Synthetic data
LORE	Yes	Yes	No*	No*	No	Only
EXPLAN	Yes	No*	No*	No*	No	Only
Anchor	Yes	No	No	Yes	No	Mainly

- They focus on one aspect of explanation, not the whole spectrum
- They use synthetic data, which is perfectly fine for ML engineer, but might be confusing for domain knowledge/user
- Consecutive runs yield different results
- Lack compelling visual representation

# Surrogate models are prone to Rashomon effect



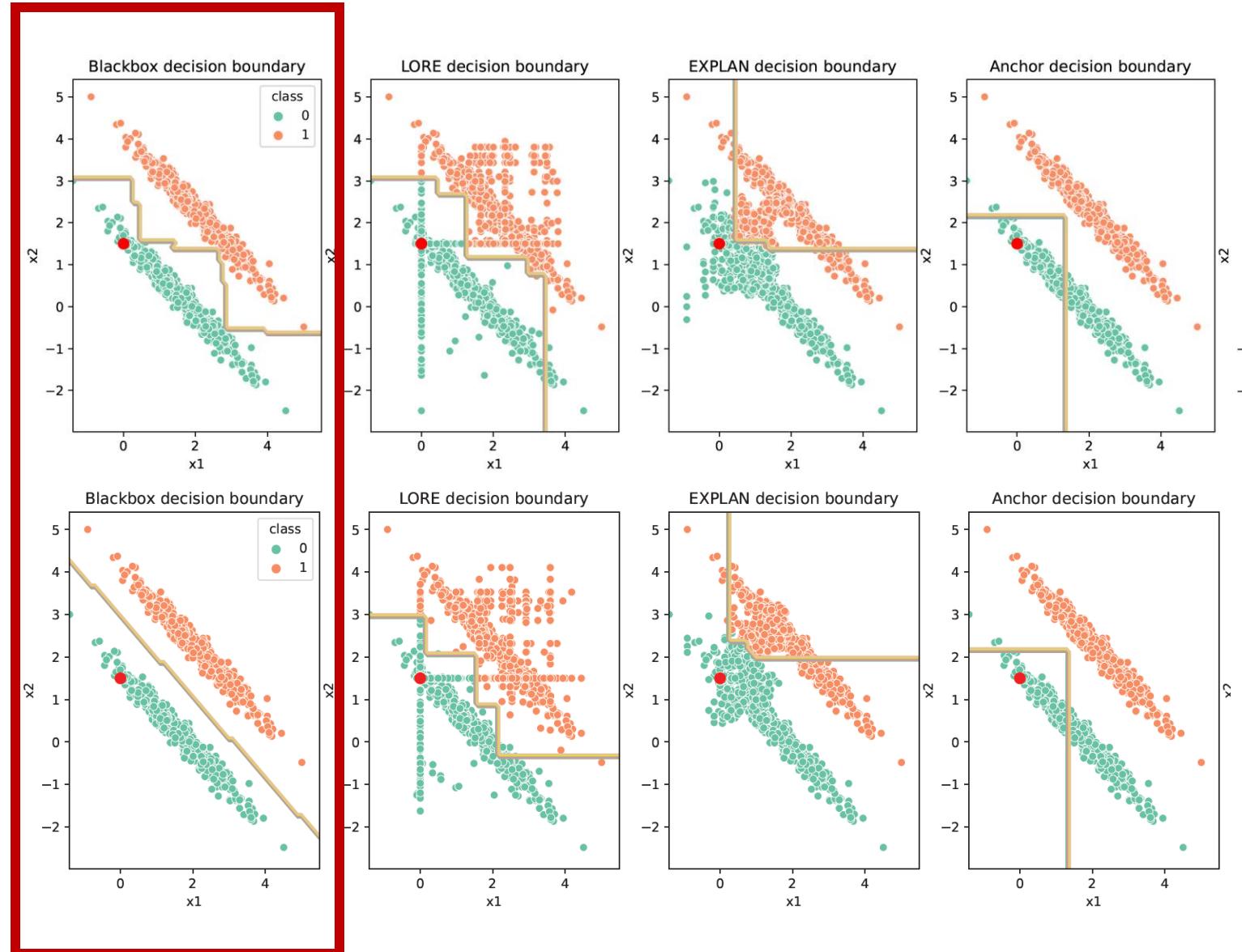
Rashomon (1950) by Akira Kurosawa

- Many models may be "right" but use very different methods to derive the "right"
- In Explainability we care about how the "right" is derived
- In such a case the more Rashomon effect the more doomed we are

# Surrogate models are prone to Rashomon effect



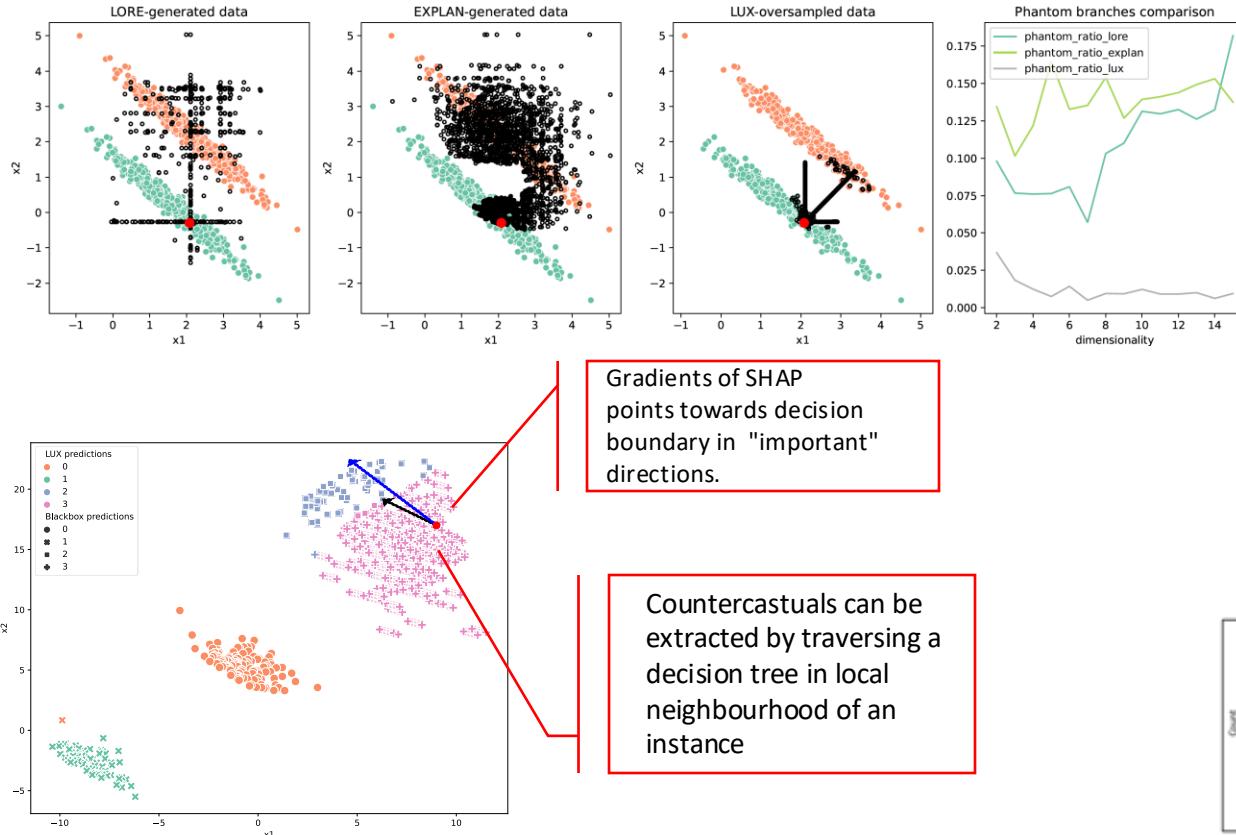
Rashomon (1950) by Akira Kurosawa



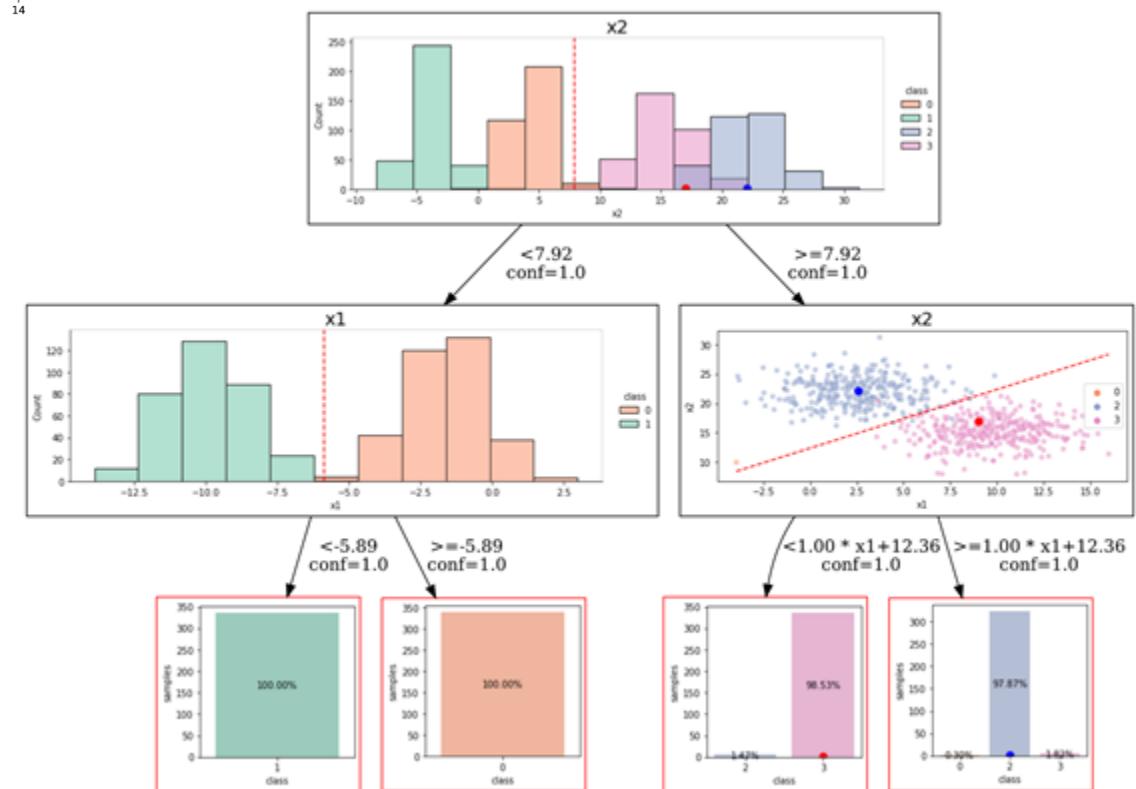


LUX: Local Universal Rule-based Explainer

# Local Universal eXplainer

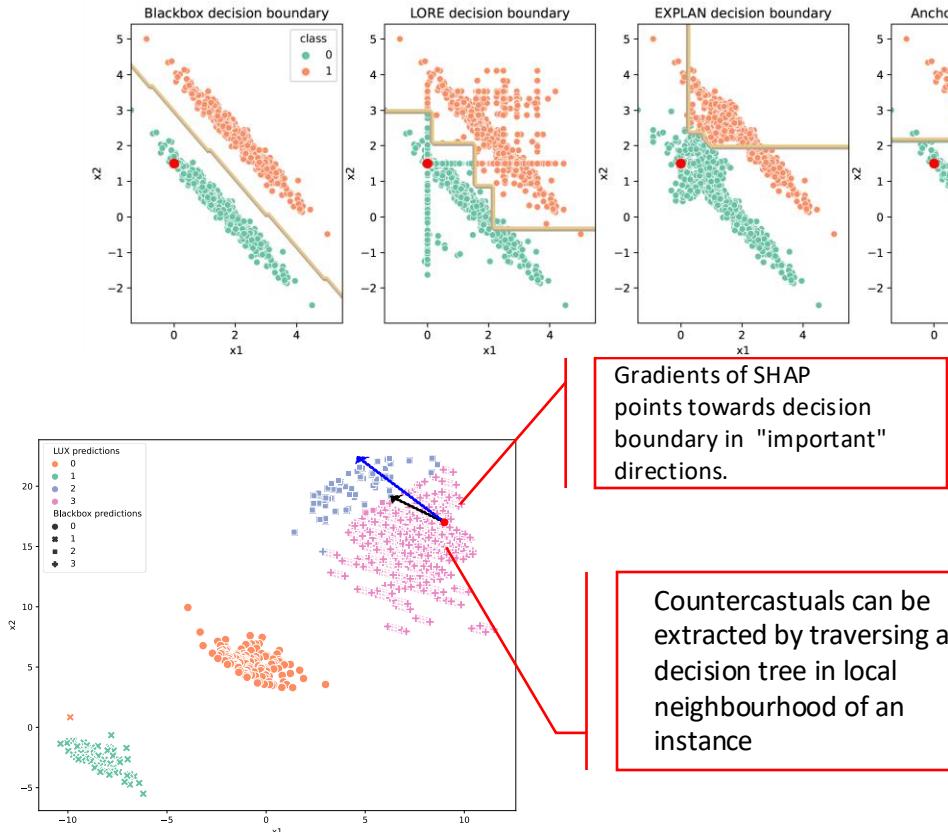


- Neighbourhood creation from original data
- Minimal artificial data generation (along SHAP-based gradients)
- Counterfactuals extraction
- Visualization
- Oblique splits

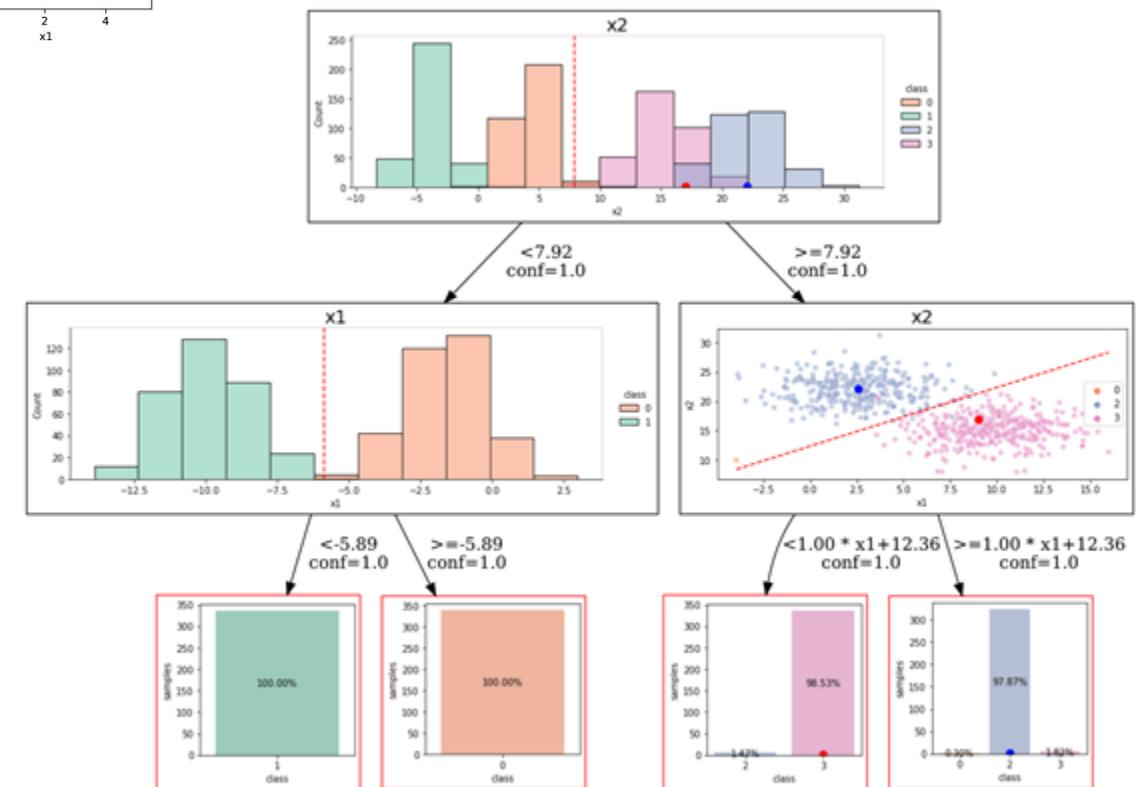


Bobek, S., & Nalepa, G. J. (2023). Local Universal Explainer (LUX) -- a rule-based explainer with factual, counterfactual and visual explanations. *arXiv [Cs.AI]*.  
Retrieved from <http://arxiv.org/abs/2310.14894>

# Local Universal eXplainer

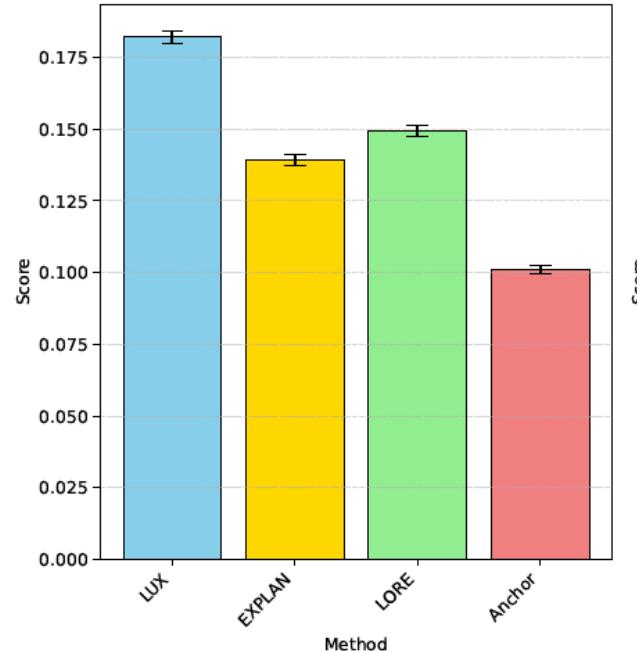


- Neighbourhood creation from original data
- Minimal artificial data generation (along SHAP-based gradients)
- Counterfactuals extraction
- Visualization
- Oblique splits

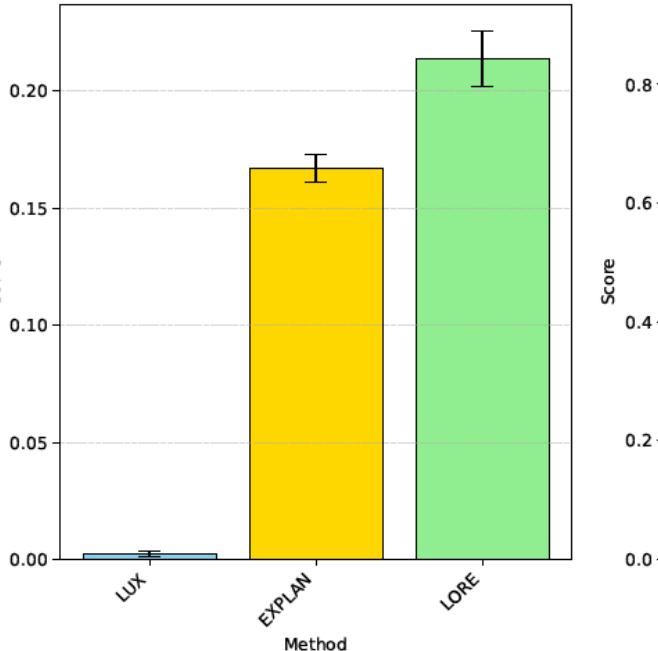


Bobek, S., & Nalepa, G. J. (2023). Local Universal Explainer (LUX) -- a rule-based explainer with factual, counterfactual and visual explanations. *arXiv [Cs.AI]*.  
Retrieved from <http://arxiv.org/abs/2310.14894>

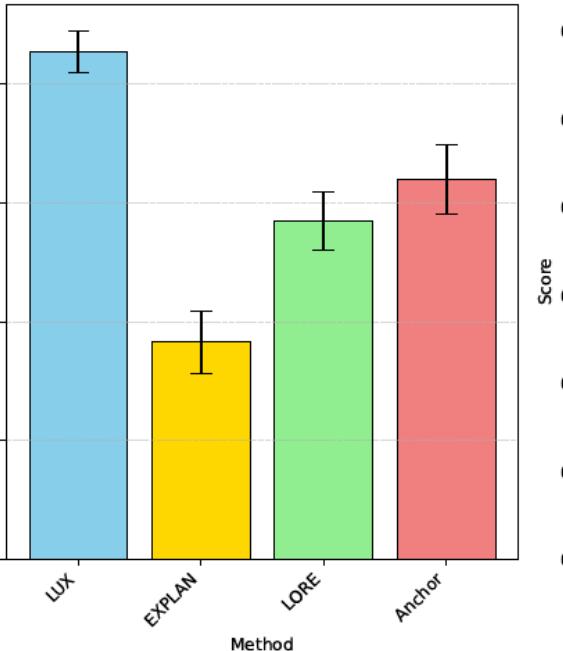
Performance with 95% confidence intervals for  
Shap Consistency ↑



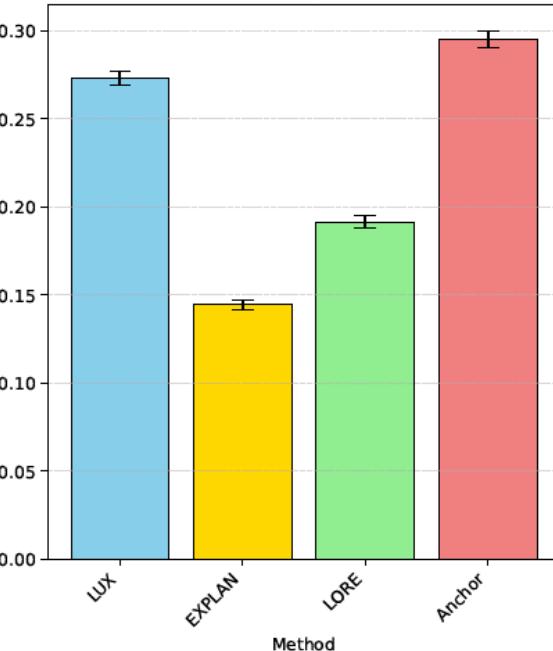
Performance with 95% confidence intervals for  
Representativeness (number of phantom branches) ↓



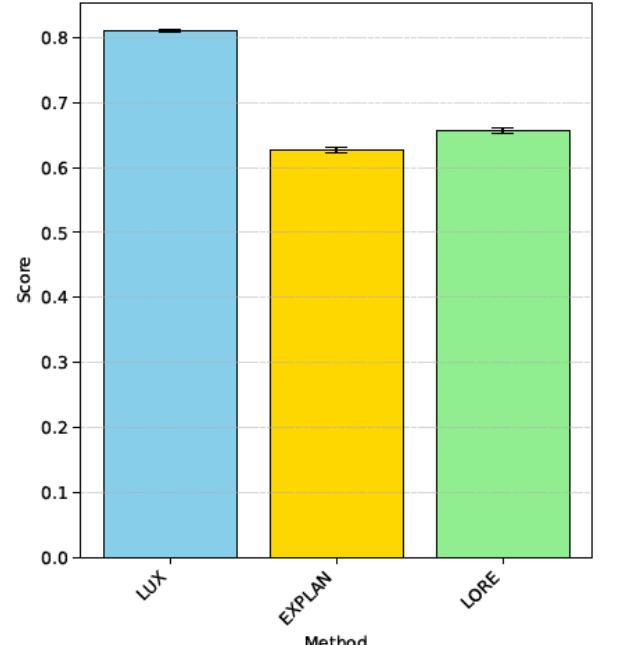
Performance with 95% confidence intervals for  
Stability ↑



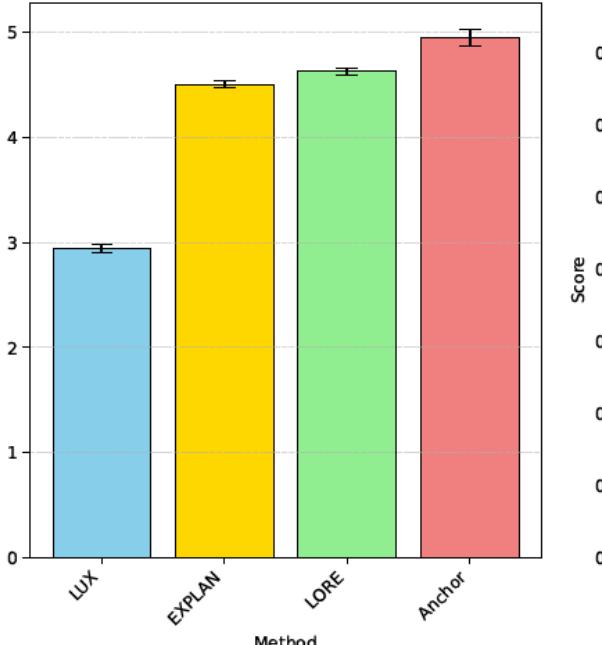
Performance with 95% confidence intervals for  
Coverage ↑



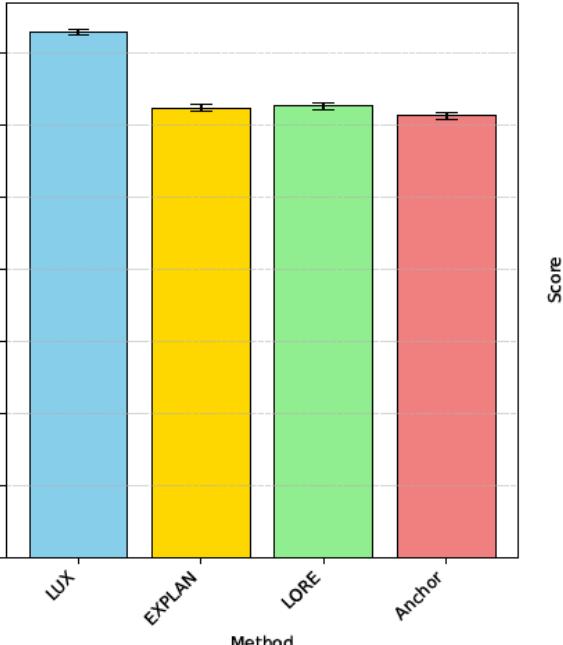
Performance with 95% confidence intervals for  
Counterfactual Fidelity ↑



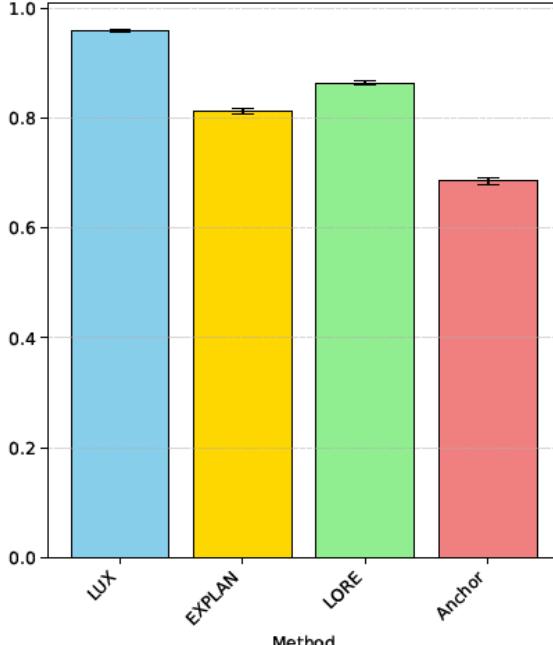
Performance with 95% confidence intervals for  
Simplicity (rule length) ↓



Performance with 95% confidence intervals for  
Fidelity ↑



Performance with 95% confidence intervals for  
Hits ↑



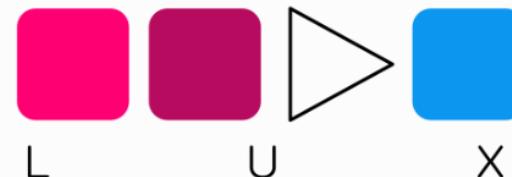
# LUX is OpenSource

Bobek, S.; Nalepa, G. J. Local Universal Rule-Based eXplainer (LUX).  
SoftwareX 2025, 30, 102102. <https://doi.org/10.1016/j.softx.2025.102102>.

The screenshot shows the homepage of the lux-explainer documentation. The header includes a logo for 'lux-explainer' and a search bar labeled 'Search docs'. The main content area is divided into sections: 'TUTORIALS' (Basic Usage examples, SHAP-guided explanation generation examples, Visualization examples, Custom model examples), 'REFERENCE' (API reference), and 'DEVELOPMENT' (Troubleshooting, Release notes, Contributing guide). A small image at the bottom left shows a dashboard with various charts and data.

The screenshot shows the 'Welcome to the LUX documentation' page. The top navigation bar includes a home icon, the text '/ Welcome to the LUX documentation', and a link 'View page source'. The main heading is 'Welcome to the LUX documentation'. Below the heading are four colored icons: a pink square with 'L', a dark red square with 'U', a white triangle in a blue square with 'X', and a blue square with 'X'.

## Welcome to the LUX documentation



LUX (Local Universal Rule-Based Explainer) is an XAI algorithm that produces explanations for any type of machine-learning model. It provides local explanations in a form of human-readable (and executable) rules, but also provide counterfactual explanations as well as visualization of the explanations.

## Install

LUX can be installed from either [PyPI](#) or directly from source code [GitHub](#)

To install from PyPI:

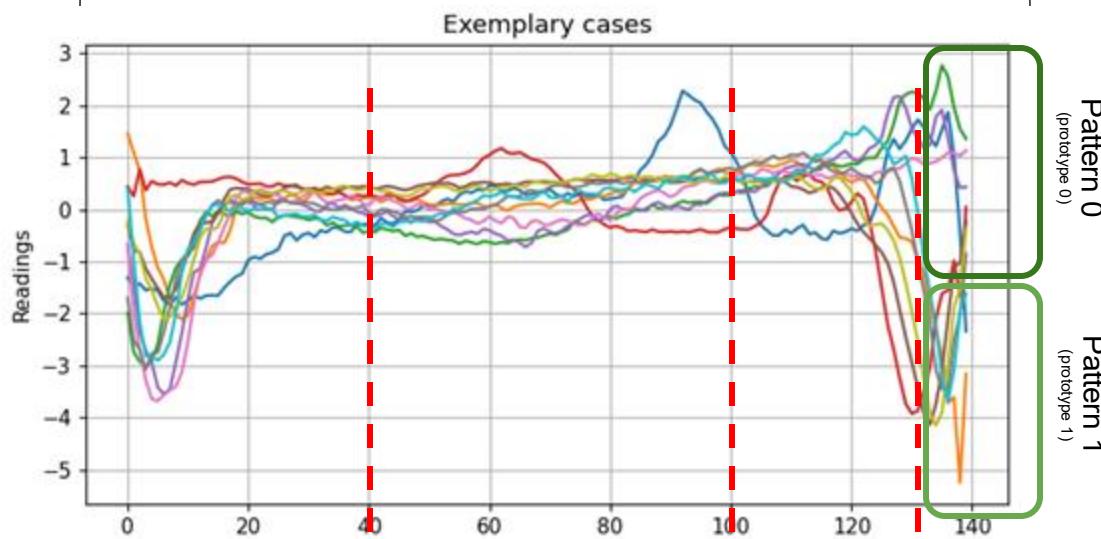
```
pip install lux-explainer
```



TSProto: Rule-based explanations for time-series

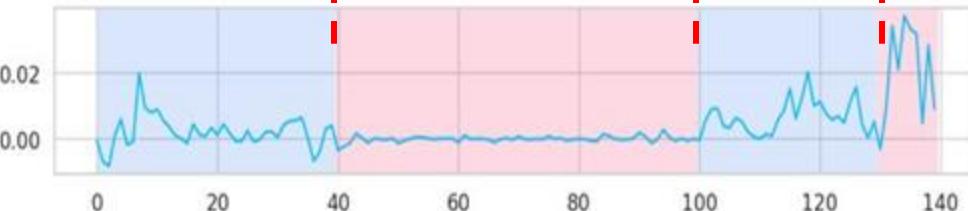
# Explainable components and prototypes discovery

The input is a dataset and a blackbox model that classifies time series



1 Calculate SHAP values for blackbox model that classifies time series

2 Take breakpoints of SHAP importances and slice original data with them to get explainable components. Then cluster slices into prototypes



The ruptures changepoint detection is used to find regions of similar importance in time series. Similar importance form dense regions that are considered interpretable components.

# Context aggregation

OHE streams with respect to presence/absence of particular pattern (prototype), enrich with statistics & build interpretable global model

3

4

Prototype 0 exists in TS?

Yes

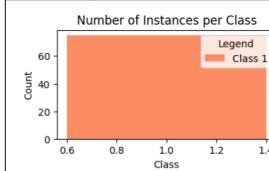
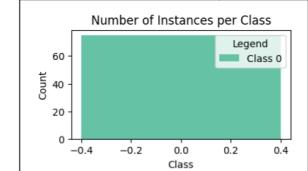
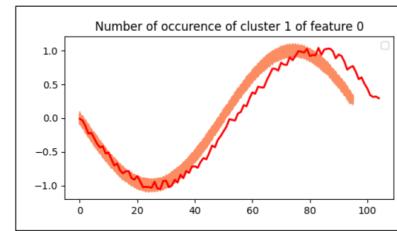
No

class 0

class 1

5

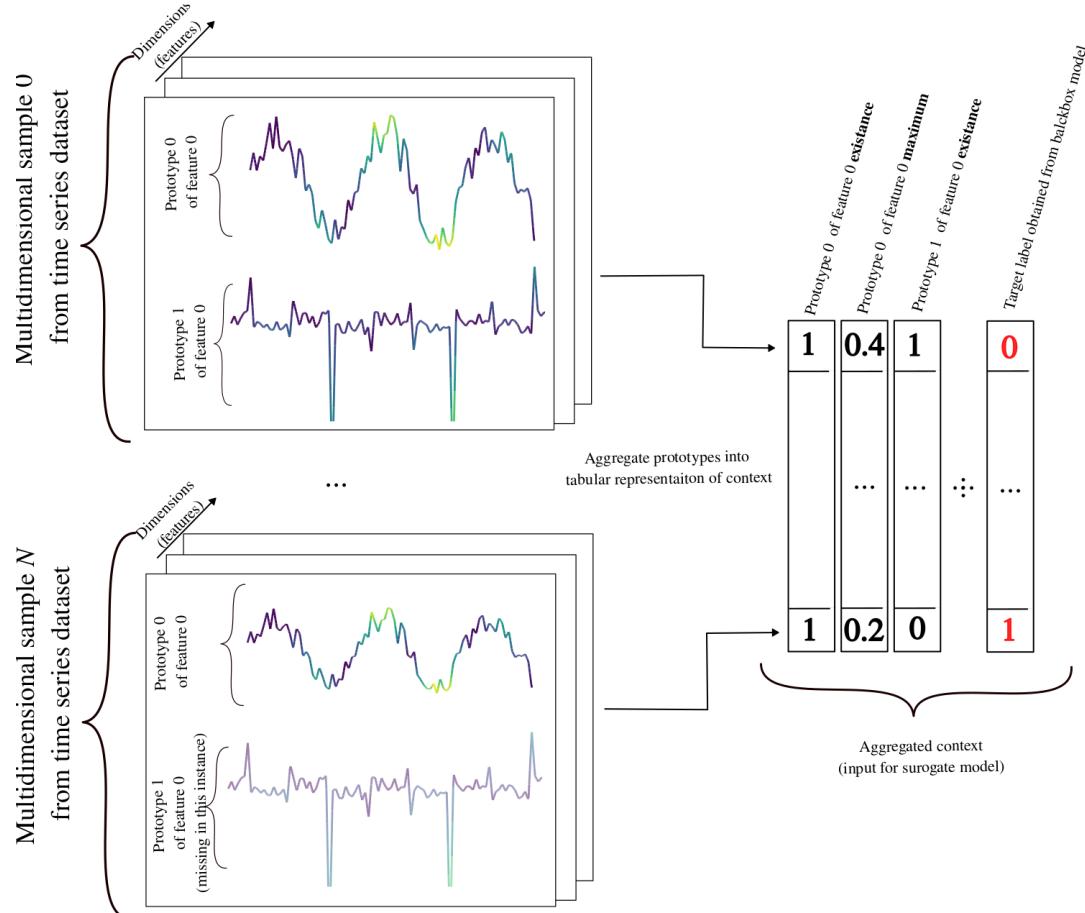
Enrich explanations with visual representation of prototypes and interpretable components



$\leq 0.50$

$> 0.50$

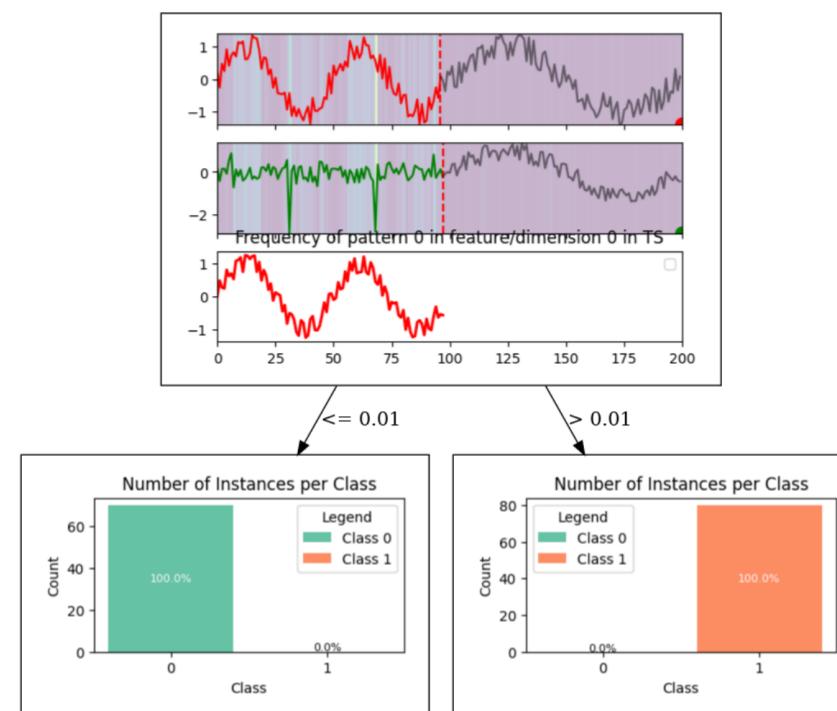
# Context aggregation



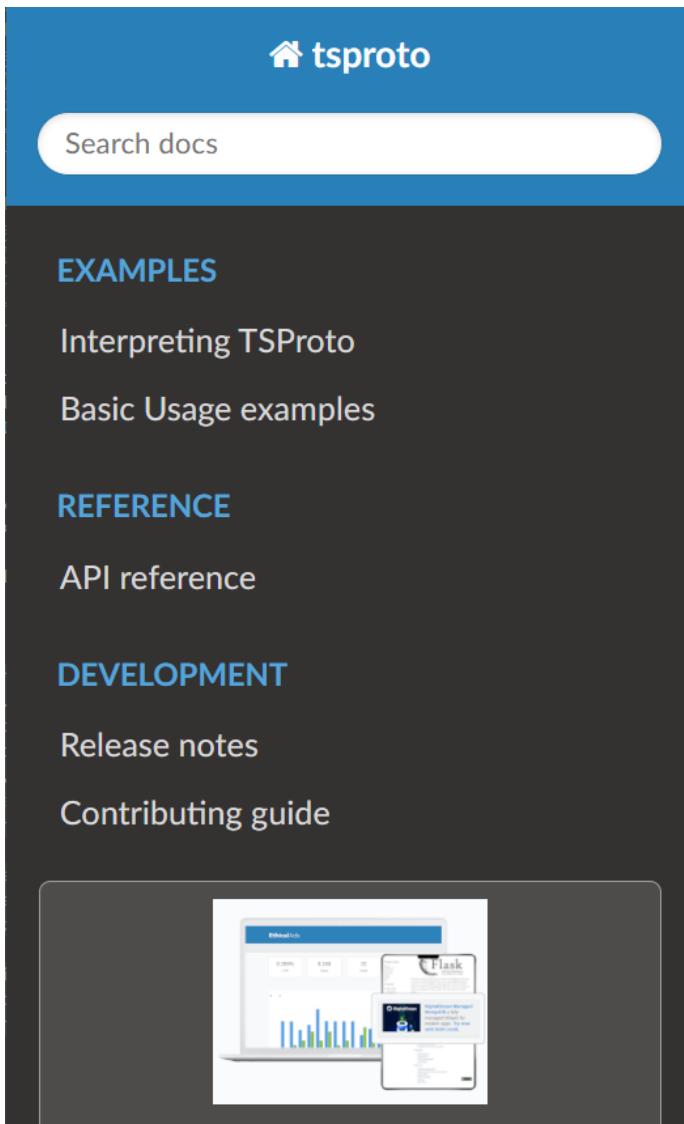
Prototype visualization/Important part similar to prototype (barycenter of cluster)

Contrastive example (most similar to prototype, but different class)

Not important component



# TSProto is OpenSource



## Welcome to the TSProto documentation

TSProto (**P**ost-host **p**rototype-based **e**xplanations **w**ith **r**ules **f**or **t**ime-series **c**lassifiers) is an XAI algorithm that produces explanations for any type of machine-learning model. It provides local explanations in a form of human-readable (and executable) rules, but also provide counterfactual explanations as well as visualization of the explanations.

## Install

TSProto can be installed from either [PyPI](#) or directly from source code [GitHub](#)

To install from PyPI:

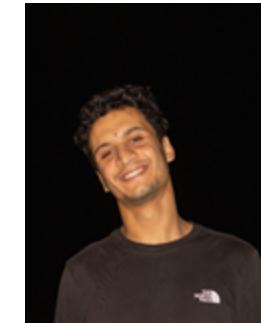
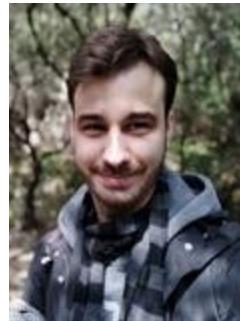
```
pip install tsproto
```

# People



Prof. Grzegorz  
J. Nalepa, PhD, Eng.

GEIST Leader



# Thank you for your attention!



JAGIELLONIAN UNIVERSITY  
IN KRAKÓW



<https://geist.re>

# References

- Bobek, S.; Nalepa, G. J. **Local Universal Explainer (LUX) -- a Rule-Based Explainer with Factual, Counterfactual and Visual Explanations.** arXiv February 9, 2024. <https://doi.org/10.48550/arXiv.2310.14894>.
- Bobek, S.; Nalepa, G. J. **Local Universal Rule-Based eXplainer (LUX).** SoftwareX 2025, 30, 102102. <https://doi.org/10.1016/j.softx.2025.102102>.
- Bobek, S.; Nalepa, G. J. **TSProto: Fusing deep feature extraction with interpretable glass-box surrogate model for explainable time-series classification.** Inf. Fusion 124, C (Dec 2025). <https://doi.org/10.1016/j.inffus.2025.103357>
- Bobek, S.; Korycińska, P.; Krakowska, M.; Mozolewski, M.; Rak, D.; Zych, M.; Wójcik, M.; Nalepa, G. J. **User-Centric Evaluation of Explainability of AI with and for Humans: A Comprehensive Empirical Study.** International Journal of Human-Computer Studies 2025, 205, 103625. <https://doi.org/10.1016/j.ijhcs.2025.103625>.
- Bobek, S., Korycińska, P., Krakowska, M. et al. **Dataset resulting from the user study on comprehensibility of explainable AI algorithms.** Sci Data 12, 1000 (2025). <https://doi.org/10.1038/s41597-025-05167-6>