# Gradual Patterns for Explainable AI

Durande KAMGA* **, Jerry LONLAC*, Anthony FLEURY*, Engelbert MEPHU NGUIFO **

*IMT Nord Europe, Institut Mines-Télécom, Univ. Lille, Centre for Digital Systems, F-59000 Lille, France

**University Clermont Auvergne, Clermont Auvergne INP, ENSMSE, CNRS, LIMOS, 63000, Clermont-Ferrand, France

# Outline

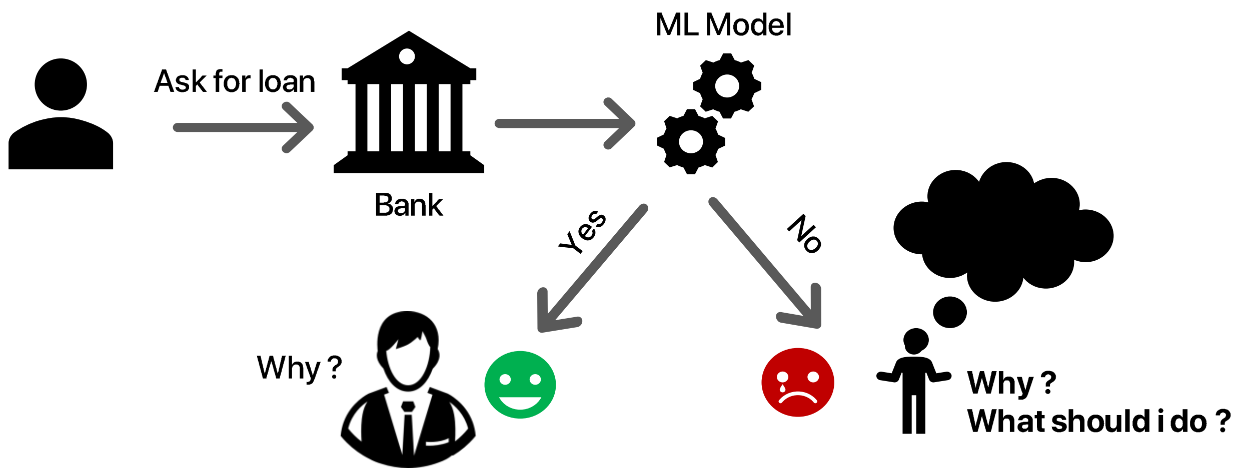
Context & Motivation

Background: Gradual Patterns

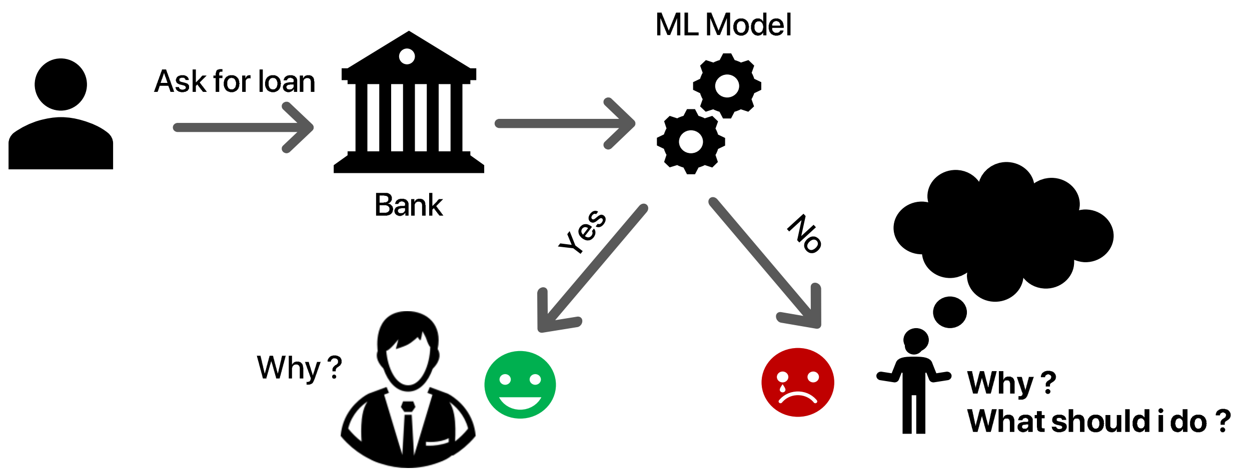
GP-Based Neighborhood Generation

Experiments, results and discussion

# Context

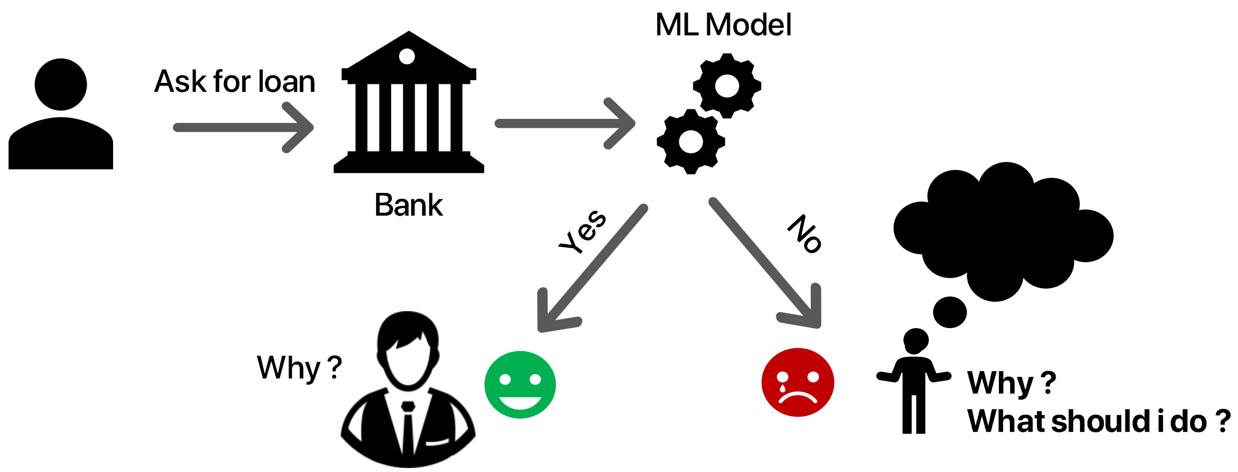


# Context




**But the most models are black box (Random Forest, XGBoost, Neural Network)**

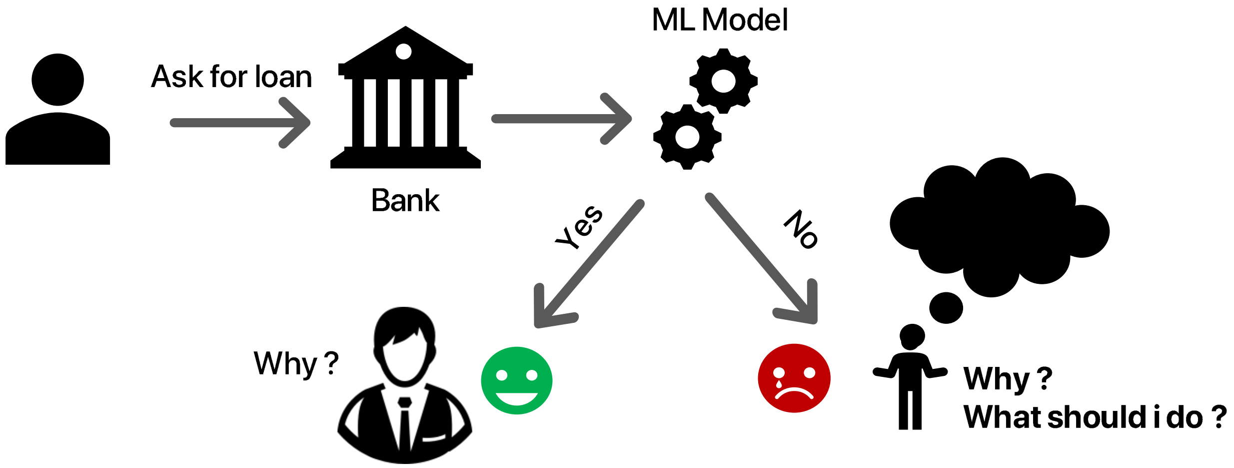
# Context



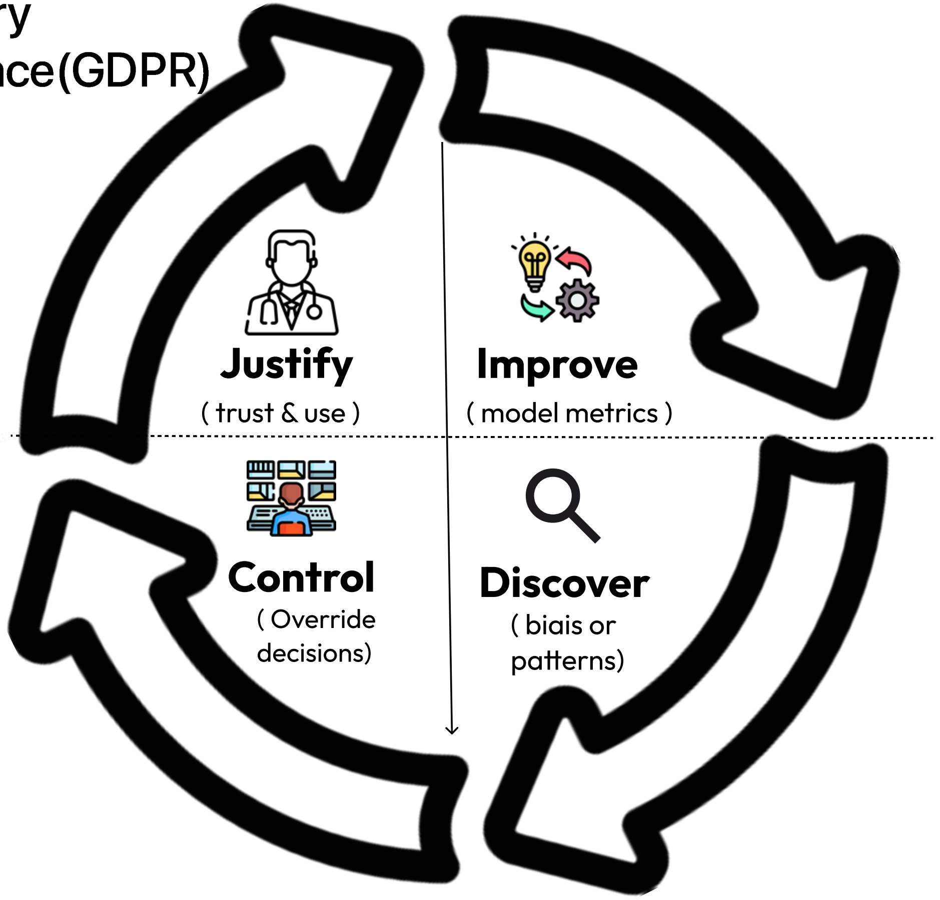

**But the most models are black box (Random Forest, XGBoost, Neural Network)**

**XAI can help in both questions**

# Context



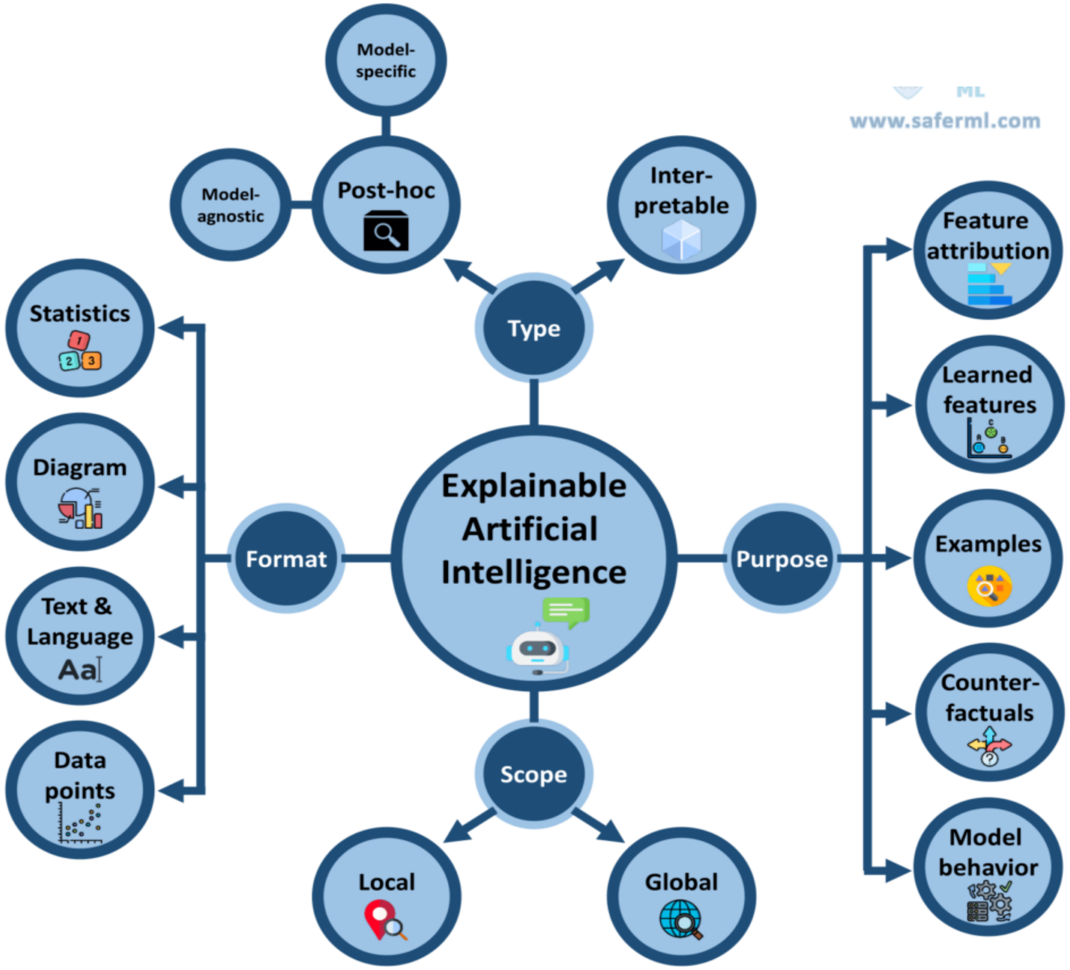

Regulatory Compliance(GDPR)




**But the most models are black box (Random Forest, XGBoost, Neural Network)**

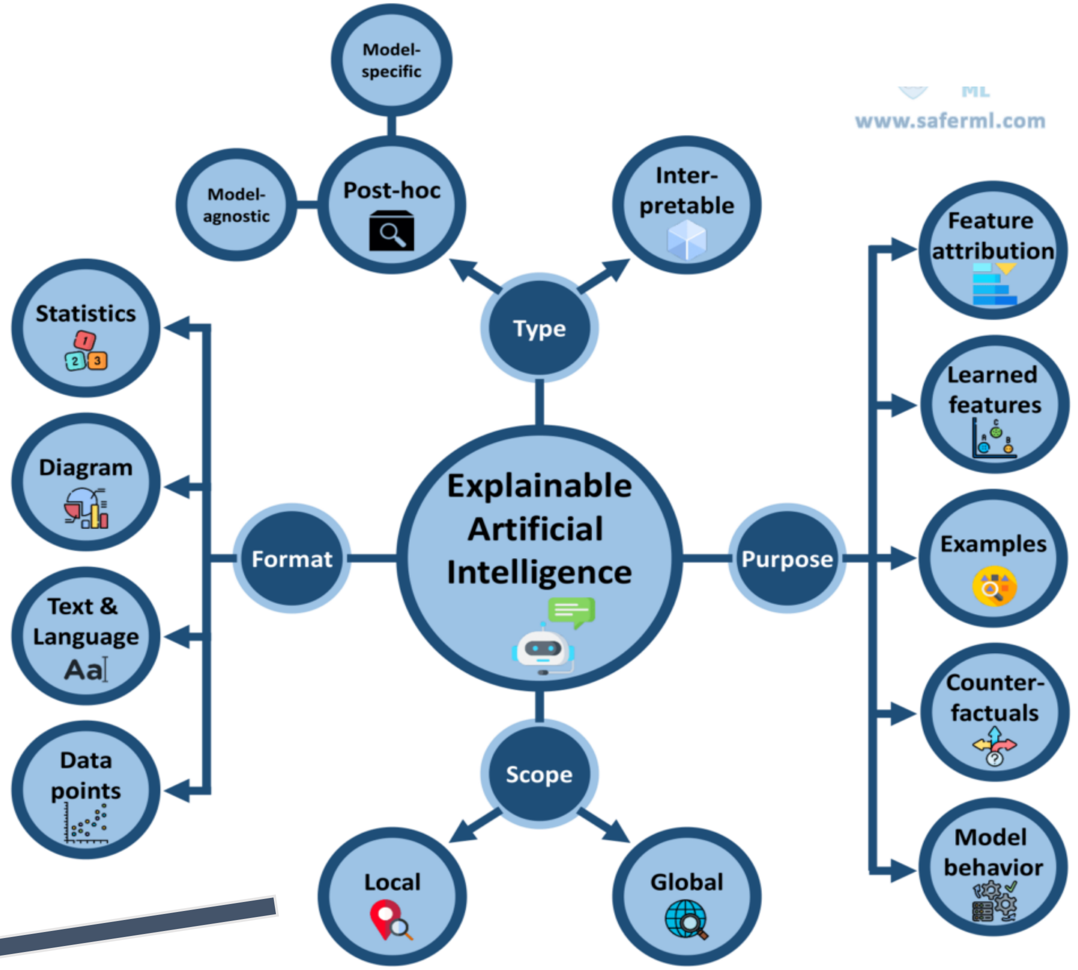
**XAI can help in both questions**

# Context



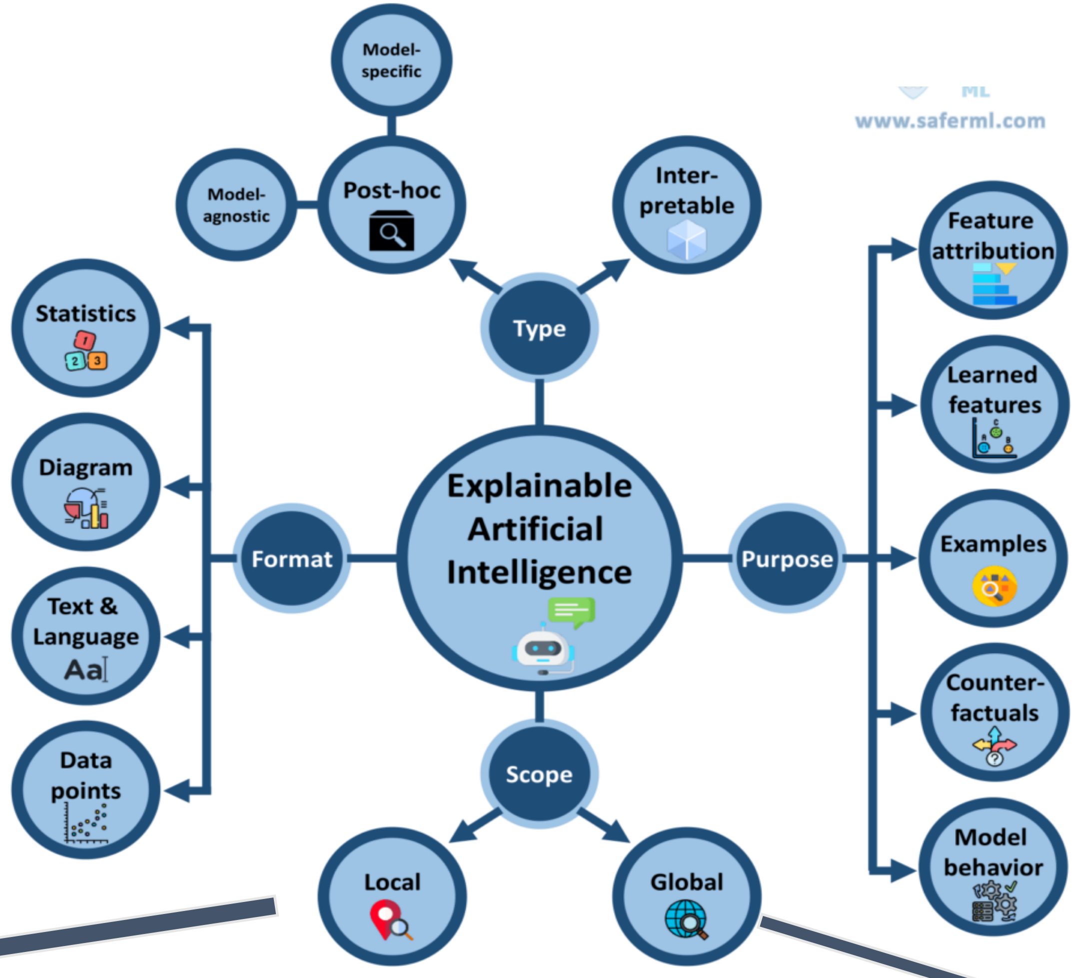
www.saferml.com
Model-specific
Model-agnostic
Post-hoc
Inter-pretable
Type
Statistics
Diagram
Text & Language
Aa
Data points
Format
Explainable Artificial Intelligence
Purpose
Feature attribution
Learned features
Examples
Counter-factuals
Model behavior
Scope
Local
Global

# Context




Explain prediction for one observation

# Context




Explain prediction for one observation

Explain prediction for a set of observations. Useful to debug a model

# Context

Instance

(1) Spilts instance into interpretables components (future explanation)

(2) Generate neighbourhood by perturbing the instance

- Randomly
- Using genetics algorithms
- Using AutoEncoder
- Knowledge graph

(3) Train an interpretable model

- Linear model
- Decision Tree
- Rules

(4) Retrieve explanations to the user

# Context

Instance

(1) Spilts instance into interpretables components (future explanation)

(2) Generate neighbourhood by perturbing the instance

- Randomly
- Using genetics algorithms
- Using AutoEncoder
- Knowledge graph

(3) Train an interpretable model

- Linear model
- Decision Tree
- Rules

(4) Retrieve explanations to the user

Local Interpretable Model-agnostic Explainations ~ LIME
(Ribeiro et al., 2016)

SHapley Additive exPlanation ~ SHAP
(Lundberg & Lee, 2017)

LOcal Rule-based Explanations ~ LORE
(Guidotti et al., 2018)

Anchors
(Ribeiro et al., 2018)

# Limitations of Current XAI Methods

## Instability

Same instance, same model, different runs
= DIFFERENT explanations!

# Limitations of Current XAI Methods

## Instability

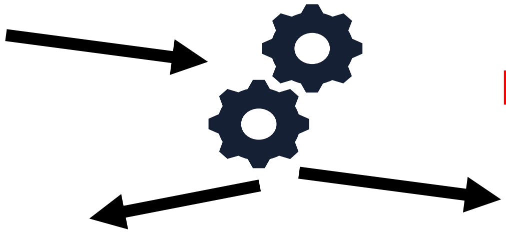
Same instance, same model, different runs
= DIFFERENT explanations!

Alice:{

"Income": 52000, "CreditScore": 645, "DebtRatio": 0.42,

"EmploymentYears": 3.5,  "Age": 34,"LoanAmount": 28000

}



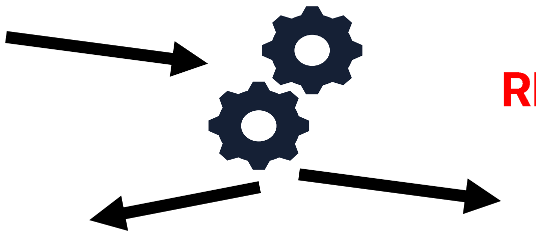
REJECTED

# Limitations of Current XAI Methods

## Instability

Same instance, same model, different runs
= DIFFERENT explanations!

Alice:{
"Income": 52000, "CreditScore": 645, "DebtRatio": 0.42,
"EmploymentYears": 3.5, "Age": 34,"LoanAmount": 28000
}



REJECTED

Run 1:

{
CreditScore <= 710.61,
Income <= 63782.38,
DebtRatio > 0.18
}

Run 2:

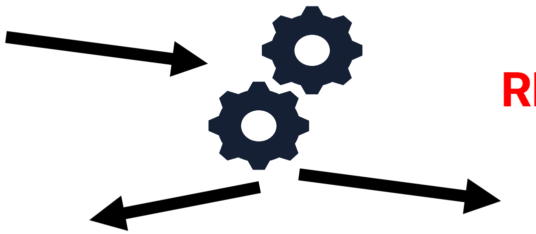
{
CreditScore <= 712.39,
Income <= 65554.80,
}

# Limitations of Current XAI Methods

## Instability

Same instance, same model, different runs
= DIFFERENT explanations!

Alice:{
"Income": 52000, "CreditScore": 645, "DebtRatio": 0.42,
"EmploymentYears": 3.5, "Age": 34,"LoanAmount": 28000
}



REJECTED

Run 1:

{
CreditScore <= 710.61,
Income <= 63782.38,
DebtRatio > 0.18
}

Run 2:

{
CreditScore <= 712.39,
Income <= 65554.80,
}

## Unrealistic neighborhoods

Random perturbations create impossible combinations

Random perturbation might generate:

Income = $150,000 (very high)

Credit Score = 300 (very low)

Debt Ratio = 0.1 (very low)

# Gradual patterns

**Gradual Patterns capture data relationships:**

"The more X increases, the more Y increases/decreases"

| ID | Income | DebtRatio | CreditScore |
|---|---|---|---|
| T1 | 100k | 0.50 | 1000 |
| T2 | 80k | 0.62 | 900 |
| T3 | 60k | 1.20 | 800 |
| T4 | 90k | 0.65 | 950 |

A gradual item is a pair $(i, v)$ in which $i$ is one attribute and $v$ its associated variation, with $v \in \{\uparrow, \downarrow\}$. Ex:$(Income, \downarrow)$

A gradual pattern G is a set of gradual items. $G = \{(i_1, v_1), \dots, (i_n, v_n)\}$. Ex:$G_1 = \{(Income, \downarrow), (DebtRatio, \uparrow)\}$

They are represented by a binary matrix to capture relationship between data

# Gradual patterns

| ID | Income | DebtRatio | CreditScore |
|---|---|---|---|
| T1 | 100k | 0.50 | 1000 |
| T2 | 80k | 0.62 | 900 |
| T3 | 60k | 1.20 | 800 |
| T4 | 90k | 0.65 | 950 |

They are represented by a binary matrix to capture relationship between data

The support of a gradual pattern represents the weight of this pattern.

| ↱ | t1 | t2 | t3 | t4 |
|---|---|---|---|---|
| t1 | 0 | 1 | 1 | 1 |
| t2 | 0 | 0 | 1 | 0 |
| t3 | 0 | 0 | 0 | 0 |
| t4 | 0 | 1 | 1 | 0 |

$(Income, \downarrow)$

$$\{G \mid sup(G) \geq minSupp\}$$

Paraminer (Negrevergne et al. 2014)

# Gradual Patterns Based Neighborhood Generation

instance

→

Generate neighbourhood by perturbing the instance

- Randomly
- Using genetics algorithms
- Expert knowledge

→

Train an interpretable model

- Linear model
- Decision Tree
- Rules

→

Retrieve explanations to the user

# Gradual Patterns Based Neighborhood Generation

instance

Generate neighbourhood by perturbing the instance

- Randomly
- Using genetics algorithms
- Expert knowledge

Train an interpretable model

- Linear model
- Decision Tree
- Rules

Retrieve explanations to the user

Use Graduals patterns here instead

# Gradual Patterns Based Neighborhood Generation

Given a dataset generate GP with threshold =0.7 training dataset
Decrease the threshold until i found a pattern of size 2

Generate neighbors by following pattern directions

$$x' = x \pm \alpha \cdot \sigma$$

**Algorithm 1** GP-Based Neighborhood Generation

**Require:** Instance $x$, Patterns $GPs$, Blackbox $f$
**Ensure:** Neighborhood $\mathcal{N}(x)$

```
1: N ← ∅
2: for GP ∈ GPs do
3:     Select attributes i_1, ..., i_k of GP
4:     for α ∈ {0.5, 1.0, 1.5} do
5:         x' ← Perturb(x, (i_1, ...i_k), α)
6:         N ← N ∪ {(x', f(x'))}
7:     end for
8: end for
9: return N
```

# Gradual Patterns Based Neighborhood Generation

Given a dataset generate GP with threshold =0.7 training dataset
Decrease the threshold until i found a pattern of size 2

**Algorithm 1** GP-Based Neighborhood Generation

**Require:** Instance $x$, Patterns $GPs$, Blackbox $f$
**Ensure:** Neighborhood $\mathcal{N}(x)$

$\mathcal{N} \leftarrow \emptyset$
**for** $GP \in GPs$ **do**
  Select attributes $i_1, \ldots, i_k$ of $GP$
  **for** $\alpha \in \{0.5, 1.0, 1.5\}$ **do**
    $x' \leftarrow \text{Perturb}(x, (i_1, \ldots i_k), \alpha)$
    $\mathcal{N} \leftarrow \mathcal{N} \cup \{(x', f(x'))\}$
  **end for**
**end for**
**return** $\mathcal{N}$

Generate neighbors by following pattern directions

$$x' = x \pm \alpha \cdot \sigma$$

**Example:**

- Instance x = {Income=$50k, CreditScore=650,DebtRatio=0.4}
- Pattern G = {(Income,↑), (CreditScore, ↑)}

**Generate variations:**

- Positive (+1 sigma) = {Income=$54,500, CreditScore=715, DebtRatio=0.4}
- Negative (-0.5 sigma)= {Income=$47,750, CreditScore=618, DebtRatio=0.4}

# Gradual Patterns Based Neighborhood Generation

Key Properties

- Statistical significance
- Plausibility (values within observed ranges)
- Deterministic (no randomness)

# Experiments

- Compare Neighborhood generated, execution time, stability

- **Material**:

  Processor: Intel Xeon Silver (64 cores), 512 GB RAM running Ubuntu 22.04.4 LTS

- **Datasets used**

| Dataset | Instances | Features | Domain |
|---|---|---|---|
| Adult Income | 48,842 | 14 | Income prediction |
| German Credit | 1,000 | 20 | Credit scoring |
| COMPAS | 4,000 | 10 | Recidivism risk |

- **Models used**

| Dataset | Random Forest | XGBoost |
|---|---|---|
| Adult Income | 0.86 | 0.87 |
| German Credit | 0.66 | 0.73 |
| COMPAS | 0.81 | 0.78 |

Source code

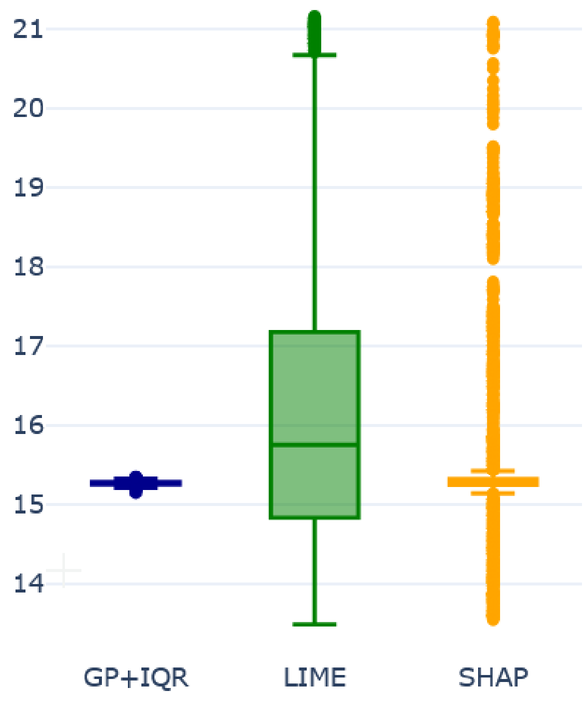
# Results : Stability

- Jaccard similarity

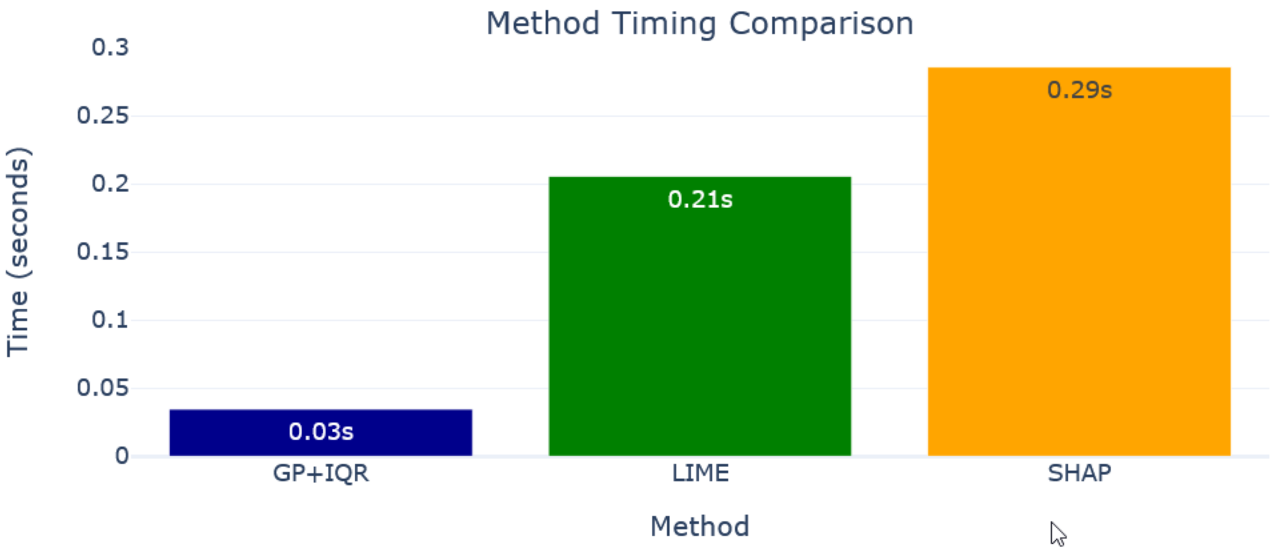
$$\text{Stability} = J(S_1, \ldots, S_n) = \frac{\left|\bigcap_{i=1}^{n} S_i\right|}{\left|\bigcup_{i=1}^{n} S_i\right|}$$

| Dataset | LIME | LORE | SHAP | Ours |
|---|---|---|---|---|
| Adult | 0.6 | 0.87 | 0.65 | 1.0 |
| COMPAS | 0.6 | 0.90 | 0.8 | 1.0 |
| German | 0.7 | 0.95 | 0.7 | 1.0 |

# Results : Adult dataset




Distance




Execution time

# Impact on LORE and Anchors

## Benefits:

- More coherent rules: grounded in realistic feature relationships
- Reduced search space:  pattern-guided perturbations converge faster
- Higher precision : rules validated within realistic neighborhoods

## Alice's Loan rejection

Income=$52k, CreditScore=645, DebtRatio=0.42, LoanAmount = 28000

## LORE with genetic algorithm

{CreditScore <= 712.39, Income <= 65554.80} |

"Increase Income to $85k AND DebtRatio to 0.5 AND reduce loanAmount to 27000" (contradictory)

## LORE with Gradual patterns

{CreditScore <= 680, Income <= 55000} |

"Increase Income to $58k AND reduce DebtRatio to 0.35"

# Discussion

## Advantages:

- Deterministic explanations which ensure stability
- faster computation
- Realistic neighborhoods - respects data structure

## Limitations:

- Currently focused on numerical tabular data
- Depends on existence of meaningful gradual patterns

# Conclusion and perspectives

**Summary:**

- Introduced gradual pattern-based neighborhood generation for XAI
- Achieved perfect stability (Jaccard = 1.0) with speedup
- Produced more realistic, trustworthy explanations

**Future Work:**

- Extend this work on time-series data

# References


- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2016. "'Why Should I Trust You?': Explaining the Predictions of Any Classifier." In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–44. San Francisco California USA: ACM.
- Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17). Curran Associates Inc., Red Hook, NY, USA, 4768–4777.
- Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. 2018. "Anchors: High-Precision Model-Agnostic Explanations." In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, 1527–35. AAAI'18/IAAI'18/EAAI'18. New Orleans, Louisiana, USA: AAAI Press.
- Vedantam, Devi Parikh, and Dhruv Batra. 2020. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." International Journal of Computer Vision 128 (2): 336–59. https://doi.org/10.1007/s11263-019-01228-7.
- Di-Jorio, Lisa, Anne Laurent, and Maguelonne Teisseire. 2009. "Mining Frequent Gradual Itemsets from Large Databases." In Advances in Intelligent Data Analysis VIII, edited by Niall M. Adams, Céline Robardet, Arno Siebes, and Jean-François Boulicaut, 297–308. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-03915-7_26.
- Negrevergne, Benjamin, Alexandre Termier, Marie-Christine Rousset, and Jean-François Méhaut. 2014. "ParaMiner: A Generic Pattern Mining Algorithm for Multi-Core Architectures." Data Mining and Knowledge Discovery 28 (3): 593–633. https://doi.org/10.1007/s10618-013-0313-2.
- Laurent, Anne, Marie-Jeanne Lesot, and Maria Rifqi. 2015. "Mining Emerging Gradual Patterns." In IFSA-EUSFLAT: International Fuzzy Systems Association - European Society for Fuzzy Logic and Technology. Gijon, Spain. https://doi.org/10.2991/ifsa-eusflat-15.2015.234.
- Uno, Takeaki, Masashi Kiyomi, and Hiroki Arimura. n.d. "LCM Ver. 2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets."
- Come, Aymeric, and Jerry Lonlac. 2021. "Extracting Frequent (Closed) Seasonal Gradual Patterns Using Closed Itemset Mining." In 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), 1442–48. Washington, DC, USA: IEEE. https://doi.org/10.1109/ICTAI52525.2021.00229.

"Explainable AI reveals how models arrive at their predictions, but it does not uncover the true causal mechanisms in the world."

# Thanks

# State of the art: GRITE
(Di-Jorio et al., 2009)

**Gradual Pattern Extraction Algorithm:**

**1.Initialization**

- Construct the binary order matrices for each attribute (i, ↑) and (i, ↓)
- gradPattern[1] = {gradual item}
- k = 2

**2.Main loop**

As long as gradPattern[k-1] is not empty:

- Generate candidates: candidates = Union(gradPattern[k-1])
- Validate candidates:gradPattern[k] = validate(candidates,minSup)
- Increment k: k += 1

M1 AND M2

**3.Final result**

Return the union of allgradPattern: return ∪gradPattern

- Construct a directed graph from the matrix M
- Calculate the length of all paths, the maximum length is the absolute frequency.

# State of the art:Paraminer

(Negrevergneet al., 2014)

- **Gradual Pattern Extraction Algorithm**:
  - Compute the closure of the empty pattern ⊥:
  - *P* ← *closure*(⊥,*D*)

  - Initialize the exclusion list EL←∅.
  - Explore closed patterns from P:
    - C ← enum_clo(P,EL)

  - Return the set of closed patterns C *C*.