UNIVERSITATEA
„ALEXANDRU IOAN CUZA“
din IAŞI

Unlocking the Black Box: Model Explainability in Machine Learning

Oriana Onicescu

Agenda

- Theoretical concepts
  - Interpretability vs Explainability
  - Motivation
  - Interpretable models
- Feature Importance
  - Demo
- LIME
  - Algorithm
  - Demo with numerical data
  - Demo with images

What are interpretability and explainability?

In the context of Machine Learning, we define ***interpretability*** as a *property of models* representing the degree to which a human can understand the cause of a prediction.

Explainability represents the ability to easily present the reasons of a prediction in understandable terms to a human. Explainability is a *stronger term* requiring interpretability and additional context.

Motivation

Fairness: Ensuring that predictions are unbiased and don't implicitly or explicitly *discriminate* against underrepresented groups.

Reliability or Robustness: Ensuring that small changes in the input don't lead to large changes in the prediction. Preventing *hacking* or adversarial attacks.

Causality: Ensure that only *causal relationships* are picked up. Assisting development by understanding errors.

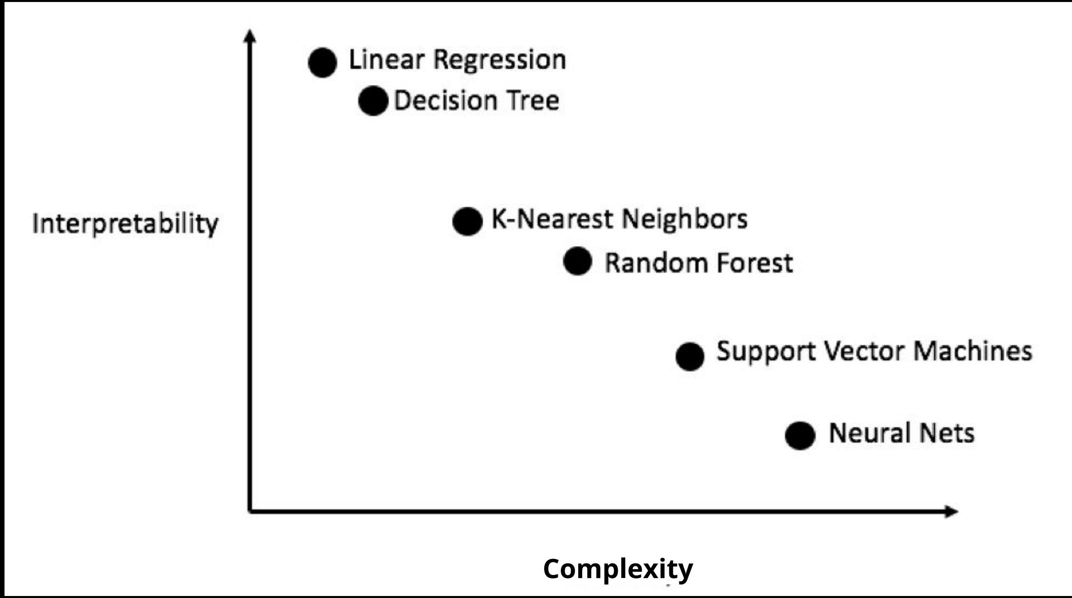
Trust: Humans *prefer* to use a system that explains its decisions compared to a black box.

Regulatory Compliance: Many *domains* require explanations (e.g., finance, legal, healthcare)

"Does your car have any idea why my car pulled it over?"

PAUL NOTH

Interpretable and uninterpretable models



Explainability and interpretability both aim to make AI models more understandable: Interpretability focuses on how straightforward it is to understand a model's workings, explainability goes further by describing why a model made a specific decision or prediction.

Interpretable models

Linear Regression

The learned relationships are linear:

$$y = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p + \epsilon$$

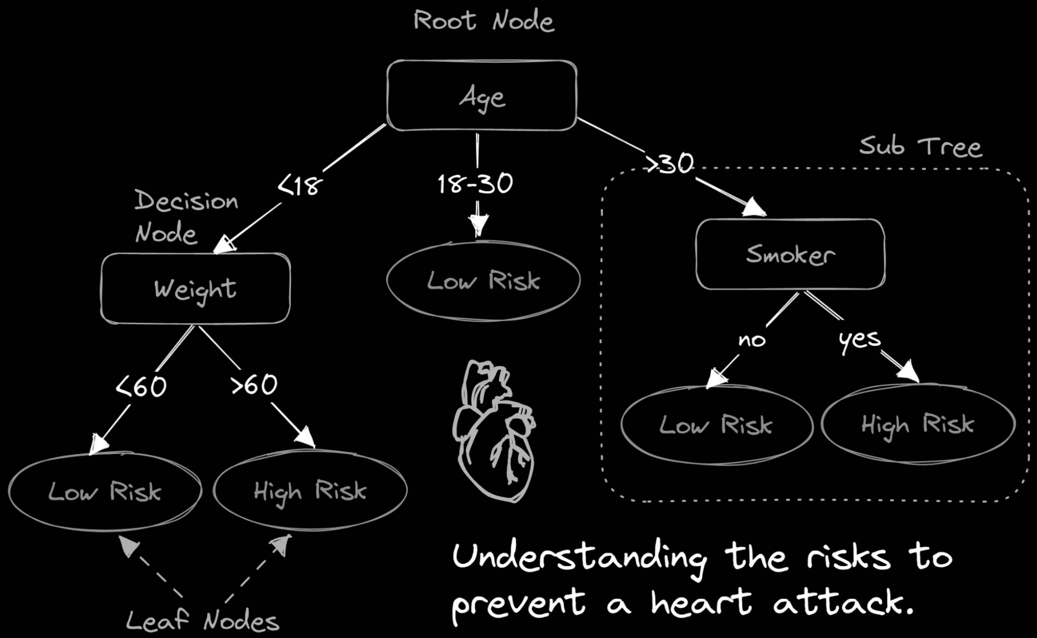
Interpretation

Different types of variables come with different types of interpretation:

Numerical features: 1 unit change = change in output equal with the weight
Binary feature: category change = change in output equal with the weight

Decision Tree

Tree based models split the data multiple times according to certain cutoff values in the features.



<u>Interpretation</u>

For each feature it is measured how much it reduced the Gini index compared to the parent node.

Feature importance

What is feature importance?

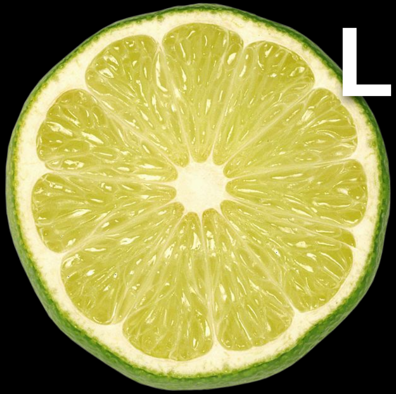
Feature importance highlights which features passed into a model have a higher degree of *impact* for generating a prediction than others.

There are various ways to calculate feature importance, such as:

1. Coefficient based feature importance
2. Permutation based feature importance
3. **Tree feature importance** (the importance scores are calculated based on the reduction in the criterion used to select split points like Gini or entropy)
4. SHAP

DEMO

LIME



LIME (Local interpretable model-agnostic explanations)

LIME focuses on training local surrogate models to explain individual predictions.

Explaining a prediction represents presenting textual or visual artifacts that provide qualitative understanding of the relationship between the input variables and the model's prediction.

$$\text{explanation}(\mathbf{x}) = \arg\min_{g \in G} L(\hat{f}, g, \pi_{\mathbf{x}}) + \Omega(g)$$

The Explanation Model is a simplified model that closely matches the original model's predictions in a neighborhood.

Algorithm

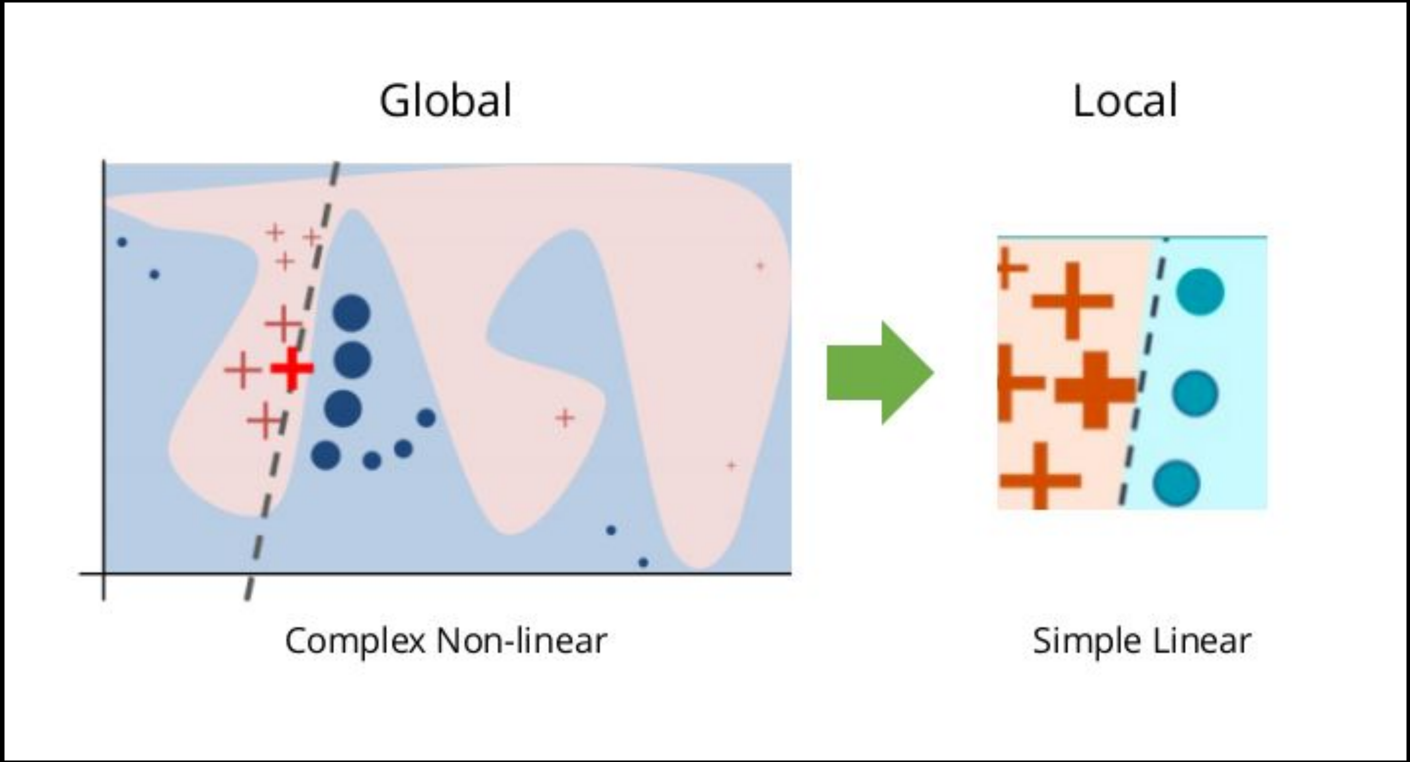
Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

```
  𝒵 ← {}
  for i ∈ {1, 2, 3, ..., N} do
      z'_i ← sample_around(x')
      𝒵 ← 𝒵 ∪ ⟨z'_i, f(z_i), π_x(z_i)⟩
  end for
  w ← K-Lasso(𝒵, K)   ▷ with z'_i as features, f(z) as target
  return w
```

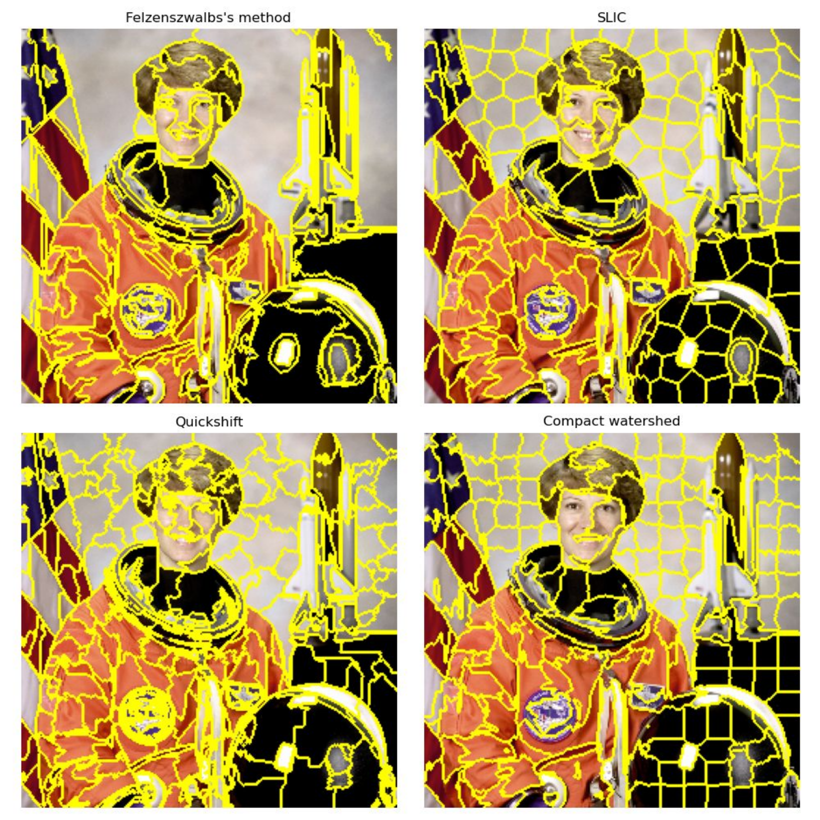
1. *Select your instance* of interest for which you want to have an explanation of its black box prediction.
2. *Perturb* your dataset and get the black box predictions for these new points.
3. Weight the new samples according to their *proximity* to the instance of interest.
4. *Train* a weighted, interpretable model on the dataset with the variations.
5. ***Explain*** the prediction by ***interpreting*** the local model.

Local versus global overview



DEMO

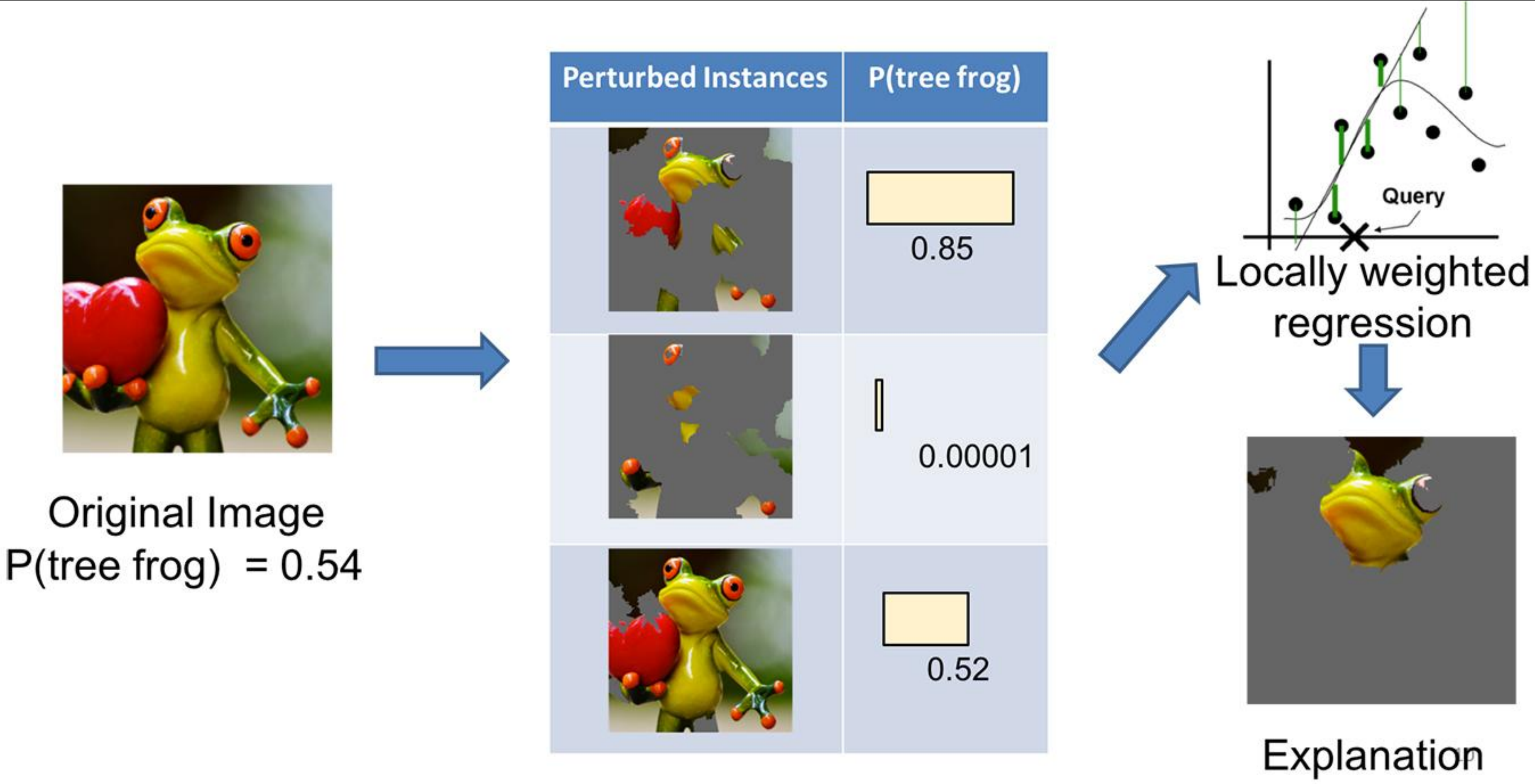
LIME for Computer Vision models - Image segmentation



1. Segments the image using superpixel from opencv
2. Build a linear model based on prediction scores against segments

Example:

- Felzenszwalb number of segments: 194
- SLIC number of segments: 196
- Quickshift number of segments: 695
- Compact watershed number of segments: 250


Original Image
P(tree frog) = 0.54
Perturbed Instances
P(tree frog)
0.85
0.00001
0.52
Query
Locally weighted regression
Explanation

DEMO

LIME for model debugging

(a) Husky classified as wolf (b) Explanation

Figure 11: Raw data and explanation of a bad model's prediction in the "Husky vs Wolf" task.

	Before	After
Trusted the bad model	10 out of 27	3 out of 27
Snow as a potential feature	12 out of 27	25 out of 27

Table 2: "Husky vs Wolf" experiment results.

Food for Thought: SHAP (SHapley Additive exPlanations)

Key Concepts:
- Based on game theory (Shapley values)
- Assigns contribution values to each feature
- Provides both local and global explanations
- Theoretically grounded with nice properties

Advantages over LIME:
1. Consistent attribution values
2. Stronger theoretical guarantees
3. Multiple visualization types
4. Better handling of feature interactions

Types of SHAP Explanations:
- Force plots (local)
- Summary plots (global)
- Dependence plots
- Decision plots

Consider exploring SHAP for:
- More consistent explanations
- Different visualization options
- Understanding feature interactions
- Cases where theoretical guarantees are important

Let’s start peeking into the AI black box!