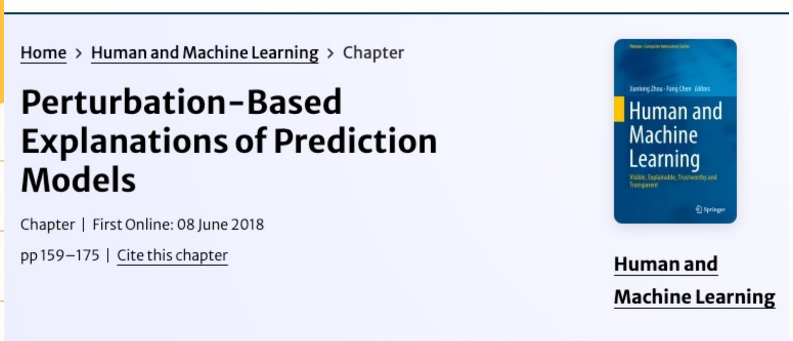
Perturbation-Based Explanations of Prediction Models

Marko Robnik-Šikonja & Marko Bohanec

Book Chapter in Human and Machine Learning
Citations: 243



Home > Human and Machine Learning > Chapter

Perturbation-Based Explanations of Prediction Models

Chapter | First Online: 08 June 2018

pp 159–175 | Cite this chapter

Jianlong Zhou · Fang Chen Editors
Human and Machine Learning
Visible, Explainable, Trustworthy and Transparent
Springer

Human and Machine Learning

Presented by: Abdullah Mamun
Date: 5/21/2025

Contents

1. Introduction to Perturbation-Based Explanations
2. Taxonomy and Properties of Explanations
3. Challenges of Interpreting Black-Box Models
4. Perturbation-Based Explanation Methods Overview
5. The EXPLAIN Method
6. The IME Method
7. The LIME Method
8. Presenting Explanations and Visualization
9. Business Application: B2B Sales Forecasting
10. Conclusion and Takeaways

Introduction to Perturbation-Based Explanations

Why it Matters:

•Machine learning powers entertainment, medicine, and finance.
•Increasing reliance → Need for **transparency** and **trust**.

The Black Box Problem:
•Models like neural networks, SVMs, and random forests lack explainability.

Explanation Techniques:
Internal Process-Based: Leverages model internals.
Perturbation-Based: Agnostic methods altering inputs to observe output changes.

Focus of This Chapter:
•Key methods: EXPLAIN, IME, LIME.
•Applications: Business, trust, and decision-making.

Taxonomy and Properties of Explanations

Two Primary Explanation Types:

Instance Explanations:

- Impact of input features on individual predictions.

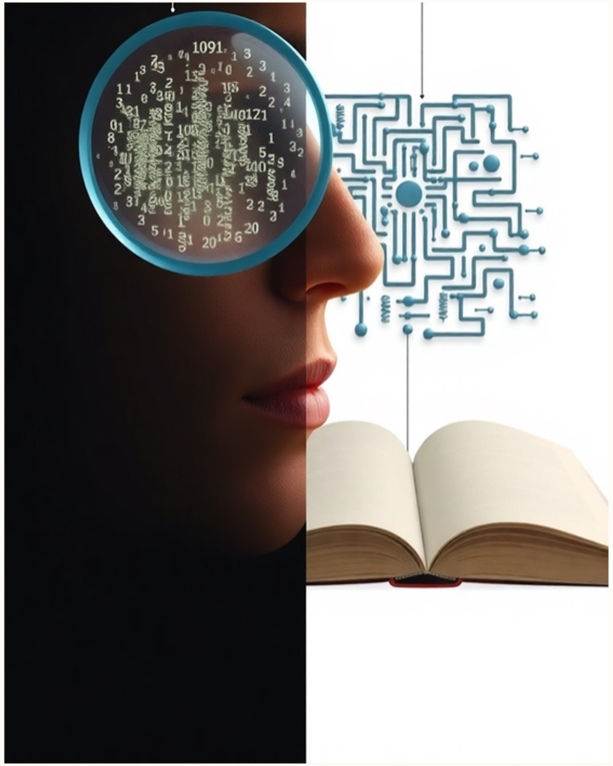
Model Explanations:

- Aggregated insights revealing overall feature influence.

Evaluation Criteria for Explanation Methods:

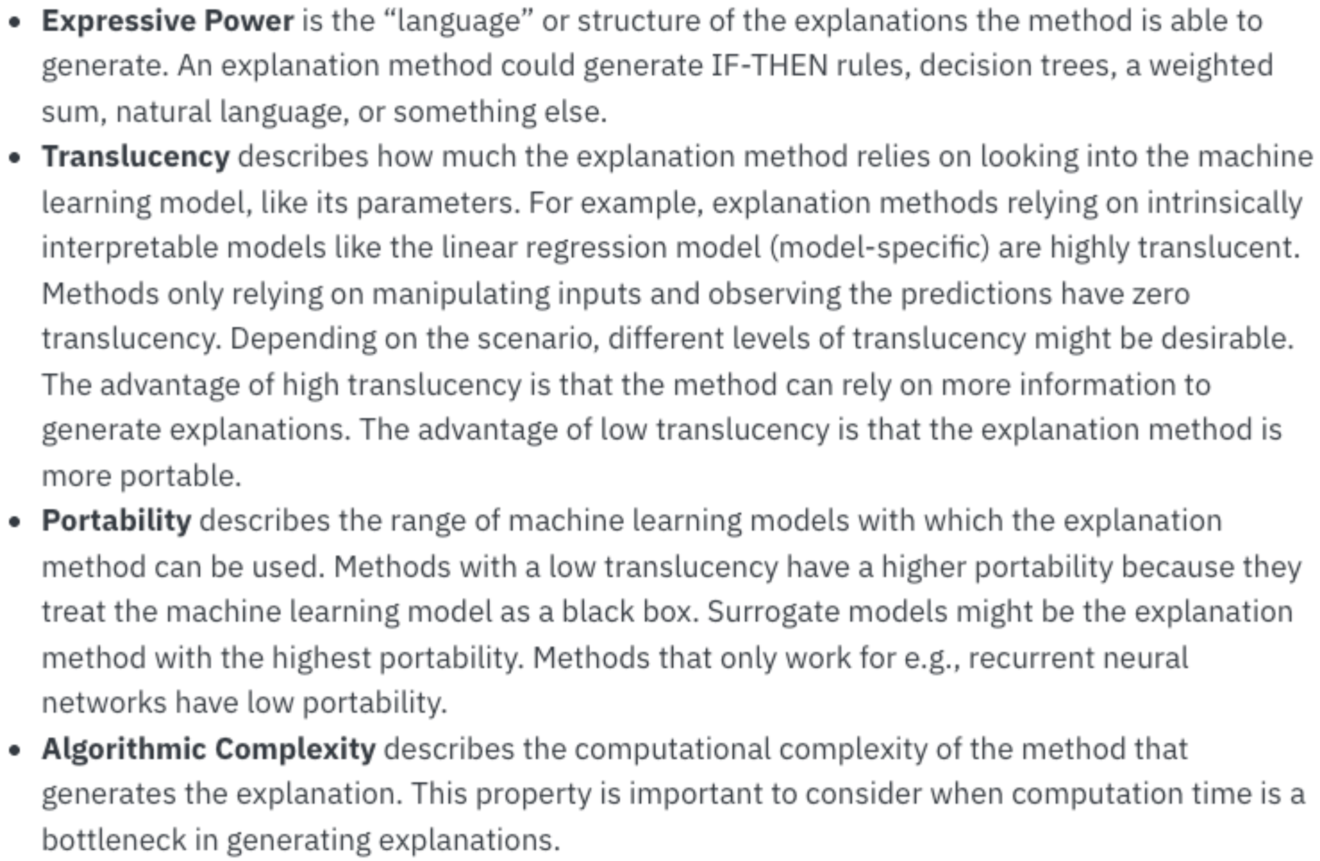
Expressive Power: Logic or form of explanations.
Translucency**:** Degree of model inspection.
Portability**:** Applicability across models.
Algorithmic Complexity**:** Resource efficiency.

Quality Attributes of Explanations:
Accuracy, fidelity, consistency, stability.
Comprehensibility, certainty, importance.
Novelty and representativeness.



Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Properties of an explanation



- **Expressive Power** is the “language” or structure of the explanations the method is able to generate. An explanation method could generate IF-THEN rules, decision trees, a weighted sum, natural language, or something else.
- **Translucency** describes how much the explanation method relies on looking into the machine learning model, like its parameters. For example, explanation methods relying on intrinsically interpretable models like the linear regression model (model-specific) are highly translucent. Methods only relying on manipulating inputs and observing the predictions have zero translucency. Depending on the scenario, different levels of translucency might be desirable. The advantage of high translucency is that the method can rely on more information to generate explanations. The advantage of low translucency is that the explanation method is more portable.
- **Portability** describes the range of machine learning models with which the explanation method can be used. Methods with a low translucency have a higher portability because they treat the machine learning model as a black box. Surrogate models might be the explanation method with the highest portability. Methods that only work for e.g., recurrent neural networks have low portability.
- **Algorithmic Complexity** describes the computational complexity of the method that generates the explanation. This property is important to consider when computation time is a bottleneck in generating explanations.

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Properties of an individual explanation

Accuracy: Measures how well the explanation predicts unseen data. High accuracy is essential if the explanation is used in place of the model. When explaining the model's behavior, fidelity can suffice even with low accuracy.

Fidelity: Reflects how closely the explanation matches the model's predictions. High fidelity is crucial for reliable explanations, especially for local fidelity methods like Shapley Values or surrogate models.

Consistency: Compares explanations across models trained on the same task. High consistency is desirable when models rely on similar features but less so if models achieve similar predictions through different features (Rashomon Effect).

High Fidelity:
A neural network predicts a loan approval. An explanation tool (e.g., LIME) says:

High **income** (+0.8) and **credit score** (+0.5) led to approval.

Small changes in income or credit score near the original inputs still give similar predictions.

Low Fidelity:
The same model predicts loan approval, but the explanation wrongly highlights **debt** (+0.7) as a strong positive factor.

Testing shows changes in debt don’t affect predictions as the explanation suggests

Properties of an individual explanation

Stability: Evaluates how explanations vary for similar data points within the same model. High stability ensures that small input changes don’t cause drastic explanation shifts unless predictions change significantly.

Comprehensibility: Gauges how easily humans understand the explanation. This depends on explanation size, simplicity, and audience familiarity with the features.

Certainty: Indicates how well the explanation reflects model confidence in predictions, as well as its own uncertainty. Essential for understanding reliability.

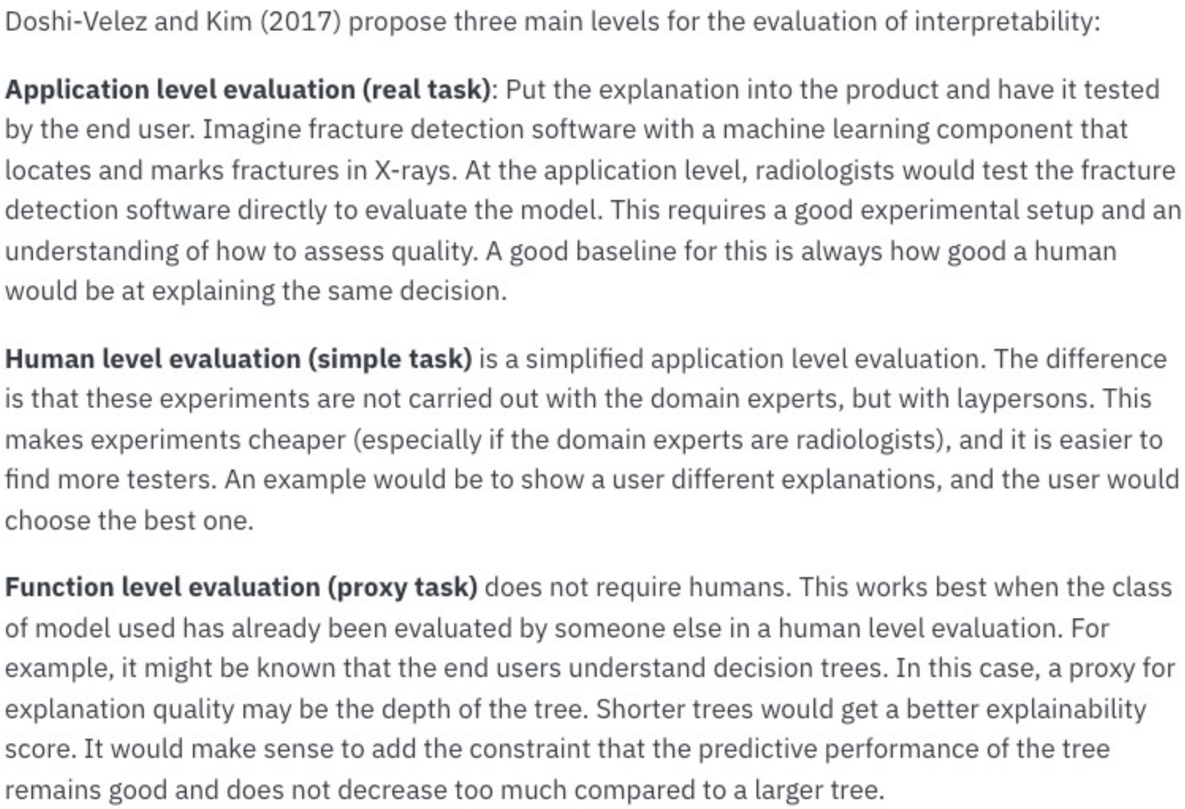
Degree of Importance: Clarifies the relative importance of features or conditions in the explanation, such as which features contributed most to a decision.

Novelty: Highlights if an instance lies outside the training data distribution, signaling potential inaccuracy in both model and explanation.

Representativeness: Shows how many instances the explanation applies to, from individual predictions (e.g., Shapley Values) to entire models (e.g., linear weights).

Levels of Evaluation

- Application level
- Human level
- Function level



Doshi-Velez and Kim (2017) propose three main levels for the evaluation of interpretability:

Application level evaluation (real task): Put the explanation into the product and have it tested by the end user. Imagine fracture detection software with a machine learning component that locates and marks fractures in X-rays. At the application level, radiologists would test the fracture detection software directly to evaluate the model. This requires a good experimental setup and an understanding of how to assess quality. A good baseline for this is always how good a human would be at explaining the same decision.

Human level evaluation (simple task) is a simplified application level evaluation. The difference is that these experiments are not carried out with the domain experts, but with laypersons. This makes experiments cheaper (especially if the domain experts are radiologists), and it is easier to find more testers. An example would be to show a user different explanations, and the user would choose the best one.

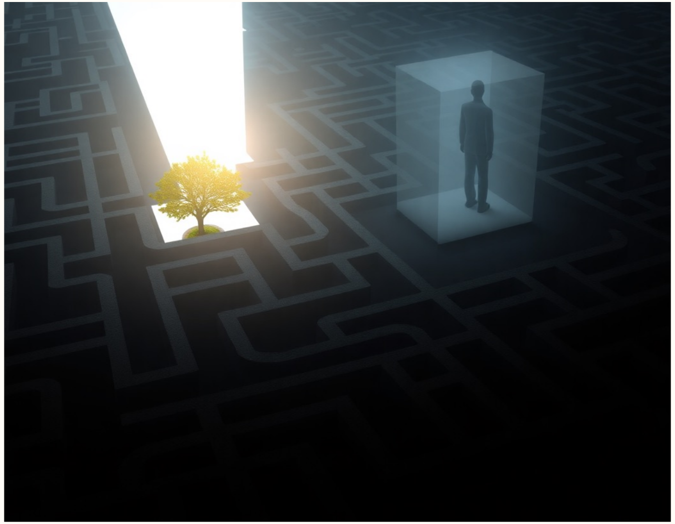
Function level evaluation (proxy task) does not require humans. This works best when the class of model used has already been evaluated by someone else in a human level evaluation. For example, it might be known that the end users understand decision trees. In this case, a proxy for explanation quality may be the depth of the tree. Shorter trees would get a better explainability score. It would make sense to add the constraint that the predictive performance of the tree remains good and does not decrease too much compared to a larger tree.

Doshi-Velez, F., & Kim, B. (2018). Considerations for evaluation and generalization in interpretable machine learning. *Explainable and interpretable models in computer vision and machine learning*, 3-17.

Challenges of Interpreting Black-Box Models

Difficulty in Non-Symbolic Models

Transparent explanations are inherent in symbolic models like decision trees but absent in non-symbolic models (SVMs, ANNs, ensembles), requiring separate explanation methodologies.



Trade-off Between Transparency and Accuracy

Highly accurate models are often difficult to interpret, often requiring a compromise between interpretability and predictive power or supplementing with explanation methods.

Perturbation-Based Explanation Methods Overview

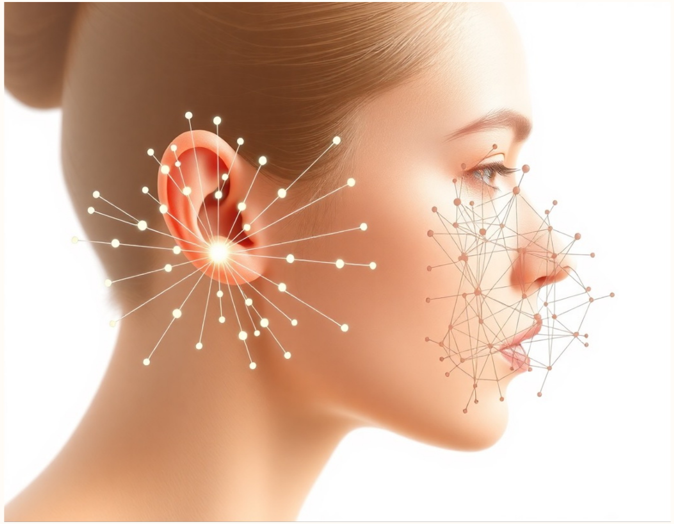
General Approach

Perturbation-based methods systematically modify the inputs of any predictive model, observing the effect on predictions to estimate feature importance. This process allows application across model types and supports both explanations and comparative model analysis.

Relationship to Other Analysis Methods

These methods are related to sensitivity and uncertainty analysis, as well as techniques like inverse classification and gradient-based attribution. Their generality makes them useful in a range of domains and for diverse data types, including text and streaming data.

The EXPLAIN Method



Principle and Computation

EXPLAIN determines the impact of each input feature by simulating the absence of that feature and measuring the effect on the model's prediction probability. Larger output changes signify greater feature importance, and the approach can distinguish between features that support or oppose a given prediction.

Limitations

The method only considers one feature at a time and cannot detect higher-order dependencies, such as interactions or feature redundancies, potentially overlooking complex relationships in data.

The IME Method

All-Subsets Approach

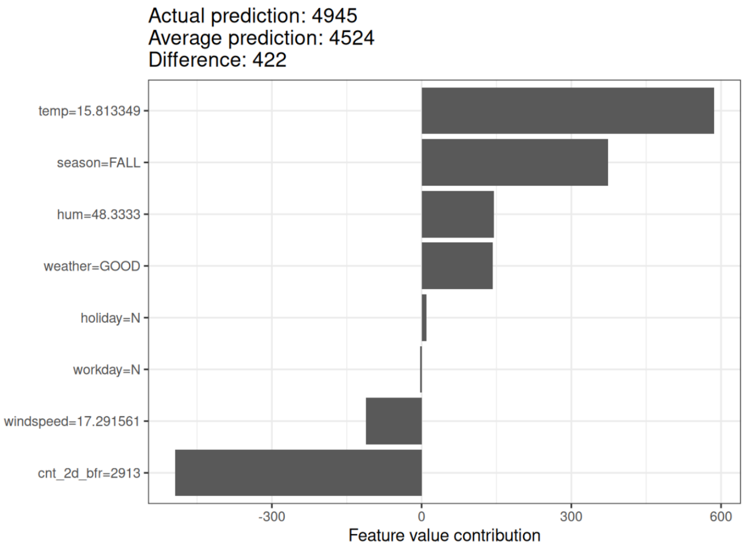
IME (Interaction-aware Model Explanation) overcomes the limitations of one-at-a-time perturbations by assessing all possible subsets of feature values.

It attributes contributions based on Shapley values from cooperative game theory, fairly reflecting the interaction and importance of each feature.

Computational Considerations

Computation involves exponential complexity (2^a, where a is the number of features), but efficient sampling-based approximations are used.

IME provides theoretical guarantees of fair feature contribution assessment but is limited to problems with fewer features due to computational load.



.6: Shapley values for day 285 of the bike data.

Explain vs IME (See Shapley below)

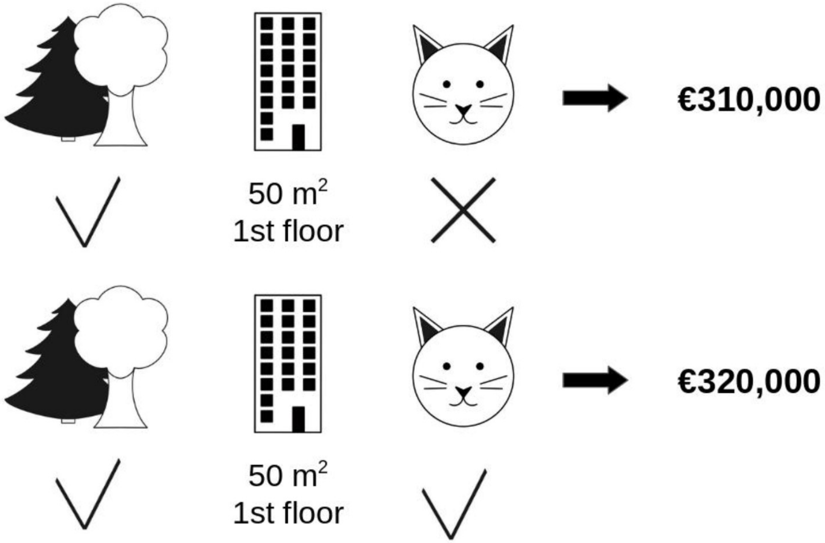
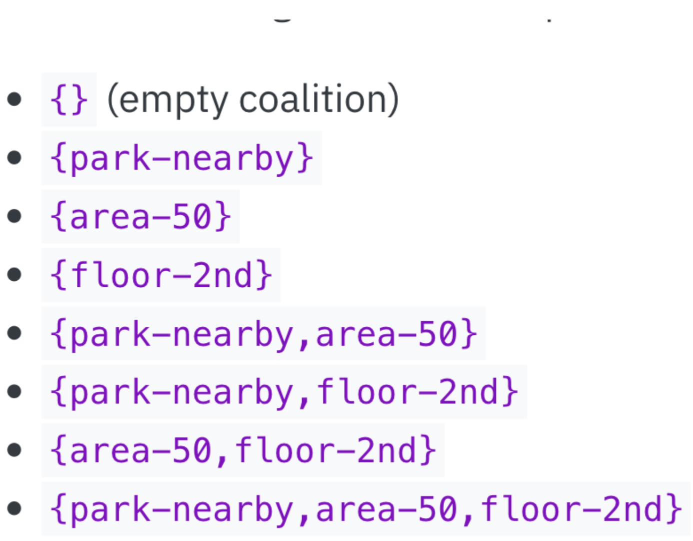


Figure 17.2: One sample repetition to estimate the contribution of `cat-banned` to the prediction when added to the coalition of `park-nearby` and `area-50`.

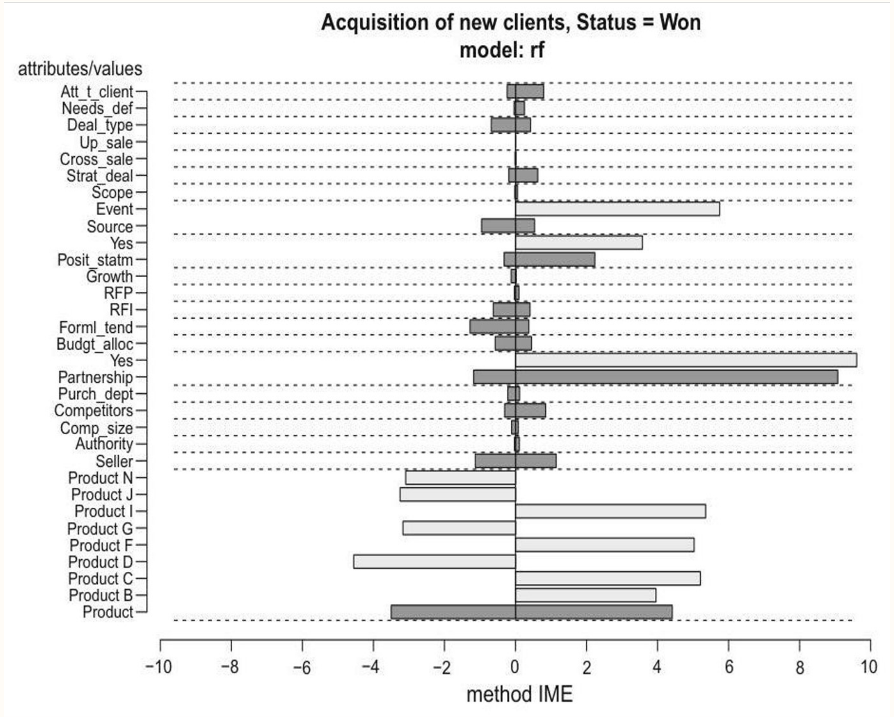


- `{}` (empty coalition)
- `{park-nearby}`
- `{area-50}`
- `{floor-2nd}`
- `{park-nearby,area-50}`
- `{park-nearby,floor-2nd}`
- `{area-50,floor-2nd}`
- `{park-nearby,area-50,floor-2nd}`

Presenting Explanations and Visualization

Visualization Techniques

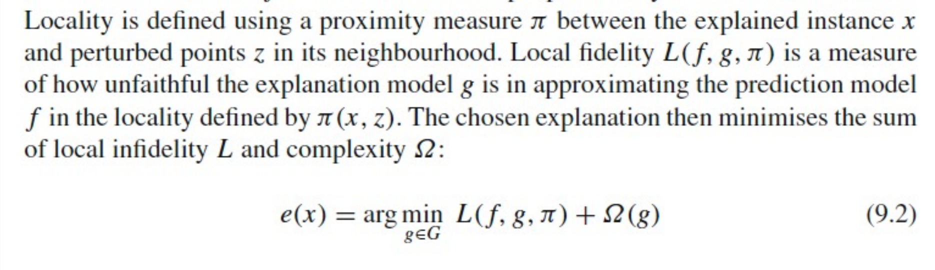
EXPLAIN and IME explanations are visualized via tools like quasi-nomograms and bar charts displaying positive and negative contributions of each input variable, both at the instance and model level. This format enables users to intuitively assess how different features drive predictions.



Applications

Visualization supports both individual predictions and global model understanding, enhancing transparency and aiding expert evaluation. Thresholds and user controls help manage complexity in high-dimensional settings.

The LIME Method



Locality is defined using a proximity measure π between the explained instance x and perturbed points z in its neighbourhood. Local fidelity $L(f, g, \pi)$ is a measure of how unfaithful the explanation model g is in approximating the prediction model f in the locality defined by $\pi(x, z)$. The chosen explanation then minimises the sum of local infidelity L and complexity Ω:

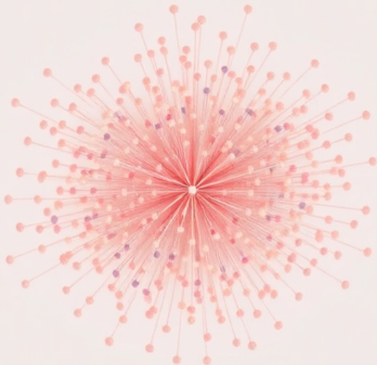
$$e(x) = \arg\min_{g \in G} L(f, g, \pi) + \Omega(g) \tag{9.2}$$

Local Interpretable Model-Agnostic Explanations

LIME (Local Interpretable Model-agnostic Explanations) generates explanations by sampling and perturbing the neighborhood of the instance to be explained.

It fits simple, interpretable models (like linear models or decision trees) to approximate the complex model locally.

LIME frames explanation as an optimization problem, trading off fidelity to the original model with complexity for interpretability.

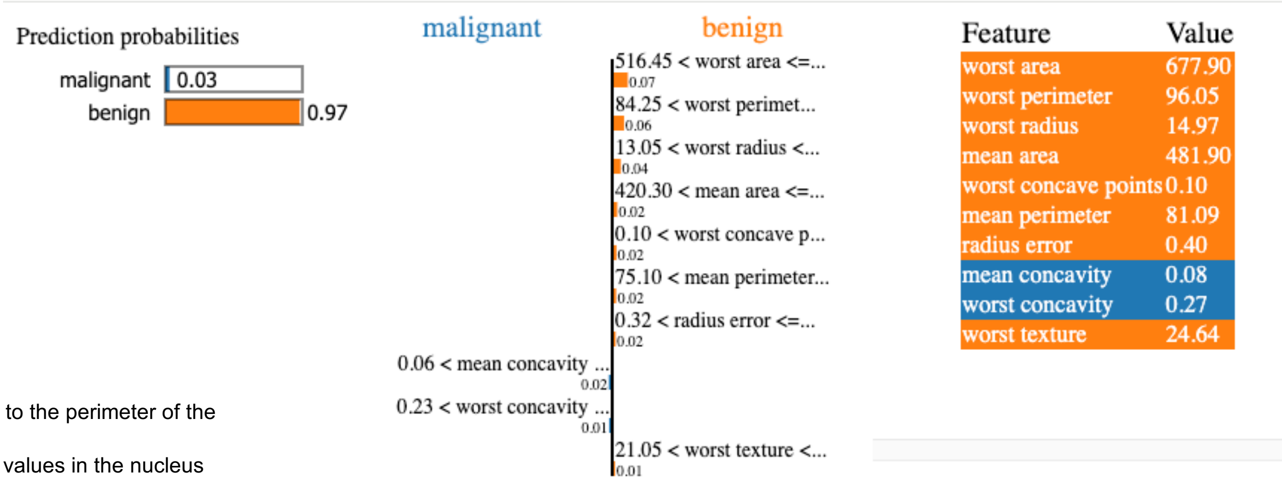


Strengths and Weaknesses

LIME efficiently handles high-dimensional and large data sets, aiding real-time explanations.

However, it offers no guarantees of explanation faithfulness or stability, and can struggle in high-dimensional spaces or with complex feature interactions.

LIME – with breast cancer.



Prediction probabilities

malignant	0.03
benign	0.97

malignant
benign
516.45 < worst area <=...
0.07
84.25 < worst perimet...
0.06
13.05 < worst radius <...
0.04
420.30 < mean area <=...
0.02
0.10 < worst concave p...
0.02
75.10 < mean perimeter...
0.02
0.32 < radius error <=...
0.02
0.06 < mean concavity ...
0.02
0.23 < worst concavity ...
0.01
21.05 < worst texture <...
0.01

Feature	Value
worst area	677.90
worst perimeter	96.05
worst radius	14.97
mean area	481.90
worst concave points	0.10
mean perimeter	81.09
radius error	0.40
mean concavity	0.08
worst concavity	0.27
worst texture	24.64

- Mean Radius: Average distance from the center to the perimeter of the nucleus.
- Mean Texture: Standard deviation of gray-scale values in the nucleus image.
- Mean Perimeter: Average perimeter of the nucleus.
- Mean Area: Average area of the nucleus.
- Mean Smoothness: Average of local variations in radius lengths (smoothness of the nucleus boundary).
- Mean Compactness: Average of (perimeter² / area - 1.0), measuring how compact the nucleus is.
- Mean Concavity: Average severity of concave portions of the nucleus contour.
- Mean Concave Points: Average number of concave portions on the nucleus contour.
- Mean Symmetry: Average symmetry of the nucleus (how mirrored it is across its center).
- Mean Fractal Dimension: Average "coastline approximation" of the nucleus shape complexity.
- Radius SE: Standard error of the nucleus radius.

Texture SE: Standard error of the gray-scale values in the nucleus.
Perimeter SE: Standard error of the nucleus perimeter.
Area SE: Standard error of the nucleus area.
Smoothness SE: Standard error of the local variations in radius lengths.
Compactness SE: Standard error of the compactness measure.
Concavity SE: Standard error of the concavity measure.
Concave Points SE: Standard error of the number of concave points.
Symmetry SE: Standard error of the symmetry measure.
Fractal Dimension SE: Standard error of the fractal dimension.
Worst Radius: Largest (worst) radius of the nucleus in the sample.
Worst Texture: Largest (worst) standard deviation of gray-scale values.
Worst Perimeter: Largest (worst) perimeter of the nucleus.
Worst Area: Largest (worst) area of the nucleus.
Worst Smoothness: Largest (worst) smoothness value.
Worst Compactness: Largest (worst) compactness value.
Worst Concavity: Largest (worst) concavity value.
Worst Concave Points: Largest (worst) number of concave points.
Worst Symmetry: Largest (worst) symmetry value.
Worst Fractal Dimension: Largest (worst) fractal dimension value.

Conclusion and Takeaways

Advantages of Perturbation-Based Explanations

EXPLAIN, IME, and LIME bring transparency to black-box models by revealing input variable contributions. They offer model-agnostic explanations, support visual insights, and enhance trust in automated decisions. EXPLAIN and LIME are efficient for large problems, while IME provides fair, game-theoretically justified contributions.

Limitations and Future Directions

Each method has trade-offs: EXPLAIN misses feature interactions, IME can be computationally intensive, and LIME may lack faithfulness. Integration of these approaches and improved visualization are important future research directions. Practical adoption is facilitated by open-source tools.

Arizona State University

Thank You!

https://abdullah-mamun.com/

a.mamun@asu.edu