Diogo Ribeiro

ESMAD - Escola Superior de Média Arte e Design

Lead Data Scientist, Mysense.ai

dfr@esmad.ipp.pt 0009-0001-2022-7072

November 16, 2025

Outline

The Interpretability Crisis

Modern ML models are powerful but opaque:

High Performance:

- Deep neural networks
- Ensemble methods
- Complex feature interactions

Low Interpretability:

- Millions of parameters
- Non-linear transformations
- Difficult to explain

Why Does Interpretability Matter?

- **Trust:** Users need to understand decisions
- **Debugging:** Identify and fix model errors
- **Regulation:** GDPR "right to explanation"
- **Fairness:** Detect and mitigate bias
- **Scientific insight:** Understand underlying phenomena

Interpretability vs. Explainability

Interpretability

The degree to which a human can **understand** the cause of a decision.
Example: Linear regression with few features is inherently interpretable.

Explainability

The degree to which a human can **consistently predict** the model's result.
Example: LIME provides explanations for black-box models.

Key distinction:

- **Interpretability:** Intrinsic property of the model
- **Explainability:** Post-hoc analysis of model behavior

The Accuracy-Interpretability Tradeoff

Traditional view: Must choose between accuracy and interpretability

Inherently Interpretable:

- Linear/logistic regression
- Decision trees (shallow)
- Generalized additive models (GAM)
- Rule-based systems

High interpretability, Lower accuracy

Black Box Models:

- Deep neural networks
- Random forests (large)
- Gradient boosting
- SVM with RBF kernel

Low interpretability, Higher accuracy

Modern Approach

Use explanation methods to make black-box models transparent!

Scope of Explanations

Global Explanations

Describe the **overall** behavior of the model across all predictions.
Question: How does the model work in general?

Local Explanations

Explain a **specific** prediction for a single instance.
Question: Why did the model make this particular prediction?

Example: Credit Scoring

Global: "Income is the most important feature overall"
Local: "Applicant X was denied because their income ($30K) is below the threshold

Global Explanation Methods

1. **Feature Importance:**
   - Ranking features by contribution to predictions
   - Methods: Permutation importance, SHAP values (aggregated), drop-column
2. **Partial Dependence Plots (PDP):**
   - Show marginal effect of features on predictions
   - $\text{PDP}(x_s) = \mathbb{E}_{x_c}[\hat{f}(x_s, x_c)]$
   - Averaged over all other features
3. **Accumulated Local Effects (ALE):**
   - Like PDP but handles correlated features better
   - Based on conditional distributions
4. **Model Distillation:**
   - Train simpler model to mimic complex model
   - Interpretable surrogate (decision tree, linear model)

Local Explanation Methods

1. **Individual Conditional Expectation (ICE):**
   - Like PDP but for individual instances
   - Shows heterogeneous effects
2. **LIME (Local Interpretable Model-agnostic Explanations):**
   - Fit simple model locally around instance
   - Perturb input and observe model response
3. **SHAP (SHapley Additive exPlanations):**
   - Game-theoretic approach
   - Distributes prediction among features fairly
4. **Counterfactual Explanations:**
   - "What would need to change for different prediction?"
   - Actionable insights

Permutation Feature Importance

Permutation Importance

Measure importance by randomly permuting a feature and observing the change in model performance.

$$\mathrm{FI}_j = \mathrm{Error}(\mathrm{permuted}(X_j)) - \mathrm{Error}(X) \tag{1}$$

Algorithm:

1. Train model on original data
2. For each feature j:
   2.1 Randomly permute values of feature j
   2.2 Compute prediction error
   2.3 Calculate difference from baseline error
3. Rank features by importance

Advantages:

LIME: Local Interpretable Model-agnostic Explanations

Core idea: Approximate complex model locally with interpretable model.

LIME Objective

$$\xi(x) = \arg\min_{g \in \mathcal{G}} \mathcal{L}(f, g, \pi_x) + \Omega(g) \tag{2}$$

where:

- f: Black-box model
- g: Interpretable model (e.g., linear)
- $\mathcal{L}$: Measure of how well g approximates f
- π_x: Locality kernel (weights nearby samples)
- $\Omega(g)$: Complexity penalty

LIME Algorithm

For tabular data:

1. **Sample:** Generate perturbed samples around instance x
   - Create synthetic data by perturbing features
   - Weight samples by proximity to x
2. **Predict:** Get black-box predictions for samples
3. **Fit:** Train interpretable model (linear regression) on weighted samples
4. **Explain:** Use coefficients as feature importance

Example Output

"For this loan application (approved):

- Income ($65K): +0.35 (positive contribution)
- Credit score (720): +0.28
- Debt ratio (0.25): -0.12

"

SHAP: SHapley Additive exPlanations

Based on Shapley values from cooperative game theory.

Shapley Value

Contribution of feature i to prediction:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)] \tag{3}$$

where N is set of all features, S is a subset not containing i.

Properties:

- **Efficiency:** $\sum_i \phi_i = f(x) - f(\emptyset)$ (prediction explained)
- **Symmetry:** Equal features get equal values
- **Dummy:** Irrelevant features get zero value
- **Additivity:** Values sum correctly

SHAP in Practice

Computing exact Shapley values is exponential!

Efficient approximations:

- **TreeSHAP:** For tree-based models (RF, XGBoost, LightGBM)
  - Polynomial time algorithm
  - Exact Shapley values
- **KernelSHAP:** Model-agnostic approximation
  - Weighted linear regression approach
  - Similar to LIME but theoretically grounded
- **DeepSHAP:** For neural networks
  - Combines DeepLIFT with Shapley values
  - Fast approximation

Visualization:

- Waterfall plots (individual predictions)
- Summary plots (feature importance)
- Dependence plots (feature interactions)

LIME vs. SHAP

Property	LIME	SHAP
Theoretical foundation	Heuristic	Game theory
Consistency	No guarantee	Guaranteed
Local accuracy	High	High
Computational cost	Low	Medium-High
Additivity	Not guaranteed	Guaranteed
Model-agnostic	Yes	Yes (KernelSHAP)
Specialized versions	Image, text	Tree, Deep

When to Use?

- **LIME:** Quick explanations, simple use case
- **SHAP:** Rigorous analysis, better properties, worth the computation

Intrinsically Interpretable Models

1. **Linear Models:**
    - Coefficients directly interpretable
    - $\hat{y} = \beta_0 + \sum_{j=1}^{p} \beta_j x_j$
    - Each β_j shows effect of one-unit change in x_j
2. **Decision Trees:**
    - Sequence of if-then rules
    - Easy to visualize and explain
    - But: unstable, high variance
3. **Generalized Additive Models (GAM):**

$$g(\mathbb{E}[y]) = \beta_0 + \sum_{j=1}^{p} f_j(x_j) \tag{4}$$

    - Flexible non-linear effects
    - Each f_j can be plotted separately

1. **Gradient-based Methods:**
   - **Saliency Maps:** $\frac{\partial f(x)}{\partial x_i}$
     - Shows which inputs affect output most
     - Commonly used for images
   - **Integrated Gradients:**

$$\mathrm{IG}_i(x) = (x_i - x_i') \int_{\alpha=0}^{1} \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} d\alpha \tag{5}$$

     - Accumulates gradients along path
     - Satisfies axioms (completeness, sensitivity)
   - **Grad-CAM:** For CNNs
     - Weighted combination of activation maps
     - Highlights important regions in images

Attention Mechanisms

Built-in interpretability in Transformers!

Attention Weights

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

Attention weights α_{ij} show how much position i attends to position j.

Interpretation:

- Visualize attention matrices
- Identify which inputs are important
- Understand model reasoning

Caveat

Algorithmic Fairness

ML models can perpetuate or amplify bias:

- Training data reflects historical discrimination
- Proxy variables encode protected attributes
- Feedback loops reinforce biases

Example: COMPAS

ProPublica (2016) found COMPAS recidivism algorithm:

- Higher false positive rate for Black defendants
- Lower false positive rate for white defendants
- Despite being "race-blind"

Legal and Ethical Implications

Fairness Definitions

Let A be protected attribute (race, gender, etc.)

1. Statistical Parity:

$$P(\hat{Y}=1|A=0)=P(\hat{Y}=1|A=1) \tag{7}$$

Equal positive prediction rates across groups.

2. Equal Opportunity:

$$P(\hat{Y}=1|Y=1,A=0)=P(\hat{Y}=1|Y=1,A=1) \tag{8}$$

Equal true positive rates (equal recall).

3. Equalized Odds:

$$P(\hat{Y}=1|Y=y,A=0)=P(\hat{Y}=1|Y=y,A=1), \quad y \in \{0,1\} \tag{9}$$

Equal TPR and FPR across groups.

4. Calibration:

$$P(Y=1|\hat{P}=p,A=a)=p \quad \forall p,a \tag{10}$$

Predicted probabilities are accurate within groups

Impossibility Results

Fairness Impossibility (Chouldechova, 2017)

Except in degenerate cases, a model cannot simultaneously satisfy:

- Equalized odds (equal TPR and FPR)
- Calibration (accurate probabilities)
- Different base rates across groups

Implication: Must choose which notion of fairness to prioritize!

Trade-offs:

- Fairness metrics often conflict
- No universally "fair" solution
- Context-dependent choices
- Need stakeholder input

Detecting and Mitigating Bias

Detection:

1. Compute fairness metrics for each group
2. Check for disparate impact
3. Analyze feature importance by group
4. Use interpretability tools (SHAP, LIME)

Mitigation strategies:

- **Pre-processing:**
  - Re-weight training data
  - Remove bias from features
- **In-processing:**
  - Add fairness constraints to objective
  - Adversarial debiasing
- **Post-processing:**
  - Adjust prediction thresholds per group
  - Calibration techniques

Python Tools for XAI

Popular libraries:

1. SHAP:

```python
import shap

# For tree models
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test)

# Visualizations
shap.summary_plot(shap_values, X_test)
shap.waterfall_plot(shap_values[0])
shap.dependence_plot("feature_name", shap_values, X_test)

```

2. LIME:

```python
from lime.lime_tabular import LimeTabularExplainer
```

More Tools

3. InterpretML (Microsoft):

```python
from interpret.glassbox import ExplainableBoostingClassifier
from interpret import show

# Train interpretable model
ebm = ExplainableBoostingClassifier()
ebm.fit(X_train, y_train)

# Global explanation
ebm_global = ebm.explain_global()
show(ebm_global)

# Local explanation
ebm_local = ebm.explain_local(X_test, y_test)
show(ebm_local)

```

Best Practices

1. **Choose the right explanation method:**
   - Model type (tree-based, neural network, etc.)
   - Audience (technical vs. non-technical)
   - Purpose (debugging, transparency, fairness)
2. **Validate explanations:**
   - Check consistency across methods
   - Test with synthetic data
   - Compare to domain knowledge
3. **Communicate effectively:**
   - Use visualizations
   - Provide context
   - Avoid over-interpretation
4. **Consider computational cost:**
   - SHAP can be expensive for large datasets

Summary

Key Takeaways:

- Interpretability is crucial for trust, debugging, and fairness
- **Global explanations:** Understand model behavior overall
- **Local explanations:** Explain individual predictions
- **SHAP:** Theoretically sound, widely applicable
- **LIME:** Fast, intuitive, model-agnostic
- **Fairness:** Multiple definitions, inherent trade-offs

The Future of XAI

- Regulatory requirements increasing
- Better integration into ML pipelines
- Standardization of methods and metrics
- Causal explanations beyond correlations

Further Reading

Books:

- Molnar (2022). *Interpretable Machine Learning*
- Barocas, Hardt, Narayanan (2019). *Fairness and Machine Learning*

Key Papers:

- Ribeiro et al. (2016). "Why Should I Trust You? Explaining Predictions" (LIME)
- Lundberg & Lee (2017). "A Unified Approach to Interpreting Model Predictions" (SHAP)
- Doshi-Velez & Kim (2017). "Towards A Rigorous Science of Interpretable ML"
- Chouldechova (2017). "Fair Prediction with Disparate Impact"

Tools:

- SHAP: `https://github.com/slundberg/shap`
- LIME: `https://github.com/marcotcr/lime`
- InterpretML: `https://interpret.ml/`
- Fairlearn: `https://fairlearn.org/`

Acknowledgments

- ESMAD for institutional support
- Mysense.ai for applied AI ethics work
- XAI research community

Generated with LaTeX Beamer

Theme: ESMAD Professional Academic Style

Contact Information

Diogo Ribeiro

ESMAD - Escola Superior de Média Arte e Design
Lead Data Scientist, Mysense.ai

dfr@esmad.ipp.pt

0009-0001-2022-7072

github.com/diogoribeiro7

linkedin.com/in/diogoribeiro7

This presentation is part of the academic materials repository.
Available at: `https://github.com/diogoribeiro7/academic-presentations`