

DDA4210/AIR6002 Advanced Machine Learning

Lecture 12 Explainable AI

Tongxin Li

School of Data Science, CUHK-Shenzhen

Spring 2024

Overview

- 1 Motivation
- 2 Permutation and Occlusion
- 3 LIME
- 4 RISE
- 5 SHAP
- 6 Interpretable ML

1 Motivation

2 Permutation and Occlusion

3 LIME

4 RISE

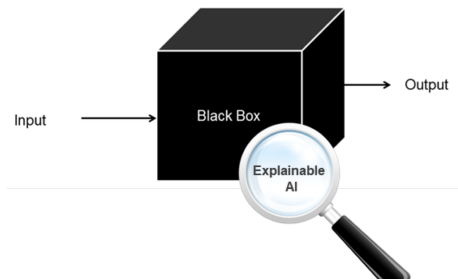
5 SHAP

6 Interpretable ML

Example

- You work at a jail, and your boss asks you to automate potential recidivism risk evaluation. You examine the various model options and select a complex model because it gets the best performance: a gradient boosted tree XGBoost or LightGBM.
- Boss: Nice job, can I ask some questions about how the model works?
 - Which features are most important overall?
 - For the people with high risk ($y = 1$), can you explain why?

**Explain a blackbox
or use an interpretable
model instead**



Motivation

- Explainable AI (XAI), or Interpretable AI, or Explainable Machine Learning (XML), is artificial intelligence (AI) in which humans can understand the reasoning behind decisions or predictions made by the AI.
- XAI algorithms follow three principles.
 - **Transparency**: the processes that extract model parameters from training data and generate labels from testing data can be described and motivated by the approach designer.
 - **Interpretability**: the possibility of comprehending the ML model and presenting the underlying basis for decision-making in a way that is understandable to humans.
 - **Explainability**: a concept that is recognized as important, but a consensus definition is not available. One possibility is: "the collection of features of the interpretable domain, that have contributed for a given example to produce a decision (e.g., classification or regression)".
- Algorithms fulfilling these principles provide a basis for justifying decisions, tracking and thereby verifying them, improving the algorithms, and exploring new facts.

Slide adapted from https://en.wikipedia.org/wiki/Explainable_artificial_intelligence

Motivation

Machine learning (ML) algorithms used in AI can be categorized as **white-box** or **black-box**.



Black box - we do not
know anything



White box - we know
everything

- White-box models provide results that are understandable for experts in the domain.
 - e.g.: linear regression, logistic regression, knn, linear SVM, decision tree, GLMs, GAMs, kmeans, PCA

Motivation

Machine learning (ML) algorithms used in AI can be categorized as **white-box** or **black-box**.



Black box - we do not know anything



White box - we know everything

- White-box models provide results that are understandable for experts in the domain.
 - e.g.: linear regression, logistic regression, knn, linear SVM, decision tree, GLMs, GAMs, kmeans, PCA
- Black-box models are extremely hard to explain and can hardly be understood even by domain experts.
 - e.g.: ensemble learning (XGBoost, LightGBM, etc), kernel methods, neural networks (MLP, CNN, RNN, transformer, GNN)

Definitions

- A model explanation attempts to highlight why a model made a prediction
- A feature importance explanation focuses on each feature's role
 - Feature attribution: each feature x_i receives a score $a_i \in \mathbb{R}$
 - Feature selection: a subset of important features $x_S \in \{x_1, \dots, x_d\}$

Definitions

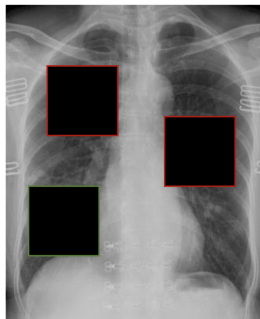
- A model explanation attempts to highlight why a model made a prediction
- A feature importance explanation focuses on each feature's role
 - Feature attribution: each feature x_i receives a score $a_i \in \mathbb{R}$
 - Feature selection: a subset of important features $x_S \in \{x_1, \dots, x_d\}$
- Explanations may relate to an individual prediction (**local**) or a broader model behavior (**global**)
- An explanation algorithm is a method that generates explanations given input data and an ML model

Removal-based Explanations

Idea: to understand a feature's importance, remove it and see how the prediction changes This is the underlying idea behind many popular approaches SHAP, LIME, RISE, etc.

Doctor analogy:

- Suppose we want to understand a doctor's diagnosis.
- We can probe the doctor's reasoning by covering parts of the scan.
- The diagnosis should change when we cover important regions.



In ML, we can analyze models by withholding features.
Challenge: How to design the features to be removed?
Given d features, we have 2^n possible ways to remove!

- 1 Motivation
- 2 Permutation and Occlusion
- 3 LIME
- 4 RISE
- 5 SHAP
- 6 Interpretable ML

Permutation Test

Permutation test (also known as permutation feature importance) is an "old" method introduced for random forests.

- It determines **overall** (global) importance of each input feature via:
 - First, evaluate the model's accuracy using the original data
 - Then, one at a time, corrupt features and record the drop in accuracy
 - Corruption: randomize/permute/shuffling a column of the dataset (corresponding to the feature)
 - Accuracy drop: $a_i = \text{Acc}(\text{original}) - \text{Acc}(x_i \text{ corrupted})$
- * It can be applied to training, validation, or testing datasets.

Height at age 20 (cm)	Height at age 10 (cm)	...	Socks owned at age 10
182	155	...	20
175	147	...	10
...
156	142	...	8
153	130	...	24

Permutation Test

Permutation test (also known as permutation feature importance) is an "old" method introduced for random forests.

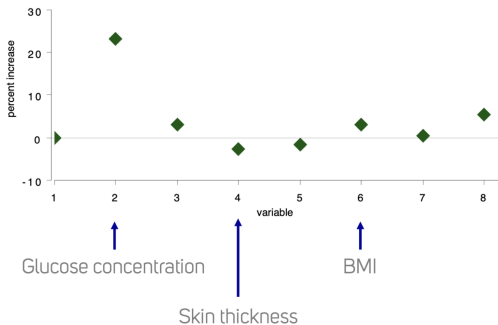
- It determines **overall** (global) importance of each input feature via:
 - First, evaluate the model's accuracy using the original data
 - Then, one at a time, corrupt features and record the drop in accuracy
 - Corruption: randomize/permute/shuffling a column of the dataset (corresponding to the feature)
 - Accuracy drop: $a_i = \text{Acc}(\text{original}) - \text{Acc}(x_i \text{corrupted})$
- * It can be applied to training, validation, or testing datasets.
- More precisely, it computes

$$a_i = \frac{1}{n} \sum_{j=1}^n \ell \left(f \left(x_1^j, \dots, \tilde{x}_i^j, \dots, x_d^j \right), y^j \right) - \frac{1}{n} \sum_{j=1}^n \ell \left(f \left(x_1^j, \dots, x_d^j \right), y^j \right)$$

- $\ell(\cdot, \cdot)$: an arbitrary loss function
- \tilde{x}_i^j : the i -th feature is corrupted

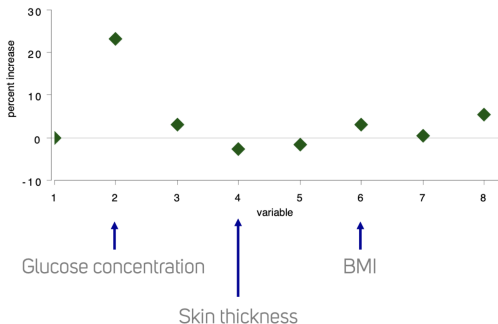
Permutation Test

Example: variable importance in random forest on diabetes data



Permutation Test

Example: variable importance in random forest on diabetes data



Properties of permutation test based explanation

- Works for any model
- Can use with continuous or categorical features
- Fast, easy to implement
- Empirical, no theoretical guarantees
- Problematic for correlated features

Occlusion is an early approach for deep learning models

- Occlusion: block/mask something
- Explain **individual** (local) predictions for image classifiers
- Calculate pixel (or super-pixel) importance

Occlusion process:

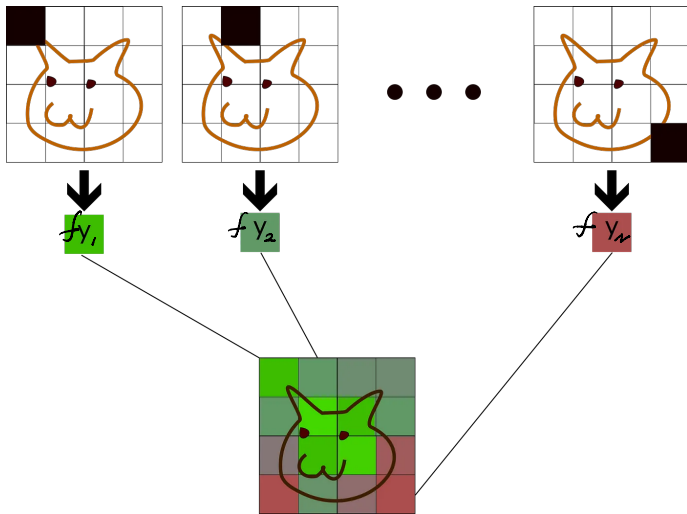
- Make prediction given full image
- Occlude various image regions, record how the prediction changes
 - Occlude by replacing with uninformative (zero) pixels
 - Potentially occlude 2x2, 4x4, etc., or super-pixels
- Mathematically,

$$a_i = f_y(x_1, \dots, x_d) - f_y(x_1, \dots, 0, \dots, x_d)$$

Zeiler & Fergus, "Visualizing and understanding convolutional networks" (2014)

Occlusion

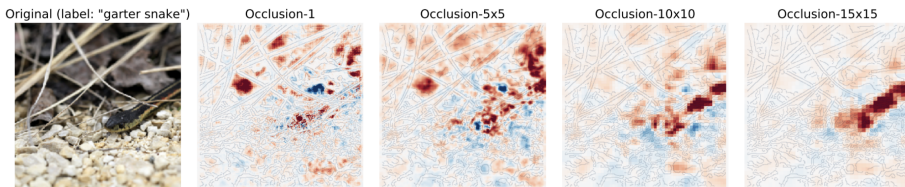
An intuitive example



<https://towardsdatascience.com/inshort-occlusion-analysis-for-explaining-dnns-d0ad3af9aeb6>

Occlusion

Example: attribution generated by occlusion with squared grey patches of different sizes



Properties of occlusion based explanation

- Works with any model, even non-image data
- Moderately fast: $d + 1$ model evaluations to explain each prediction
- Easy to implement
- Empirical, no theoretical guarantees

Ancona et al. Towards better understanding of gradient-based attribution methods for Deep Neural Networks (2018)

Permutation Test vs Occlusion

	Permutation tests	Occlusion
Corrupt input	Randomize features	Set to zero
Observe model change	Change in accuracy	Change in prediction
Calculate impact	Remove single feat.	Remove single feat.

A General Framework

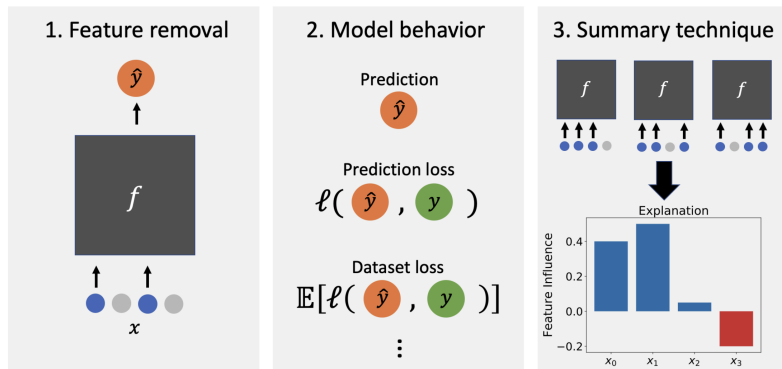


Figure 1: A unified framework for *removal-based explanations*. Each method is determined by three choices: how it removes features, what model behavior it analyzes, and how it summarizes feature influence.

Options for feature removal approach: 1. replace with default values (zero); 2. replace with random values; 3. train separate models with each feature set; 4. use a model that accepts missing features; 5. application-based operations (e.g. blurring for images).

Covert et al., "Explaining by removing: a unified framework for model explanation" (2021)

A General Framework

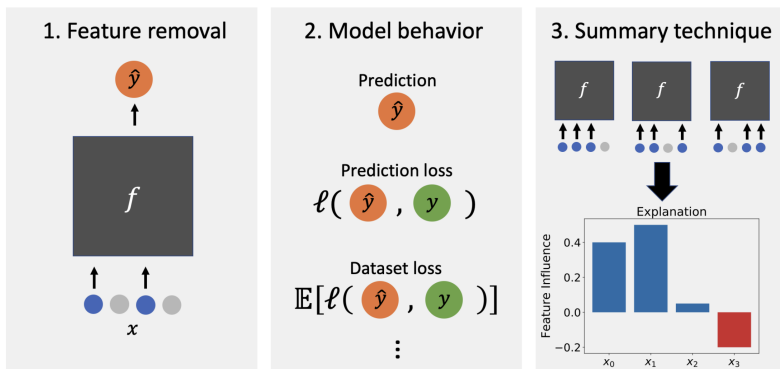


Figure 1: A unified framework for *removal-based explanations*. Each method is determined by three choices: how it removes features, what model behavior it analyzes, and how it summarizes feature influence.

- At least 26 published papers follow this recipe
- Example methods include SHAP, LIME, etc.
- Suggested the term removal-based explanations

A General Framework

	Permutation tests	Occlusion
1. Feature removal	Sample new values	Set to zero
2. Model behavior	Dataset loss	Individual prediction
3. Summarization	Remove single feat.	Remove single feat.

Options for feature removal approach: 1.replace with default values (zero); 2.replace with random values; 3.train separate models with each feature set; 4.use a model that accepts missing features; 5.application-based operations (e.g. blurring for images).

Covert et al., "Explaining by removing: a unified framework for model explanation" (2021)

A General Framework

METHOD	REMOVAL	BEHAVIOR	SUMMARY
IME (2009)	Separate models	Prediction	Shapley value
IME (2010)	Marginalize (uniform)	Prediction	Shapley value
QII	Marginalize (marginals product)	Prediction	Shapley value
SHAP	Marginalize (conditional/marginal)	Prediction	Shapley value
KernelSHAP	Marginalize (marginal)	Prediction	Shapley value
TreeSHAP	Tree distribution	Prediction	Shapley value
LossSHAP	Marginalize (conditional)	Prediction loss	Shapley value
SAGE	Marginalize (conditional)	Dataset loss (label)	Shapley value
Shapley Net Effects	Separate models (linear)	Dataset loss (label)	Shapley value
SPVIM	Separate models	Dataset loss (label)	Shapley value
Shapley Effects	Marginalize (conditional)	Dataset loss (output)	Shapley value
→ Permutation Test	Marginalize (marginal)	Dataset loss (label)	Remove individual
Conditional Perm. Test	Marginalize (conditional)	Dataset loss (label)	Remove individual
Feature Ablation (LOCO)	Separate models	Dataset loss (label)	Remove individual
Univariate Predictors	Separate models	Dataset loss (label)	Include individual
L2X	Surrogate	Prediction loss (output)	High-value subset
REAL-X	Surrogate	Prediction loss (output)	High-value subset
INVASE	Missingness during training	Prediction mean loss	High-value subset
→ LIME (Images)	Default values	Prediction	Linear model
LIME (Tabular)	Marginalize (replacement dist.)	Prediction	Linear model
→ PredDiff	Marginalize (conditional)	Prediction	Remove individual
→ Occlusion	Zeros	Prediction	Remove individual
CXPlain	Zeros	Prediction loss	Remove individual
→ RISE	Zeros	Prediction	Mean when included
MM	Default values	Prediction	Partitioned subsets
MIR	Extend pixel values	Prediction	High-value subset
→ MP	Blurring	Prediction	Low-value subset
EP	Blurring	Prediction	High-value subset
FIDO-CA	Generative model	Prediction	High-value subset

- 1 Motivation
- 2 Permutation and Occlusion
- 3 LIME**
- 4 RISE
- 5 SHAP
- 6 Interpretable ML

Local Interpretable Model-agnostic Explanations (LIME): feature attribution for an individual sample x

$$\xi(x) = \operatorname{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

- $f : \mathbb{R}^d \rightarrow \mathbb{R}$. In classification, $f(x)$ is the probability (or a binary indicator) that x belongs to a certain class η .
- $g : \{0, 1\}^{d'} \rightarrow \mathbb{R}$ denotes an interpretable model, e.g., $g(z') = w_g \cdot z'$
- $\pi_x(z)$: a proximity measure between z to x , e.g., $\exp(-\|x - z\|^2/\sigma^2)$
- $\mathcal{L}(f, g, \pi_x)$: a fidelity function measuring of how unfaithful g is in approximating f in the locality defined by π_x , e.g., $\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2$
- $\Omega(g)$: complexity measure (Example?)

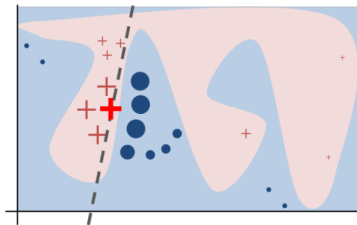
Ribeiro et al., "Why should I trust you? Explaining the predictions of any classifier" (2016)

Local Interpretable Model-agnostic Explanations (LIME)

$$\xi(x) = \operatorname{argmin}_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

- $x, z \in \mathbb{R}^d$, $x', z' \in \{0, 1\}^{d'}$
- $f: \mathbb{R}^d \rightarrow \mathbb{R}$, $g(z') = w_g \cdot z'$
- $\pi_x(z) = \exp(-\|x - z\|^2 / \sigma^2)$
- $\mathcal{L}(f, g, \pi_x) = \sum_{z, z' \in \mathcal{Z}} \pi_x(z) (f(z) - g(z'))^2$

Generate z' : sample instances around x' by drawing nonzero elements of x' uniformly at random (where the number of such draws is also uniformly sampled).

**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

$\mathcal{Z} \leftarrow \{\}$

for $i \in \{1, 2, 3, \dots, N\}$ **do**

$z'_i \leftarrow \text{sample_around}(x')$

$\mathcal{Z} \leftarrow \mathcal{Z} \cup \{z'_i, f(z_i), \pi_x(z_i)\}$

end for

$w \leftarrow \text{K-Lasso}(\mathcal{Z}, K)$ \triangleright with z'_i as features, $f(z)$ as target

return w

Ribeiro et al., "Why should I trust you? Explaining the predictions of any classifier" (2016)

LIME: Explain Random Forest

White Wine Quality dataset (4898 samples and 11 features):

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	bad
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	bad
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	good
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad

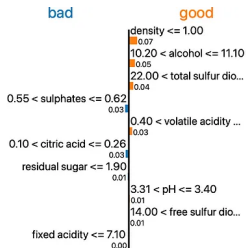
LIME: Explain Random Forest

White Wine Quality dataset (4898 samples and 11 features):

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	bad
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	bad
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	good
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad

Results:

Prediction probabilities



Feature Value

density	0.99
alcohol	10.60
total sulfur dioxide	34.00
sulphates	0.60
volatile acidity	0.46
citric acid	0.24
residual sugar	1.70
pH	3.39
free sulfur dioxide	18.00

The example is from <https://betterdatascience.com/lime/>
The dataset is from <https://archive.ics.uci.edu/ml/datasets/wine+quality>

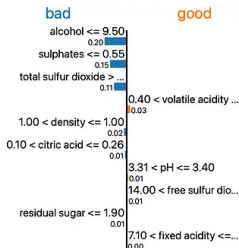
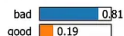
LIME: Explain Random Forest

White Wine Quality dataset (4898 samples and 11 features):

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	bad
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	bad
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	good
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	bad

Results:

Prediction probabilities



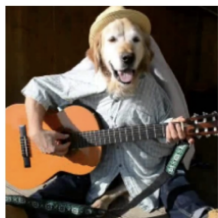
Feature Value

alcohol	9.50
sulphates	0.48
total sulfur dioxide	102.00
volatile acidity	0.50
density	1.00
citric acid	0.17
pH	3.39
free sulfur dioxide	21.00
residual sugar	1.60

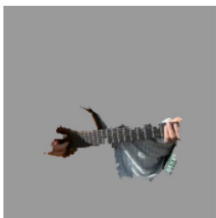
The example is from <https://betterdatascience.com/lime/>
The dataset is from <https://archive.ics.uci.edu/ml/datasets/wine+quality>

LIME: Explain Deep Neural Network

Model: Google's pre-trained Inception neural network



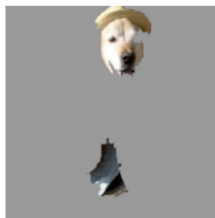
(a) Original Image



(b) Explaining *Electric guitar*



(c) Explaining *Acoustic guitar*



(d) Explaining *Labrador*

Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)

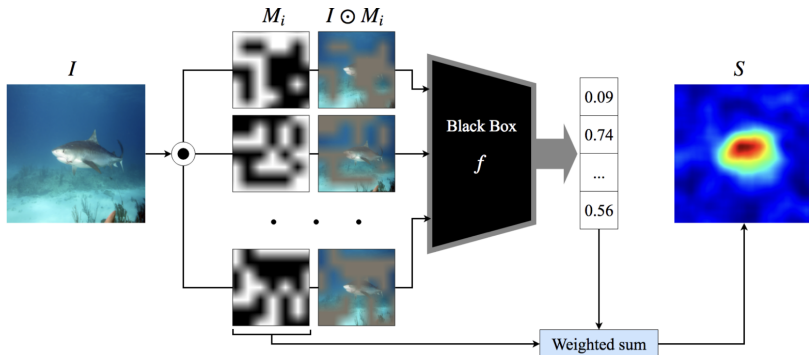
Ribeiro et al., "Why should I trust you? Explaining the predictions of any classifier" (2016)

- 1 Motivation
- 2 Permutation and Occlusion
- 3 LIME
- 4 RISE**
- 5 SHAP
- 6 Interpretable ML

Randomized Input Sampling for Explanation

- Samples many subsets of missing features
- Calculates mean prediction when x_i included

$$S_{I,f}(\lambda) = \frac{1}{\mathbb{E}[M] \cdot N} \sum_{i=1}^N f(I \odot M_i) \cdot M_i(\lambda) \quad \text{saliency or importance map}$$



Petsiuk et al., "RISE: Randomized input sampling for explanation of black-box models" (2018)

- 1 Motivation
- 2 Permutation and Occlusion
- 3 LIME
- 4 RISE
- 5 SHAP**
- 6 Interpretable ML

SHapley Additive exPlanations (SHAP)

- Based on Shapley values (Cooperative game theory, 1953) (Nash equilibrium is from noncooperative game theory)
- Unify various existing XAI methods
- Axiomatic definitions (the axioms of SHAP slightly differ from Shapley values)
- A prediction can be explained by assuming that each feature value of the instance is a “player” in a game where the prediction is the payout.



Lloyd Shapley

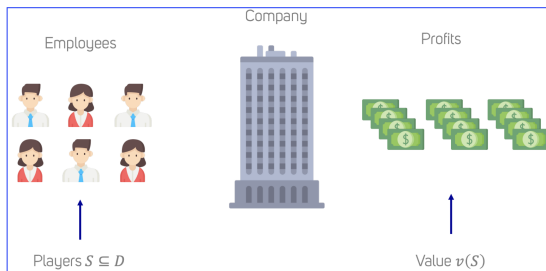
Won 2012 Nobel Memorial Prize in economics

Lundberg, Scott M., and Su-In Lee. A unified approach to interpreting model predictions. (2017).

Lloyd Shapley, "A value for n-person games" (1953)

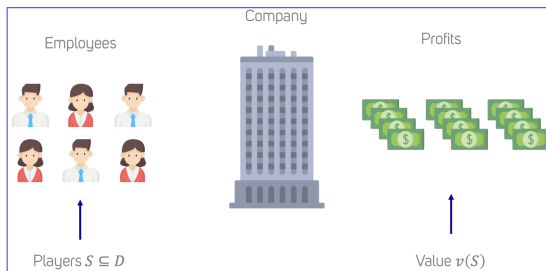
SHAP: Shapley Values

- Set of players $D = \{1, \dots, d\}$
- A game is given by specifying a value for every coalition $S \subseteq D$
- Represented by a characteristic function: $v : 2^D \rightarrow \mathbb{R}$
- Grand coalition value $v(D)$, null coalition $v(\emptyset)$, arbitrary coalition $v(S)$



SHAP: Shapley Values

- Set of players $D = \{1, \dots, d\}$
- A game is given by specifying a value for every coalition $S \subseteq D$
- Represented by a characteristic function: $v : 2^D \rightarrow \mathbb{R}$
- Grand coalition value $v(D)$, null coalition $v(\emptyset)$, arbitrary coalition $v(S)$



- * Which players will participate vs. break off on their own?
- * How to allocate credit among players?

Shapley values allocate credit to players in a cooperative game. Shapley values were famously derived from a set of fairness axioms.

SHAP: Shapley Values

- The Shapley value assigns a vector of credits to each game (in \mathbb{R}^d , one credit per player), mathematically, a function of the form

$$\phi : G \rightarrow \mathbb{R}^d$$

game

- For a game v , Shapley values are $\phi_1(v), \dots, \phi_d(v)$
- Axiomatic uniqueness: The Shapley value is the only attribution method ϕ that satisfies the following four properties
 - (Efficiency) The credits sum to the grand coalition's value, or $\sum_{i \in D} \phi_i(v) = v(D) - v(\emptyset)$
 - (Symmetry) If two players (i, j) are interchangeable, or $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $S \subseteq D$, then $\phi_i(v) = \phi_j(v)$
 - (Null player) If a player contributes no value, or $v(S \cup \{i\}) = v(S)$ for all $S \subseteq D$, then $\phi_i(v) = 0$
 - (Linearity) The credits for linear combinations of games behave linearly, or $\phi(c_1 v_1 + c_2 v_2) = c_1 \phi(v_1) + c_2 \phi(v_2)$, where $c_1, c_2 \in \mathbb{R}$

Lloyd Shapley, "A value for n-person games" (1953)

SHAP: Shapley Values

- The Shapley value of a feature value is its contribution to the payout, weighted and summed over all possible feature value combinations:

$$\phi_i(v) = \sum_{S \subseteq D \setminus i} \frac{|S|!(d-1-|S|)!}{d!} [v(S \cup \{i\}) - v(S)]$$

SHAP: Shapley Values

- The Shapley value of a feature value is its contribution to the payout, weighted and summed over all possible feature value combinations:

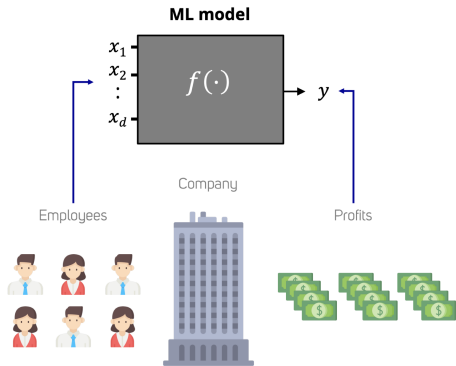
$$\phi_i(v) = \sum_{S \subseteq D \setminus i} \frac{|S|!(d-1-|S|)!}{d!} [v(S \cup \{i\}) - v(S)]$$

- Interpretation
 - Intuitive meaning in terms of player orderings
 - Given an ordering π , each player contributes when added to the preceding ones
 - SV is the average contribution across all orderings

$$\phi_i(v) = \frac{1}{d!} \sum_{\pi \in \Pi} \left[v(\{j \mid \pi^{-1}(j) \leq \pi^{-1}(i)\}) - v(\{j \mid \pi^{-1}(j) < \pi^{-1}(i)\}) \right]$$

SHapley Additive exPlanations

- Consider features as players
- Consider model behavior as profit
- Use Shapley values to quantify each feature's impact predictions



Change the notation from game to ML model:

$$\phi_i = \sum_{S \subseteq D \setminus \{i\}} \frac{|S|!(d - |S| - 1)!}{d!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)]$$

Require retraining the model on all feature subsets $S \subseteq D$

Other Shapley Value-based Methods

- Shapley Net Effects: Owen, "Sobol' indices and Shapley value" (2014)
- QII: Datta et al., "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems" (2016)
- IME: Strumbelj & Kononenko, "Explaining instance classifications with interactions of subsets of feature values" (2009)
- SAGE: Covert et al., "Understanding global feature contributions with additive importance measures" (2020)
- Causal Shapley values: Heskes et al., "Causal Shapley values: Exploiting causal knowledge to explain individual predictions of complex models" (2020)
- ASV: Frye et al., "Asymmetric Shapley values: incorporating causal knowledge into modelagnostic explainability" (2020)
- SP-VIM: Williamson & Feng, "Efficient nonparametric statistical inference on population feature importance using Shapley values" (2020)
- Shapley Flow, GraphSVX, Asymmetric Shapley, ...

$$\phi_i(v) = \sum_{S \subseteq D \setminus i} \frac{|S|!(d-1-|S|)!}{d!} [v(S \cup i) - v(S)]$$

- Exponential running time $O(2^d)$
- Intractable for even moderate d (e.g., $d > 20$) (Universe has about 10^{82} atoms (current estimate))
- Approximation methods exist

SHAP: Connection with LIME

- LIME has a weighting kernel $\pi(\mathcal{S})$ on feature subsets. It fits linear/additive proxy/surrogate model

$$\min_{a_0, \dots, a_d} \sum_{\mathcal{S} \subseteq D} \pi(\mathcal{S}) \left(a_0 + \sum_{i \in \mathcal{S}} a_i - f(x_{\mathcal{S}}) \right)^2 + \Omega(a_1, \dots, a_d)$$

- Shapley values minimize the following objective:

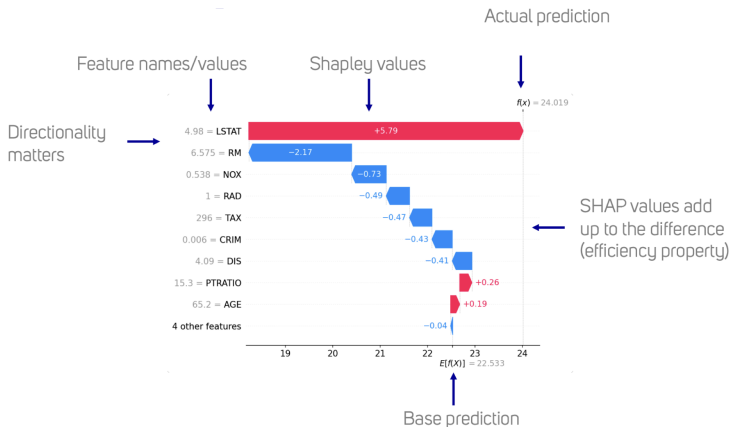
$$\min_{\beta_0, \dots, \beta_d} \sum_{\mathcal{S} \subseteq D} \mu(\mathcal{S}) \left(\beta_0 + \sum_{i \in \mathcal{S}} \beta_i - v(\mathcal{S}) \right)^2, \quad \mu(\mathcal{S}) = \frac{d-1}{\binom{d}{|\mathcal{S}|} |\mathcal{S}| (d-|\mathcal{S}|)}$$

- SHAP is a special case of LIME with $\pi(\mathcal{S}) = \mu(\mathcal{S})$ and $\Omega = 0$

Lundberg, Scott M., and Su-In Lee. A unified approach to interpreting model predictions. (2017).

Case Study: Boston Housing Dataset

- Predict median house price in a neighborhood using 14 features: mean number of rooms, crime rate, distance to employment centers, etc.
- Trained an XGBoost model (gradient boosted decision tree)

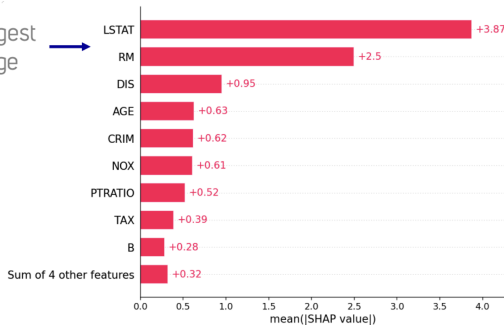


Lundberg, Scott M., and Su-In Lee. A unified approach to interpreting model predictions. (2017).

Case Study: Boston Housing Dataset

- Predict median house price in a neighborhood using 14 features: mean number of rooms, crime rate, distance to employment centers, etc.
- Trained an XGBoost model (gradient boosted decision tree)

Features with largest impact, on average

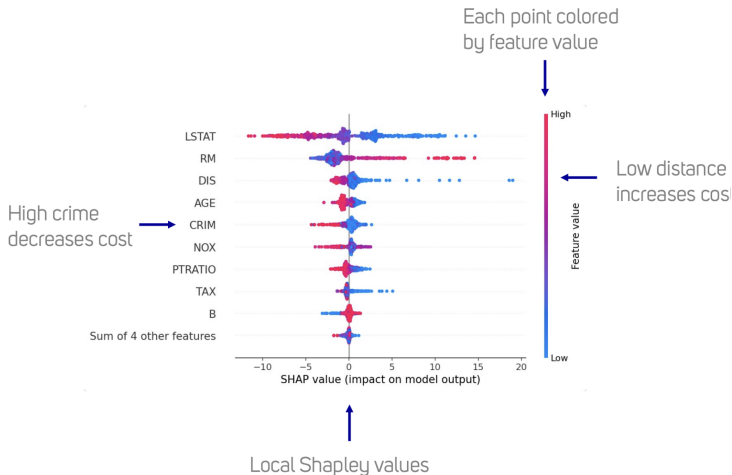


Aggregating local SHAP values

Lundberg, Scott M., and Su-In Lee. A unified approach to interpreting model predictions. (2017).

Case Study: Boston Housing Dataset

- Predict median house price in a neighborhood using 14 features: mean number of rooms, crime rate, distance to employment centers, etc.
- Trained an XGBoost model (gradient boosted decision tree)



- Removal-based methods are very common in XAI
- Other structural based approaches exist
 - Gradient-based Methods (e.g. Grad-CAM, Gradient \times Input)
 - Consider the gradient of the network output w.r.t. each input variable
 - Propagation-based Methods (e.g. Deep Taylor, Deep SHAP)
 - Use backpropagation idea to quantify feature importance
 - Estimate attributions of intermediate features at a layer and then back-propagate these attributions to the previous layer

- 1 Motivation
- 2 Permutation and Occlusion
- 3 LIME
- 4 RISE
- 5 SHAP
- 6 Interpretable ML**

Interpretable ML vs XAI

- Interpretable ML: when you use a model that is not a blackbox
- Explainable AI: when you use a black-box model and use another model to explain afterwards
 - Start with a black-box
 - Create another model that approximates it
 - Compute derivatives of the proxy model
 - Visualize what part of the input the model is paying attention to


nature machine intelligence

Explore content ▾ About the journal ▾ Publish with us ▾ | [Subscribe](#)

[nature](#) > [nature machine intelligence](#) > [perspectives](#) > [article](#)

Perspective | [Published: 13 May 2019](#)

Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead

[Cynthia Rudin](#) 

[Nature Machine Intelligence](#) 1, 206–215 (2019) | [Cite this article](#)

61k Accesses | 1957 Citations | 470 Altmetric | [Metrics](#)

Interpretable ML vs XAI

- Interpretable ML: when you use a model that is not a blackbox
- Explainable AI: when you use a black-box model and use another model to explain afterwards
 - Start with a black-box
 - Create another model that approximates it
 - Compute derivatives of the proxy model
 - Visualize what part of the input the model is paying attention to


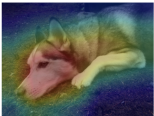

	Test Image	Evidence for Animal Being a Siberian Husky	Evidence for Animal Being a Transverse Flute
Explanations Using Attention Maps			

Figure 2: Saliency does not explain anything except where the network is looking. We have no idea why this image is labeled as either a dog or a musical instrument when considering only saliency. The explanations look essentially the same for both classes. Figure credit: Chaofan Chen and [28].

Interpretable ML vs XAI

- Interpretable ML \neq XAI
 - Trusting a black-box means you trust the database it was built from
 - Double Trouble: Need to rely on two models, instead of one
 - Those models may make mistakes and may disagree with each other
 - If we can produce an interpretable ML model, why explain a black-box?

Interpretable ML vs XAI

- Interpretable ML \neq XAI
 - Trusting a black-box means you trust the database it was built from
 - Double Trouble: Need to rely on two models, instead of one
 - Those models may make mistakes and may disagree with each other
 - If we can produce an interpretable ML model, why explain a black-box?
- Example: COMPAS vs CORELS

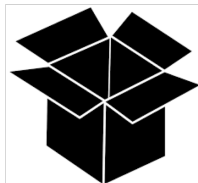
COMPAS

Correctional Offender Management Profiling for Alternative Sanctions



CORELS

Learning Certifiably Optimal Rule Lists



Building Predictive Models with Rule Lists

[Click here to enter the CORELS website](#)

Transparent

Rule lists are fully human-interpretable, giving them distinct advantages over black box models.

Optimized

Our algorithms utilize highly optimized vector operations, allowing them to run in reasonable time on commodity laptops.

Accurate

On many datasets, rule lists have been shown to be comparable in accuracy to much more complex black box models.

Free!

All our code is free, open source, and under the [GNU General Public License v3.0](#). You can find it [on GitHub](#).

More information about CORELS can be found in the following papers:

- Elaine Angelino, Nicholas Larus-Stone, Daniel Alabi, Margo Seltzer, and Cynthia Rudin. **Learning Certifiably Optimal Rule Lists for Categorical Data**. KDD 2017. Journal of Machine Learning Research, 2018; 19: 1-77. [arXiv:1704.01701](#), 2017.
- Nicholas Larus-Stone, Elaine Angelino, Daniel Alabi, Margo Seltzer, Vassilios Kaxiras, Aditya Saligrama, and Cynthia Rudin. **Systems Optimizations for Learning Certifiably Optimal Rule Lists**. [SysML Conference, 2018](#).
- Nicholas Larus-Stone. **Learning Certifiably Optimal Rule Lists: A Case For Discrete Optimization in the 21st**

COMPAS vs CORELS

COMPAS	CORELS
black box 130+ factors might include socio-economic info expensive (software license), within software used in U.S. Justice System	full model is in Figure 3 only age, priors, (optional) gender no other information free, transparent



```
IF      age between 18-20 and sex is male      THEN predict arrest (within 2 years)
ELSE IF age between 21-23 and 2-3 prior offenses THEN predict arrest
ELSE IF      more than three priors            THEN predict arrest
ELSE      predict no arrest.
```

Are the following models interpretable?

- Decision trees
- Linear regression
- Generalized additive models (GAMs)
- Attention as explanation

Open question: how to comprehensively evaluate interpretability/transparency?

- Understand why explanations are needed for black-box models
- Understand the difference between XAI and interpretable ML
- Be able to list a few removal-based XAI methods and know exactly what they do
- Know the definition of Shapley value and how it can be used in XAI