WB-XIC, Lab6:

Wstęp do wyjaśnień konwolucyjnych sieci neuronowych

Hubert Baniecki

hbaniecki@gmail.com | http://hbaniecki.com

Explainability in AI

Researchers say Amazon face-detection technology shows bias

Two researchers say Amazon's facial-recognition technology has a lot of trouble identifying darker-skinned women

By TALI ARBEL AP Technology Writer
25 January 2019, 22:24 • 3 min read

STOCK PHOTO/Getty Images

Amazon login screen on a mobile device.

ARTIFICIAL INTELLIGENCE

Hundreds of AI tools have been built to catch covid. None of them helped.

Some have been used in hospitals, despite not being properly tested. But the pandemic could help make medical AI better.

By Will Douglas Heaven July 30, 2021

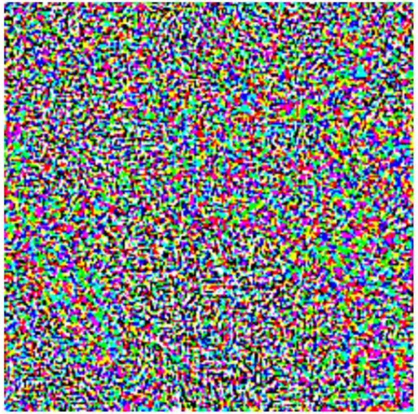
https://github.com/daviddao/awful-ai

Adversary in AI: Security & Safety

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$



$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$

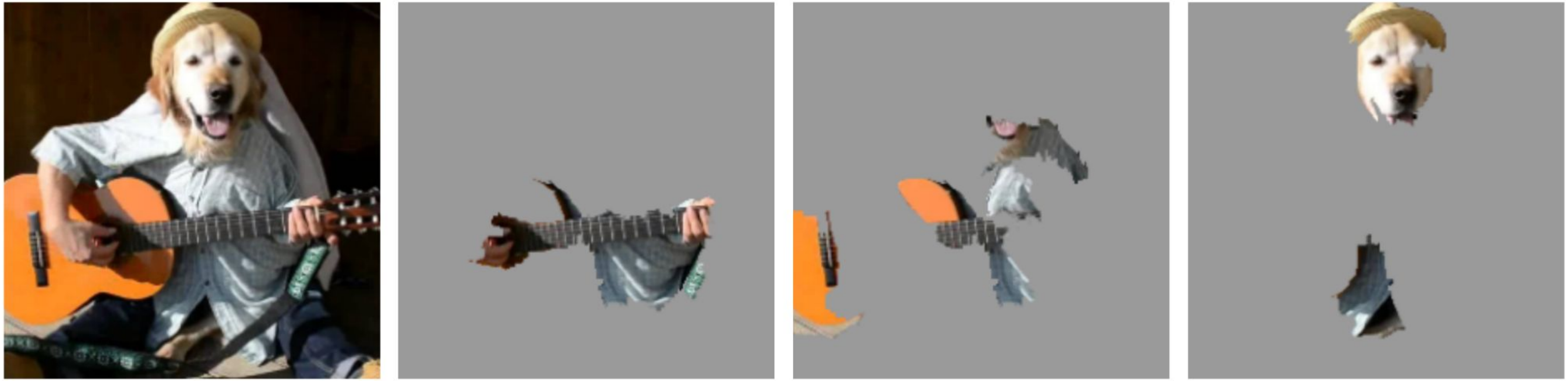
$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"gibbon"
99.3 % confidence

https://arxiv.org/abs/1412.6572

Explanations of neural networks

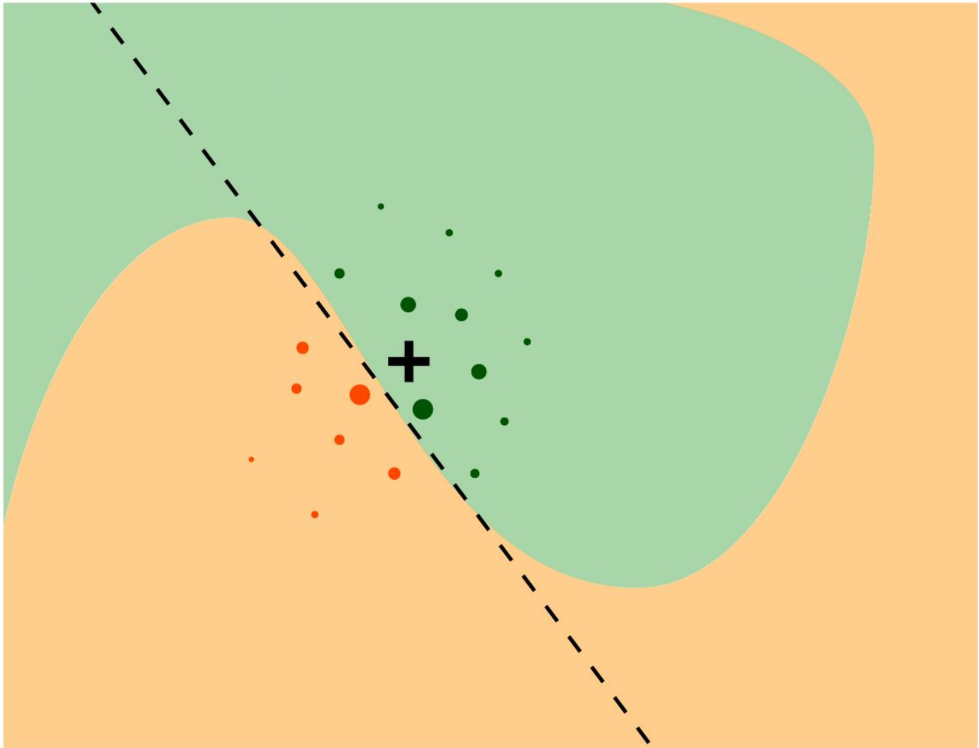
Local interpretable model-agnostic explanations (LIME)



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

M. T. Ribeiro et al. **"Why Should I Trust You?": Explaining the Predictions of Any Classifier**. *KDD*, 2016. https://doi.org/10.1145/2939672.2939778

LIME: local surrogate model



https://ema.drwhy.ai/LIME

LIME: intuition

Mathematically, local surrogate models with interpretability constraint can be expressed as follows:

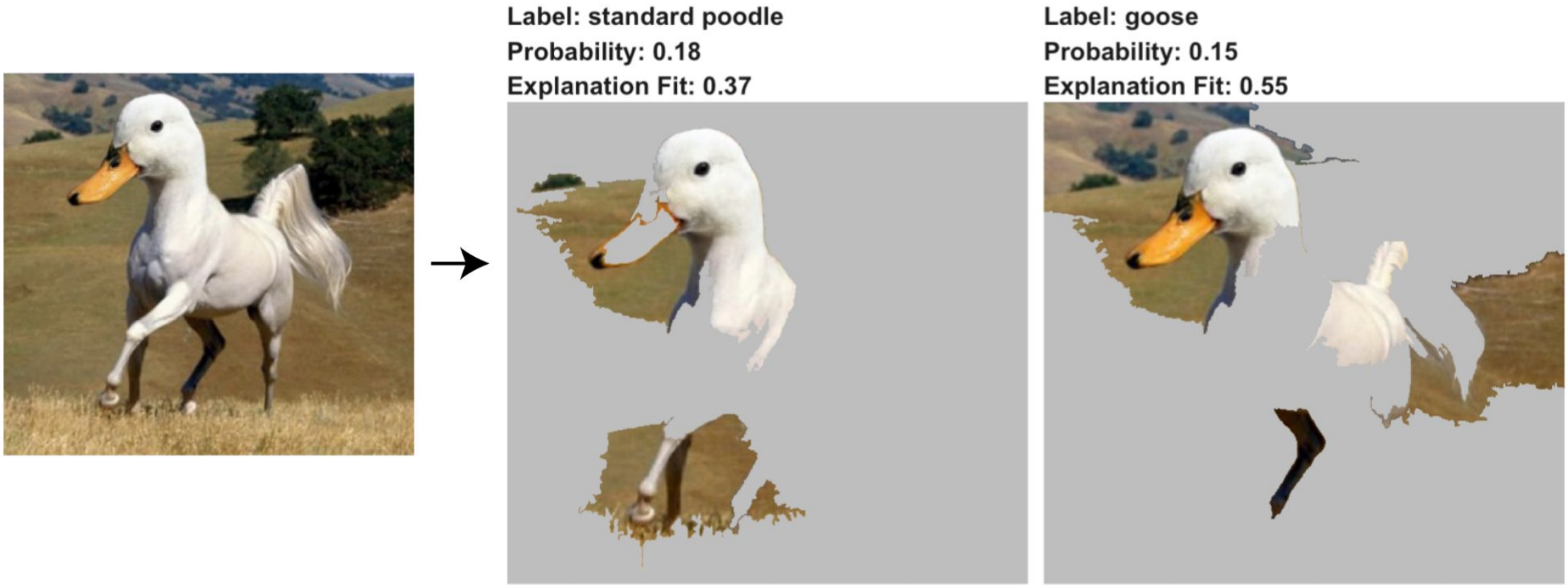
$$\text{explanation}(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

The recipe for training local surrogate models:

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

https://christophm.github.io/interpretable-ml-book/lime

LIME for image: superpixels and image segmentation



https://ema.drwhy.ai/LIME

Google Colab

Saliency maps (*vanilla* gradients)

The recipe for this approach is:

1. Perform a forward pass of the image of interest.
2. Compute the gradient of class score of interest with respect to the input pixels:

$$E_{grad}(I_0) = \frac{\delta S_c}{\delta I}\Big|_{I=I_0}$$

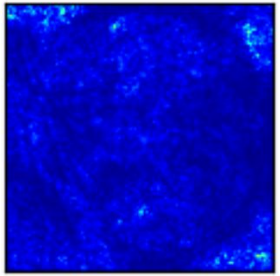
   Here we set all other classes to zero.

3. Visualize the gradients. You can either show the absolute values or highlight negative and positive contributions separately.

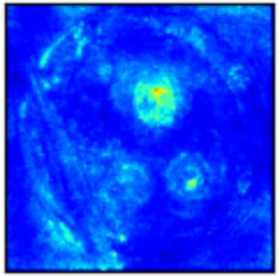
Smoothgrad: average multiple explanations for an image with added noise

Grad-Cam: gradient explanation tailored to CNN (ReLU, last Conv2d)

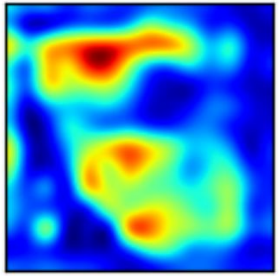
Soup Bowl (vanilla)



Soup Bowl (Smoothgrad)

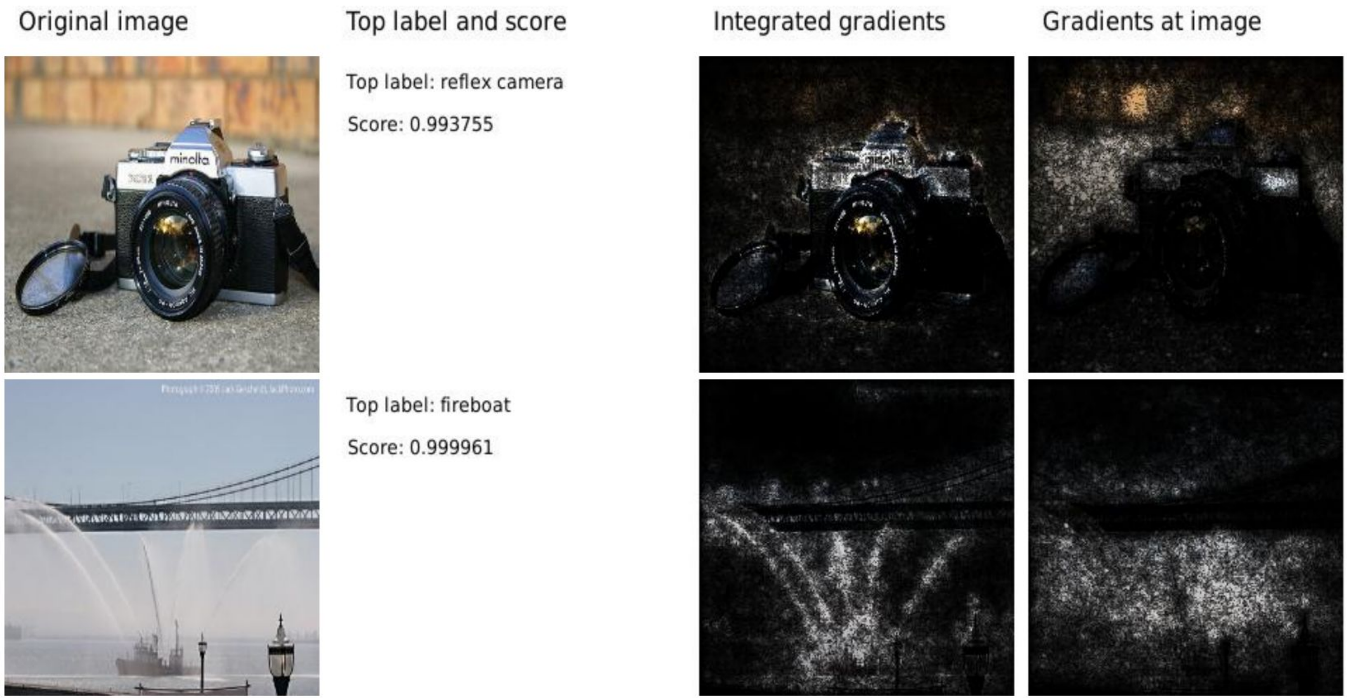


Soup Bowl (Grad-Cam)



https://christophm.github.io/interpretable-ml-book/pixel-attribution

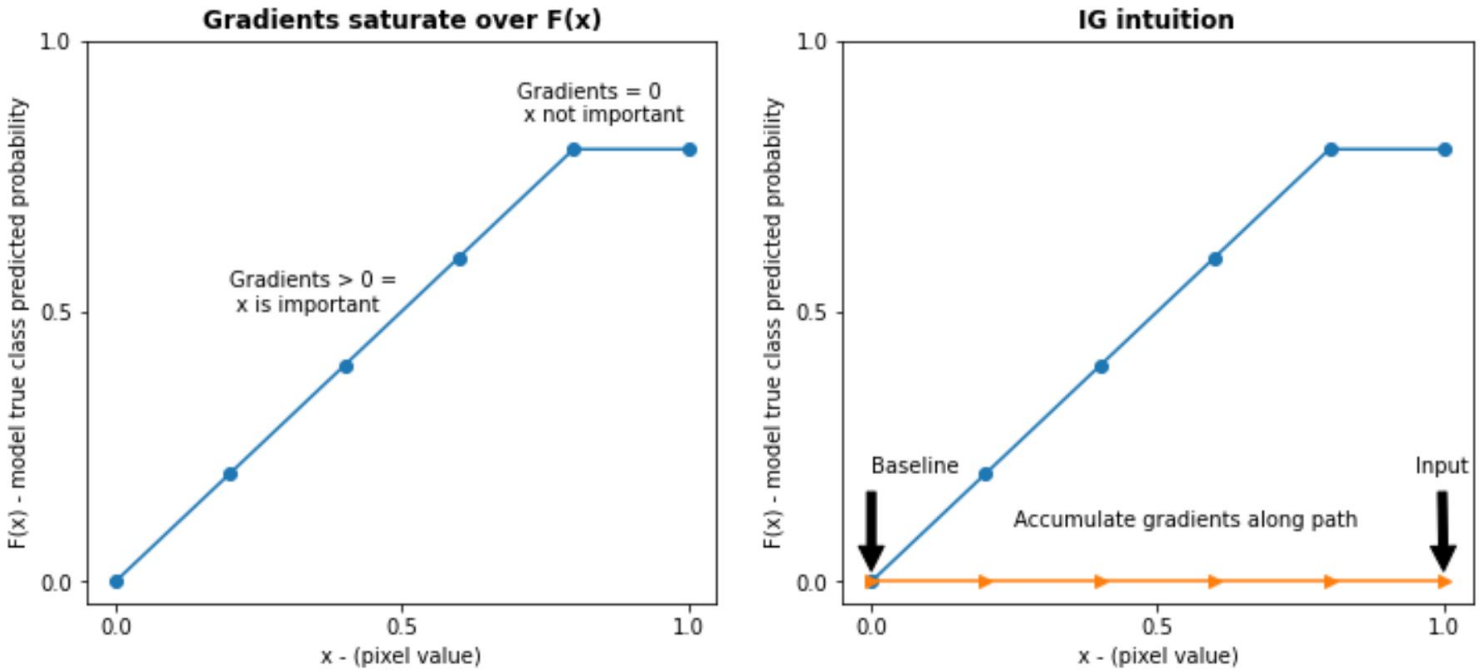
Integrated gradients (IG)



M. Sundararajan et al. **Axiomatic attribution for deep networks**. *ICML*, 2017. https://dl.acm.org/doi/10.5555/3045118.3045167

IG: integral over gradients

Baseline: dark/grey/white image or pixel distribution over a subset of data.



https://www.tensorflow.org/tutorials/interpretability/integrated_gradients

IG: intuition

Formally, let $\gamma = (\gamma_1, \ldots, \gamma_n) : [0, 1] \to \mathsf{R}^n$ be a smooth function specifying a path in R^n from the baseline x' to the input x, i.e., $\gamma(0) = x'$ and $\gamma(1) = x$.

Given a path function γ, *path integrated gradients* are obtained by integrating the gradients along the path $\gamma(\alpha)$ for $\alpha \in [0, 1]$. Formally, path integrated gradients along the i^{th} dimension for an input x is defined as follows.

$$\mathsf{PathIntegratedGrads}_i^{\gamma}(x) ::= \int_{\alpha=0}^{1} \frac{\partial F(\gamma(\alpha))}{\partial \gamma_i(\alpha)} \frac{\partial \gamma_i(\alpha)}{\partial \alpha}\, d\alpha \tag{2}$$

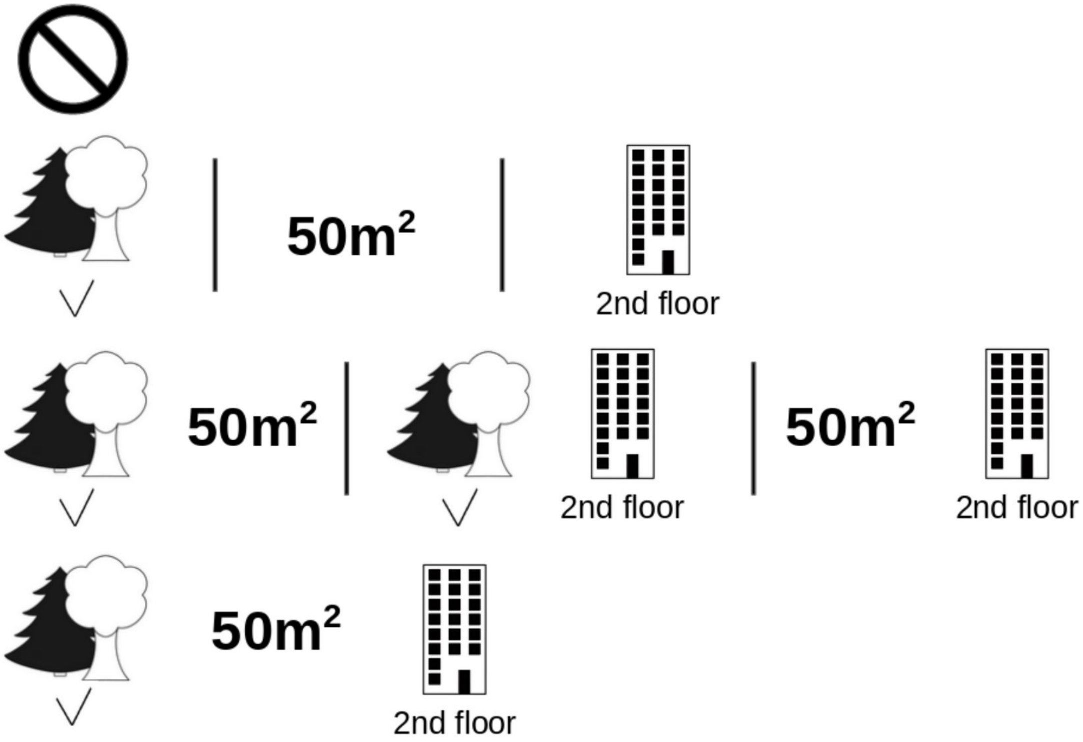
where $\frac{\partial F(x)}{\partial x_i}$ is the gradient of F along the i^{th} dimension at x.

$$\mathsf{IntegratedGrads}_i^{approx}(x) ::= (x_i - x_i') \times \Sigma_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m}$$

(path-attribution methods) https://christophm.github.io/interpretable-ml-book/pixel-attribution

Google Colab

Shapley values: game theory



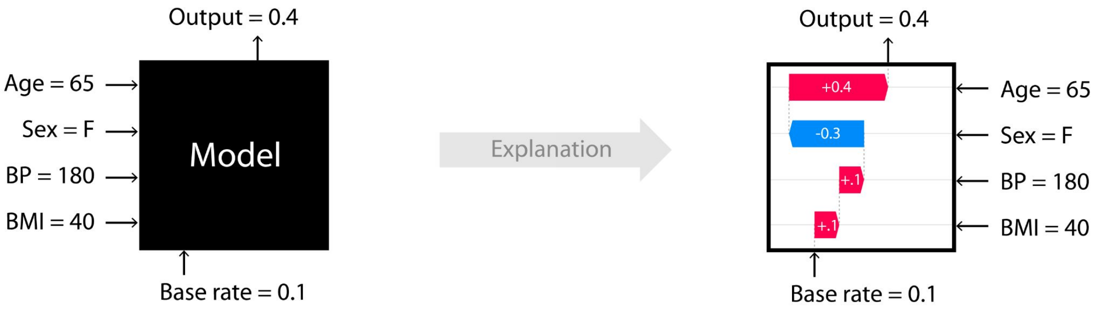
https://christophm.github.io/interpretable-ml-book/shapley

Shapley values: math

$$\phi_j(val) = \sum_{S \subseteq \{1,\ldots,p\} \setminus \{j\}} \frac{|S|!\,(p - |S| - 1)!}{p!} \left(val\left(S \cup \{j\}\right) - val(S)\right)$$

https://christophm.github.io/interpretable-ml-book/shapley

SHapley Additive exPlanations (SHAP)



Definition 1 **Additive feature attribution methods** *have an explanation model that is a linear function of binary variables:*

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i, \tag{1}$$

where $z' \in \{0,1\}^M$, M *is the number of simplified input features, and* $\phi_i \in \mathbb{R}$.

S. M. Lundberg & S. Lee. **A Unified Approach to Interpreting Model Predictions**. *NeurIPS*, 2017.
https://dl.acm.org/doi/10.5555/3295222.3295230

SHAP

1. (model-agnostic) **KernelSHAP**: LIME + SHAP kernel
2. TreeSHAP: fast SHAP values for tree-ensemble models)
3. Gradient: SHAP based on IG and Smoothgrad
4. *SHAP based on DeepLIFT https://arxiv.org/abs/1704.02685

https://christophm.github.io/interpretable-ml-book/shap

Google Colab

Praca domowa