DECSAI
Departamento de Ciencias de la Computación e I.A.
Universidad de Granada

XAI [eXplainable AI]

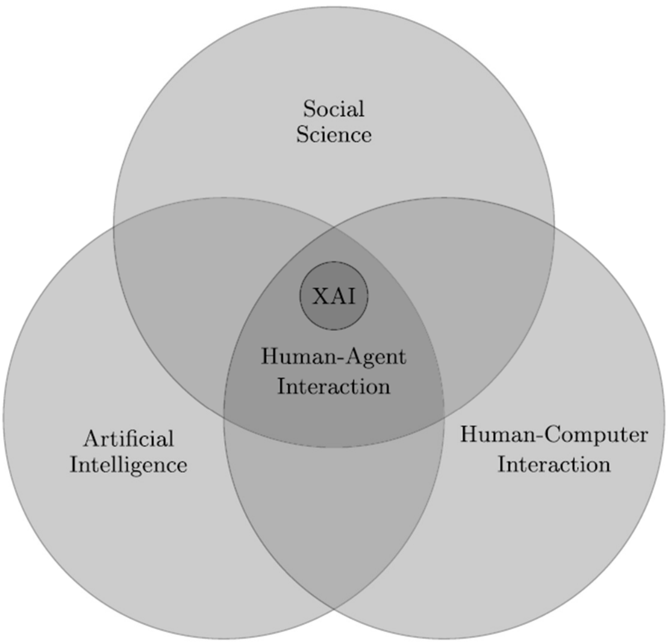
Fernando Berzal, berzal@acm.org

XAI

Artificial Intelligence
Volume 267, February 2019, Pages 1-38

Explanation in artificial intelligence: Insights from the social sciences

Tim Miller



XAI

1. Explanations are **contrastive** — they are sought in response to particular counterfactual cases [foilsin]: people do not ask why P happened, but rather why P happened instead of Q.

2. Explanation are **selected** (in a biased manner) — people rarely, if ever, expect an explanation that consists of an actual and complete cause of an event. Humans are adept at selecting one or two causes (this selection is influenced by cognitive biases)

XAI

3. Probabilities probably don’t matter —referring to probabilities or statistical relationships in explanation is not as effective as referring to causes. The most likely explanation is not always the best explanation for a person (using statistical generalizations to explain why events occur is unsatisfying, unless accompanied by an underlying causal explanation for the generalisation itself).

4. Explanations are social — they are presented relative to the explainer’s beliefs about the explainee’s beliefs.

XAI

Explanations are not just
the presentation of associations
and causes (causal attribution),
they are contextual.


ELSEVIER

Artificial Intelligence
Volume 267, February 2019, Pages 1-38

Explanation in artificial intelligence:
Insights from the social sciences

Tim Miller

While an event may have many causes, often the explainee cares only about a small subset (relevant to the context), the explainer selects a subset of this subset (based on several different criteria), and explainer and explainee may interact and argue about this explanation.

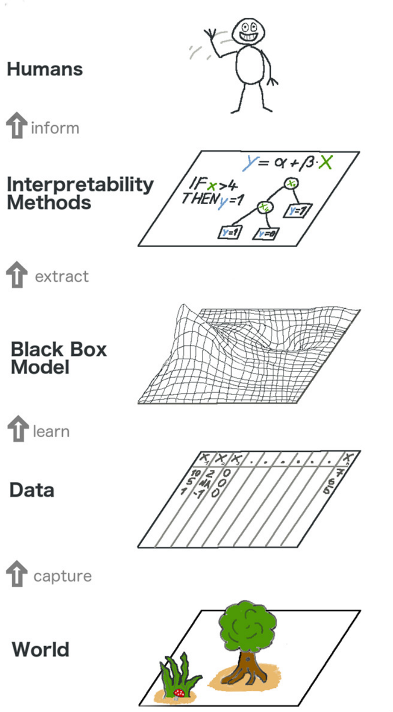
XAI

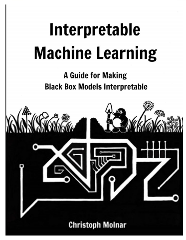
Interpretability:

The degree to which an observer
can understand the cause of a decision.
Or Biran & Courtenay Cotton: "Explanation and justification in machine learning: A survey," IJCAI-17 Workshop on Explainable AI (XAI), Melbourne, Australia, 20 August 2017.

The degree to which a human observer
can consistently predict the model's output.
Been Kim, Rajiv Khanna & Oluwasanmi Koyejo: "Examples are not enough, learn to criticize! Criticism for interpretability." NIPS'2016

XAI





XAI

Model-Agnostic Methods

- PDP [Partial Dependence Plot], a.k.a. PD plot
- ICE [Individual Conditional Expectation]
- ALE [Accumulated Local Effects]
- LIME [Local Interpretable Model-agnostic Explanations]
- Anchors
- SHAP [Shapley Additive exPlanations]

PDP [Partial Dependence Plot]

- Función de dependencia parcial

$$\hat{f}_{x_S}(x_S) = E_{x_C}\left[\hat{f}(x_S, x_C)\right] = \int \hat{f}(x_S, x_C) d\mathbb{P}(x_C)$$

- Estimación a partir del conjunto de entrenamiento:

$$\hat{f}_{x_S}(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)})$$

Jerome H. Friedman: "Greedy function approximation: A gradient boosting machine." Annals of Statistics (2001): 1189-1232

Qingyuan Zhao & Trevor Hastie: "Causal interpretations of black-box models." Journal of Business & Economic Statistics, 2021

ICE [Individual Conditional Expectation]

Local method equivalent to the PDP global method, i.e. how the instance's prediction changes when a feature changes.

Variants

- Centered ICE [c-ICE]

$$\hat{f}_{cent}^{(i)} = \hat{f}^{(i)} - \mathbf{1}\hat{f}(x^a, x_C^{(i)})$$

- Derivative ICE [d-ICE]

$$\hat{f}(x) = \hat{f}(x_S, x_C) = g(x_S) + h(x_C), \quad \text{with} \quad \frac{\delta \hat{f}(x)}{\delta x_S} = g'(x_S)$$

Alex Goldstein et al.: "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation." Journal of Computational and Graphical Statistics 24.1 (2015): 44-65.

ALE [Accumulated Local Effects]

i.e. how features influence the prediction of a machine learning model on average

- M plots average the predictions over the conditional distribution.
$$\hat{f}_{x_S,M}(x_S) = E_{X_C|X_S}\left[\hat{f}(X_S, X_C)|X_S = x_s\right] = \int_{x_C} \hat{f}(x_S, x_C)\mathbb{P}(x_C|x_S)dx_C$$

- ALE plots average the changes in the predictions and accumulate them
$$\hat{f}_{x_S,ALE}(x_S) = \int_{z_{0,1}}^{x_S} E_{X_C|X_S}\left[\hat{f}^S(X_s, X_c)|X_S = z_S\right] dz_S - \text{constant} = \int_{z_{0,1}}^{x_S} \int_{x_C} \hat{f}^S(z_s, x_c)\mathbb{P}(x_C|z_S)dx_C dz_S - \text{constant}$$

Daniel W. Apley: "Visualizing the effects of predictor variables in black box supervised learning models." arXiv preprint, 2016, arXiv:1612.08468

Feature Importance

IDEA: Measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature. A feature is "important" if shuffling its values increases the model error (i.e. the model relied on the feature for the prediction).

e.g. **Model Reliance**

Aaron Fisher, Cynthia Rudin & Francesca Dominici. "Model Class Reliance: Variable importance measures for any machine learning model class, from the 'Rashomon' perspective." arXiv, 2018. https://arxiv.org/abs/1801.01489

Feature Interaction

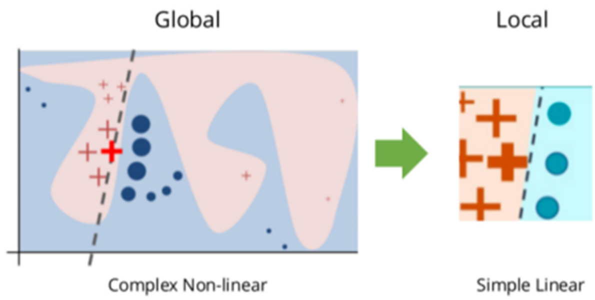
- Friedman's H-statistic
  Jerome H. Friedman & Bogdan E. Popescu: "Predictive learning via rule ensembles." The Annals of Applied Statistics. JSTOR, 916–54. (2008)

- VIN [Variable Interaction Networks]
  Giles Hooker: "Discovering additive structure in black box functions." KDD'2004, 10th ACM International Conference on Knowledge Discovery and Data Mining

- Partial dependence-based feature interaction
  Brandon M. Greenwell, Bradley C. Boehmke & Andrew J. McCarthy: "A simple and effective model-based variable importance measure." arXiv preprint arXiv:1805.04755 (2018)

LIME [Local interpretable model-agnostic explanations]

Local surrogate model

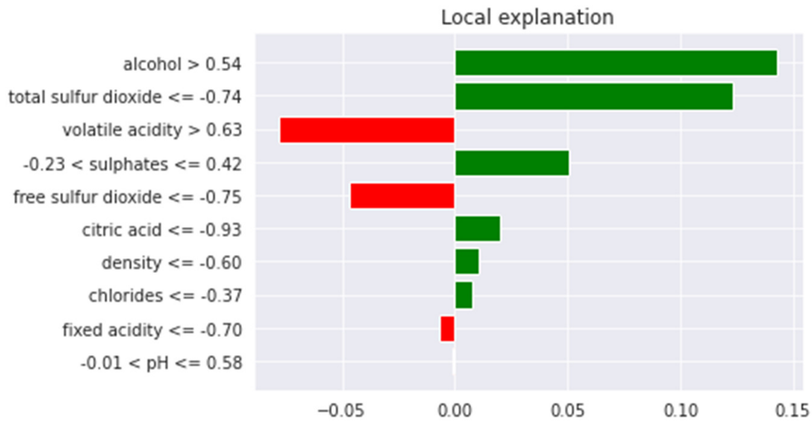


Marco Tulio Ribeiro, Sameer Singh & Carlos Guestrin:
"Why should I trust you?: Explaining the predictions of any classifier."
KDD'2016, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

https://github.com/marcotcr/lime

LIME [Local interpretable model-agnostic explanations]

Example: Wine quality



https://github.com/Milan-Chicago/Explainable-AI-Examples/blob/master/Explainable%20Notebooks/LIME/LIME.ipynb

LIME [Local interpretable model-agnostic explanations]

Warnings!

- Instability of the explanations.
  David Alvarez-Melis & Tommi S. Jaakkola: "On the robustness of interpretability methods." arXiv preprint arXiv:1806.08049 (2018)

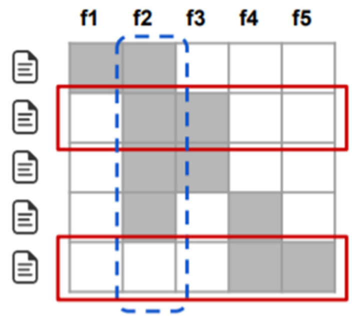
- Hidden biases (LIME explanations can be manipulated)
  Dylan Slack et al.: "Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods." Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, 2020

SP-LIME [Submodular Pick LIME]

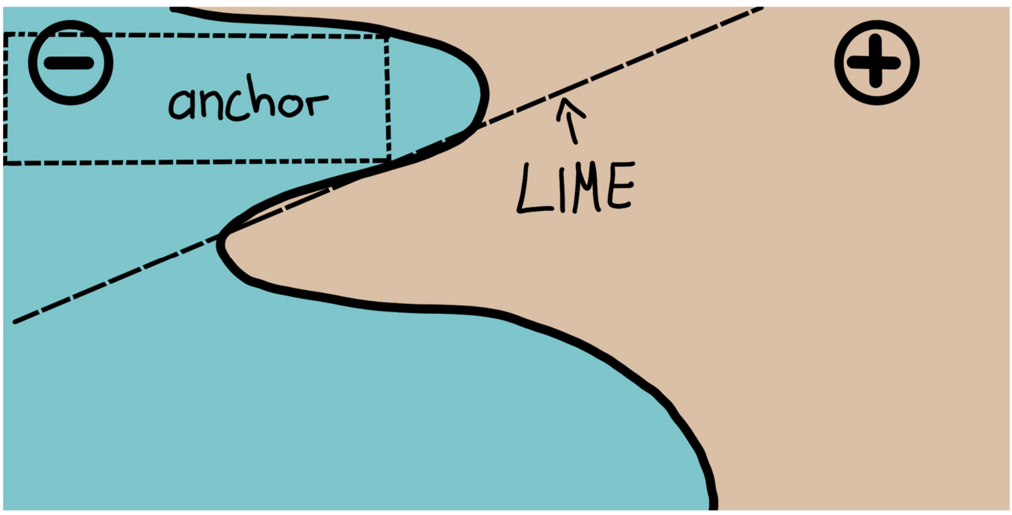
Submodular pick for explaining models

1. Run the explanation model on all instances (all x's).
2. Compute the global importance of individual features.
3. Maximize the coverage function by iteratively adding the instance with the highest maximum coverage gain
4. Return a representative nonredundant explanation set.

NOTE: Greedy algorithm, since coverage maximization is NP-hard.



Anchors

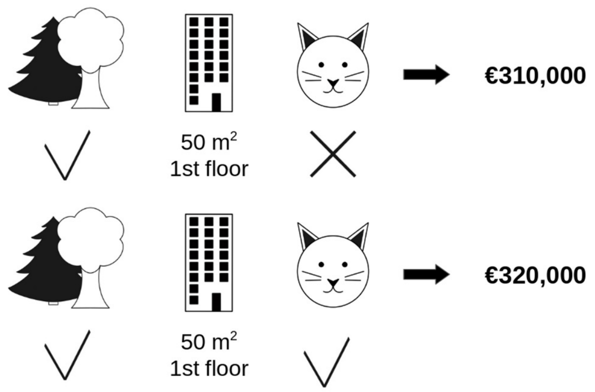


Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin.
"Anchors: High-Precision Model-Agnostic Explanations."
AAAI Conference on Artificial Intelligence (AAAI), 2018

SHAP [SHapley Additive exPlanations]

Shapley value [@ coallitional game theory]

a method for assigning payouts to players
depending on their contribution to the total payout



Lloyd S. Shapley: "A value for n-person games."
Contributions to the Theory of Games 2.28 (1953): 307-317

SHAP [SHapley Additive exPlanations]

Shapley value

- Linear model: $\hat{f}(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p$
- Contribution of the j-th feature to the prediction:

$$\phi_j(\hat{f}) = \beta_j x_j - E(\beta_j X_j) = \beta_j x_j - \beta_j E(X_j)$$

Sum of all the feature contributions
= predicted value - average predicted value

$$\begin{aligned}\sum_{j=1}^{p} \phi_j(\hat{f}) &= \sum_{j=1}^{p} (\beta_j x_j - E(\beta_j X_j)) \\ &= (\beta_0 + \sum_{j=1}^{p} \beta_j x_j) - (\beta_0 + \sum_{j=1}^{p} E(\beta_j X_j)) \\ &= \hat{f}(x) - E(\hat{f}(X))\end{aligned}$$

SHAP [SHapley Additive exPlanations]

Shapley value

Estimating the Shapley value through Monte Carlo sampling:

$$\hat{\phi}_j = \frac{1}{M}\sum_{m=1}^{M}\left(\hat{f}\left(x_{+j}^m\right) - \hat{f}\left(x_{-j}^m\right)\right)$$

Warnings!

- Only approximate solutions are feasible.
- Interpretation of the estimated Shapley value: the contribution of a feature value to the difference between the actual prediction and the mean prediction given the current set of feature values

Erik Štrumbelj & Igor Kononenko:
"Explaining prediction models and individual predictions with feature contributions." Knowledge and Information Systems 41.3 (2014): 647-665

SHAP [SHapley Additive exPlanations]

SHAP, a method to explain individual predictions.

Scott M. Lundberg & Su-In Lee:
"A unified approach to interpreting model predictions." NIPS'2017

- **KernelSHAP**, a kernel-based estimation approach for Shapley values inspired by local surrogate models.

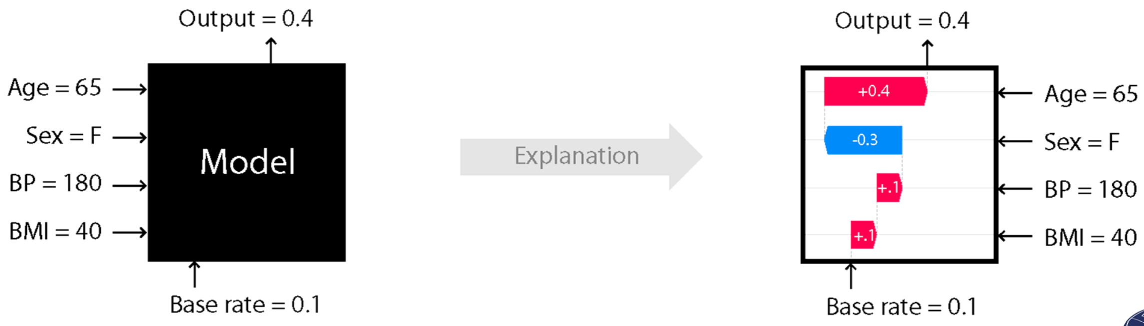
$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j$$

- SHAP feature importance:

$$I_j = \sum_{i=1}^{n} |\phi_j^{(i)}|$$

SHAP [SHapley Additive exPlanations]

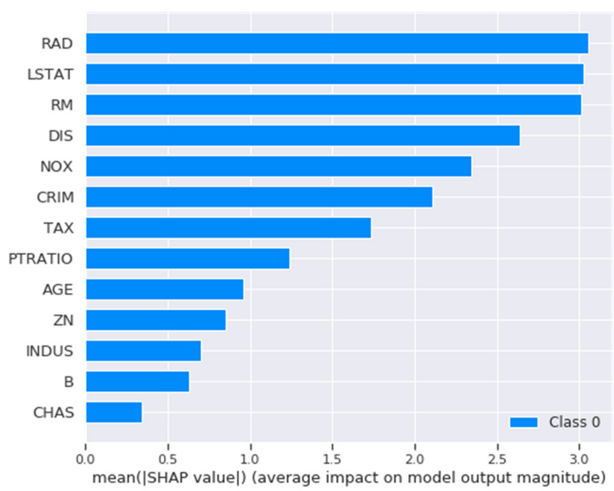
SHAP

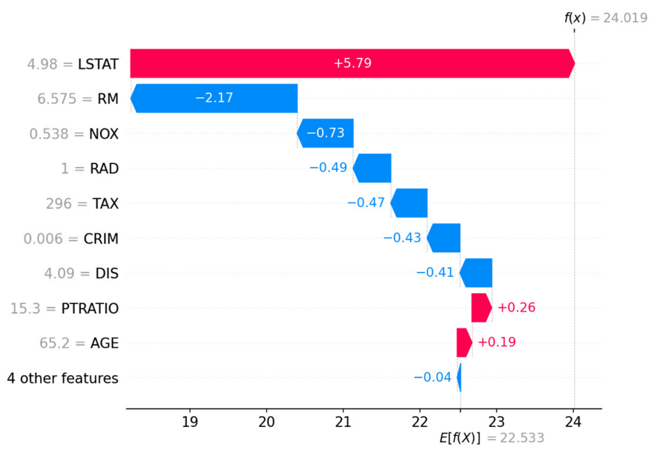


https://github.com/slundberg/shap

SHAP [SHapley Additive exPlanations]

Example: House Pricing





https://towardsdatascience.com/keras-101-a-simple-and-interpretable-neural-network-model-for-house-pricing-regression-31b1a77f05ae

SHAP [SHapley Additive exPlanations]

SHAP pros:

- computes Shapley values (solid theoretical foundation)
- connects LIME and Shapley values (in KernelSHAP)

SHAP cons:

- slow KernelSHAP
- ignores feature dependence
- can be misinterpreted
- can be used to create intentionally misleading interpretations to hide biases (as LIME).

XAI

Example-Based Methods

explain a model by selecting instances of the dataset and not by creating summaries of features

- Counterfactual explanations
- Adversarial examples
- Prototypes
- Influential instances
- Nearest neighbors (i.e. k-NN)

Counterfactual Explanations

i.e. how an instance has to change
to significantly change its prediction.
(the opposite to anchors)

"If X had not occurred, Y would not have occurred"

e.g.

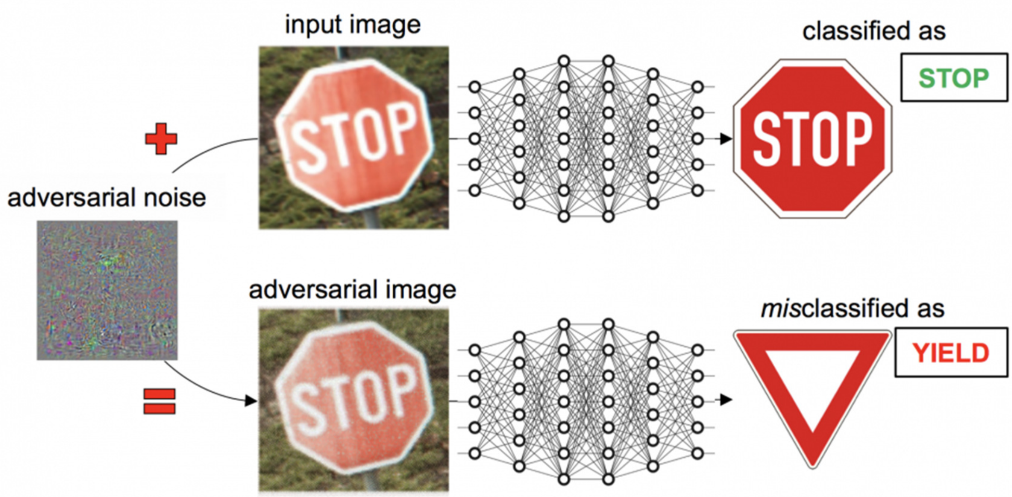
Sandra Wachter, Brent Mittelstadt & Chris Russell: "Counterfactual explanations without opening the black box: Automated decisions and the GDPR." Harvard Journal of Law & Technology, 2018

Susanne Dandl, Christoph Molnar, Martin Binder & Bernd Bischl: "Multi-Objective Counterfactual Explanations," Parallel Problem Solving from Nature, PPSN'2020.

Arnaud Van Looveren & Janis Klaise: "Interpretable Counterfactual Explanations Guided by Prototypes." arXiv, 2019. arXiv:1907.02584

Adversarial Examples

i.e. counterfactuals used to fool machine learning models

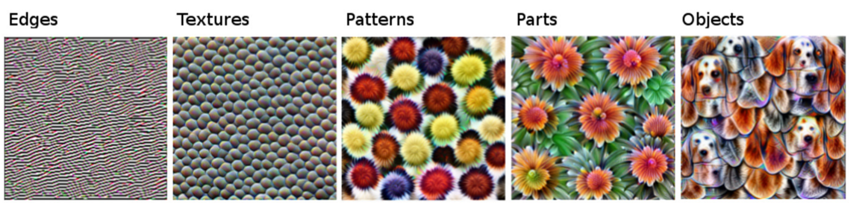


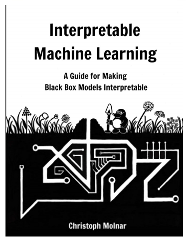
XAI

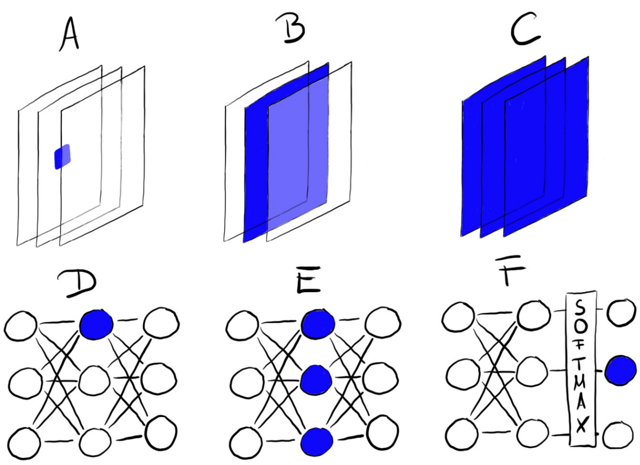
Neural Network Interpretation Methods

- Feature Visualization
- Pixel Attribution
  - Saliency Maps
  - Path-Attribution Methods
    - DeepLIFT
    - Deep Taylor
    - Integrated Gradients
    - XRAI
- Concepts

Feature Visualization







Feature Visualization

Lucid

https://github.com/tensorflow/lucid

Negative Neurons [colab]

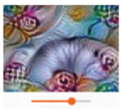
What is the *opposite* of what a neuron is looking for? This can reveal interesting things about the representation.

Diversity Visualization [colab]

Neurons generally respond to multiple things -- sometimes similar and sometimes wildly different. How can we visaulize this diversity?

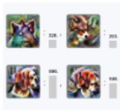
Neuron Interactions [colab]

Explore how neurons combine and interact. Linear combinations, random directions in neuron space, and interpolation.



Regularizing Visualizations [colab]

One of the main challenges to visualizing features is regularizing the feature visualizations. Try different techniques and fiddle with hyperparameters.

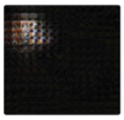


Semantic Dictionaries [colab]

Saying "neuron 312 fired" isn't very meaningful to humans. Combining neuron activations with feature visualization can make things much more meaningful.

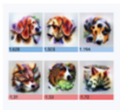
Activation Grids [colab]

Activation grids can help us see how the network understood each spatial position.



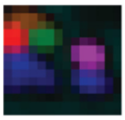
Spatial Attribution [colab]

Do attribution to spatial positions in hidden layers -- either from the output or other hidden layers. This is similar to traditional saliency maps.



Channel Attribution [colab]

How did different features effect the output? We can use attribution between channels in hidden layers and the output, along with feature visualization, to explore this.



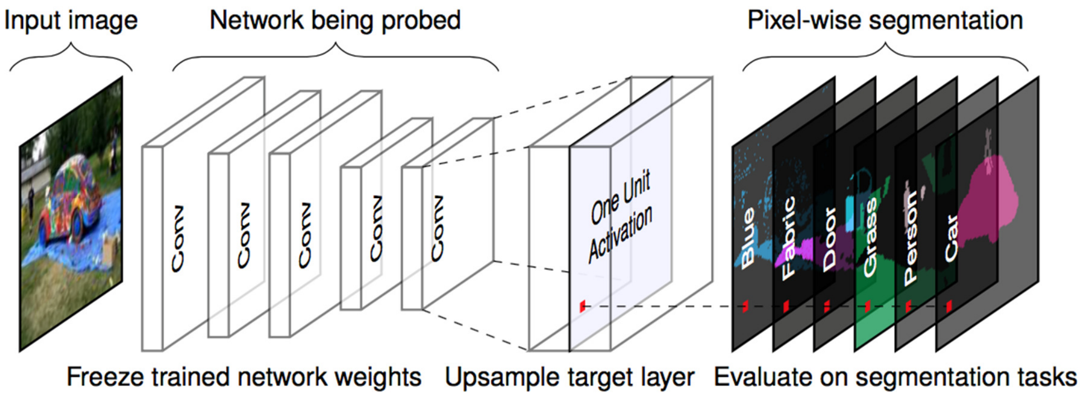
Neuron Groups [colab]

Explore how groups of neurons work together to represent objects in an image. Automatically extract neuron groups and then visualize them.

Feature Visualization

Network Dissection (CVPR'2017)

http://netdissect.csail.mit.edu/



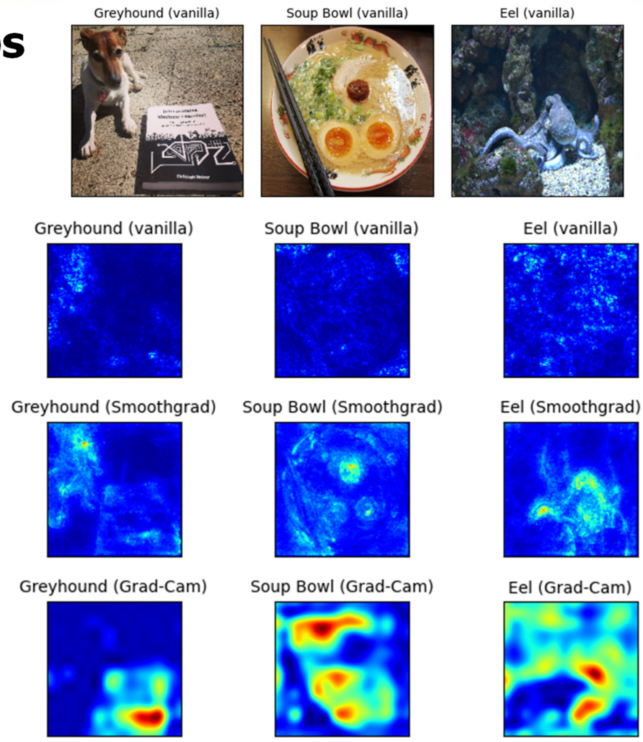
"By measuring the concept that best matches each unit, Net Dissection can break down the types of concepts represented in a layer"

Pixel Attribution

- **Gradient-only methods** tell us whether a change in a pixel would change the prediction [saliency maps]
  - Vanilla Gradient
  - DeconvNet [LRP: Layer-wise Relevance Propagation]
  - Grad-CAM [Gradient-weighted Class Activation Map]
  - SmoothGrad
- **Path-attribution methods** compare the current image to a reference image [baseline]
  - Deep Taylor
  - DeepLIFT
  - Integrated Gradients
  - XRAI

Pixel Attribution

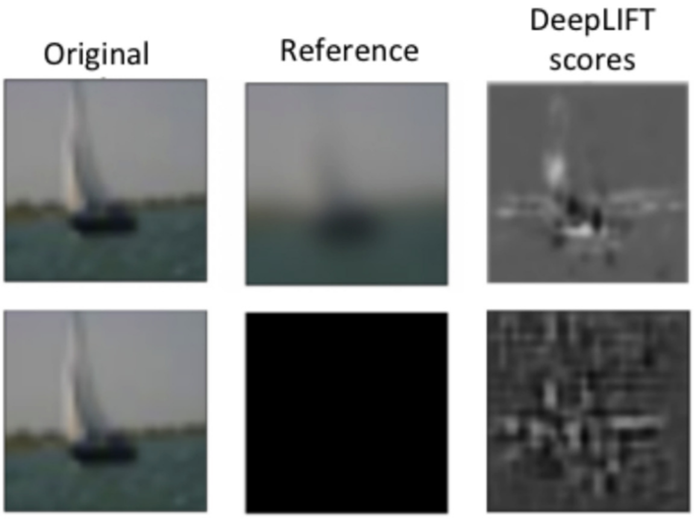
Saliency Maps



Pixel Attribution

DeepLIFT [Deep Learning Important FeaTures]

ICML'2017



Avanti Shrikumar, Peyton Greenside, Anshul Kundaje

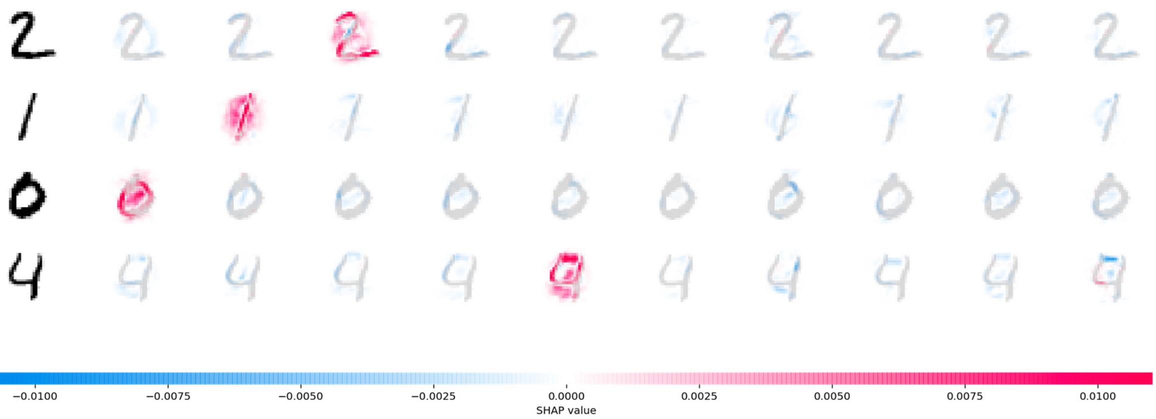
"Learning Important Features Through Propagating Activation Differences"

ICML'2017, https://arxiv.org/abs/1704.02685

https://github.com/kundajelab/deeplift

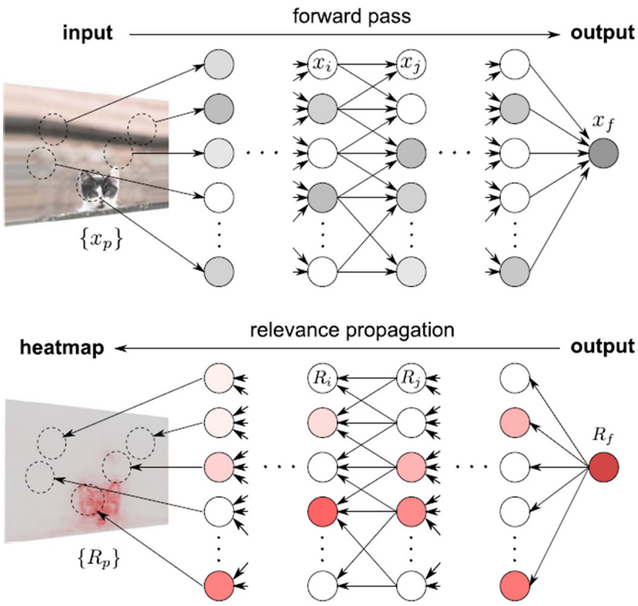
Pixel Attribution

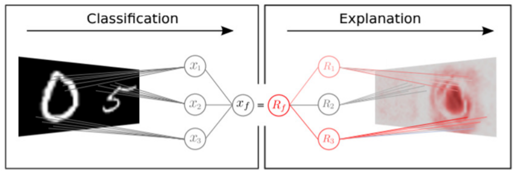
DeepSHAP = DeepLIFT + Shapley values



Pixel Attribution

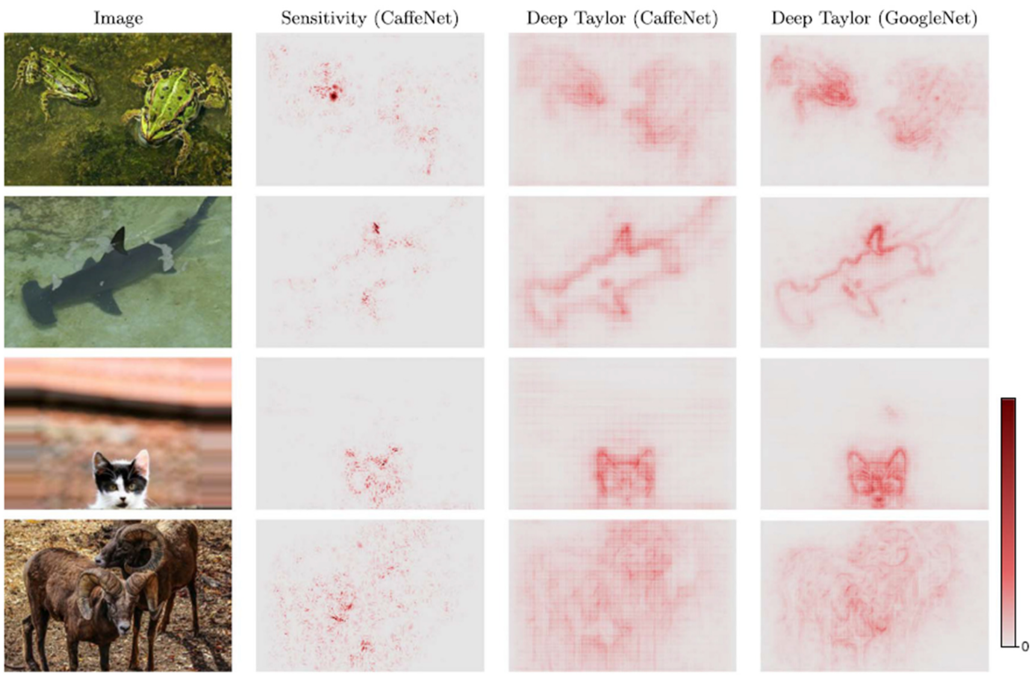
Deep Taylor
Pattern Recognition '2017





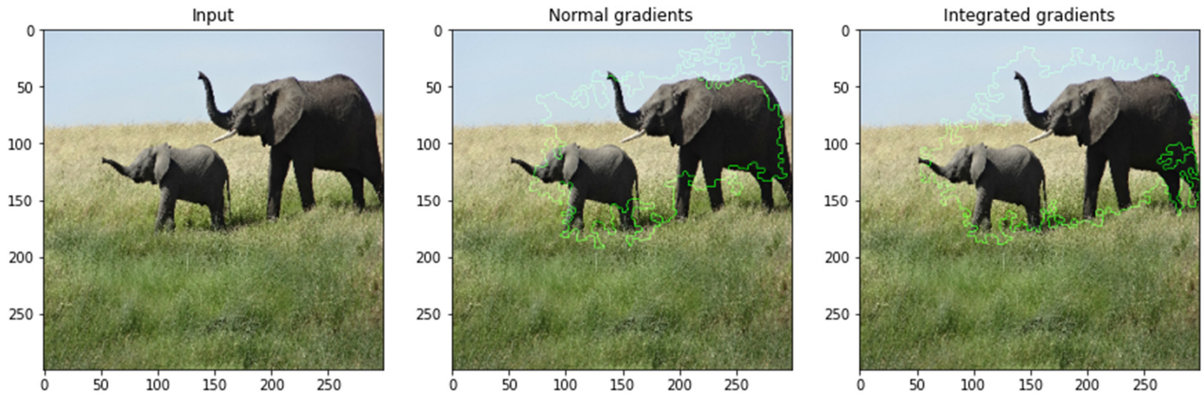
Pixel Attribution

Deep Taylor



Pixel Attribution

Integrated Gradients
ICML'2017

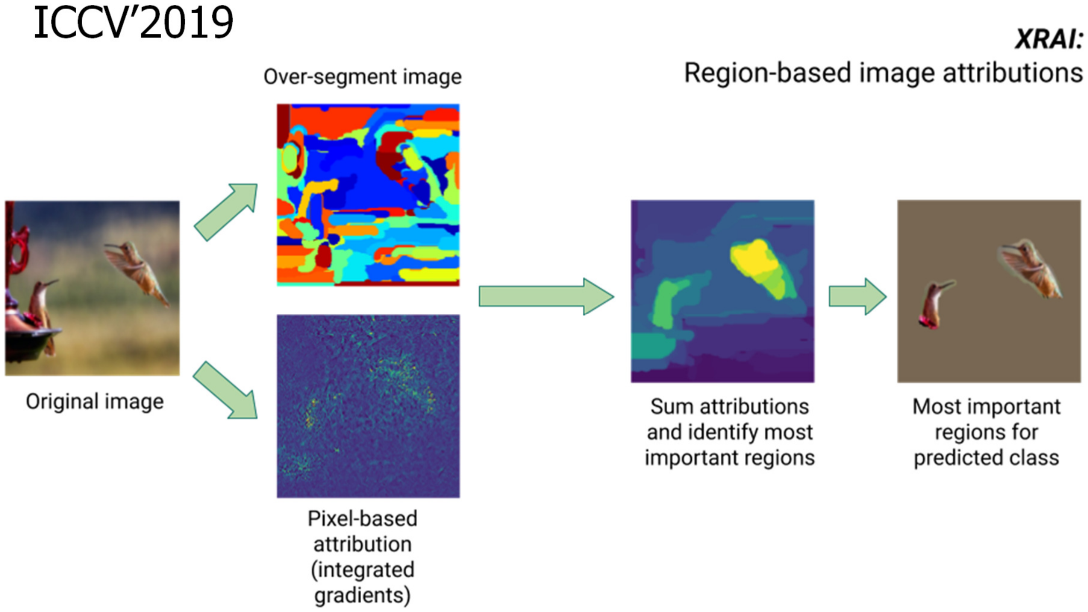


https://keras.io/examples/vision/integrated_gradients/

Mukund Sundararajan, Ankur Taly, Qiqi Yan
"Axiomatic Attribution for Deep Networks"
ICML'2017, https://arxiv.org/abs/1703.01365
https://github.com/ankurtaly/Integrated-Gradients

Pixel Attribution

XRAI: Better Attributions Through Regions
ICCV'2019



Concepts

TCAV [Testing with Concept Activation Vectors]
ICML'2018

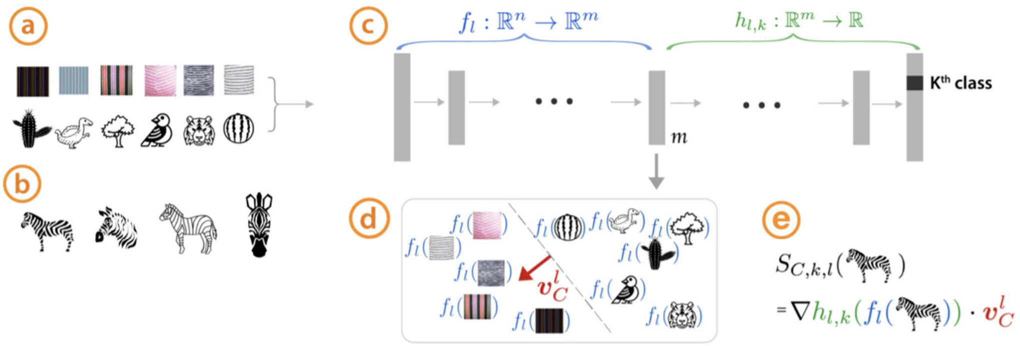


Figure 1. **Testing with Concept Activation Vectors:** Given a user-defined set of examples for a concept (e.g., 'striped'), and random examples ⓐ, labeled training-data examples for the studied class (zebras) ⓑ, and a trained network ⓒ, TCAV can quantify the model's sensitivity to the concept for that class. CAVs are learned by training a linear classifier to distinguish between the activations produced by a concept's examples and examples in any layer ⓓ. The CAV is the vector orthogonal to the classification boundary (v_C^l, red arrow). For the class of interest (zebras), TCAV uses the directional derivative $S_{C,k,l}(\boldsymbol{x})$ to quantify conceptual sensitivity ⓔ.

Concepts

ACE [Automatic Concept-based Explanations]
NIPS'2019

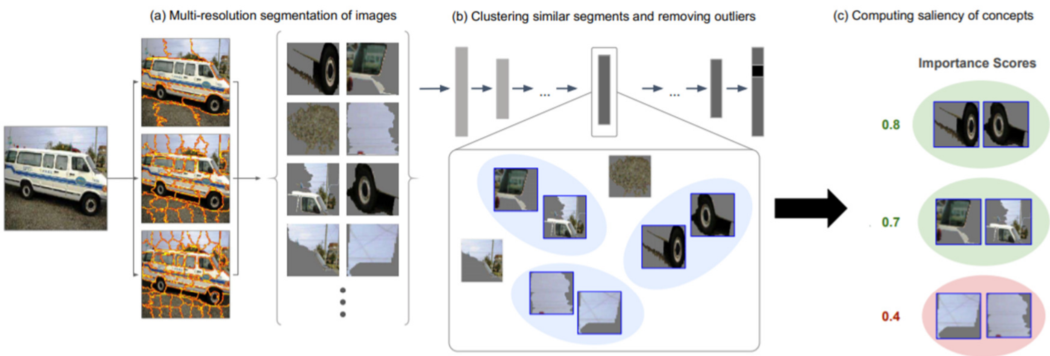


Figure 1: ***ACE* algorithm** (a) A set of images from the same class is given. Each image is segmented with multiple resolutions resulting in a pool of segments all coming from the same class. (b) The activation space of one bottleneck layer of a state-of-the-art CNN classifier is used as a similarity space. After resizing each segment to the standard input size of the model, similar segments are clustered in the activation space and outliers are removed to increase coherency of clusters. (d) For each concept, its TCAV importance score is computed given its examples segments.

Concepts

CBM [Concept Bottleneck Models]
ICML’2020

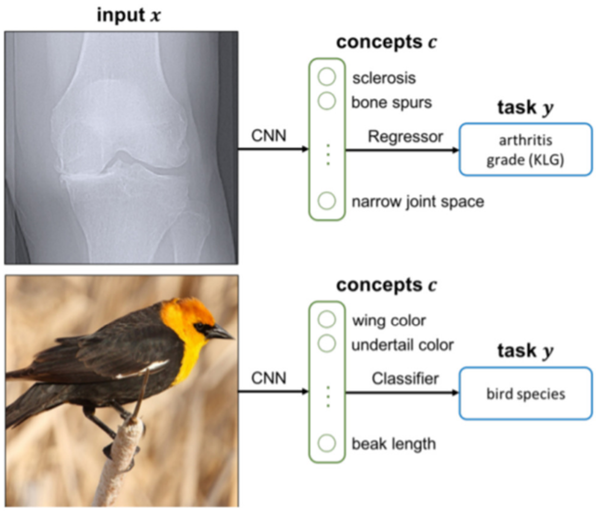
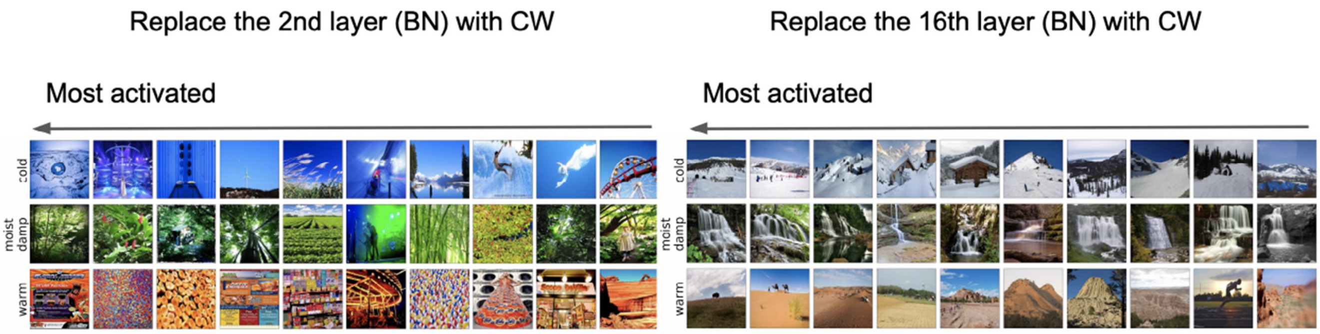


Figure 1. We study concept bottleneck models that first predict an intermediate set of human-specified concepts c, then use c to predict the final output y. We illustrate the two applications we consider: knee x-ray grading and bird identification.

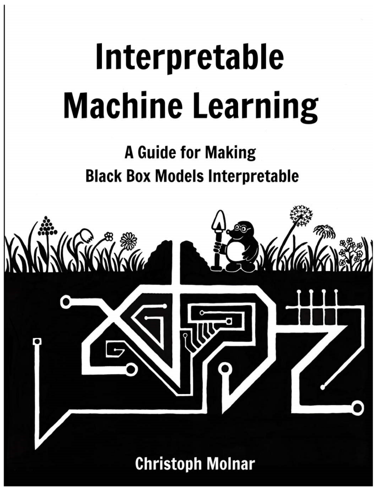
Concepts

CW [Concept Whitening]
Nature Machine Intelligence, 2020



“When a concept whitening module is added to a CNN, the axes of the latent space are aligned with known concepts of interest.”

Bibliografía



- Christoph Molnar:
  Interpretable Machine Learning:
  A Guide for Making Black Box Models Interpretable
  https://christophm.github.io/interpretable-ml-book/
  2021. ISBN 0244768528

  En español (versión 2019): https://fedefliguer.github.io/AAI/