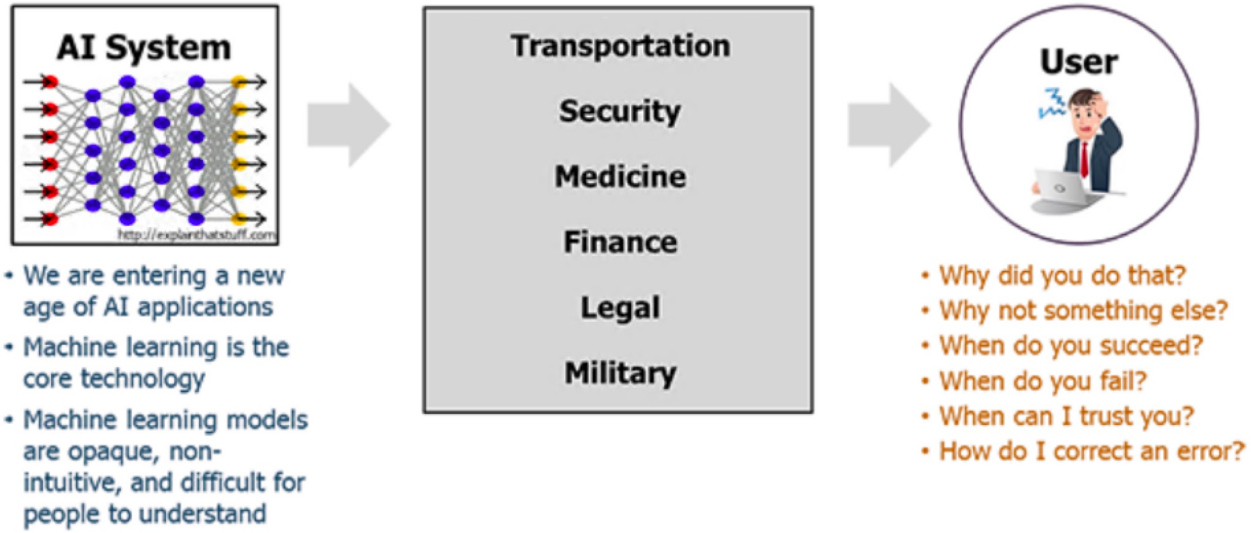
# Lecture 18: Explainable AI




- **Explainable AI (XAI)** refers to methods and techniques in the application of artificial intelligence technology (AI) such that the results of the solution can be understood by human experts and users.
- It contrasts with the concept of the "black box" in machine learning where even their designers cannot explain why the AI arrived at a specific decision

## The Need for XAI: User Perspective

**Explanations foster trust and verifiability**

- Patients trust well-explained therapy.
- Doctors trust well-explained suggestions.


"Honey, drink the medicine."
Man died later of poison.

## The Need for XAI: Society Perspective

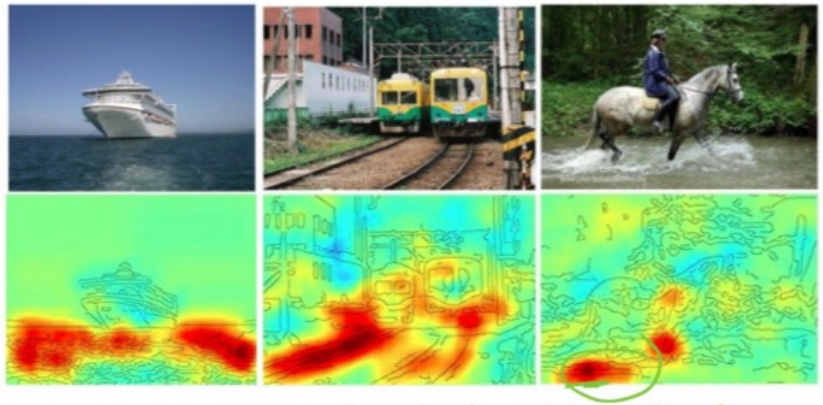
**Explanations are required by regulations for fairness and accountability**

- The EUs General Data Protection Regulation (GDPR) confers a right of explanation for all individuals to obtain meaningful explanations of the logic involved for automated decision making.


Lundberg (2019): Explaining interest rate of loan.

## The Need for XAI: ML Expert Perspective

**Explanations help to determine if predication is based on the wrong reason** (Clever Hans)



Samek (2019)

copyright tag

Wrong cue

# Types of Explanations

**Local vs Global explanations**:

- Local XAI: Explains one particular prediction made by a model.
- Global XAI: Explains general behaviour of a model.

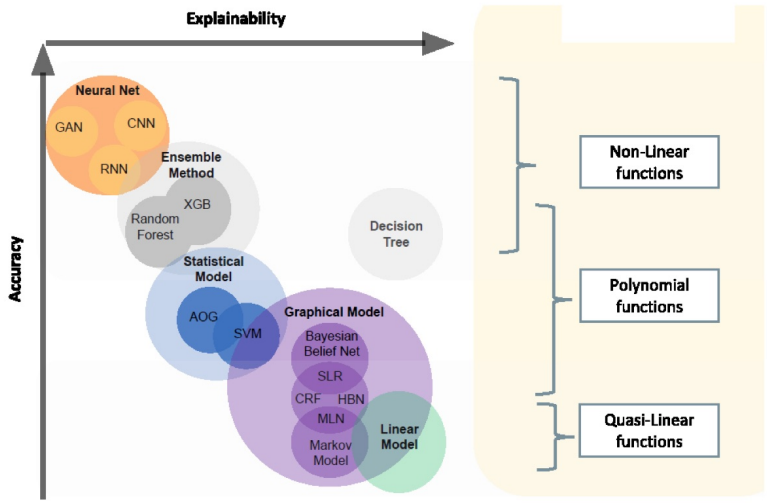
**Model-specific or model-agnostic**:

- Model-agnostic XAI: Treats models as black-box.
- Model-specific XAI: Depends on the type of selected model

**Ante Hoc. vs Post Hoc.**:

- Ante Hoc. XAI: Learn models that are interpretable.
- Post Hoc. XAI: Interpret models that are not interpretable by themselves.

## The Interpretability and Accuracy Tradeoff




Lecue *et al.* (2020)

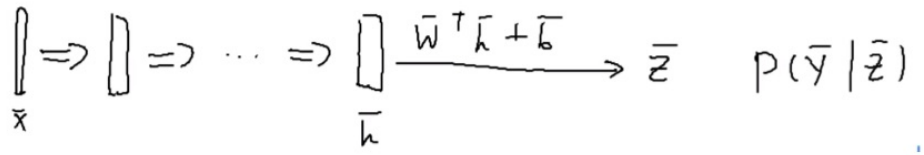
# Models to be explained

- **Image classifiers**
- **Tabular data classifiers**
- Large language models
- Reinforcement learning models
- Clustering algorithms
- ...

An XAI method is typically applicable to multiple models. We will focus on two tasks, image classification and tabular data classification.

# Outline

# The Setup




- An image $\mathbf{x} = (x_1, \ldots, x_D)^\top$ is fed to a DNN to produce a latent feature vector $\mathbf{h}$.
- An affine transformation is performed on $\mathbf{h}$ to get a **logit vector** $\mathbf{z} = (z_1, \ldots, z_C)$, which is used to define a probability distributions over the classes via softmax.
- Question: How important is a pixel $x_i$ to the score $z_c(\mathbf{x})$?
  - **Sensitivity**: How sensitive is the score $z_c(\mathbf{x})$ to changes in $x_i$?
  - **Attribution**: How much does $x_i$ contribution to the score $z_c(\mathbf{x})$?

## Sensitivity vs Attribution

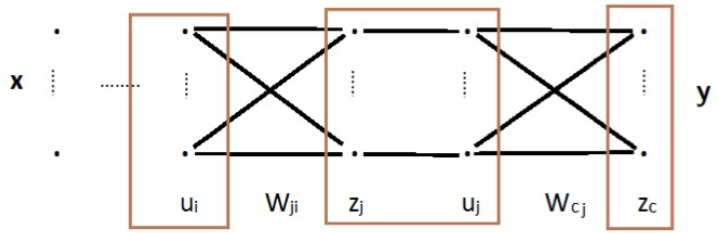
- **Sensitivity**: How sensitive is the score $z_c(\mathbf{x})$ to changes in $x_i$?
- **Attribution**: How much does $x_i$ contribution to the score $z_c(\mathbf{x})$?

# Saliency Map

- In the general case, we can still determine the sensitivity of $z_c$ to $x_i$ using $\frac{\partial z_c}{\partial x_i}$. Vanilla Gradient
- **Saliency Map** ( Simonyan *et al.* 2013) is a way to visualize the gradients w.r.t all the pixel.

A saliency map highlights the pixels that have the largest impact on class score if perturbed.

# Guided Backpropagation

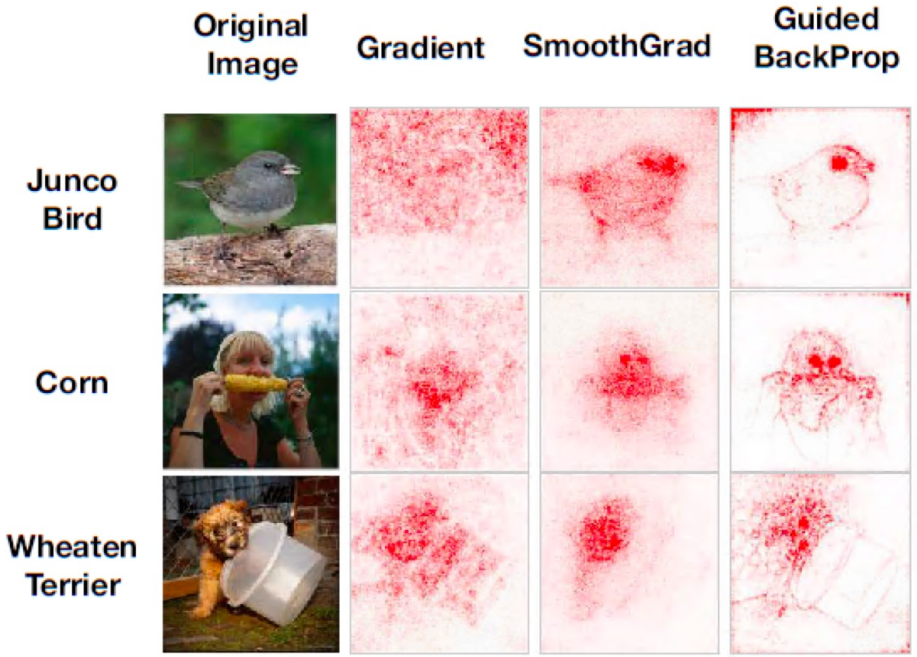



**Vanilla Backprop**: $\frac{\partial z_c}{\partial u_i} \leftarrow \sum_j W_{ji} \frac{\partial u_j}{\partial z_j} \frac{\partial z_c}{\partial u_j}$

- If the gradient $\frac{\partial z_c}{\partial u_j} < 0$, then $u_j (>= 0)$ contributes to $z_c$ negatively.
- If we want to find the pixels the contribution to $z_c$ positively, we can ignore negative gradients.
- This gives rise to **Guided Backpropagation** (Springenberg *et al.* 2014):

$$\frac{\partial z_c}{\partial u_i} \leftarrow \sum_j W_{ji} \frac{\partial u_j}{\partial z_j} ReLU(\frac{\partial z_c}{\partial u_j}).$$

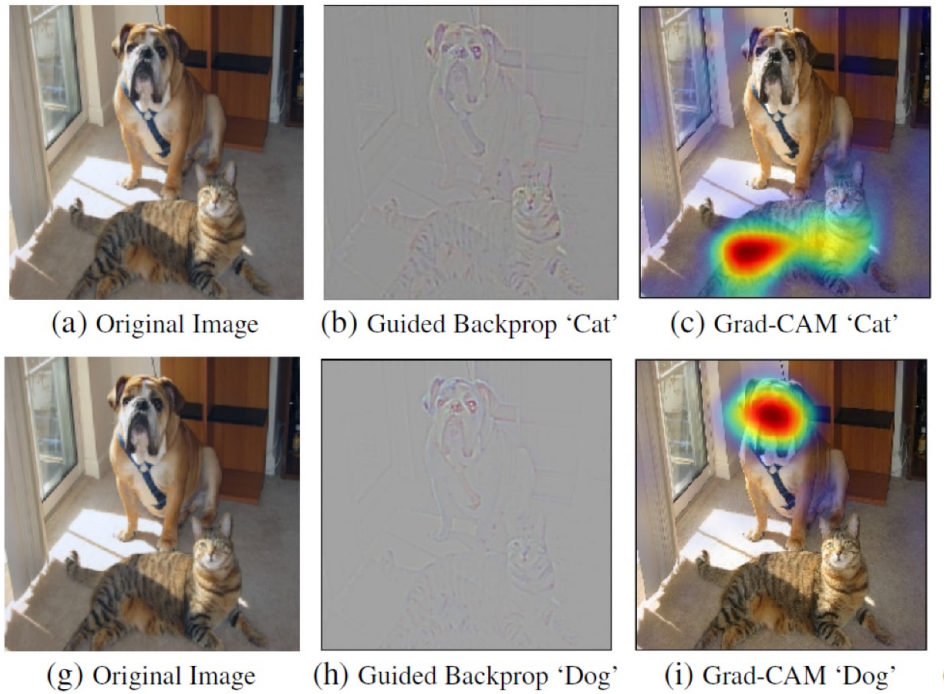
ReLU is applied to gradients at all layers.




Adebayo *et al.* (2018)

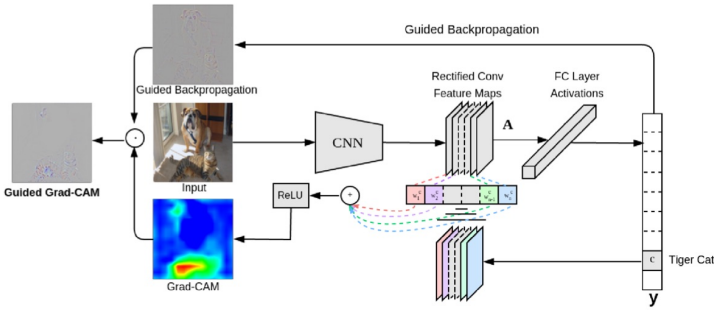
# Grad-CAM (Selvaraju *et al.* 2017)

- Unlike previous methods, **Grad-CAM (Gradient-weighted Class Activation Mapping)** is class-discriminative: It localizes class in the image.

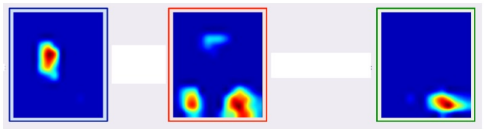


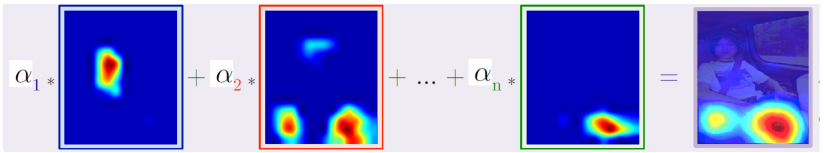
(a) Original Image (b) Guided Backprop 'Cat' (c) Grad-CAM 'Cat'

(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog'

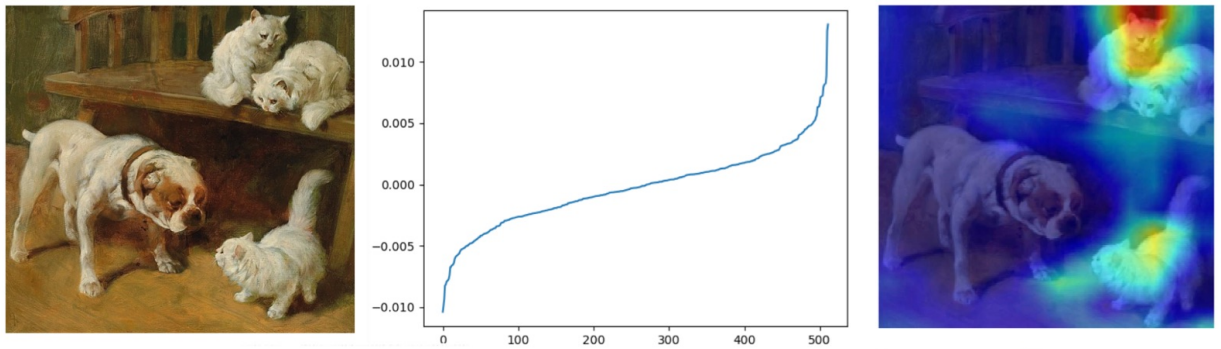



- Let $A^k = [a_{ij}^k]$ be a feature map in the last convolutional layer for an input $\mathbf{x}$. The activations are local.






Grad-CAM essentially combines only those feature maps that contribute positively to $c$, and hence is effective in localize it.



# Importance of Feature Map to Class

# Tutorial

* Tests: dog, zebra, boat
* Model: resnet50 from torchvision
* Get access to a layer:

  register_forward_hook
* Compute heatmap

$$L^c_{Grad-CAM} = ReLU(\sum_k \alpha^c_k A^K).$$

$$\alpha^c_k = \frac{1}{Z} \sum_{i,j} \frac{\partial z_c}{\partial a^k_{ij}},$$

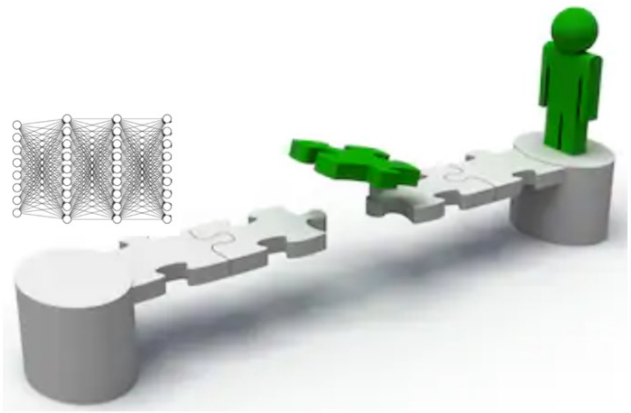
```
handle = model.layer4.register_forward_hook(probe.get_hook())

    last_conv_output = probe.data[0]
    last_conv_output.retain_grad() #make sure the intermediate result sa

    #backprop
    logits[0, target].backward(retain_graph=True)
    grad = last_conv_output.grad
    #taking average on the H-W panel
    weight = grad.mean(dim = (-1, -2), keepdim = True)
    saliency = (last_conv_output * weight).sum(dim = 1, keepdim = True)
    #relu
    saliency = saliency.clamp(min = 0)
```

ReLU

$\sum_k$

## Faithfulness and Interpretability

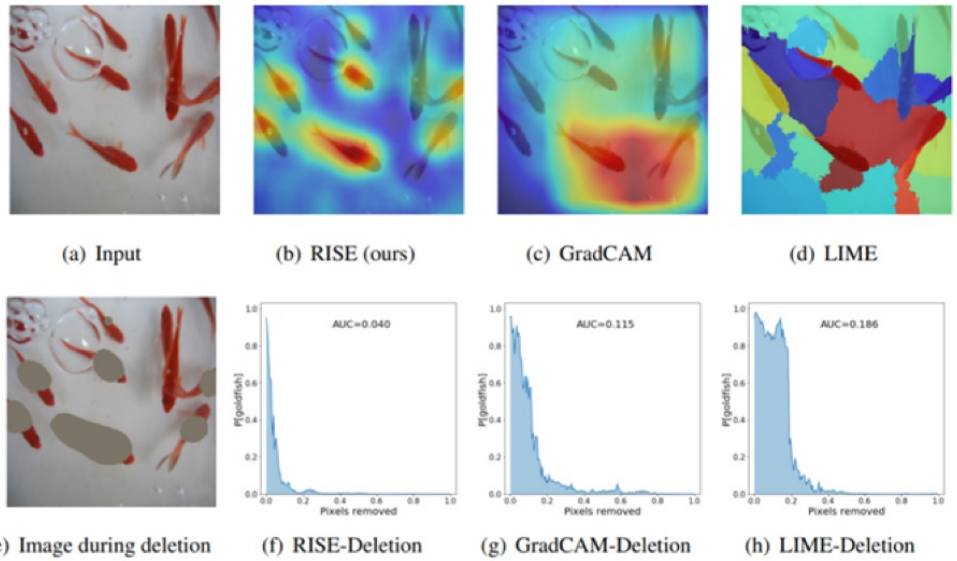
- XAI is a bridge between model and human.



- **Faithfulness**: How well it ties in with the model, revealing key evidence and reasoning that model uses for prediction.
- **Interpretability**: How well it is received by human, providing comprehensible and meaningful information, and allowing a good understanding of model behavior.

# Local Faithfulness

- If input pixels are deemed important, removing them should cause a drop in class probability for the given example (local).
- **Local faithfulness via perturbation** (Samek *et al.* 2016)
  - Sort pixels by importance
  - Perturb them one by one (replace them by random values)
  - Measure drop in probability of true class




(a) Input (b) RISE (ours) (c) GradCAM (d) LIME

(e) Image during deletion (f) RISE-Deletion (g) GradCAM-Deletion (h) LIME-Deletion

# Interpretability via Human Studies Selvaraju *et al.* (2019)

**What do you see?**

**Your options:**

- ○ Horse
- ○ Person

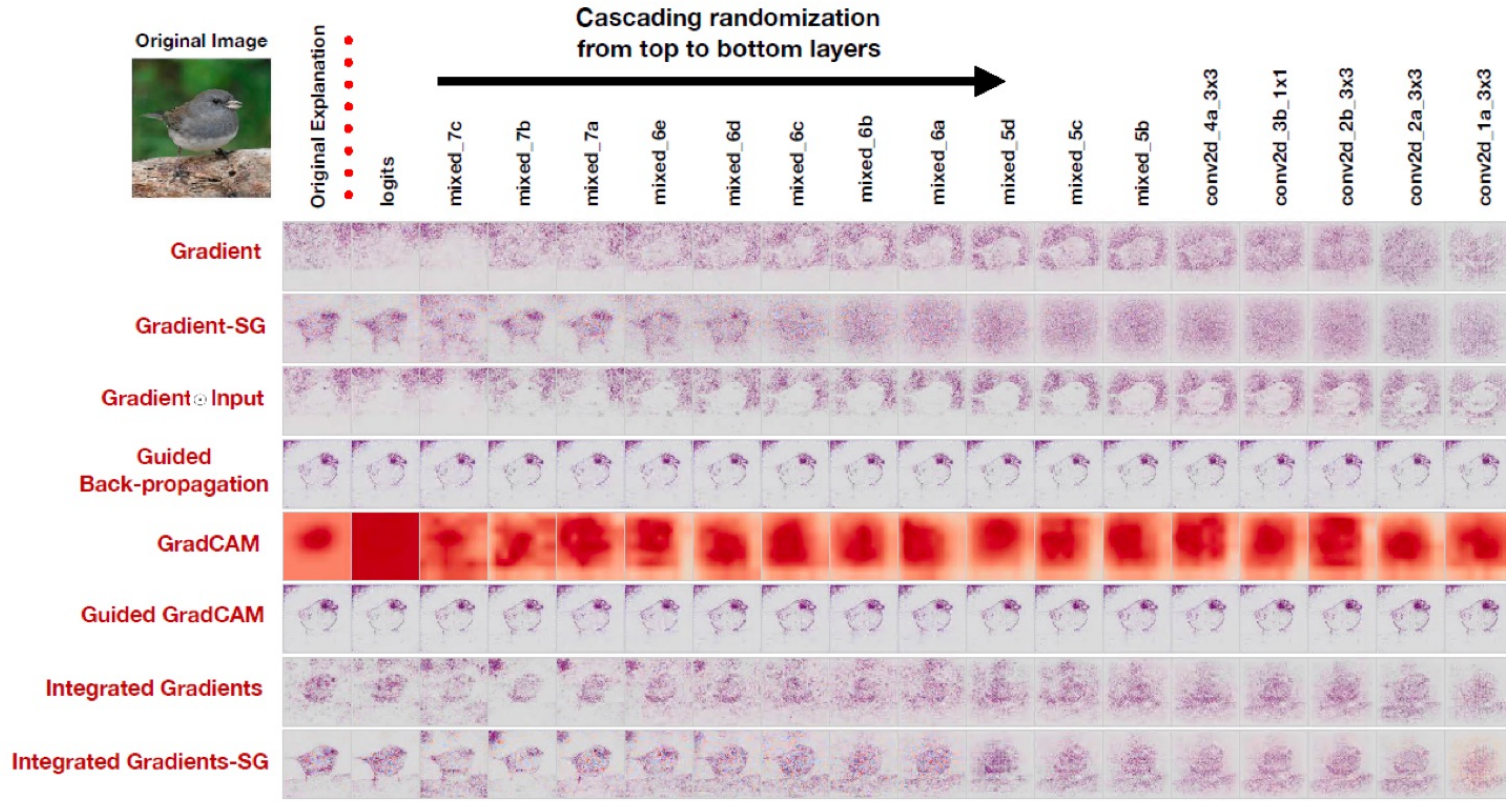
How well can heatmap help user predict behavior of model?

- Heatmaps shown to workers on Amazon Mechanical Turk.
- If human can correctly identify the class predicted by model from heatmap, then the heatmap is meaningful to human and faithful to the model.

| Guided Grad-CAM | Guided Backprop | Deconv Grad-cam | Deconv |
|---|---|---|---|
| 0.61 | 0.44 | 0.60 | 0.53 |

# Sanity Checks for Saliency Maps Adebayo *et al.* (2018)

Some saliency methods give similar results even when many weights are randomly re-initialized. (Inception V3)

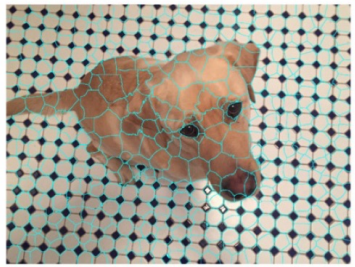


# Outline

## Features

In this lecture, **features** refer to

- **Super-pixels** in images data obtained by standard image segmentation algorithms such as SLIC (Achanta *et al.* 2012).



- **Presence or absence of words** in text data.
- **Input variables** in tabular data.

model f(x)

For XAI

$x$ $z_x = (1, 1, \ldots, 1)$

$z = (0, 1, \ldots, 0)$

$x_z$

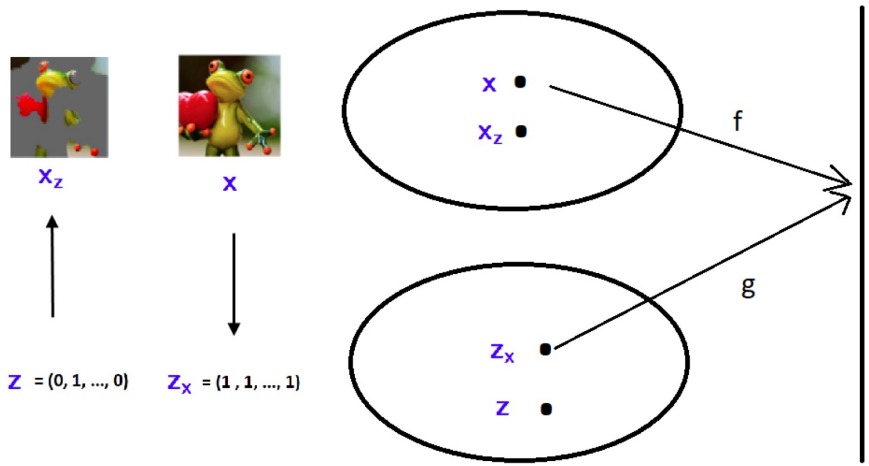
LIME

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

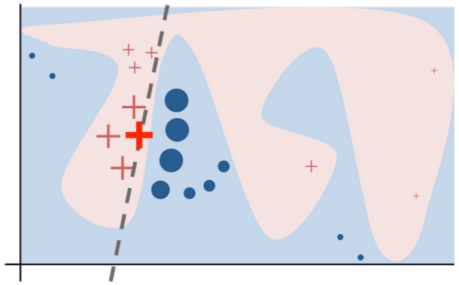
**Figure 4: Explaining an image classification prediction made by Google's Inception neural network. The top 3 classes predicted are "Electric Guitar" ($p = 0.32$), "Acoustic guitar" ($p = 0.24$) and "Labrador" ($p = 0.21$)**

# LIME (Ribeiro *et al.* 2016)

- **LIME** stands for Local Interpretable Model-Agnostic Explanations. It is for explaining a binary classifier.

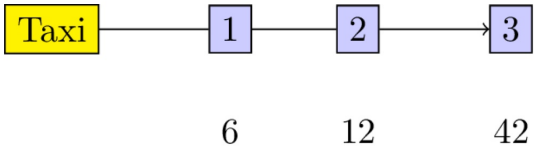



- Although a model might be very complex globally, it can still be faithfully approximated using a linear model locally.



## The Shapley values (Wikipedia)

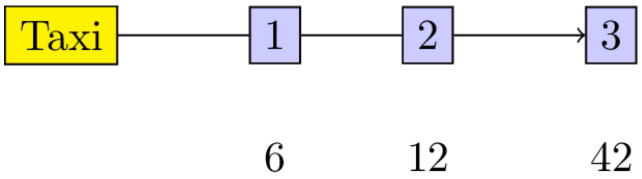
- The Shapley value is a solution concept in cooperative game theory. It was named in honor of Lloyd Shapley, who introduced it in 1951 and won the Nobel Prize in Economics for it in 2012.
- To each cooperative game it assigns a unique distribution (among the players) of a total surplus generated by the coalition of all players.
- The Shapley value is characterized by a collection of desirable properties.




- **Example**: 3 persons share a taxi. Here are the costs for each individual journey:
  - Person 1: 6
  - Person 2: 12
  - Person 3: 42

  How much should each individual contribute?

- A **characteristic function game**: $f : 2^{\{1,\ldots,M\}} \to \mathbb{R}$
  - For any subset $S$ of players, $f(S)$ is their payoff if they act as a coalition.
- **Question**: What is the fair way to divide the total payoff $f(\{1,\ldots,M\})$

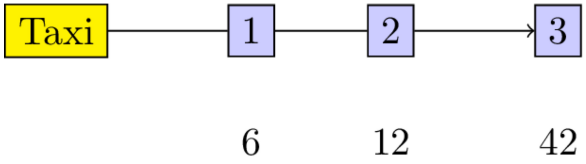



| $S$ | $f(S)$ |
|---|---|
| $\phi$ | |
| $\{1\}$ | |
| $\{2\}$ | |
| $\{3\}$ | |
| $\{1,2\}$ | |
| $\{1,3\}$ | |
| $\{2,3\}$ | |
| $\{1,2,3\}$ | |

- **The Shapley Value** for player $i$ is:

$$\begin{aligned}\phi_i(f) &= \frac{1}{M!} \sum_{\pi:\text{permutation of}} \Delta_\pi^f(i) \\ &= \frac{1}{M!} \sum_{\pi:\text{permutation of } \{1,\ldots,M\}} [f(S_\pi^i \cup i) - f(S_\pi^i)]\end{aligned}$$

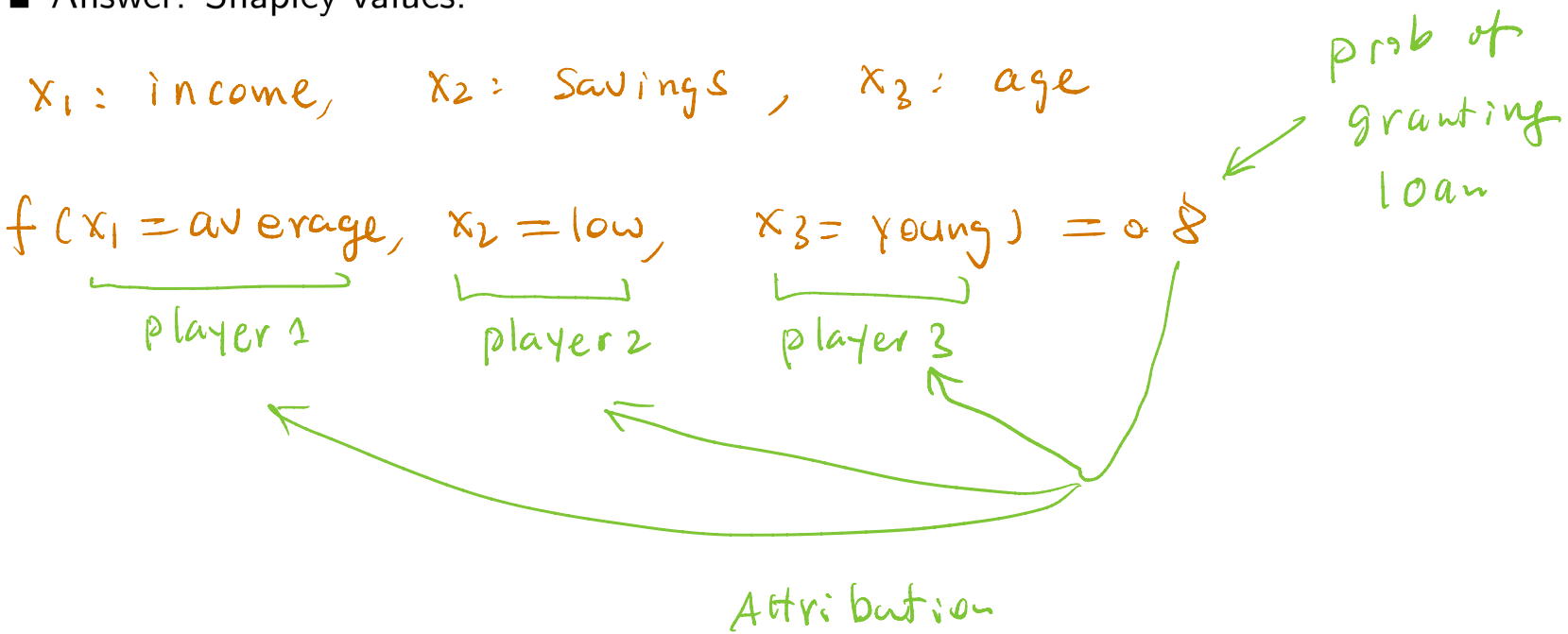
  - $S_\pi$ is the set of predecessors of $i$ in $\pi$,
  - $\Delta_\pi^f(i)$ is the **marginal contribution** of player $i$ w.r.t $\pi$.




| $\pi$ | $\Delta_\pi(1)$ | $\Delta_\pi(2)$ | $\Delta_\pi(3)$ |
|---|---|---|---|
| 1, 2, 3 | | | |
| 1, 3, 2 | | | |
| 2, 1, 3 | | | |
| 2, 3, 1 | | | |
| 3, 1, 2 | | | |
| 3, 2, 1 | | | |

# Use of Shapley Values in XAI (Lundberg and Lee 2017)

- Consider **explaining the prediction $f(\mathbf{x})$ of a complex model $f$ on an input $\mathbf{x}$.**
- We regard each feature $i$ as a player. Their joint "payoff" is $f(\mathbf{x})$.
- How do we divide the "payoff" $f(\mathbf{x})$ among the features?
- Answer: Shapley values.



# Missing Value Imputation

$f(x_1 = \text{average}, x_2 = \text{low}, x_3 = \text{young}) = 0.8$

player 1 player 2 player 3

* The total payoff for 3 "players" is 0.8
* What is the payoff for subset of players?

$f(x_1 = \text{average}, x_2 = \text{low}, \_\_) = ?$

$f(x_1 = \text{average}, \_\_, x_3 = \text{young}) = ?$

$f(\_\_, \_\_, \_\_) = ?$

Need to impute the missing values "__"

- Given a set function $f_{\mathbf{x}} : 2^{\{1,\ldots,M\}} \to \mathbb{R}$, the **Shapley value for feature** $i$ is:

$$\phi_i(f, \mathbf{x}) = \frac{1}{M!} \sum_{\pi:\text{permutation of } \{1,\ \ldots,\ M\}} [f_{\mathbf{x}}(S_\pi^i \cup i) - f_{\mathbf{x}}(S_\pi^i)]$$

  where $S_\pi$ is the set of predecessors of $i$ in $\pi$.

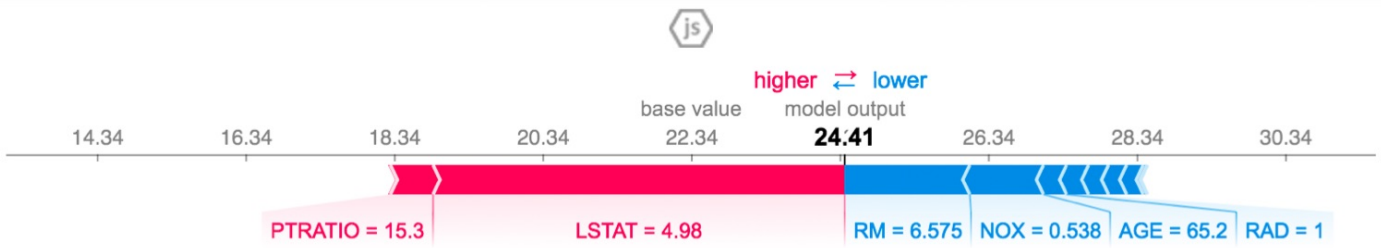
- $\phi_0 = f(\mathbf{x}) - \sum_{i=1}^{M} \phi_i(f, \mathbf{x})$ is the **base value**, the value of $f$ when no feature is present, i.e., $f_{\mathbf{x}}(\emptyset)$.

- **Additive Feature Attribution**:

$$f(\mathbf{x}) = \phi_0 + \sum_{i=1}^{M} \phi_i(f, \mathbf{x})$$

- $\phi_i(f, \mathbf{x})$ is the contribution of feature $i$ to $f(\mathbf{x})$. It is called the **SHapley Additive exPlanation (SHAP) value** of $i$.

# SHAP Values for Individual Input Shown as Force Plots




The above explanation shows features each contributing to push the model output from the base value (the average model output over the training dataset we passed) to the model output. Features pushing the prediction higher are shown in red, those pushing the prediction lower are in blue.