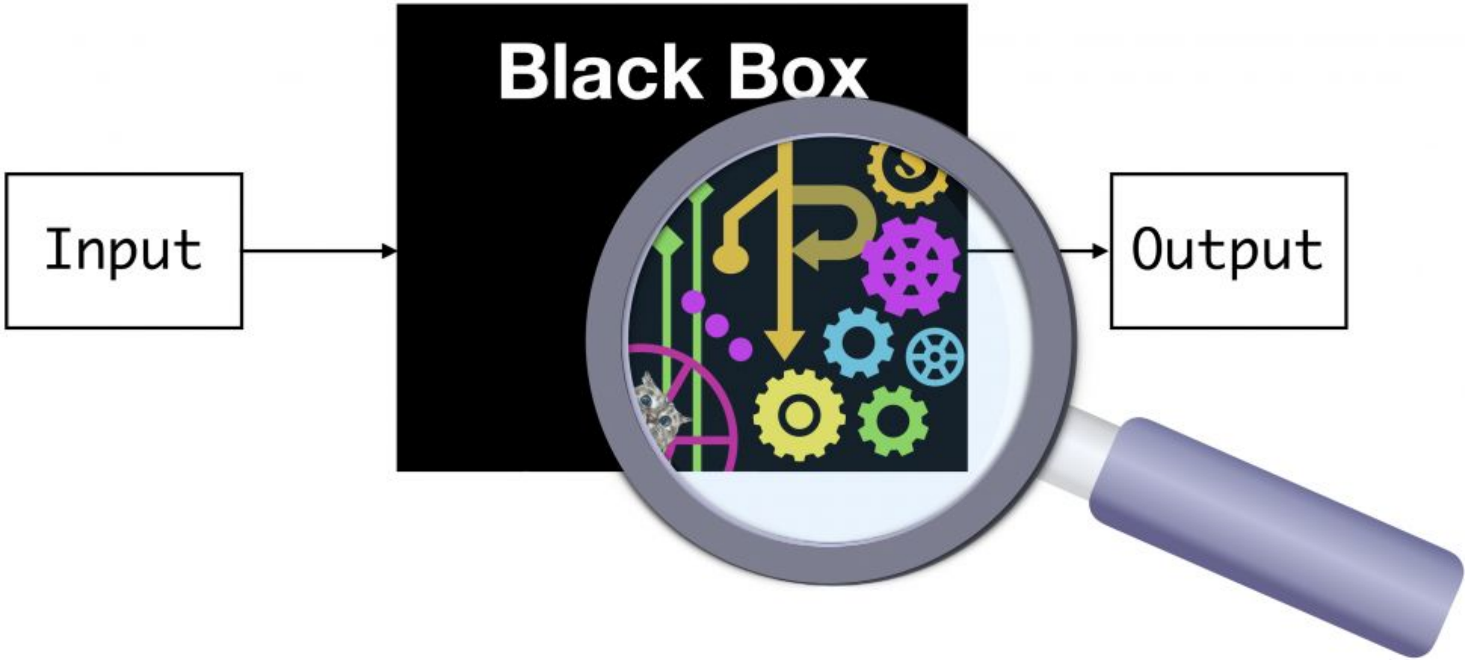
# Deep Learning and Explainability

University of Victoria - PHYS-555


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

# Explainable or Interpretable?



# Terminology (as of today)

- **Interpretability**: a *passive* characteristic of a model referring to the level at which it makes sense to humans.

- ≠ **Explainability**: an *active* characteristic of a model, denoting any action or procedure taken by a model with the intent of clarifying or detailing its internal functions.

- **Explainable AI (XAI)** is a set of tools and frameworks to help humans understand predictions made by AI systems.

# Motivation

Model understanding is absolutely critical in several domains -- particularly those involving *high stakes decisions*

# Motivation: Why Model Understanding?

Input

This model is
incorrect
make
on!! Let
model

Model understanding facilitates debugging.

Predictive
Model

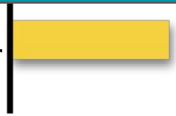
Prediction = Siberian Husky

# Motivation: Why Model Understanding?

This prediction is
biased. Race and
ing
the
!

Defendant Detai

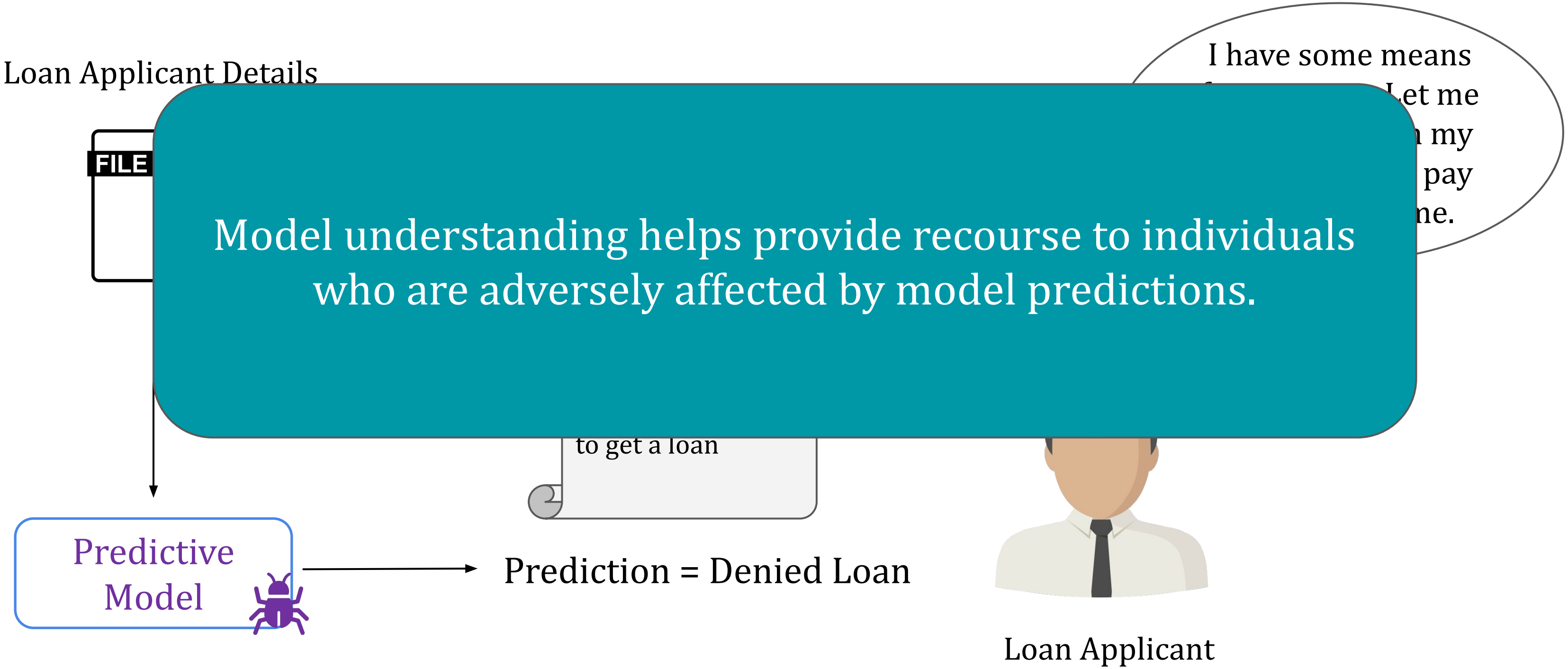
Model understanding facilitates bias detection.




Predictive Model

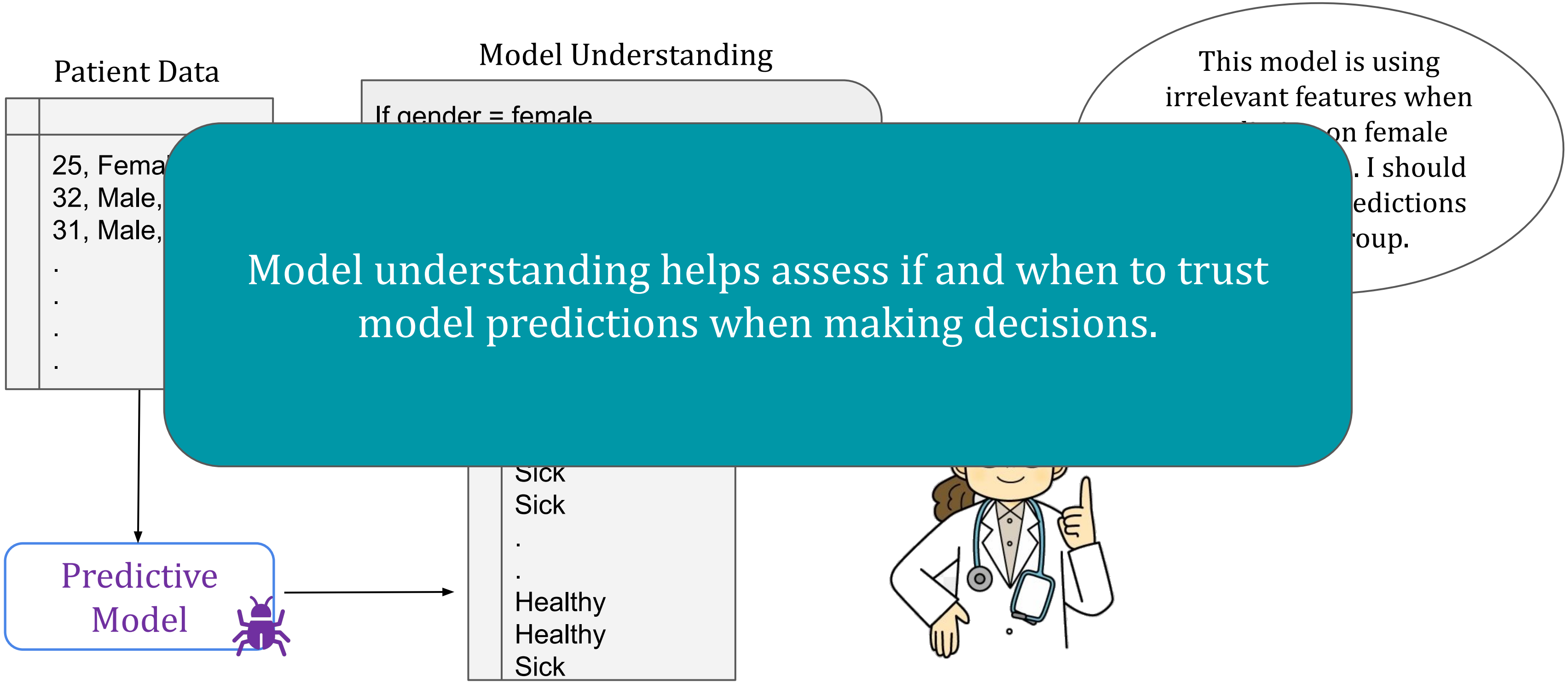
Prediction = Risky to Release

# Motivation: Why Model Understanding?



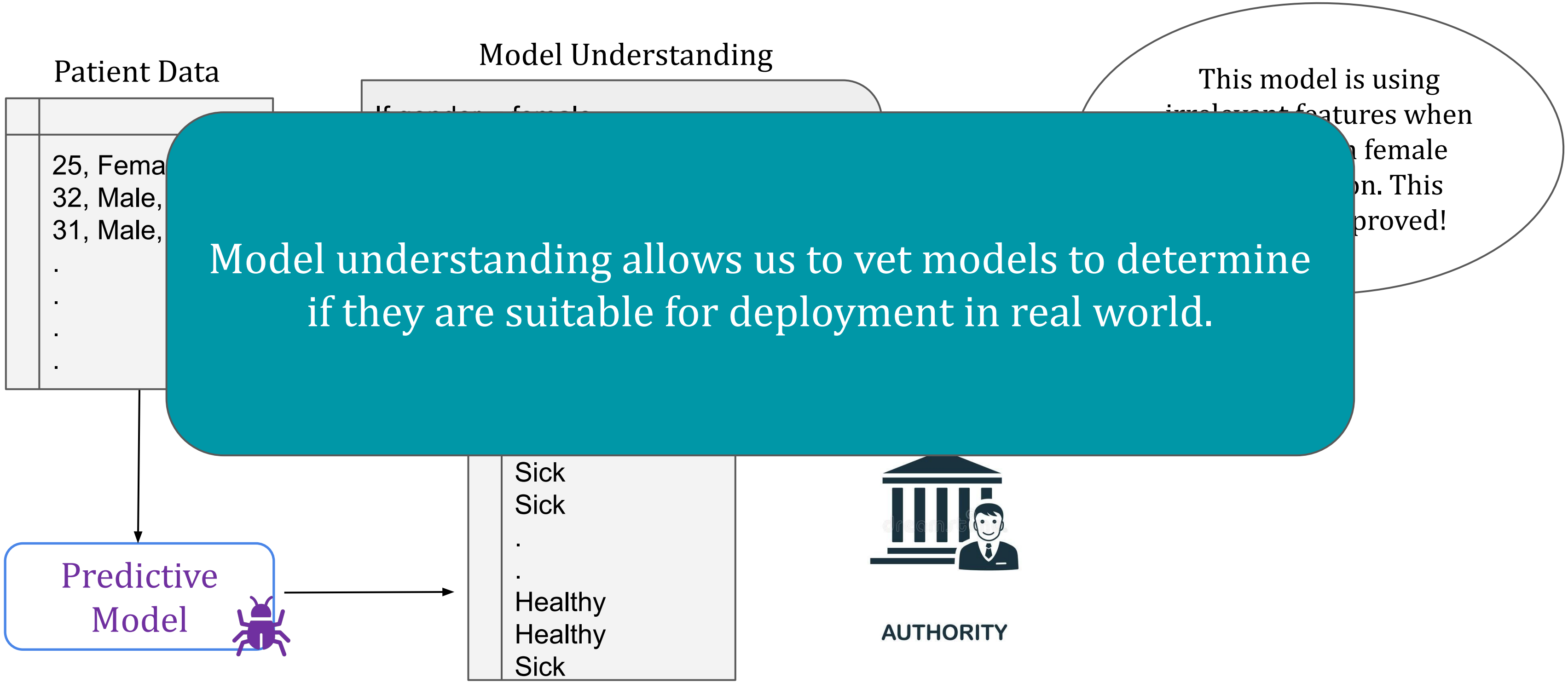

Model understanding helps provide recourse to individuals who are adversely affected by model predictions.

# Motivation: Why Model Understanding?




Model understanding helps assess if and when to trust model predictions when making decisions.

# Motivation: Why Model Understanding?



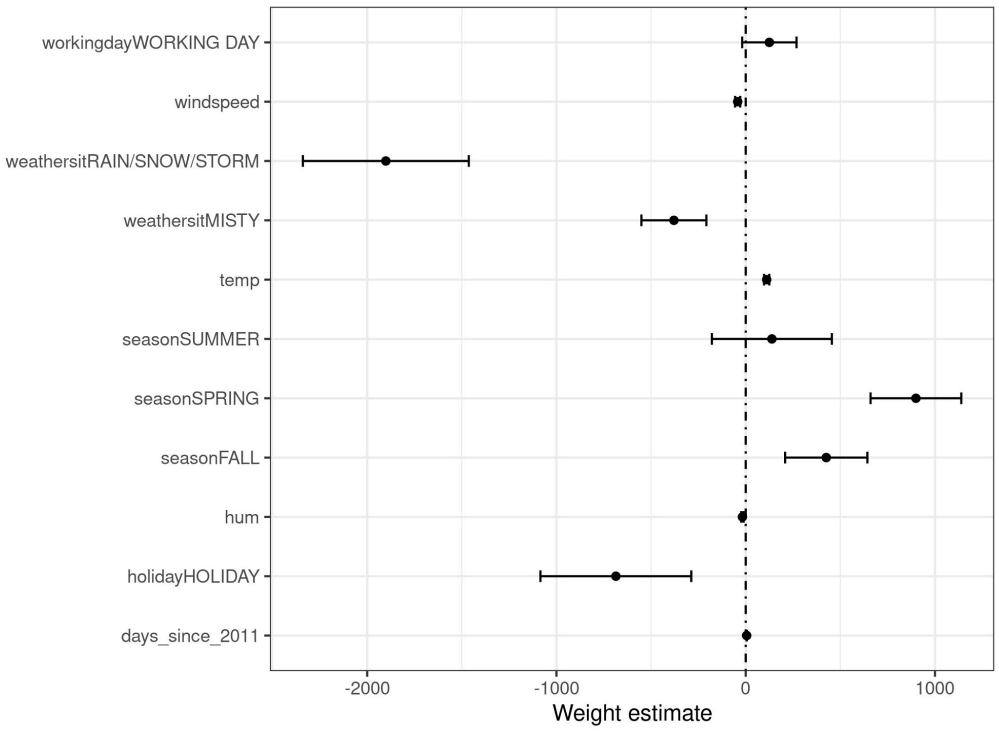

Model understanding allows us to vet models to determine if they are suitable for deployment in real world.

# Linear Models

- Both Logistic regression and linear regression are easily interpretable with the weights

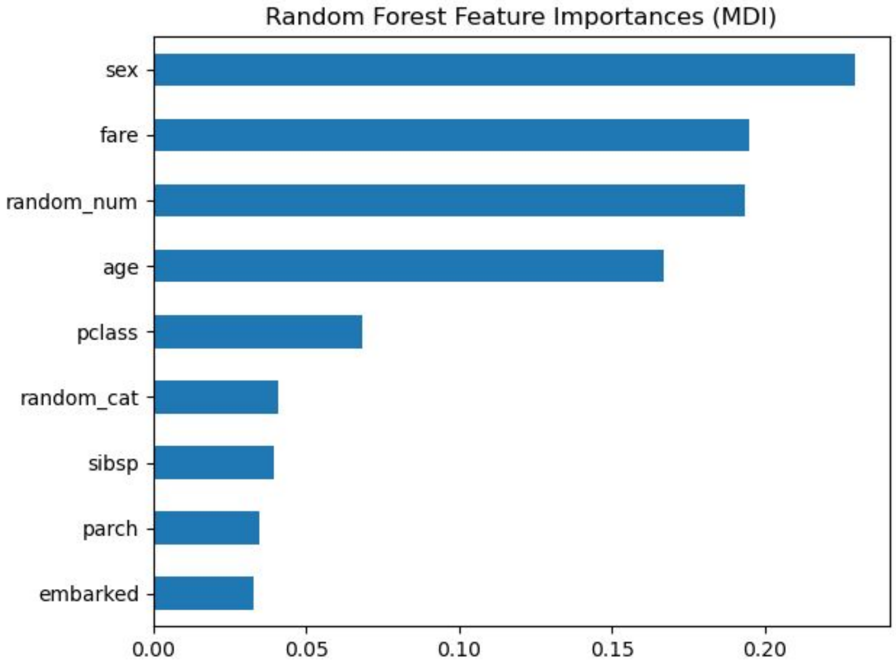
- Proceed with caution with the scale and correlation of features. See [scikit-learn ](scikit-learn) pitfall example.
.

Prediction of rented bikes

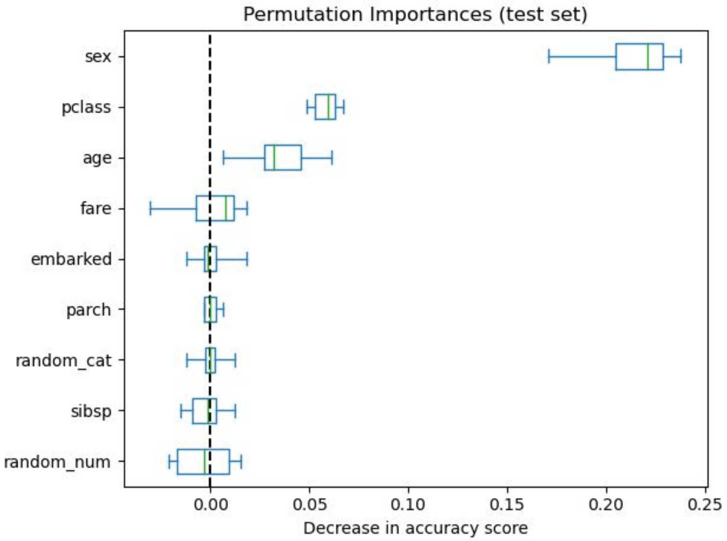


# Decision Trees - Global Explanations

Features Importance with Mean Decrease Impurity

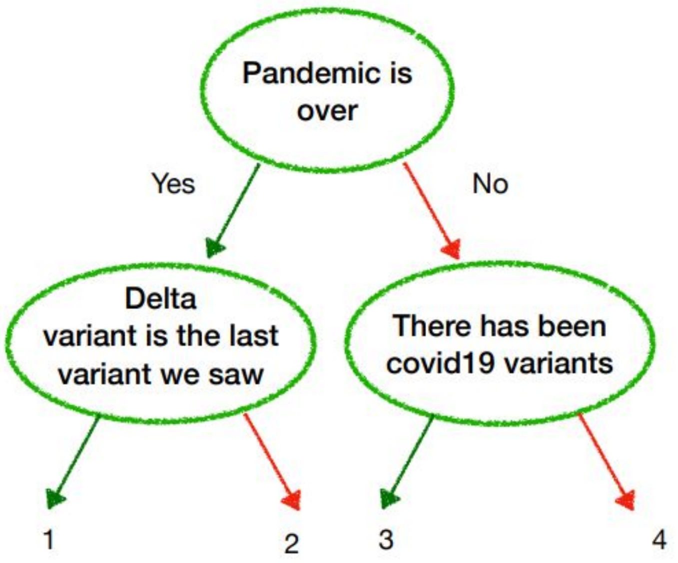



Permutation Importance

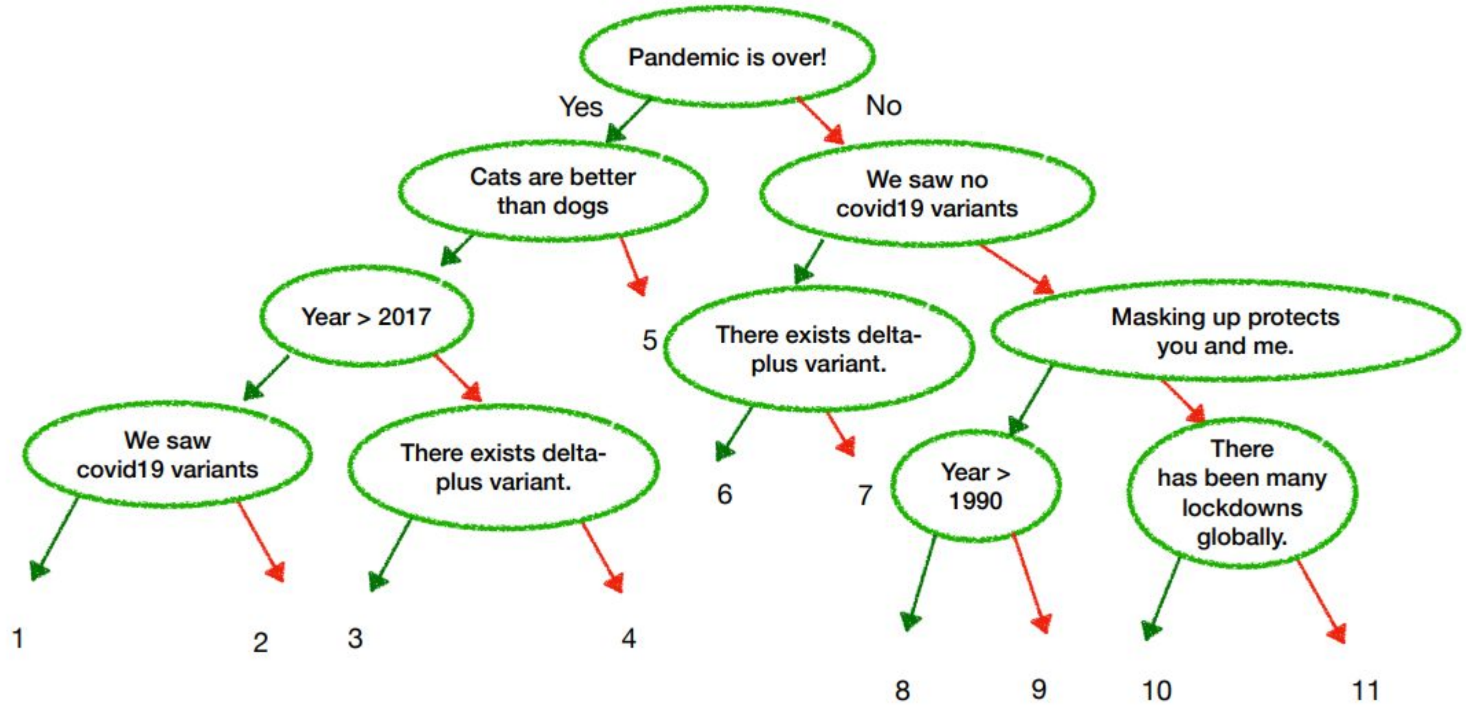



Ex: [scikit-learn RF with Titanic dataset]()

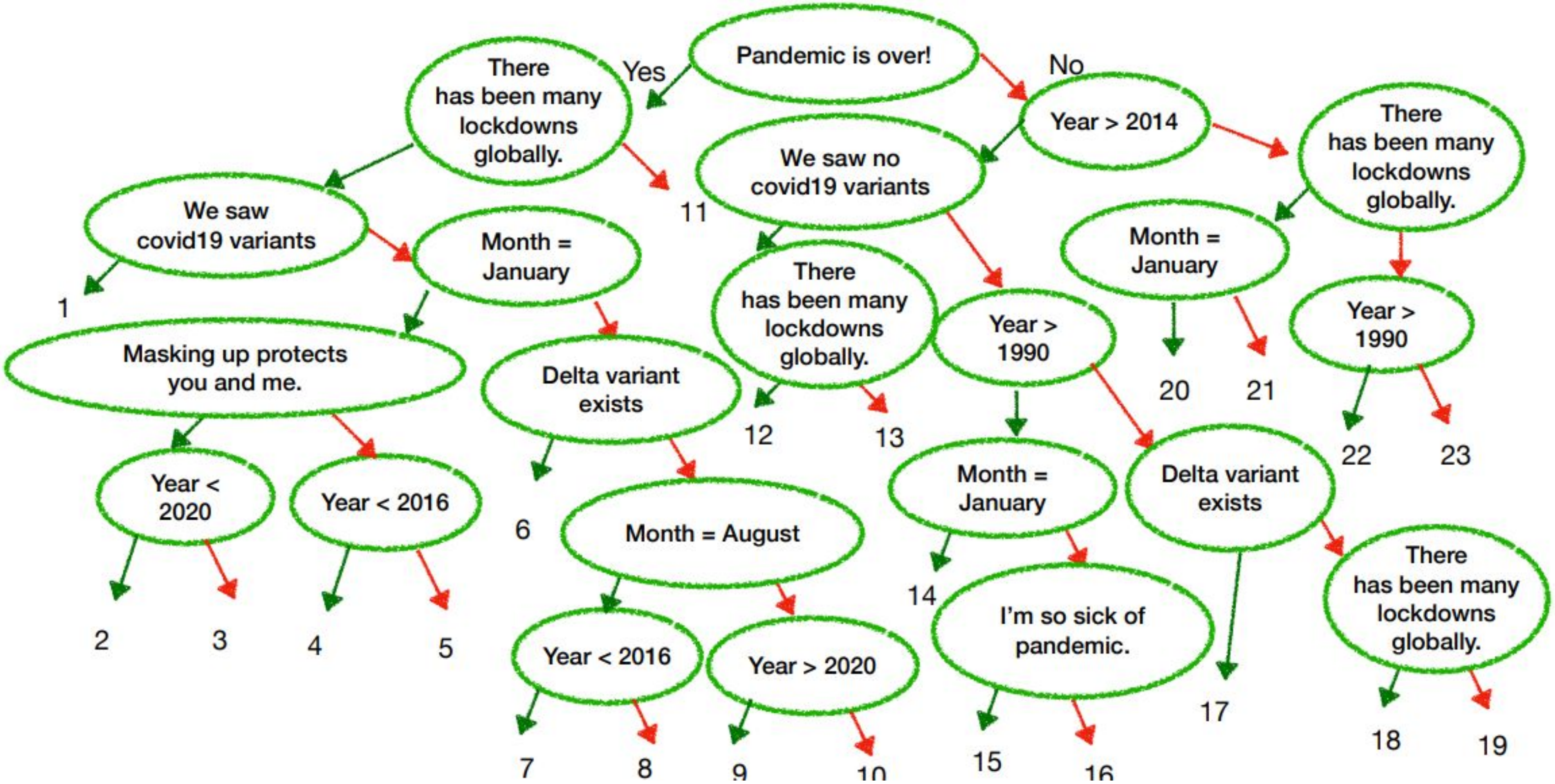
# Sample Tree #1



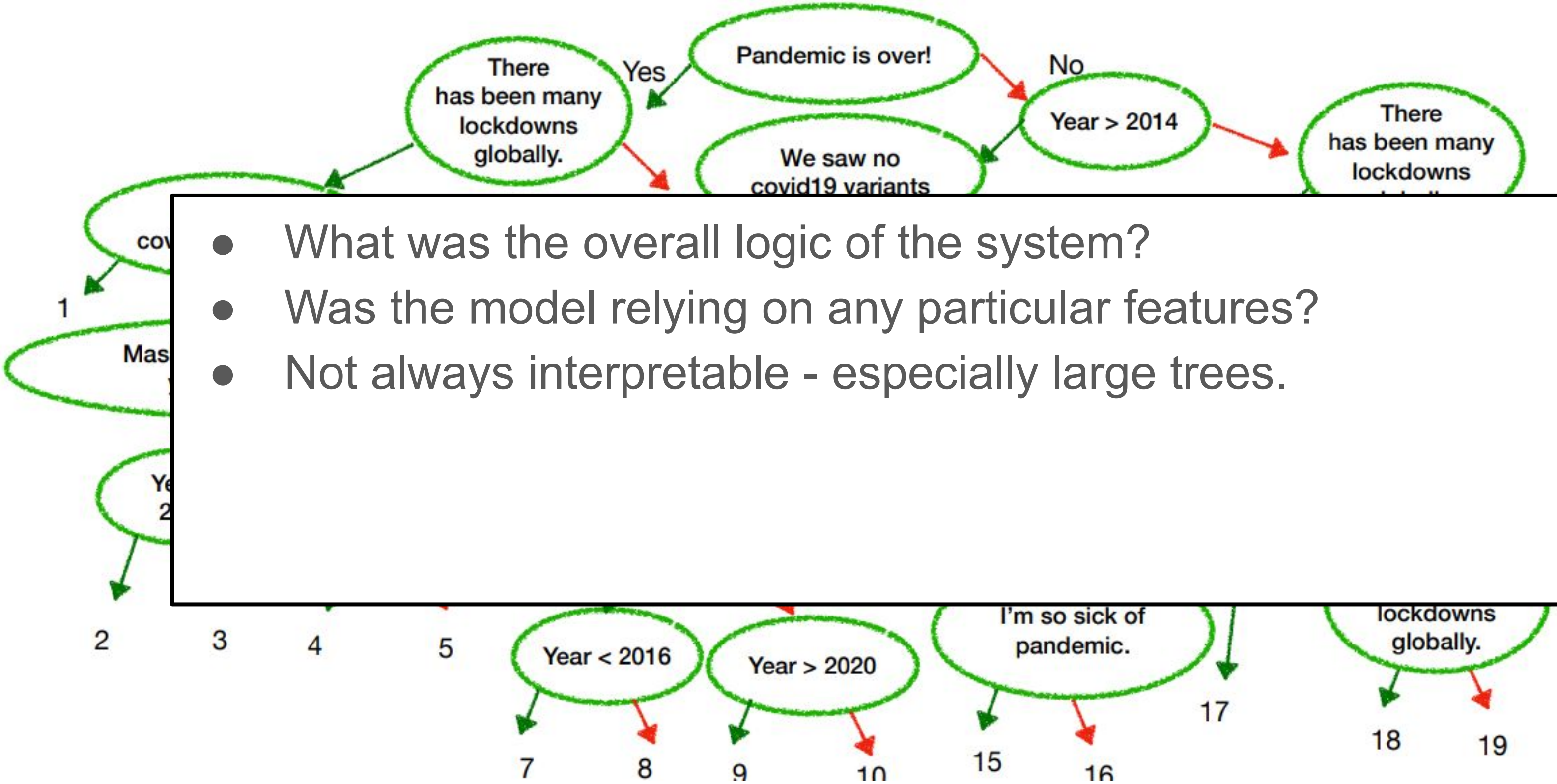
# Sample Tree #2



# Sample Tree #3

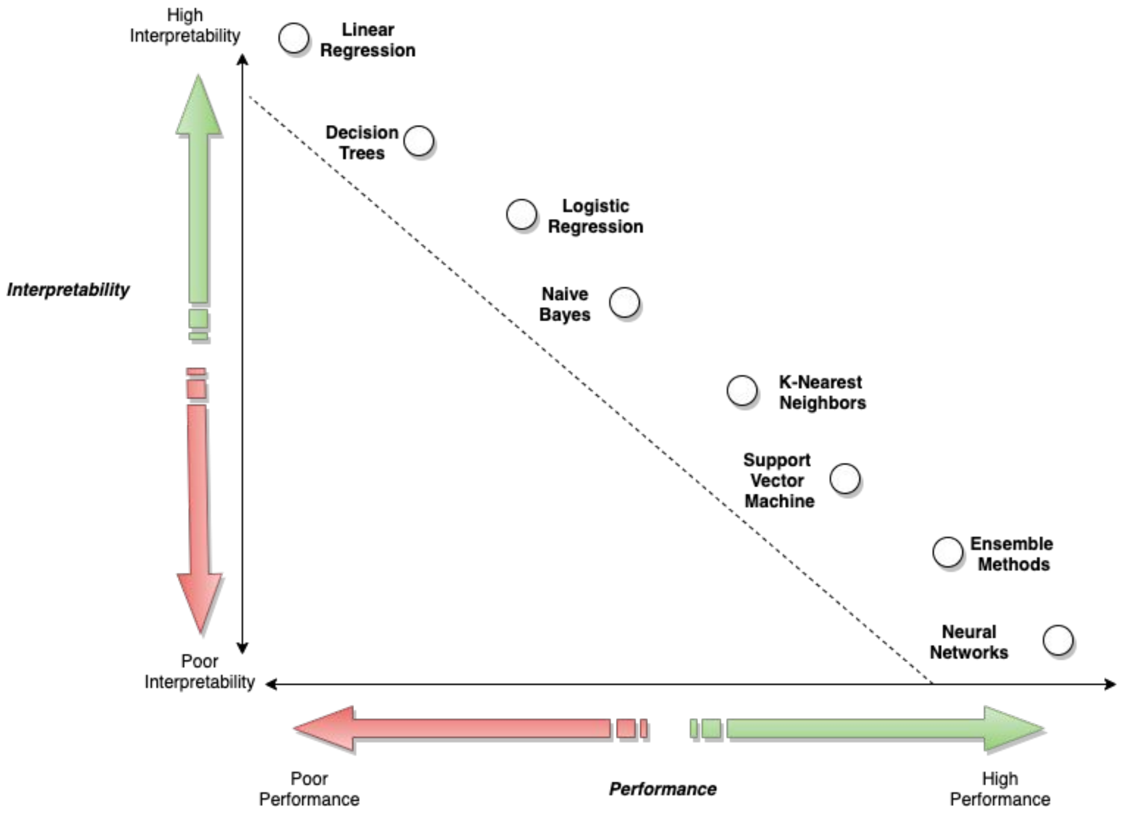


# Sample Tree #3

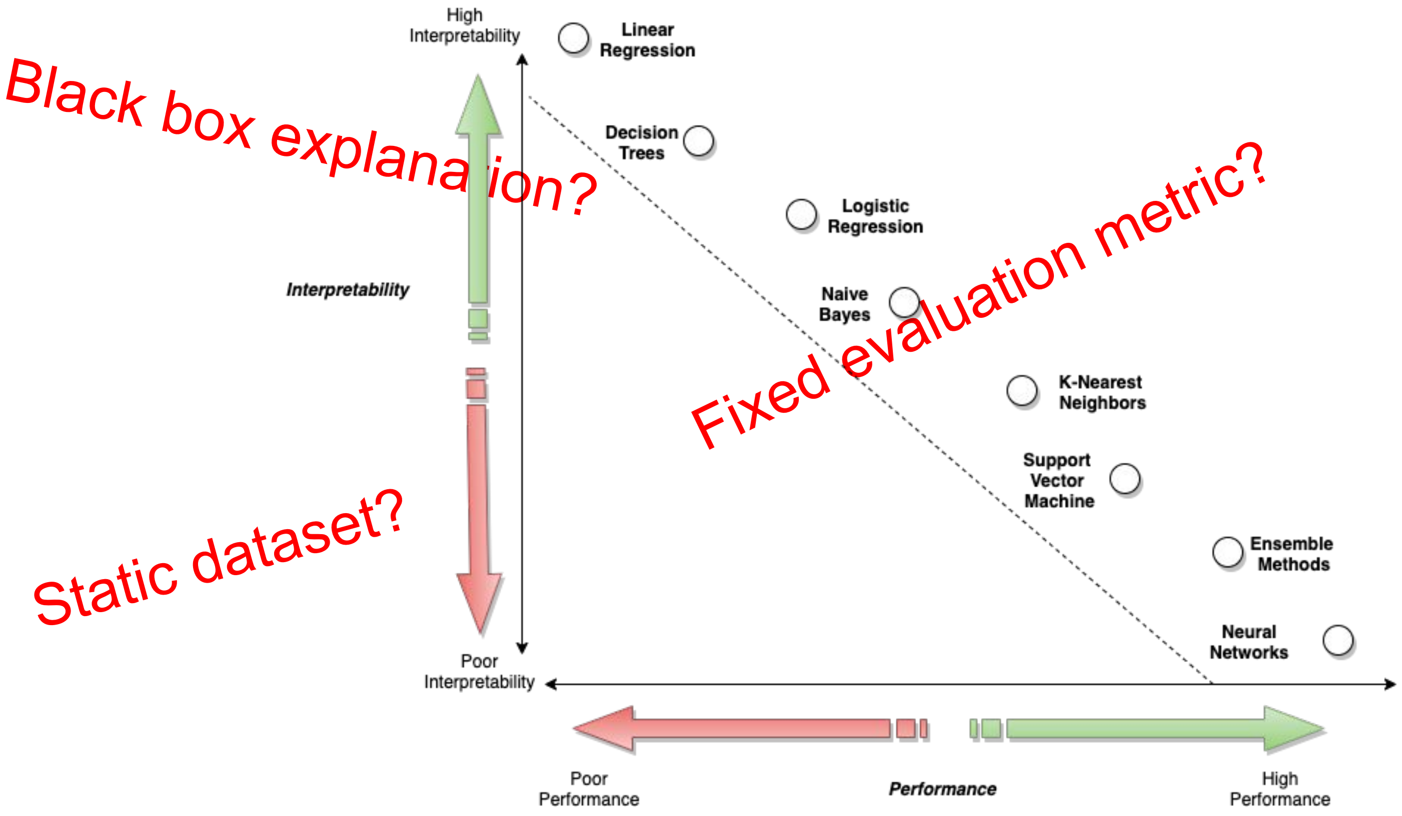



- What was the overall logic of the system?
- Was the model relying on any particular features?
- Not always interpretable - especially large trees.

# Power / Interpretability Tradeoff ?



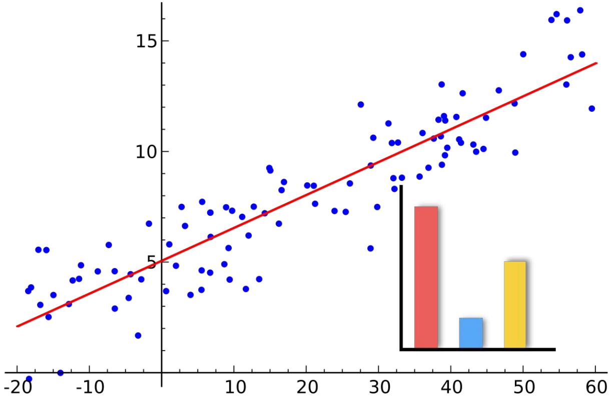
# Power / Interpretability Tradeoff ?

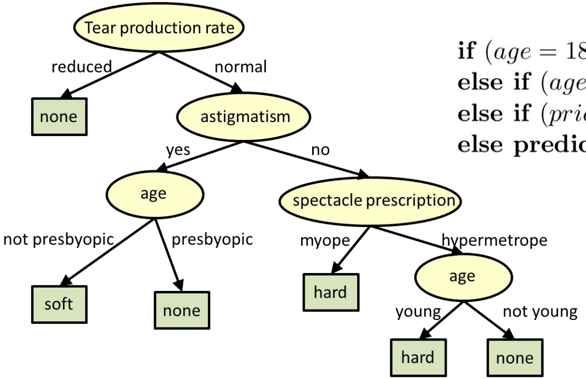


[ Letham and Rudin 2015; Lakkaraju et. al. 2016 ]

# Achieving Model Understanding

Take 1: Build *inherently interpretable* predictive models



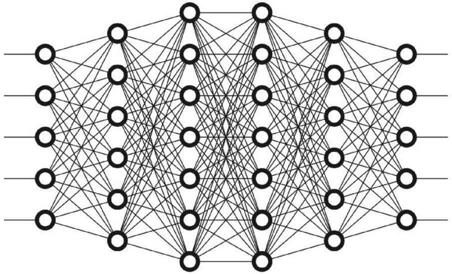




**if** $(age = 18 - 20)$ **and** $(sex = male)$ **then predict** *yes*
**else if** $(age = 21 - 23)$ **and** $(priors = 2 - 3)$ **then predict** *yes*
**else if** $(priors > 3)$ **then predict** *yes*
**else predict** *no*

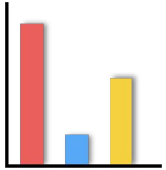
[ Ribeiro et. al. 2016, 2018; Lakkaraju et. al. 2019]

# Achieving Model Understanding

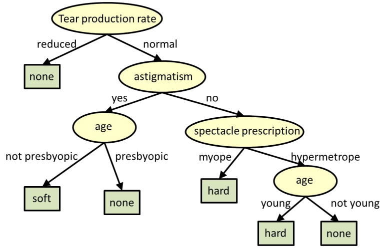
Take 2: *Explain* pre-built models *in a post-hoc manner*



Explainer



**if** $(age = 18 - 20)$ **and** $(sex = male)$ **then predict** $yes$
**else if** $(age = 21 - 23)$ **and** $(priors = 2 - 3)$ **then predict** $yes$
**else if** $(priors > 3)$ **then predict** $yes$
**else predict** $no$



[Rudin 2019]

# Inherently Interpretable Models vs. Post hoc Explanations

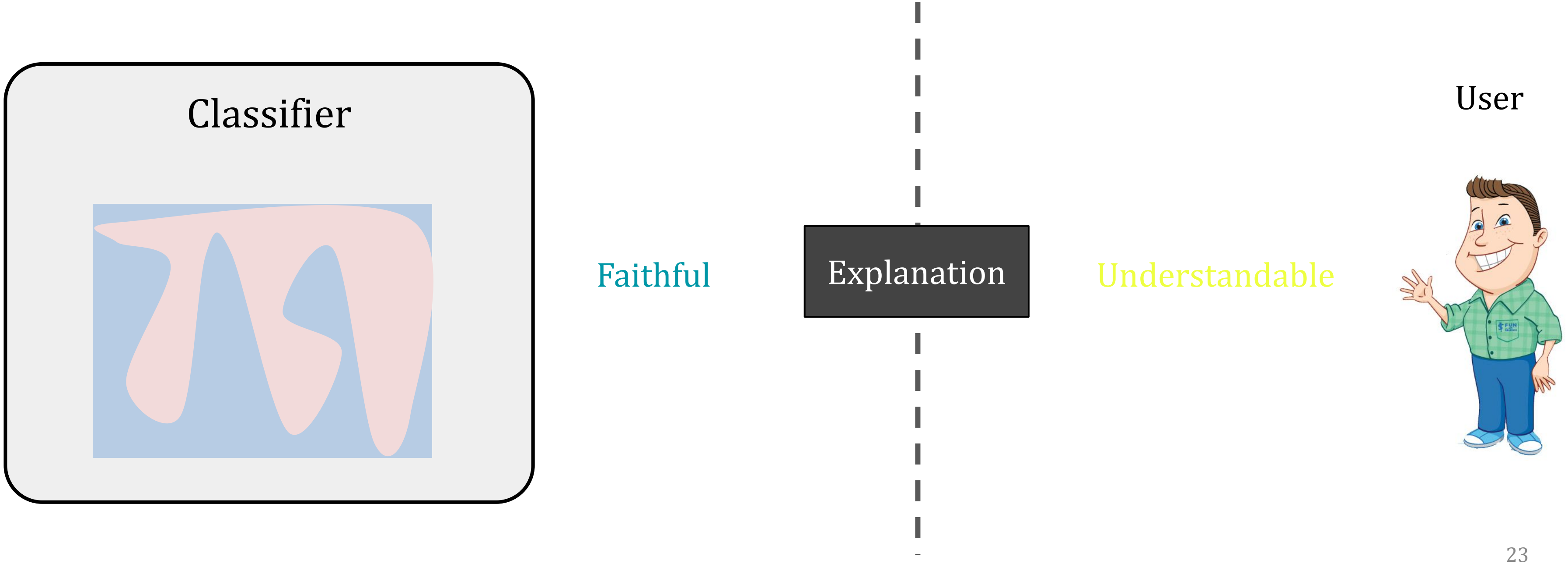
If you *can build* an interpretable model which is also adequately accurate for your setting, DO IT!

Otherwise, ***post hoc explanations*** come to the rescue!

# What is an Explanation?

# What is an Explanation?

Definition: Interpretable description of the model behavior



# What is an Explanation?

Definition: Interpretable description of the model behavior

Classifier

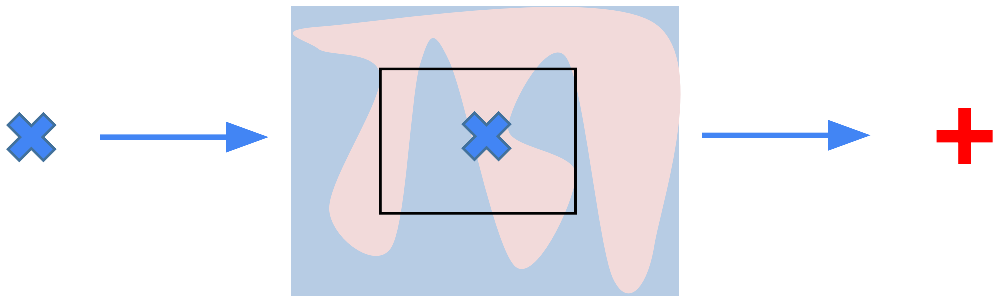


- Send all the model parameters $\theta$?
- Send many example predictions?
- Summarize with a program/rule/tree
- Select most important features/points
- Describe how to *flip* the model prediction
- ...

User

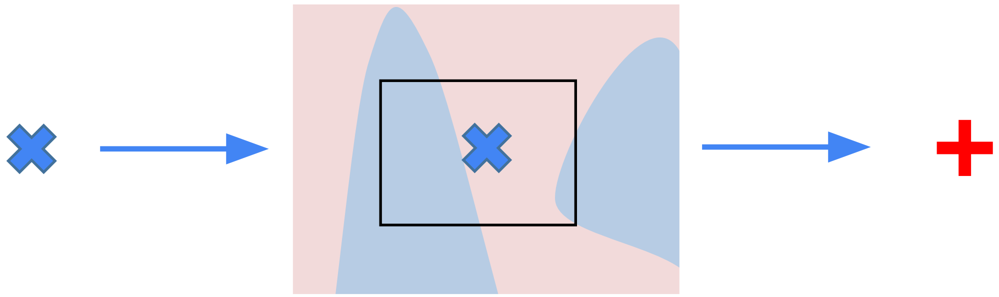
# Local vs. Global Explanations

Global explanation may be too complicated



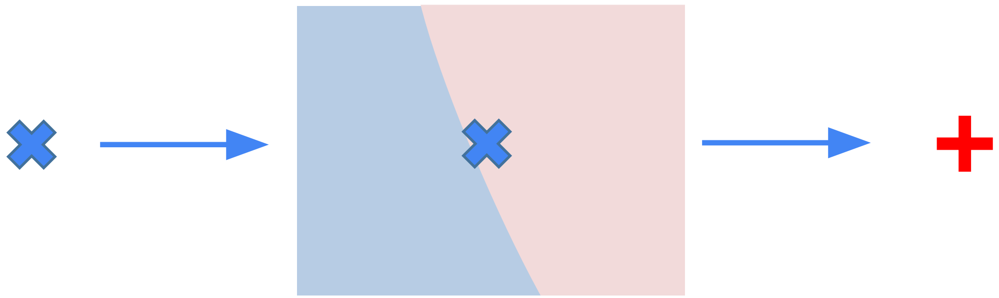
# Local vs. Global Explanations

Global explanation may be too complicated



# Local vs. Global Explanations

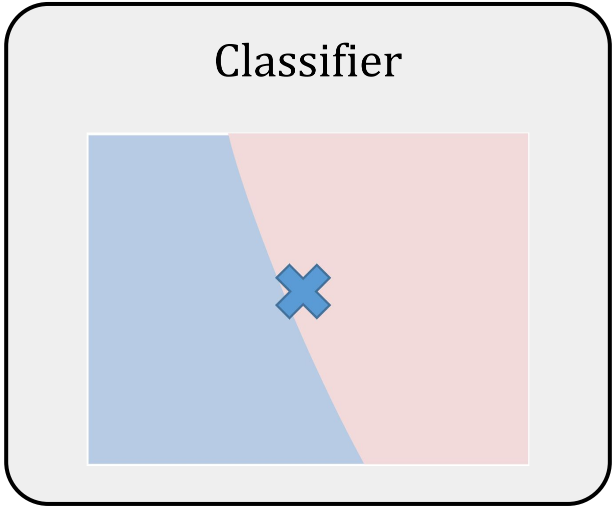
Global explanation may be too complicated



Definition: Interpretable description of the model behavior *in a target neighborhood*.

# Local Explanations

Definition: Interpretable description of the model behavior *in a target neighborhood*.




Send many example predictions?

Summarize with a program/rule/tree

Select most important features/points

Describe how to *flip* the model prediction

...

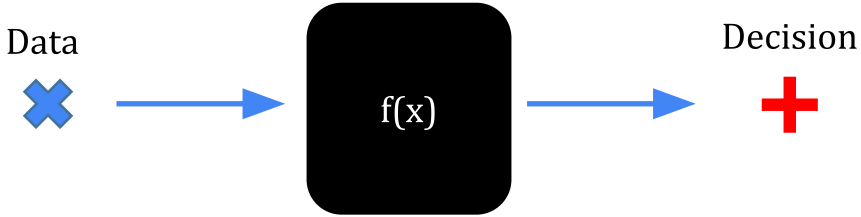
User

# Local Explanations vs. Global Explanations

| Local Explanations | Global Explanations |
| --- | --- |
| Explain individual predictions | Explain complete behavior of the model |
| Help unearth biases in the *local neighborhood* of a given instance | Help shed light on *big picture biases* affecting larger subgroups |
| Help vet if individual predictions are being made for the right reasons | Help vet if the model, at a high level, is suitable for deployment |

# Being Model-Agnostic…

No access to the internal structure…



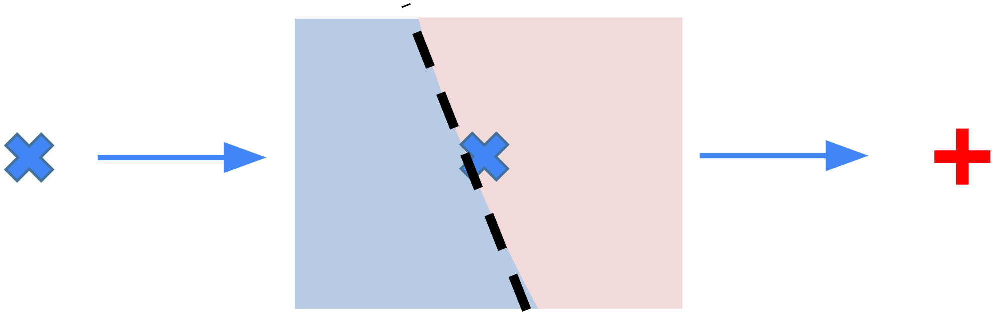

Not restricted to specific models

Practically easy: not tied to PyTorch, Tflow, etc.

Study models that you don't have access to!

# LIME: Sparse, Approximate Linear Explanations

Identify the important dimensions, and present their relative importance

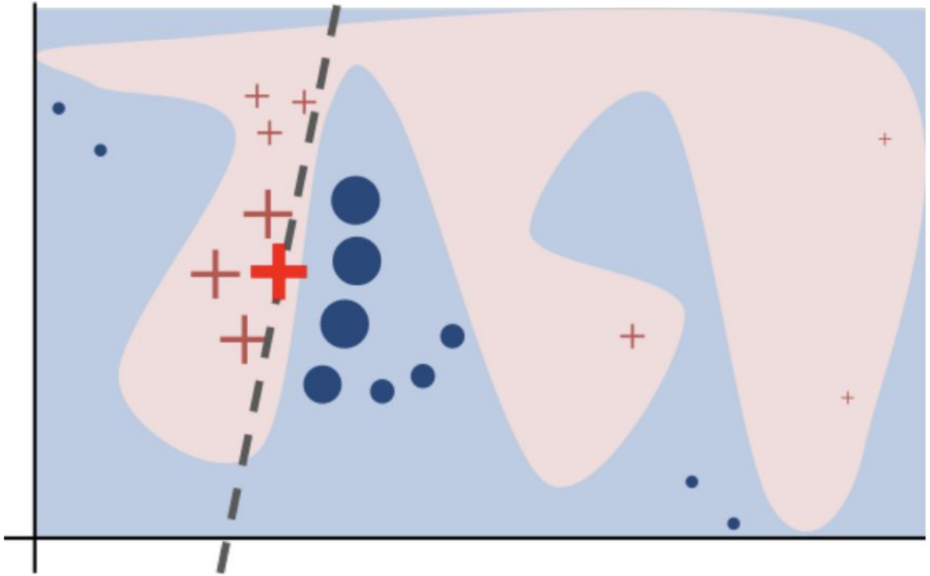


Ribeiro+ (2016)

# LIME

- Approximate model with something explainable (e.g. linear model)

- The approximation only needs to hold locally, i.e. on similar inputs.

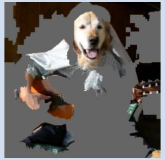


[ Ribeiro et al. 2016 ]

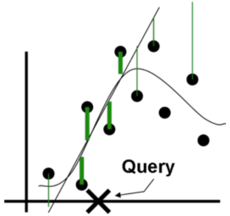
# LIME Example - Images

Original Image

P(labrador) = 0.21

| Perturbed Instances | P(Labrador) |
|---|---|
|  | 0.92 |
|  | 0.001 |
|  | 0.34 |

Maybe to a fault?




Locally weighted regression

Explanation

LIME is quite customizable:

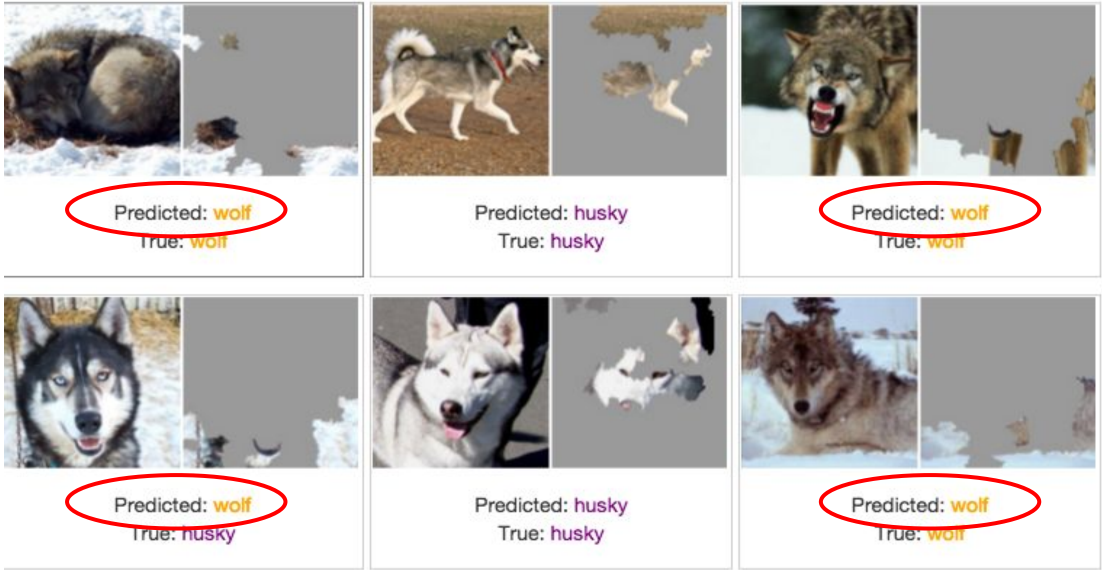
- How to perturb?
- Distance/similarity?
- How *local* you want it to be?
- How to express explanation

[ Ribeiro et al. 2016 ]

# Predict Wolf vs Husky

Only 1 mistake!

# Predict Wolf vs Husky




We've built a great snow detector…

# Shapley Values (from Game Theory)

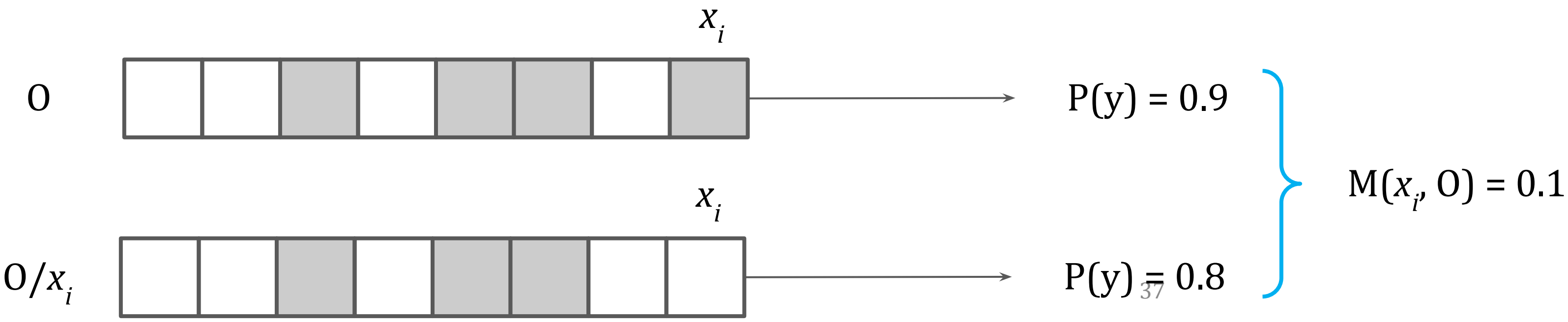
Main Analogy:

Features are “players” who play the cooperative game of making a prediction

- Prediction can be explained by assuming that each feature value of the instance is a “player” in a game.
- The contribution of each player is measured by adding and removing the player from all subsets of the rest of the players.
- The Shapley Value for one player is the weighted sum of all its contributions.

# SHAP: Shapley Values as Importance

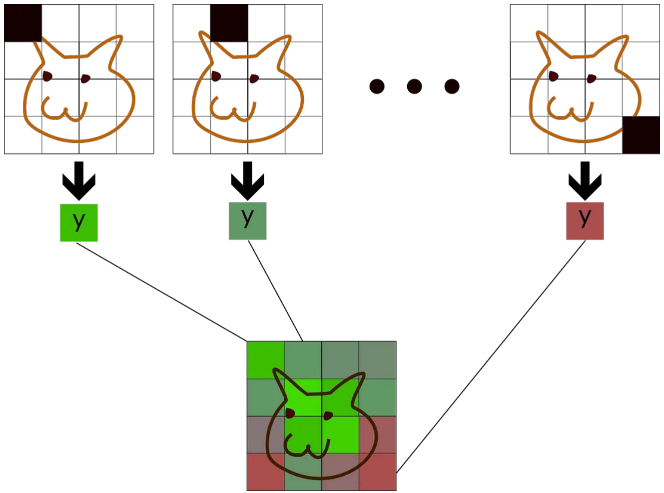
Marginal contribution of each feature towards the prediction, averaged over all possible permutations.

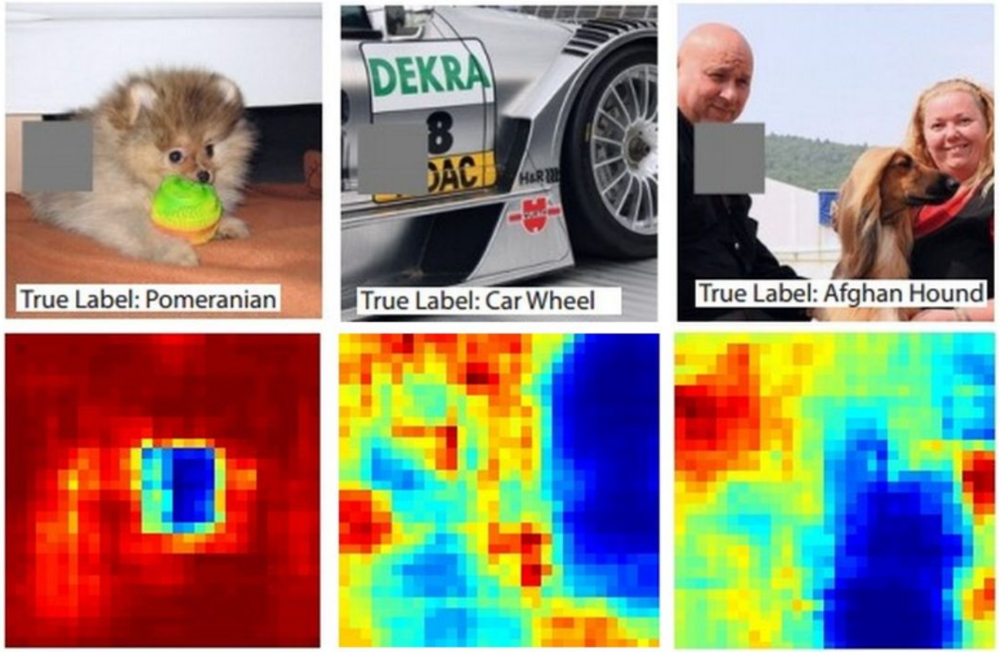



Fairly attributes the prediction to all the features.

[ Lundberg & Lee 2017 ]

# Explanation by Occlusion




Three input images (top). Notice that the occluder region is shown in grey. As we slide the occluder over the image we record the probability of the correct class and then visualize it as a heatmap (shown below each image). For instance, in the left-most image we see that the probability of Pomeranian plummets when the occluder covers the face of the dog, giving us some level of confidence that the dog's face is primarily responsible for the high classification score. Conversely, zeroing out other parts of the image is seen to have relatively negligible impact.

Zeiler+ (2013)

# Feature/Pixel Attribution

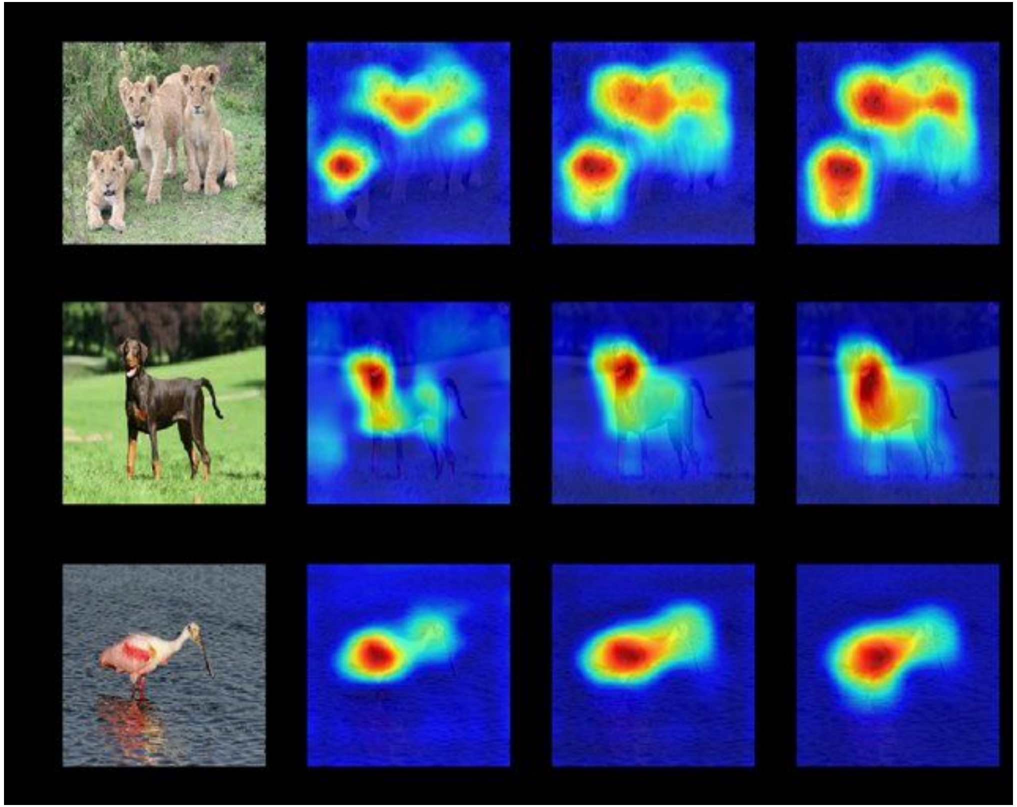
A standard neural network implicitly “focus” its attention on relevant input features to obtain the target outputs.

Can we check where it focused?

See the [CNN Explainer demo]

# Saliency Maps / Feature Attribution

Task: Classification
Network: CNN



# Neural Network Jacobian

**x** = input vector, size k

**y** = output vector, size m

**J** = m X k Jacobian

Compute with back-prop:

set output 'errors' = output activations

$$J_{ij} = \frac{\partial y_i}{\partial x_j}$$

$$J = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \frac{\partial y_1}{\partial x_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial y_m}{\partial x_1} & \cdots & \frac{\partial y_m}{\partial x_k} \end{bmatrix}$$

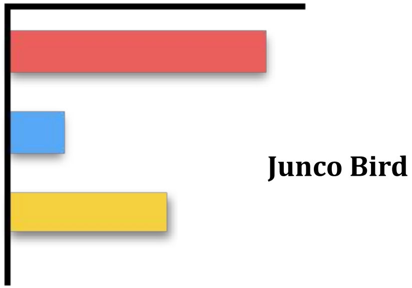
# Saliency Map Overview

**Input**

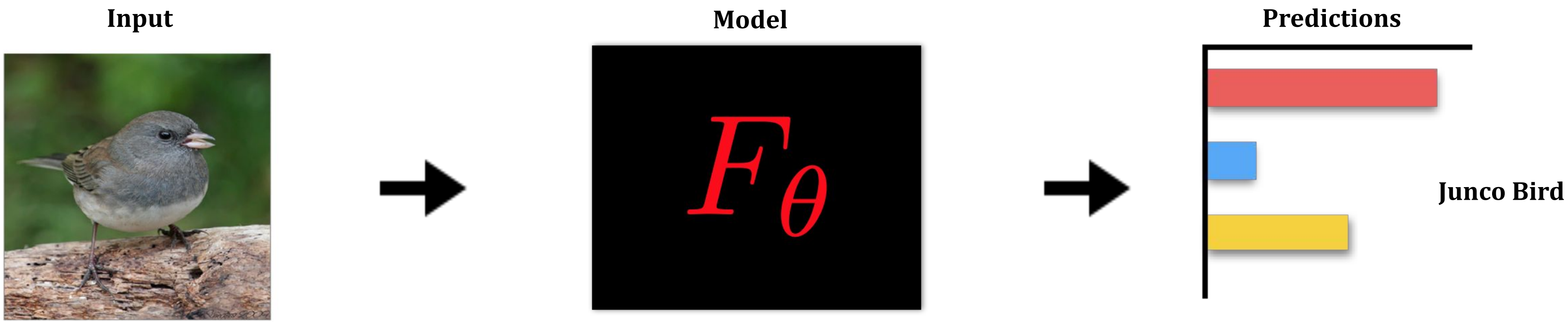
**Model**

$F_\theta$

**Predictions**

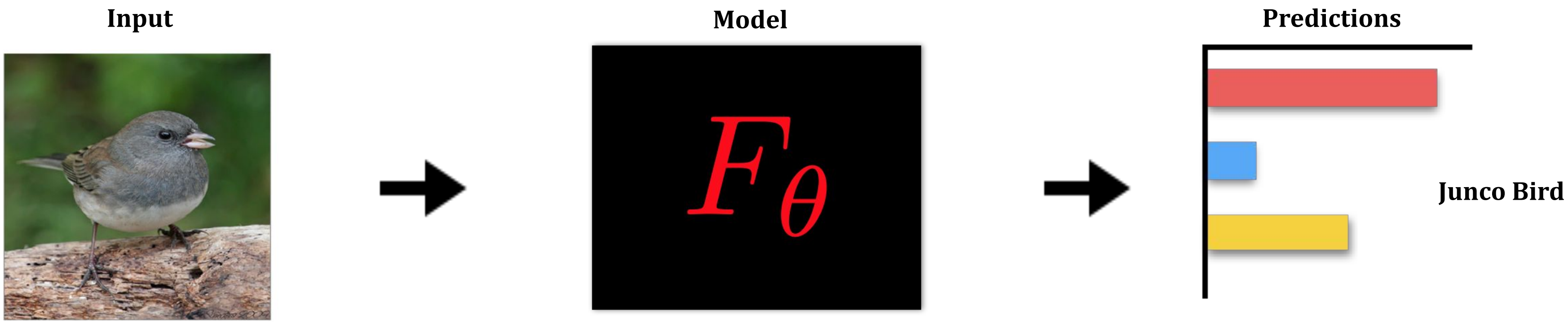


# Saliency Map Overview

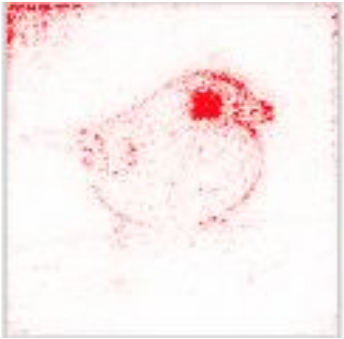



**What parts of the input are most relevant for the model's prediction: ‘Junco Bird’?**

# Saliency Map Overview




**What parts of the input are most relevant for the model's prediction: 'Junco Bird'?**



- **Feature Attribution**
- **'Saliency Map'**
- **Heatmap**

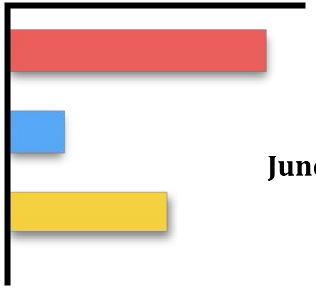
# Input-Gradient

Input

Model

$$F_\theta$$

Predictions

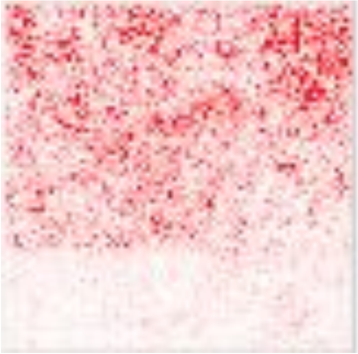



Input-Gradient

$$\nabla_x F_i(x)$$

Logit

Input



Visualize as a heatmap



Baehrens et. al. 2010; Simonyan et. al. 2014 .

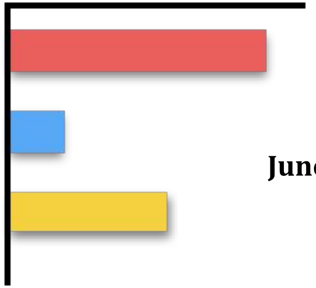
# Input-Gradient

**Input**

**Model**

$F_\theta$

**Predictions**



**Junco Bird**

**Input-Gradient**

$$\nabla_x F_i(x)$$

**Logit**

**Input**

**Challenges**

- **Visually noisy & difficult to interpret.**
- **'Gradient saturation.'**

[Shrikumar et. al. 2017](#).

[Baehrens et. al. 2010](#); [Simonyan et. al. 2014](#) .

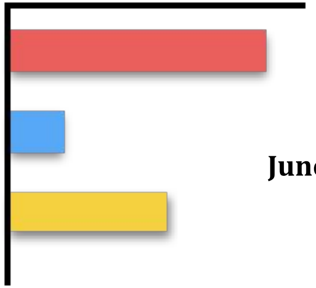
# SmoothGrad

**Input**

**Model**

$F_\theta$

**Predictions**




**SmoothGrad**

$$\frac{1}{N}\sum_{i}^{N}\nabla_{(x+\epsilon)}F_i(x+\epsilon)$$

**Gaussian noise**

**Average Input-gradient of 'noisy' inputs.**

[Smilkov et. al. 2017]

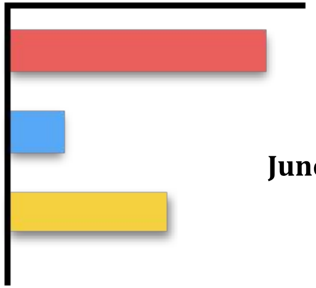
# Integrated Gradients

**Input**

**Model**

$$F_\theta$$

**Predictions**



**Junco Bird**

$$(x - \tilde{x}) \times \int_{\alpha=0}^{1} \frac{\partial F(\tilde{x} + \alpha \times (x - \tilde{x}))}{\partial x}$$

**Baseline input**

**Path integral: 'sum' of interpolated gradients**



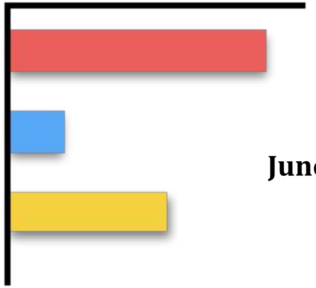
[Sundararajan et. al. 2017](#)

# Recap

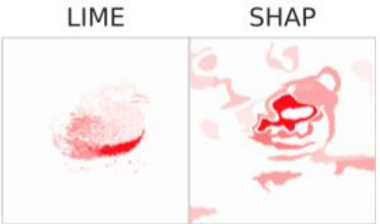
**Input**

**Model**

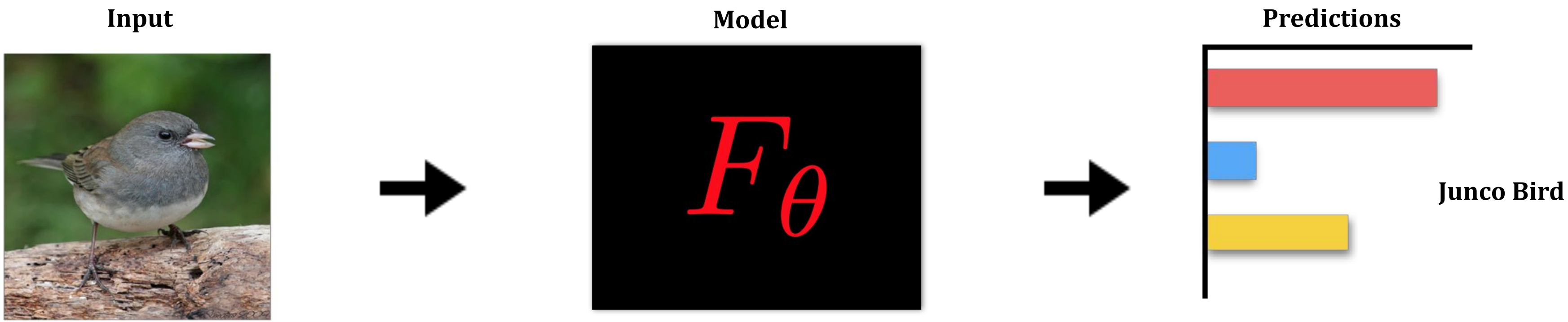

**Predictions**

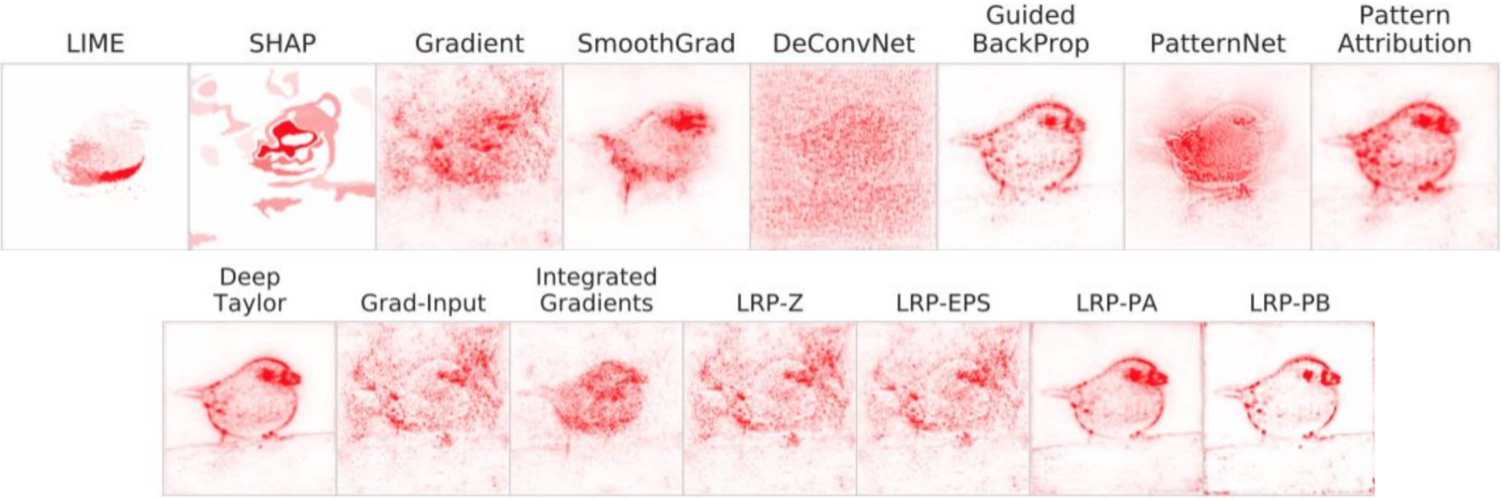


**Junco Bird**



# Recap


Input
Model
$F_\theta$
Predictions
Junco Bird


LIME
SHAP
Gradient
SmoothGrad
DeConvNet
Guided BackProp
PatternNet
Pattern Attribution
Deep Taylor
Grad-Input
Integrated Gradients
LRP-Z
LRP-EPS
LRP-PA
LRP-PB

# Limitations to post-hoc explainability

- **Faithfulness/Fidelity**
  Some explanation methods do not ‘*reflect*’ the underlying model.
- **Fragility**
  Post-hoc explanations can be easily manipulated.
- **Stability**
  Slight changes to inputs can cause large changes in explanations.
- **Useful in practice?**
  Unclear if a data scientist (ML engineer)/end-user can use explanations to isolate errors, improve ‘trust’ or simulate the model.

# Sanity Checks

Input image

A trained
machine learning model
(e.g., neural network)

prediction

$p(z)$

Junco Bird-ness

So these pixels are the **evidence** of **prediction.**

When **prediction** changes, the explanations will probably change.

When **prediction** is random, the explanations really should change!

**g(prediction) = explanation**

**g(prediction') = explanation'**

**g(random) != explanation ?**

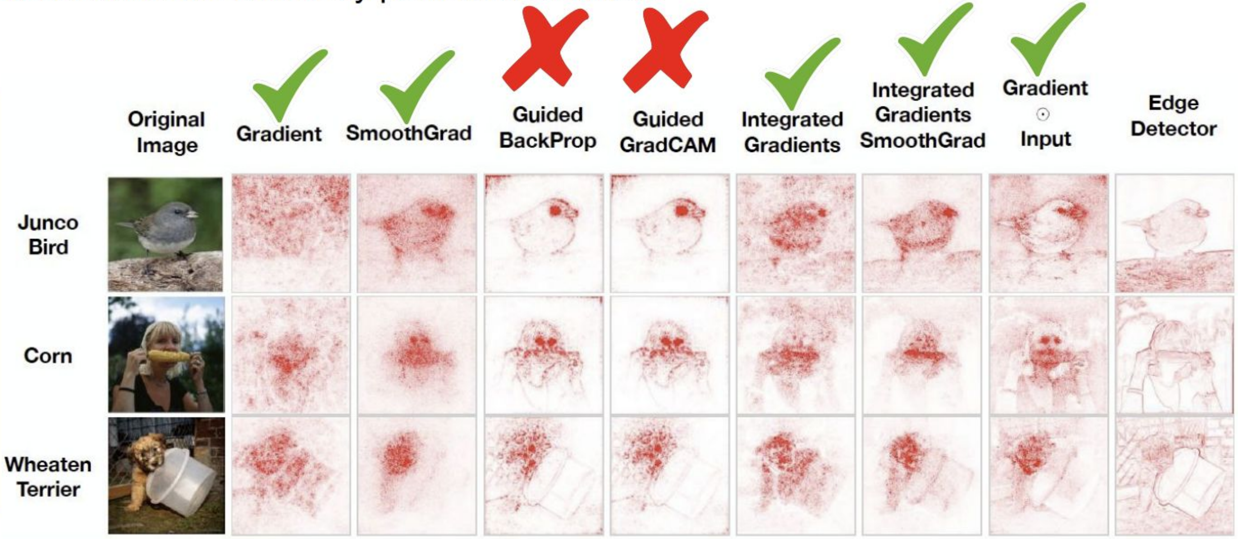
# Sanity Checks for Saliency Map

**Model parameter randomization test** compares the output of a saliency method on a trained model with the output of the saliency method on a randomly initialized untrained network of the same architecture.

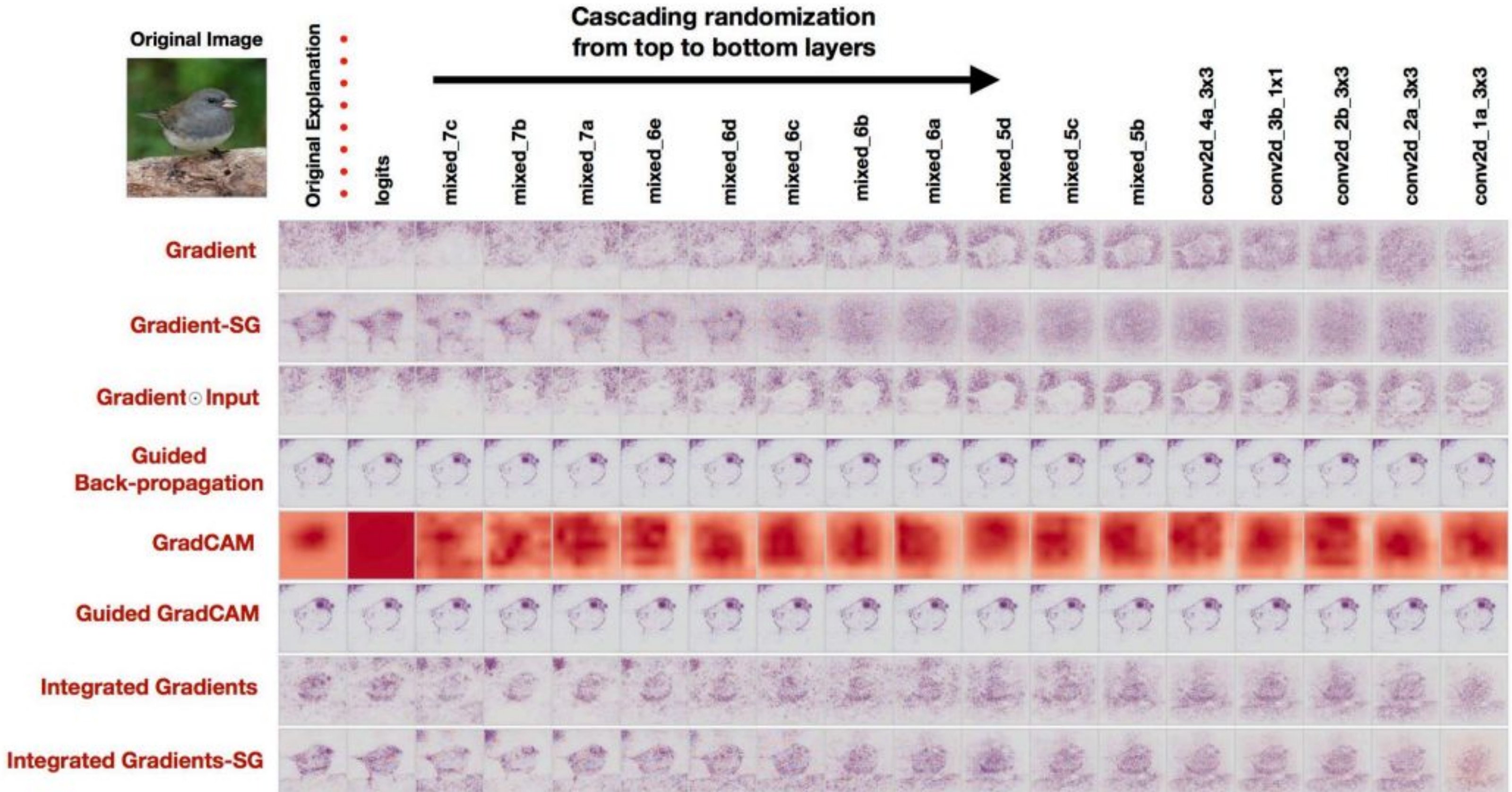
**Data randomization test** compares a given saliency method applied to a model trained on a labeled data set with the method applied to the same model architecture but trained on a copy of the data set in which we randomly permuted all labels.

"We find that reliance, solely, on visual assessment can be misleading.

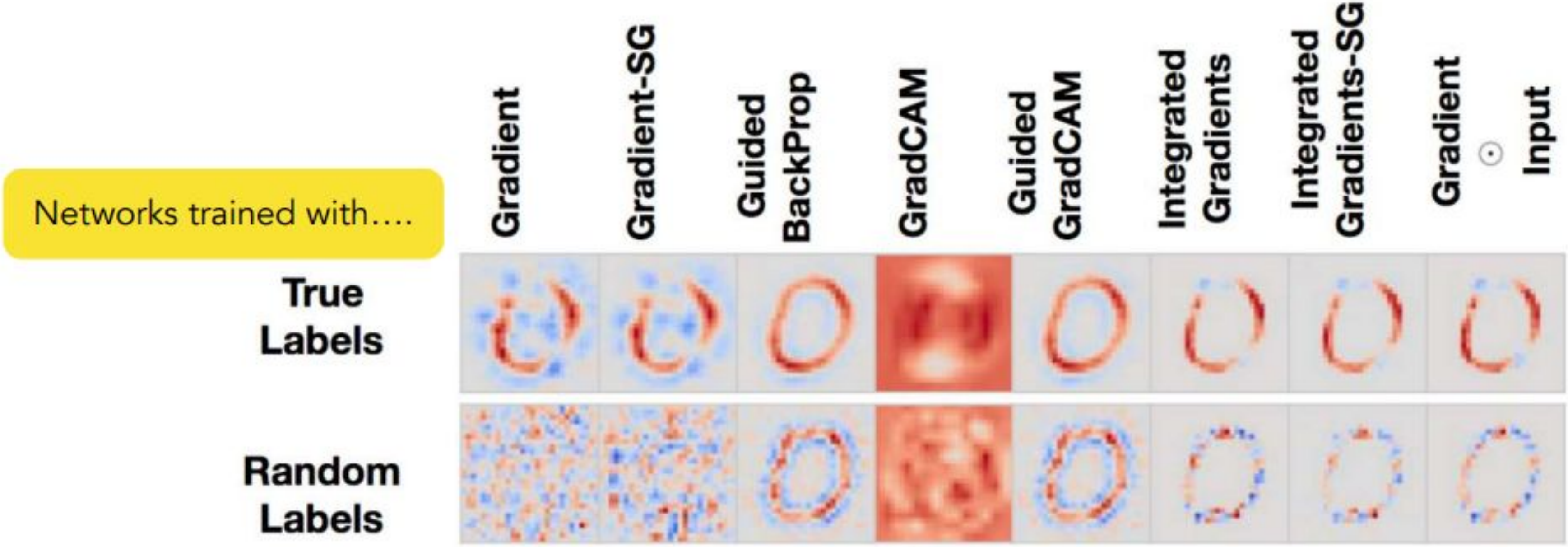
Through extensive experiments we show that some existing saliency methods are independent both of the model and of the data generating process."



# Sanity Check 1: Replace Weight with Random Layer

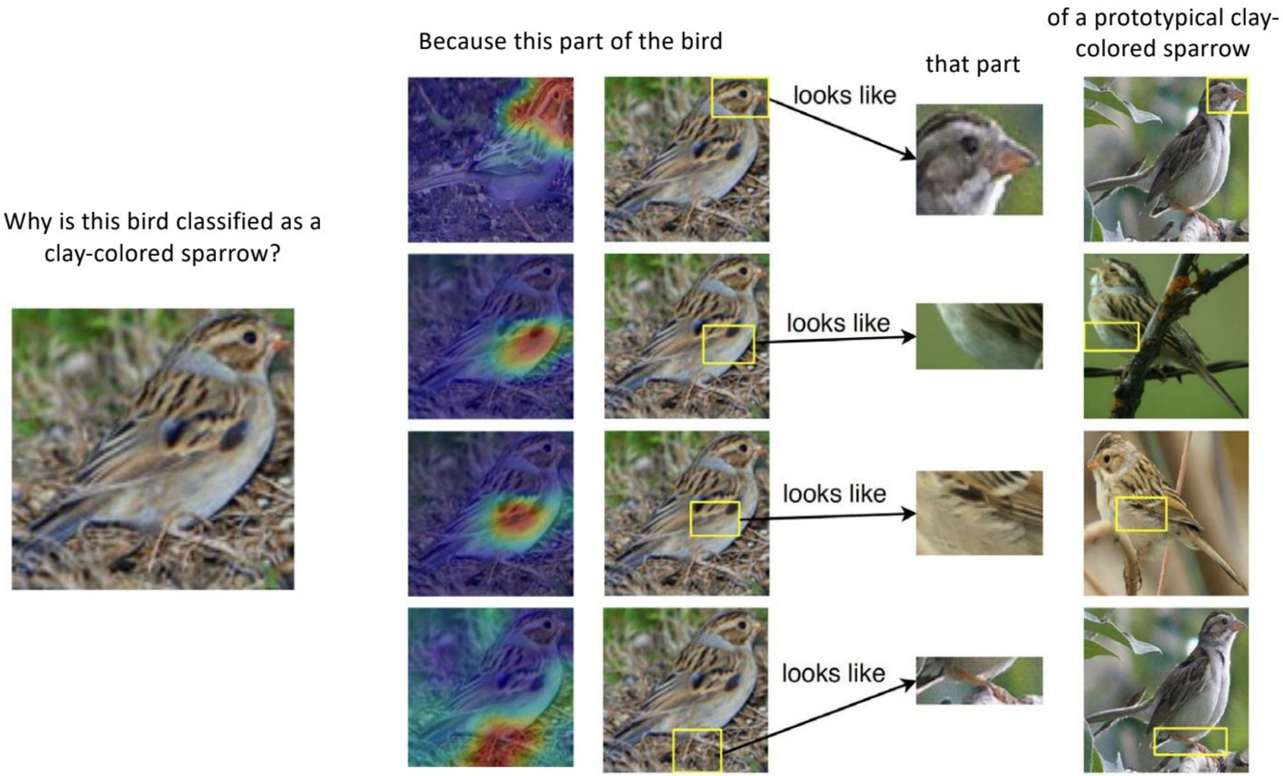


# Sanity Check 2: Replace the Trained Model with Bogus




See Adebayo+ (2018)

# Baked-in interpretable CNN

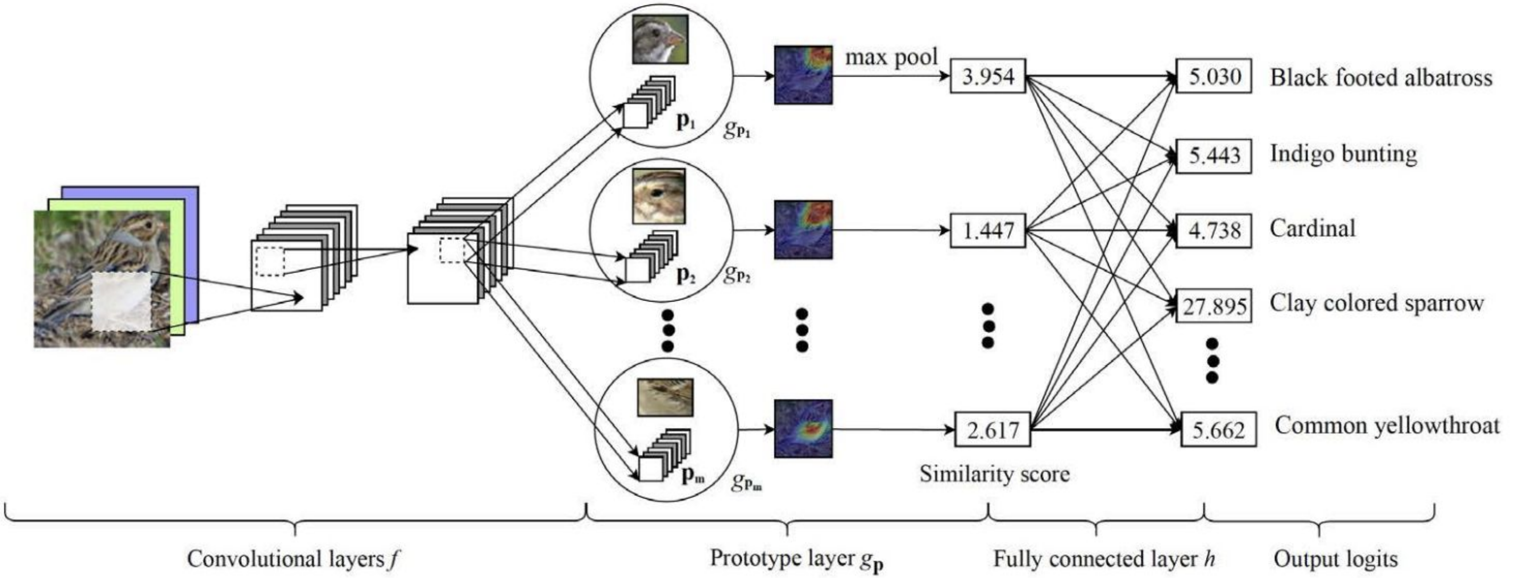


# Baked-in interpretable CNN

[Chen+ (2019)]

Take any "standard" black box CNN...

And transform it to be interpretable



# Influence Functions

- Remove specific data points or features, and measure their influence on a performance metric
- Largest change in performance indicates the most influential data points or features

Dawid+ (2020)

New Journal of Physics
The open access journal at the forefront of physics

PAPER • OPEN ACCESS

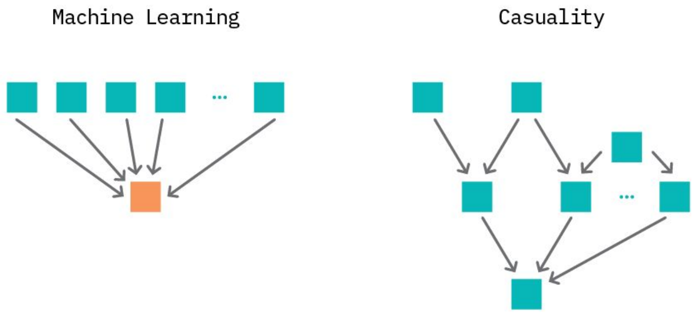
## Phase detection with neural networks: interpreting the black box

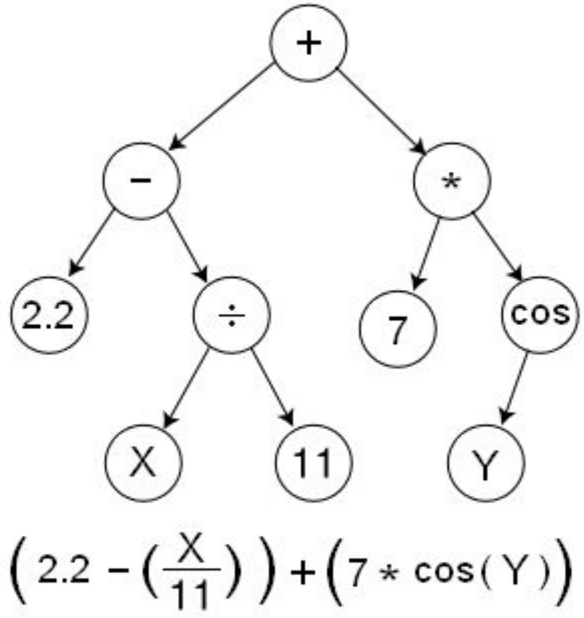
Anna Dawid[4,1,2], Patrick Huembeli[2], Michal Tomza[1], Maciej Lewenstein[2,3] and Alexandre Dauphin[2]

Published 12 November 2020 • 

# Structural Methods

Symbolic Regression tries to find the best mathematical expression to match a dataset



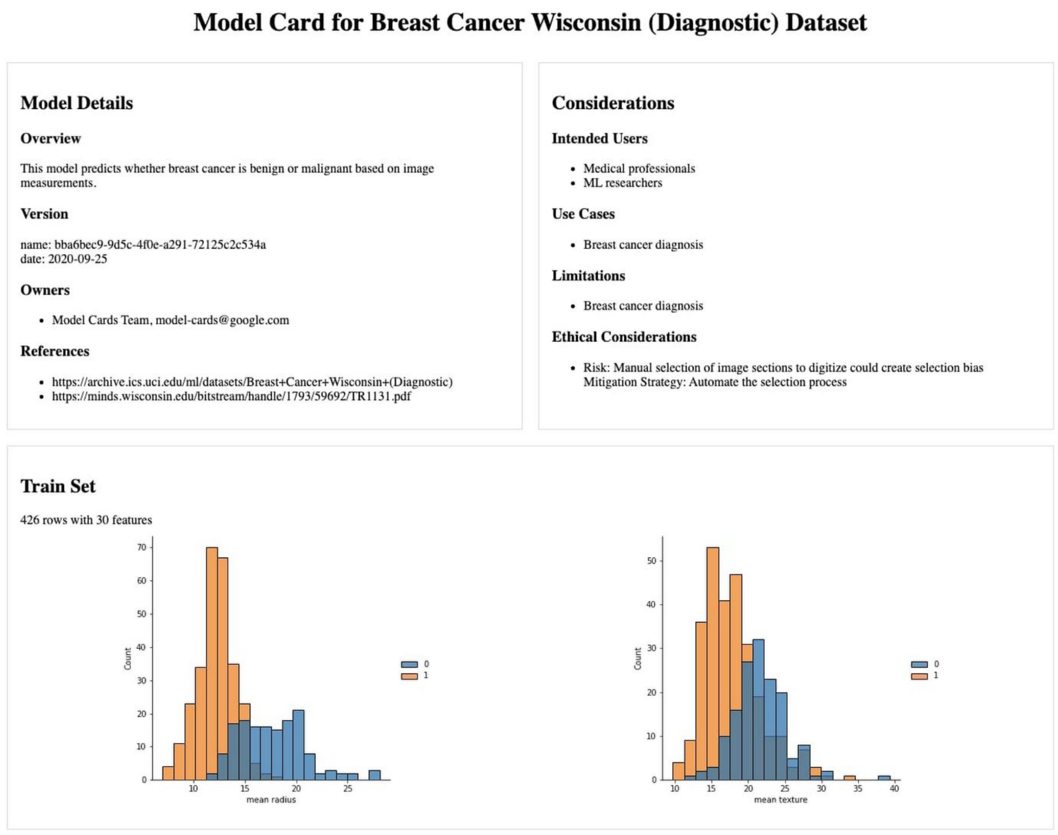

Causal Machine Learning tries to make sense on the data generation process

# Model Transparency

- Specify training set, metrics, limitations

  [Mitchell+ (2018)]()

  [Model Card creation tool]().



# Resources

- Tutorial: Interpretable Machine Learning MLSS, Kim (2021)
- Book: *Interpretable Machine Learning*, Molnar (2023)
- Software: `shap`, `interpret.ml`, `captum`, `PySR`, `causalml`