An Introduction to Explainable ML

Sign-in: http://bit.ly/bumicspring2019ws05

BOSTON UNIVERSITY
MACHINE INTELLIGENCE COMMUNITY

Devin de Hueck, Jianqi Ma
Spring, 2019

What we will cover

- Motivation for Interpretability
- Overview of Interpretable ML Methods
- Counterfactual Explanations
- A method to generate saliency maps for black-box models
- Future Work

Why do we want explainable models?

What is your problem?

Do you want to just know what predicted or do you want to know why it was predicted?

The problem of explainable ML

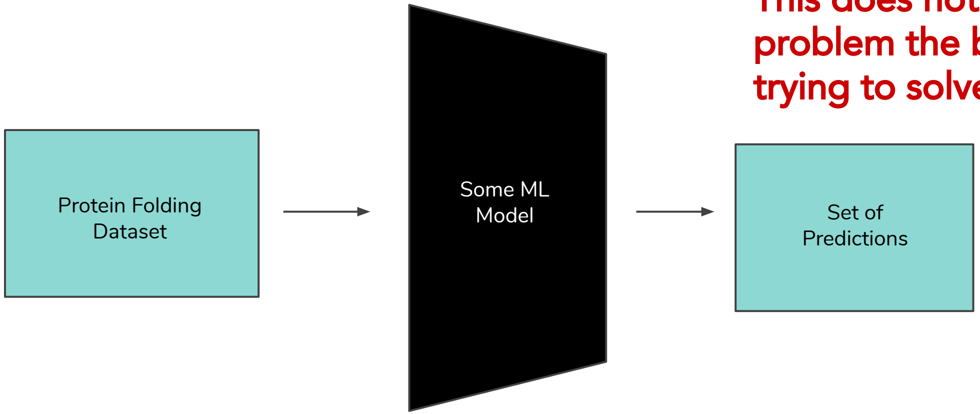
Why does a ML model make the prediction that it does?

Problems that requires interpretability

Consider the case of a biologist studying protein behavior

This biologist has the goal of understanding why/how proteins behave in the way they do

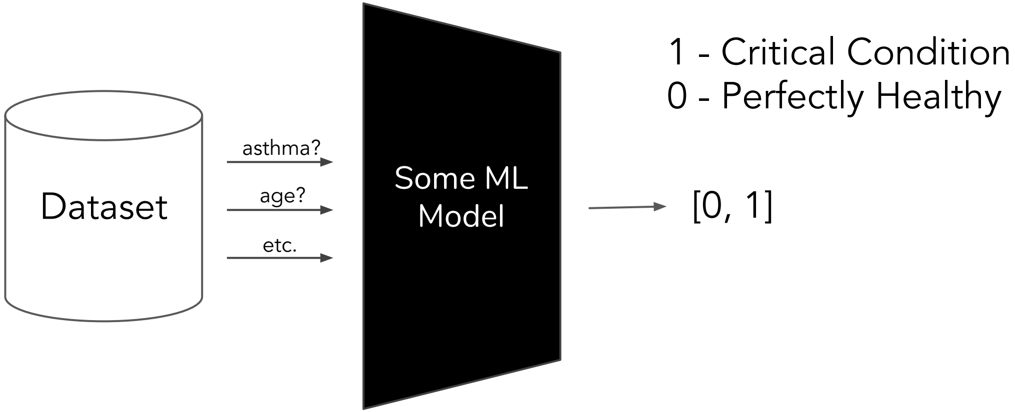
This does not solve the problem the biologist is trying to solve.



Problems that requires interpretability cont.

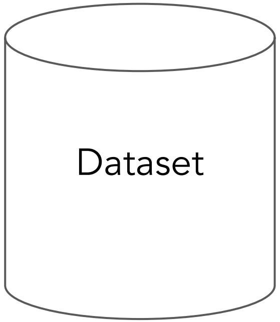
A hospital wants to predict the likelihood a patient has pneumonia.

It trains a model to predict a score between 0 and 1



Problems that requires interpretability cont.

A hospital wants to predict the likelihood a patient has pneumonia.



Consider the case that the dataset showed that patients with asthma actually **had a better prognosis than those without.**

Imagine if the hospital was prioritizing treatment with this model. Yikes!

Overview of Explainable ML Methods

3 Categories

Surrogate Models

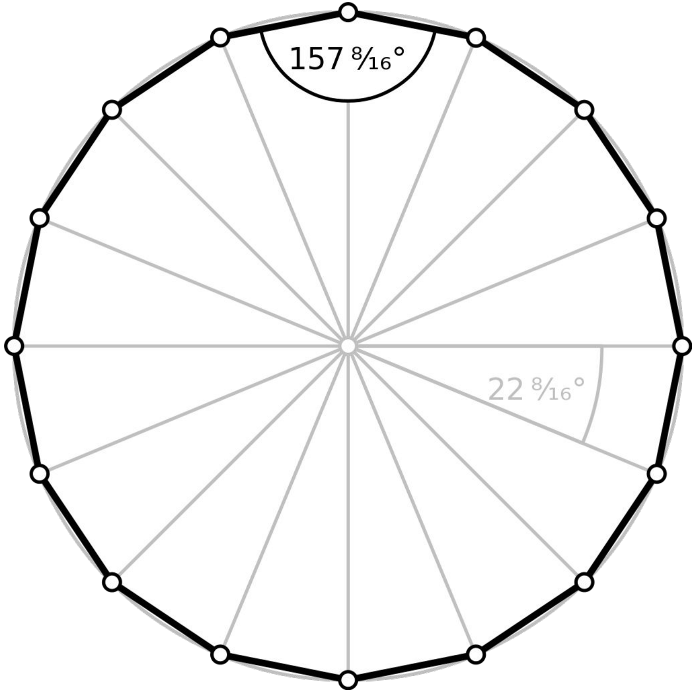
Feature Importance

Feature Effects

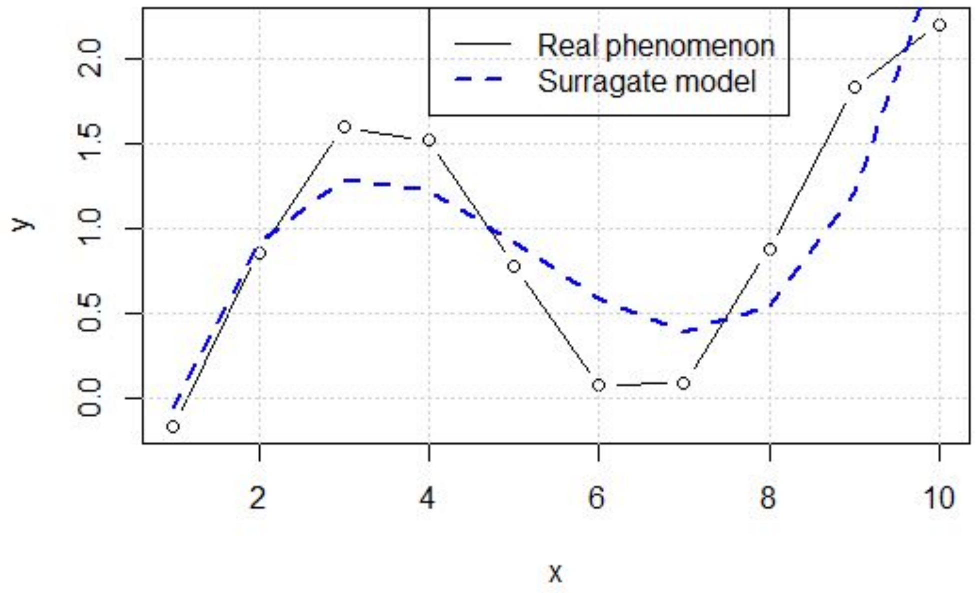
Surrogate Models

Why using Surrogate Models:

If a model is expensive, time-consuming or otherwise difficult to measure, a cheap and fast surrogate model of the outcome can be used instead.



Surrogate Models



The purpose of surrogate models is to approximate the predictions of the ML model as accurately as possible and to be interpretable at the same time.

Feature Importance

Feature Importance = Increase in Prediction Error

- A feature is “important” if shuffling its values increases the model error, because in this case the model relied on the feature for the prediction.
- A feature is “unimportant” if shuffling its values leaves the model error unchanged, because in this case the model ignored the feature for the prediction.

Feature Effects

Main Idea: summarize the average effect a feature has on the prediction

- **Partial Dependence Plot (PDP):**

  The method considers all instances and gives a statement about the global relationship of a feature with the predicted outcome.

- **Individual Conditional Expectation (ICE):**

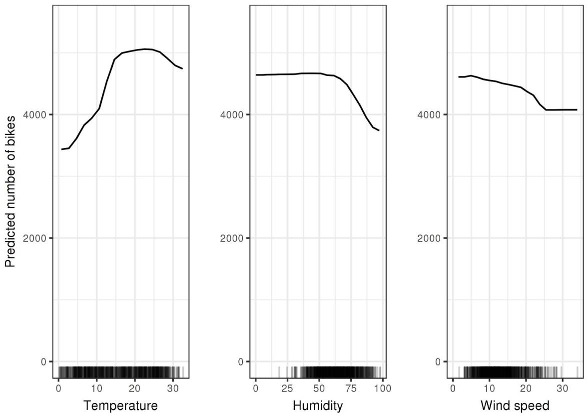
  ICE plot visualizes the dependence of the prediction on a feature for each instance separately, resulting in one line per instance.

- **Accumulated Local Effects (ALE):**

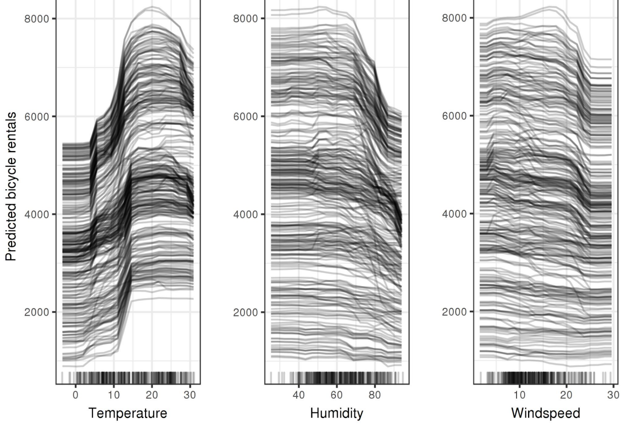
  Accumulated local effects describe how features influence the prediction of a machine learning model on average.

Visualizing Feature Effects Methods

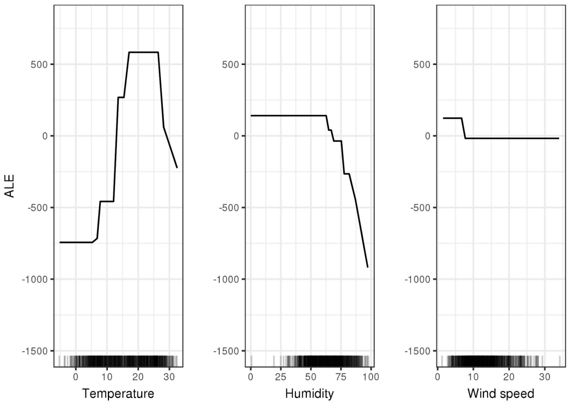
Partial Dependence Plot(PDP):



Individual Conditional Expectation(ICE):



Accumulated Local Effects(ALE):



Properties of Explainable Models

- Global vs. Local

  Is the interpretation for the entire model? Or maybe just for an individual prediction?

- Model-Specific vs. Model-Agnostic

  Whitebox vs. Blackbox

- Type of Data
  - Tabular
  - Text
  - Image

Counterfactual Explanations

What is a Counterfactual Explanation?

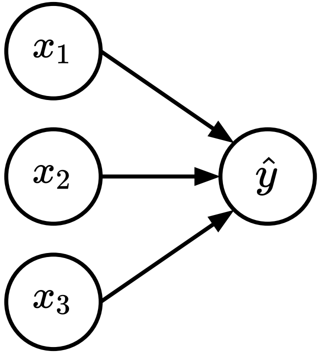
If X had not occurred, Y would not have occurred

We contradict observed facts to “imagine” a reality in which these facts did not occur

E.g. If I hadn’t taken a sip of this hot coffee, I wouldn’t have burned my tongue

What is a Counterfactual Explanation in the Context of ML?

Our inputs cause our outputs so an input explains an output.



We explain one output with examples

- Example Based/Local

Some Scenarios for Counterfactual Explanations

Given a model which outputs a decision of denial or acceptance for a loan based on factors such as income, assets, age, etc.

We ask the question: What would have to change for an individual who has been rejected to be granted the loan?

Plausible Counterfactual:

> "You were denied a loan because your annual income was $30,000. If your income had been $45,000, you would have been offered a loan."

https://arxiv.org/pdf/1711.00399.pdf

Some Scenarios for Counterfactual Explanations

Given a model which outputs a diabetes risk score for a woman of Pima heritage based on factors such as age, BMI, insulin level, etc.

We ask the question: What would have to change for an individual to have a risk score of 0.5?

Plausible Counterfactual:

> "If the individuals 2-Hour serum insulin level was 169.5, the risk score would be 0.51"

This is already very similar to how doctors communicate risks. E.g. "A BMI greater than X indicates obesity and an increase in health risks."

https://arxiv.org/pdf/1711.00399.pdf

Generating Counterfactuals - Notation

We asked the question: *What would have to change for an individual to have a risk score of 0.5?*

General Case: What in the input $\boldsymbol{x}$ would have to change such that the output has the desired outcome $\boldsymbol{y'}$?

$\boldsymbol{x'}$ - The counterfactual that we are generating

$\boldsymbol{x}$ - An input value

$\boldsymbol{y'}$ - The desired outcome

$\boldsymbol{x'}$ is the smallest change in $\boldsymbol{x}$ such that the prediction is the desired outcome $\boldsymbol{y'}$.

Generating Counterfactuals - The Loss Function

Our assertion that ***x'*** is the smallest change in ***x*** such that the prediction is the desired outcome ***y'*** is characterized by the loss function:

$$L(x, x', y', \lambda)$$

Generating Counterfactuals - The Loss Function

Our assertion that ***x'*** is the smallest change in ***x*** such that the prediction is the desired outcome ***y'*** is characterized by the loss function:

$$L(x, x', y', \lambda) = \underbrace{\lambda}_{\text{The hyperparameter lambda}} \cdot \underbrace{(\hat{f}(x') - y')^2}_{\text{Distance between the prediction from the current counterfactual and desired outcome}}$$

Generating Counterfactuals - The Loss Function

Our assertion that ***x'*** is the smallest change in ***x*** such that the prediction is the desired outcome ***y'*** is characterized by the loss function:

$$L(x, x', y', \lambda) = \underset{\text{The hyperparameter lambda}}{\lambda} \cdot \underbrace{(\hat{f}(x') - y')^2}_{\text{Distance between the prediction from the current counterfactual and desired outcome}} + \underbrace{d(x, x')}_{\text{Distance between the counterfactual example and the original example}}$$

Generating Counterfactuals - The Loss Function

Our assertion that $\boldsymbol{x'}$ is the smallest change in $\boldsymbol{x}$ such that the prediction is the desired outcome $\boldsymbol{y'}$ is characterized by the loss function:

$$L(x, x', y', \lambda) = \lambda \cdot (\hat{f}(x') - y')^2 + d(x, x')$$

The hyperparameter lambda

Distance between the prediction from the current counterfactual and desired outcome

Distance between the counterfactual example and the original example

Find the values of $\boldsymbol{x'}$ and $\boldsymbol{\lambda}$ such that: $\arg\min_{x'} L(x, x', y', \lambda)$

Generating Counterfactuals - The Algorithm

1. **Initialize:**

x - You input that is to be explained

y' - The desired outcome

ϵ - The tolerance (an acceptable distance for ***f(x')*** to be from ***y'***)

λ - Hyperparameter to balance the two parts of the loss function (usually set to a low value, > 1)

x' - An random input instance

Generating Counterfactuals - The Algorithm

2. **Minimize the loss function (find an x' really close to x):**

$$L(x, x', y', \lambda) = \lambda \cdot (\hat{f}(x') - y')^2 + d(x, x')$$

Generating Counterfactuals - The Algorithm

2. **Minimize the loss function (find an x' really close to x):**

$$L(x, x', y', \lambda) = \lambda \cdot (\hat{f}(x') - y')^2 + d(x, x')$$

Until satisfied:

Use an optimization algorithm like gradient descent to update x' to minimize the loss:

$$x' = x' - \alpha \nabla_{x'} L(x, x', y', \lambda)$$

Generating Counterfactuals - The Algorithm

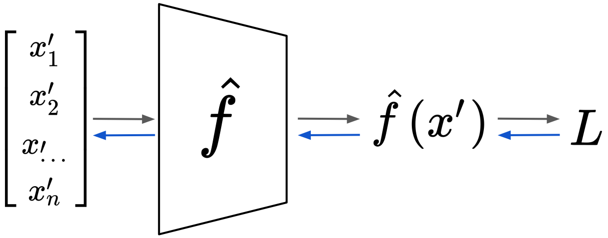
2. **Minimize the loss function (find an *x'* really close to *x*):**

$$L(x, x', y', \lambda) = \lambda \cdot (\hat{f}(x') - y')^2 + d(x, x')$$

Until satisfied:

Use an optimization algorithm like gradient descent to update ***x'*** to minimize the loss:

$$x' = x' - \alpha \nabla_{x'} L(x, x', y', \lambda)$$



Generating Counterfactuals - The Algorithm

3. **Maximize λ until we have our desired outcome while maintaining a *x'* similar to *x***

While $|f(x') - y'| > \epsilon$

$$\lambda = \lambda + \alpha$$ - Increase lambda by a small step size **α**

Generating Counterfactuals - The Algorithm

3. **Maximize λ until we have our desired outcome while maintaining a x' similar to x**

While $|f(x') - y'| > \epsilon$

$$\lambda = \lambda + \alpha$$ - Increase lambda by a small step size α

Use an optimization algorithm like gradient descent to update x' to minimize the loss:

$$x' = x' - \alpha \nabla_{x'} L(x, x', y', \lambda)$$

Generating Counterfactuals - The Algorithm

3. **Maximize λ until we have our desired outcome while maintaining a x' similar to x**

While $|f(x') - y'| > \epsilon$

$\lambda = \lambda + \alpha$ - Increase lambda by a small step size α

Use an optimization algorithm like gradient descent to update x' to minimize the loss:

$$x' = x' - \alpha \nabla_{x'} L(x, x', y', \lambda)$$

Return x' and Repeat

What is a Good Counterfactual Explanation?

x' is the smallest change in ***x*** such that the prediction is the desired outcome ***y'*.**

All features must be of reasonable scale

- The counterfactual: "If the apartment had 300 bathrooms the rent prediction would 1500/month" is not a meaningful counterfactual

Limit features to be within the max and min values in the dataset

Advantages

- Very clear explanations
  - Humans understand these type of explanations very well
- Does not require much domain knowledge to understand
  - Don’t need to be an ML expert to understand them
  - Don’t need to be a doctor to understand a counterfactual concerning a medical model

MIC

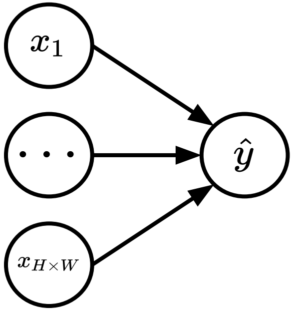
Disadvantages

- Rashmon Effect
  - There can be multiple *valid* counterfactual examples for each input
- No guarantee of finding a counterfactual within the set tolerance (**ϵ**)
  - Will have to increase epsilon until you do

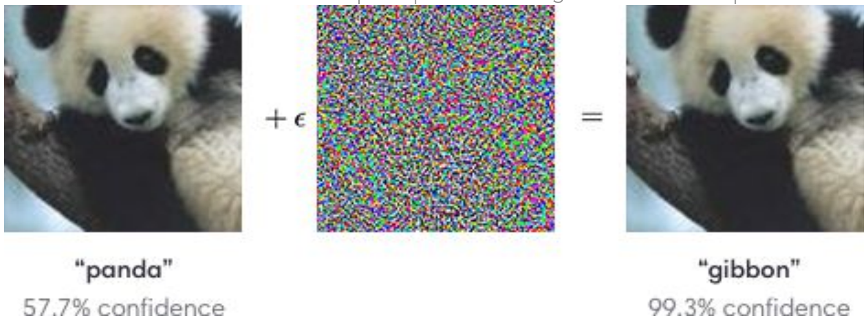
Adversarial Examples
vs Counterfactual Explanations

x' is the smallest change in ***x*** such that the prediction is the desired outcome ***y'*.**

What happens when our input is an image? Finding a counterfactual would give us an adversarial example.



https://openai.com/blog/adversarial-example-research/

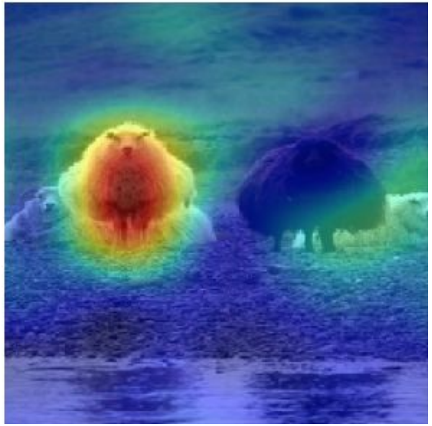


Saliency Map Generation

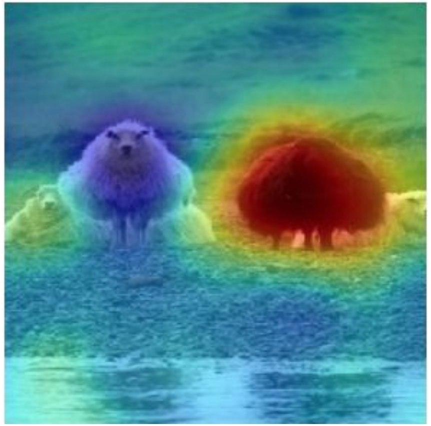
What Are Saliency Maps?

Saliency maps tell us how important each pixel of an image is for a prediction

Saliency map for prediction of class **sheep**



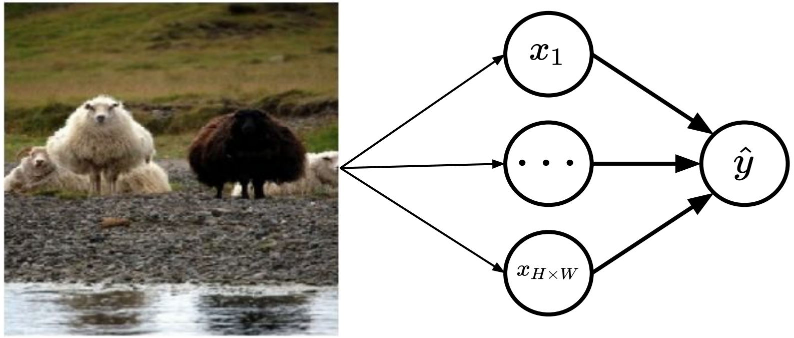
Saliency map for prediction of class **cow**

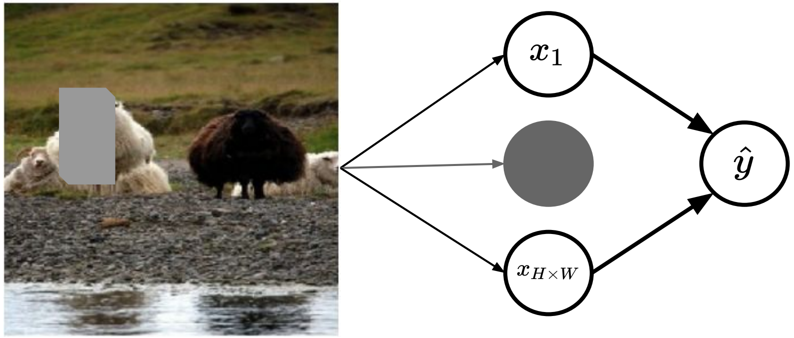


https://arxiv.org/pdf/1806.07421.pdf

Causality and Saliency Maps

What will happen if we delete some inputs/pixels? How will the predicted class score change?



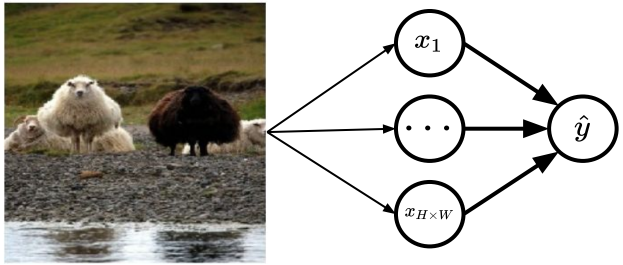


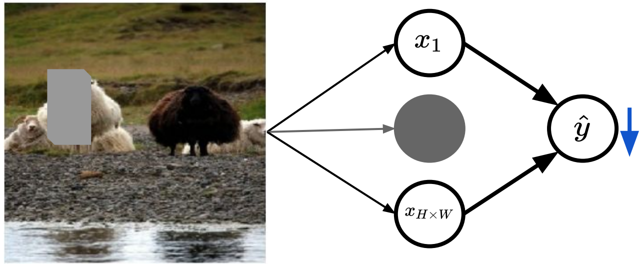
Causality and Saliency Maps

What will happen if we delete some inputs/pixels? How will the predicted class score change? **It depends!**

If the inputs/pixel that were removed **were important** for making the prediction then the predicted class score would **decrease** based on the importance.

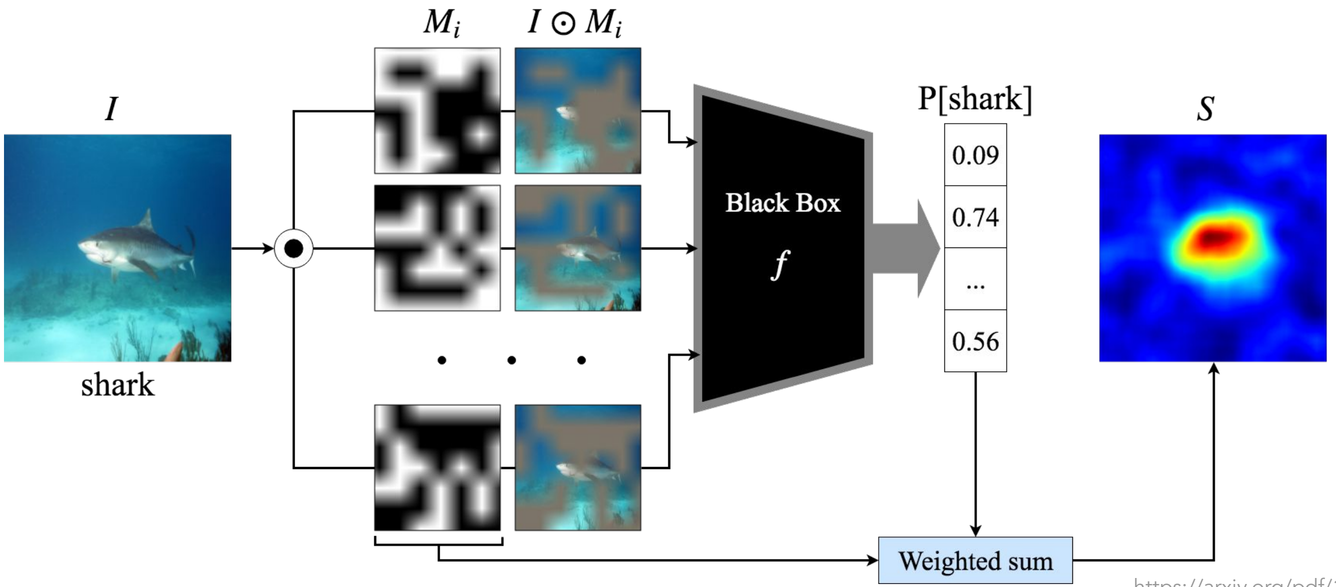
If the inputs/pixel that were removed **were not important** for making the prediction then the predicted class score would **stay about the same**.





RISE: Randomized Input Sampling for Explanation of Black-box Models

If we randomly remove pixels from different parts of an image and make a prediction on these modified images then we will eventually learn the importance of each pixel.



https://arxiv.org/pdf/1806.07421.pdf

RISE - Notation

I - The image to be explained which is of size H x W x 3

M - A binary mask which is of size H x W

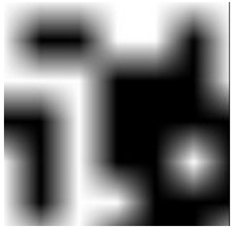
$I \odot M$ - A masked image which is of size H x W x 3

$f(I \odot M)$ - The prediction from a masked image

I

$\odot$

M



$\Rightarrow$

$I \odot M$

$\rightarrow f \Rightarrow$ **0.09**

$f(I \odot M)$

https://arxiv.org/pdf/1806.07421.pdf

RISE - The importance of a pixel

"We define importance of pixel $\boldsymbol{\lambda}$ as the expected score over all possible masks $\boldsymbol{M}$ conditioned on the event that pixel $\boldsymbol{\lambda}$ is observed, i.e., $\boldsymbol{M}(\boldsymbol{\lambda}) = 1$"

$$\underbrace{S_{I,f}(\lambda)}_{\text{Saliency/Importance of pixel } \boldsymbol{\lambda}}$$

Saliency/Importance
of pixel $\boldsymbol{\lambda}$

https://arxiv.org/pdf/1806.07421.pdf

RISE - The importance of a pixel

"We define importance of pixel **λ** as the expected score over all possible masks $\boldsymbol{M}$ conditioned on the event that pixel **λ** is observed, i.e., $\boldsymbol{M}(\boldsymbol{\lambda}) = 1$"

$$\underbrace{S_{I,f}(\lambda)}_{\text{Saliency/Importance of pixel } \boldsymbol{\lambda}} = \mathbb{E}_M[\underbrace{f(I \odot M)}_{\text{Black-box model's prediction of a masked image}}]$$

Saliency/Importance of pixel **λ**

Black-box model's prediction of a masked image

https://arxiv.org/pdf/1806.07421.pdf

RISE - The importance of a pixel

"We define importance of pixel **λ** as the expected score over all possible masks $\boldsymbol{M}$ conditioned on the event that pixel **λ** is observed, i.e., $\boldsymbol{M}(\boldsymbol{\lambda}) = 1$"

$$\underbrace{S_{I,f}(\lambda)}_{\text{Saliency/Importance of pixel } \boldsymbol{\lambda}} = \mathbb{E}_M[\underbrace{f(I \odot M)}_{\text{Black-box model's prediction of a masked image}} | \underbrace{M(\lambda) = 1}_{\text{Visibility of pixel } \boldsymbol{\lambda} \text{ - 0 if hidden by mask, 1 if visible}}]$$

https://arxiv.org/pdf/1806.07421.pdf

RISE - The importance of a pixel

"We define importance of pixel **λ** as the expected score over all possible masks $\boldsymbol{M}$ conditioned on the event that pixel **λ** is observed, i.e., $\boldsymbol{M}(\boldsymbol{\lambda}) = 1$"

$$S_{I,f}(\lambda) = \mathbb{E}_M[f(I \odot M) | M(\lambda) = 1]$$

Saliency/Importance of pixel **λ**

Black-box model's prediction of a masked image

Visibility of pixel **λ** - 0 if hidden by mask, 1 if visible

To get the complete saliency map we solve this equation for every pixel in the image

https://arxiv.org/pdf/1806.07421.pdf

RISE - The importance of a pixel - In practice

In practice we perform a Monte-Carlo simulation.

The importance of one pixel is the average predicted score of N masked images where that pixel is visible normalized by the expected value of a mask (0.5).

$$S_{I,f}(\lambda) \overset{\mathrm{MC}}{\approx}$$

https://arxiv.org/pdf/1806.07421.pdf

RISE - The importance of a pixel - In practice

In practice we perform a Monte-Carlo simulation.

The importance of one pixel is the average predicted score of N masked images where that pixel is visible normalized by the expected value of a mask (0.5).

$$S_{I,f}(\lambda) \overset{\text{MC}}{\approx} \underbrace{\sum_{i} f(I \odot M_i)}_{\text{Sum of masked images prediction scores}}$$

RISE - The importance of a pixel - In practice

In practice we perform a Monte-Carlo simulation.

The importance of one pixel is the average predicted score of N masked images where that pixel is visible normalized by the expected value of a mask (0.5).

$$S_{I,f}(\lambda) \overset{\text{MC}}{\approx} \sum_i f(I \odot M_i) \cdot \underline{M_i(\lambda)}$$

Only want to sum if visible

RISE - The importance of a pixel - In practice

In practice we perform a Monte-Carlo simulation.

The importance of one pixel is the average predicted score of N masked images where that pixel is visible normalized by the expected value of a mask (0.5).

$$S_{I,f}(\lambda) \overset{\mathrm{MC}}{\approx} \underbrace{\frac{1}{N} \sum_{i=1}^{N}}_{\text{Get the average predicted score}} f(I \odot M_i) \cdot M_i(\lambda)$$

RISE - The importance of a pixel - In practice

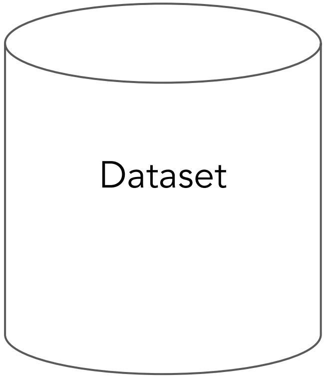
In practice we perform a Monte-Carlo simulation.

The importance of one pixel is the average predicted score of N masked images where that pixel is visible normalized by the expected value of a mask (0.5).

$$S_{I,f}(\lambda) \overset{\text{MC}}{\approx} \underbrace{\frac{1}{\mathbb{E}[M]}}_{\text{Normalize by the expected value of a mask (0.5)}} \cdot \underbrace{\frac{1}{N} \sum_{i=1}^{N} f(I \odot M_i) \cdot M_i(\lambda)}_{\text{The average predicted score of N masked images}}$$

RISE - The algorithm

1. **Select and image to have explained**

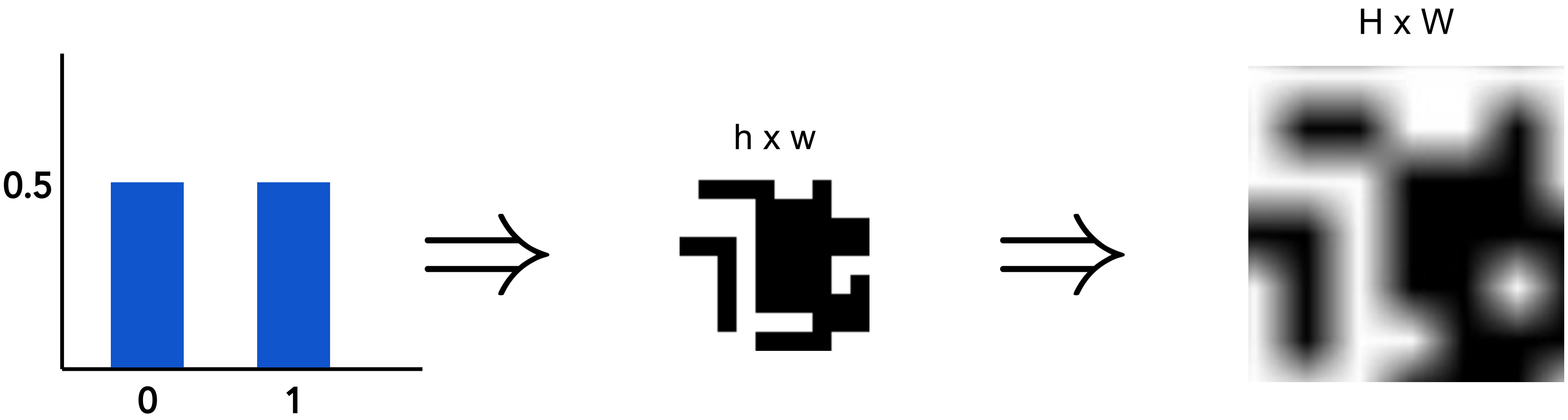


$\Rightarrow$

I

RISE - The algorithm

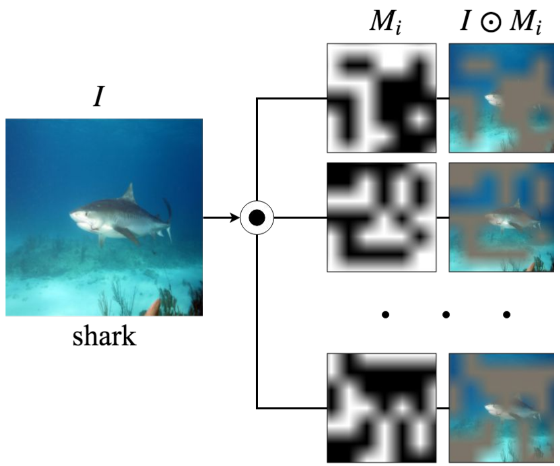
2. **Generate a set of N masks**



https://arxiv.org/pdf/1806.07421.pdf

RISE - The algorithm

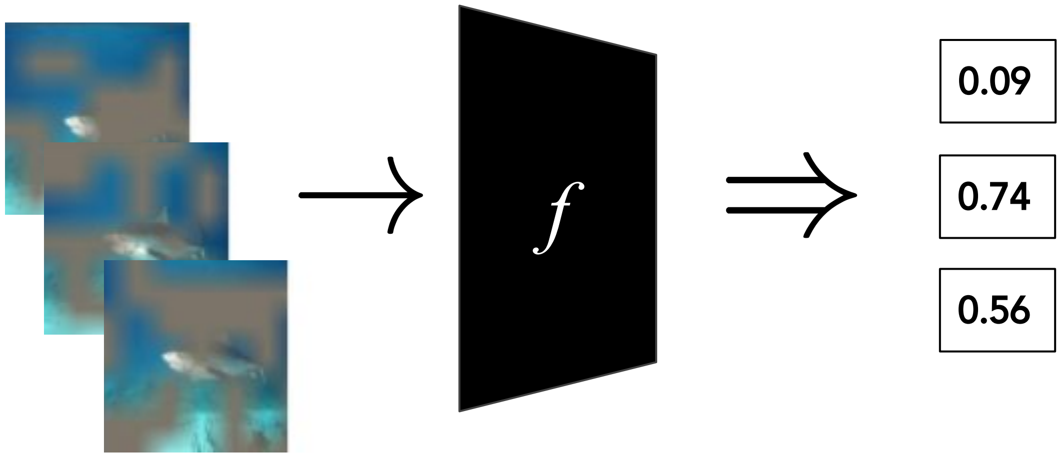
3. **Generate N masked images through element-wise multiplication of a mask and the image.**



https://arxiv.org/pdf/1806.07421.pdf

RISE - The algorithm

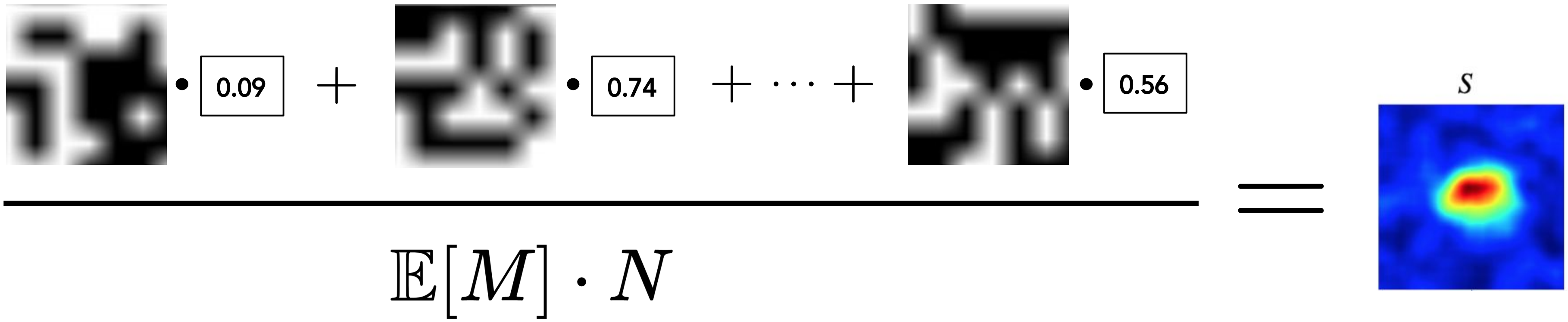
4. **Get N predictions from your black-box model on the each masked image**



https://arxiv.org/pdf/1806.07421.pdf

RISE - The algorithm

5. **Take the dot product between the N masks and N predictions then normalize to get one saliency map for each possible prediction**



And that's it!

What does the future hold?

The Future of Explainable ML

Focus will be on Model-Agnostic methods

Explainable ML add-ons to currently existing libraries

References & Further Reading

1. https://christophm.github.io/interpretable-ml-book/counterfactual.html
2. Petsiuk, Vitali, Abir Das, and Kate Saenko. "RISE: Randomized Input Sampling for Explanation of Black-box Models." arXiv preprint arXiv:1806.07421 (2018).
3. Doshi-Velez, Finale, and Been Kim. "Towards a rigorous science of interpretable machine learning." arXiv preprint arXiv:1702.08608 (2017).
4. Hall, Patrick, and Navdeep Gill. An Introduction to Machine Learning Interpretability. O'Reilly Media Inc., 2018.