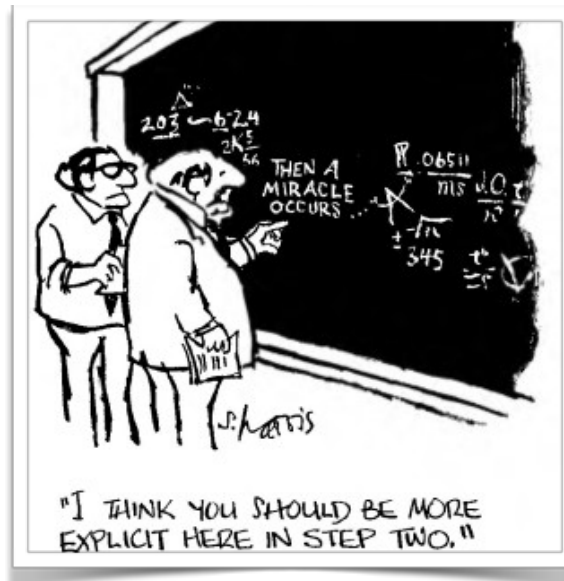


“Interpretable” Models

Explanations by “interpretable” models are produced by inspecting the inner logic of the algorithm that has given the prediction.

Linear regression, logistic regression and the decision tree are **commonly considered** interpretable models...



“Interpretable” Models

A linear regression model predicts the target as a weighted sum of the feature inputs:

$$y = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p + \epsilon$$

Estimated weights can come with confidence intervals, such as standard error values.

“Interpretable” Models

Interpretation of a linear model:

An increase of feature x_k by one unit increases the prediction for y by β_k units when all other feature values remain fixed.



The importance of a feature in a linear regression model can be measured by the absolute value of its t-statistic. The t-statistic is the estimated weight scaled with its standard error.

$$t_{\hat{\beta}_j} = \frac{\hat{\beta}_j}{SE(\hat{\beta}_j)}$$

Let us examine what this formula tells us: The importance of a feature increases with increasing weight. This makes sense. The more variance the estimated weight has (= the less certain we are about the correct value), the less important the feature is. This also makes sense.

“Interpretable” Models

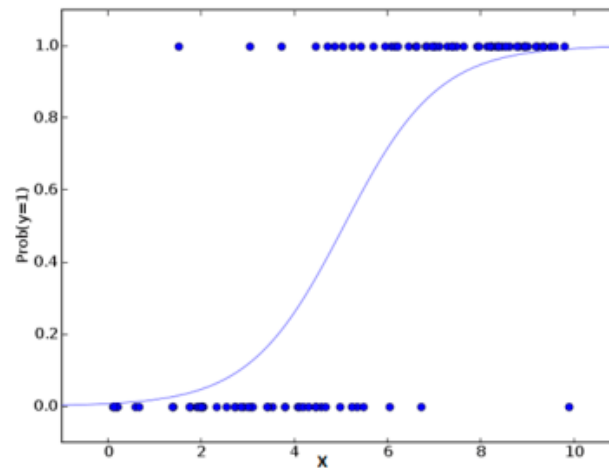
The interpretation of a weight can be unintuitive because **it depends on** all other features.

A feature with high positive correlation with the outcome y and another feature might get a negative weight in the linear model, because, given the other correlated feature, it is negatively correlated with y in the high-dimensional space.

“Interpretable” Models

A logistic regression model predicts the target as:

$$P(y^{(i)} = 1) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_1^{(i)} + \dots + \beta_p x_p^{(i)}))}$$



“Interpretable” Models

The interpretation of this model can be made by considering the “odds” function: the probability of event divided by the probability of not event.

$$\log \left(\frac{P(y = 1)}{1 - P(y = 1)} \right) = \log \left(\frac{P(y = 1)}{P(y = 0)} \right) = \beta_0 + \beta_1 x_1 + \dots + \beta_p x_p$$

We then can compare what happens when we increase one of the feature values by 1

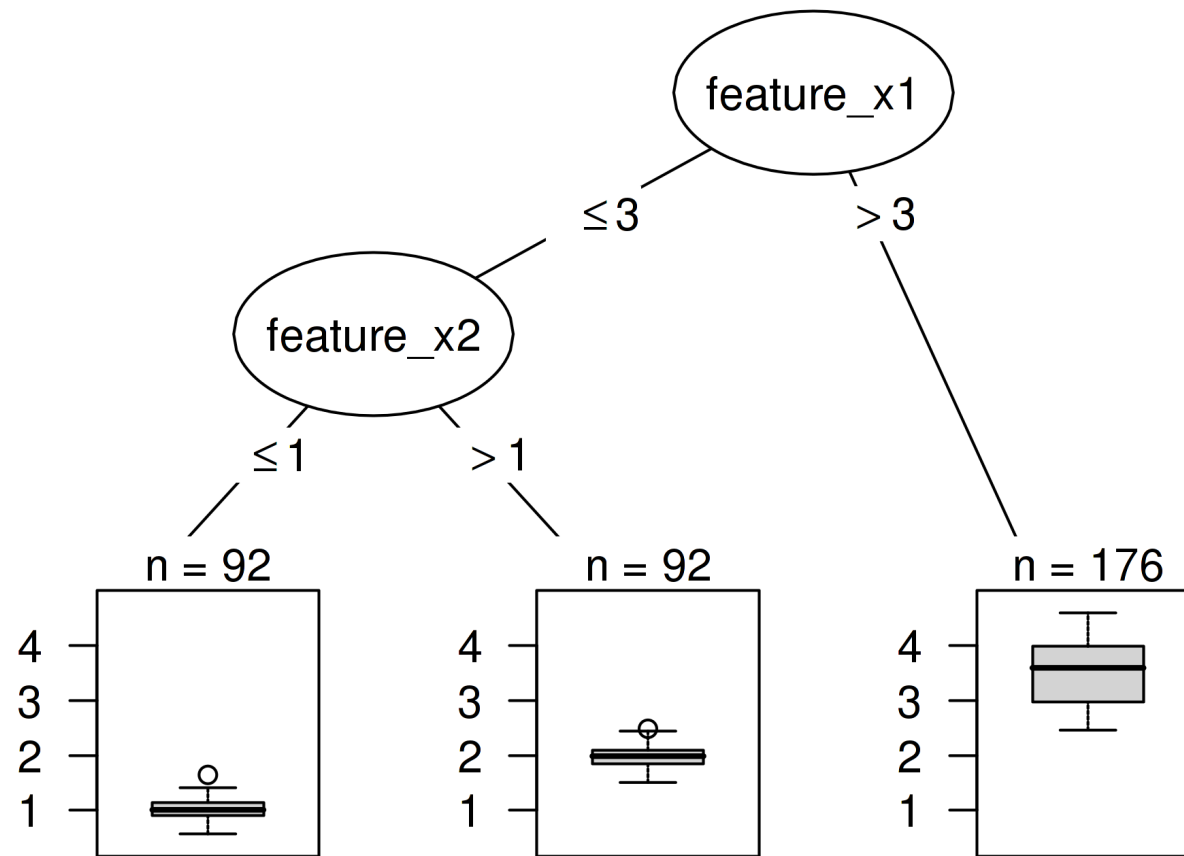
$$\frac{odds_{x_j+1}}{odds} = \frac{\exp(\beta_0 + \beta_1 x_1 + \dots + \beta_j(x_j + 1) + \dots + \beta_p x_p)}{\exp(\beta_0 + \beta_1 x_1 + \dots + \beta_j x_j + \dots + \beta_p x_p)}$$

Then, it can be shown that a change in one feature by 1 unit changes the “odds” ratio by a factor of:

$$\exp(\beta_j)$$

“Interpretable” Models

Decision trees & CART models.



“Interpretable” Models

Decision trees & CART models.

The interpretation is simple: Starting from the root node, you go to the next nodes and the edges tell you which subsets you are looking at. Once you reach the leaf node, the node tells you the predicted outcome. All the edges are connected by ‘AND’.

Template: If feature x is [smaller/bigger] than threshold c AND ... then the predicted outcome is the mean value of y of the instances in that node.

Feature importance

The overall importance of a feature in a decision tree can be computed in the following way: Go through all the splits for which the feature was used and measure how much it has reduced the variance or Gini index compared to the parent node. The sum of all importances is scaled to 100. This means that each importance can be interpreted as share of the overall model importance.

“Interpretable” Models

Other “interpretable” models:

- GLM, GAM
- Decision Rules
- Naive Bayes
- K-nearest-neighbors

“Interpretable” Models

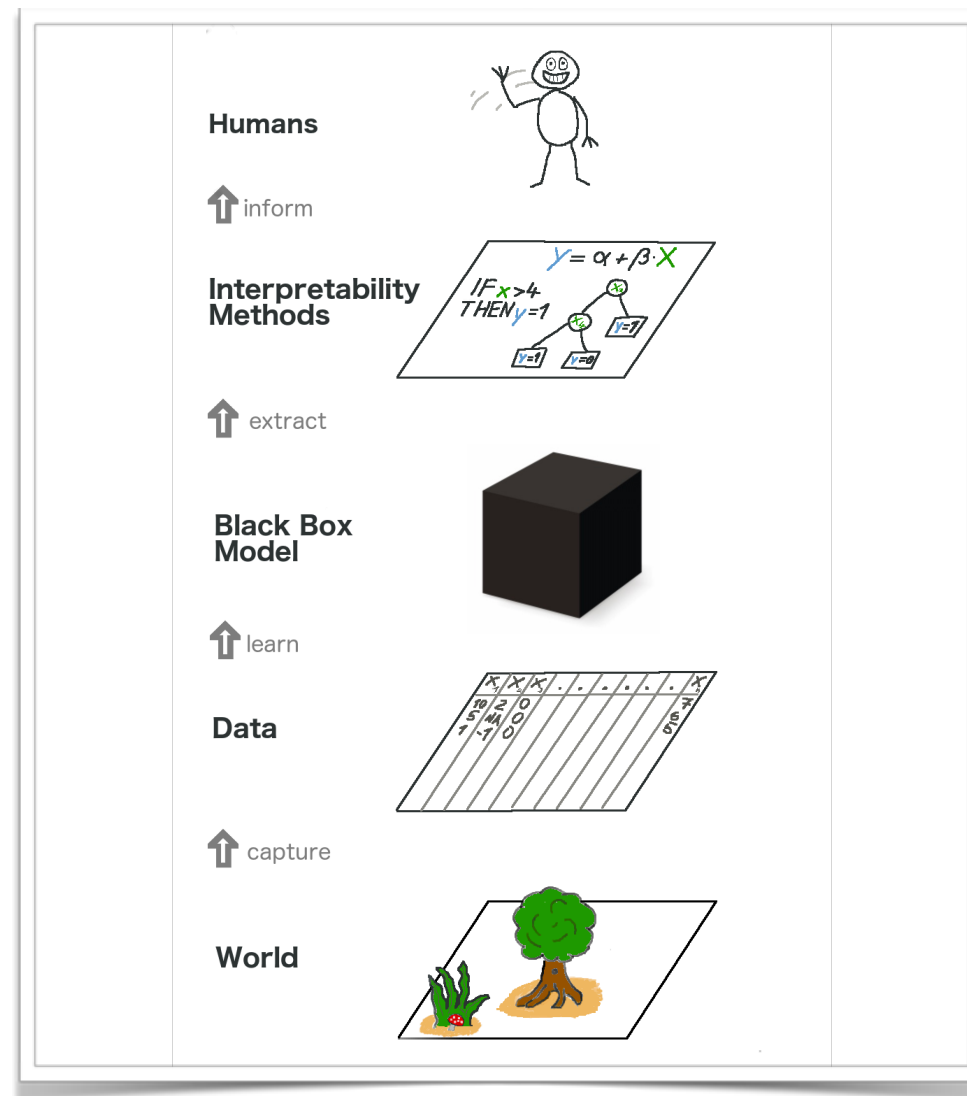
We have a “**mechanical**” explanation of the behavior of the model that can be **easily understood by a human**.

Is this the answer we were looking for when asking our WHY questions?

Example:

- Does it make sense to increase the value of a feature without taking into consideration other feature correlations?
- What about explanations that require complex combinations of features?

Model-Agnostic Models



Model-Agnostic Models

The **partial dependence plot** (PDP) shows the marginal effect one (or two features) have on the predicted outcome of a machine learning model.

Suppose S is a subset of p predictor variables, such that $S \subset \{X_1, X_2, \dots, X_p\}$. Let C be a complement to S , such that $S \cup C = \{X_1, X_2, \dots, X_p\}$. The random forest predictor function, $f(X)$, will depend upon all p predictor variables. Thus, $f(X) = f(X_S, X_C)$. The partial dependence of the S predictors on the predictive function $f(X)$ is

$$f_S(X_S) = \mathbb{E}_{X_C}[f(X_S, X_C)]$$

and can be estimated by

$$\bar{f}_S(X_S) = \frac{1}{N} \sum_{i=1}^N [f(X_S, X_{Ci})]$$

where $\{x_{C1}, x_{C2}, \dots, x_{CN}\}$ are the values of X_C occurring over all observations in the training data. In other words, in order to calculate the partial dependence of a given variable (or variables), the entire training set must be utilized for every set of joint values in X_S . As one can imagine, this can be quite computationally expensive when the data set becomes large.

Model-Agnostic Models

For example, let's assume a data set that only contains three data points and three features (A, B, C) as shown below.

A	B	C	Y
a1	b1	c1	y1
a2	b2	c2	y2
a3	b3	c3	y3

If we want to see how **feature A** is influencing the prediction **Y**, what PDP does is to generate a new data set as follows and do prediction as usual.

A	B	C	D
a1	b1	c1	y11
a1	b2	c2	y21
a1	b3	c3	y31
a2	b1	c1	y12
a2	b2	c2	y22
a2	b3	c3	y32
a3	b1	c1	y13
a3	b2	c2	y23
a3	b3	c3	y33

Model-Agnostic Models

Then, it averages the predictions for having a unique value of feature A:

A	B	C	D
a1	b1	c1	yA1
a1	b2	c2	
a1	b3	c3	
a2	b1	c1	yA2
a2	b2	c2	
a2	b3	c3	
a3	b1	c1	yA3
a3	b2	c2	
a3	b3	c3	

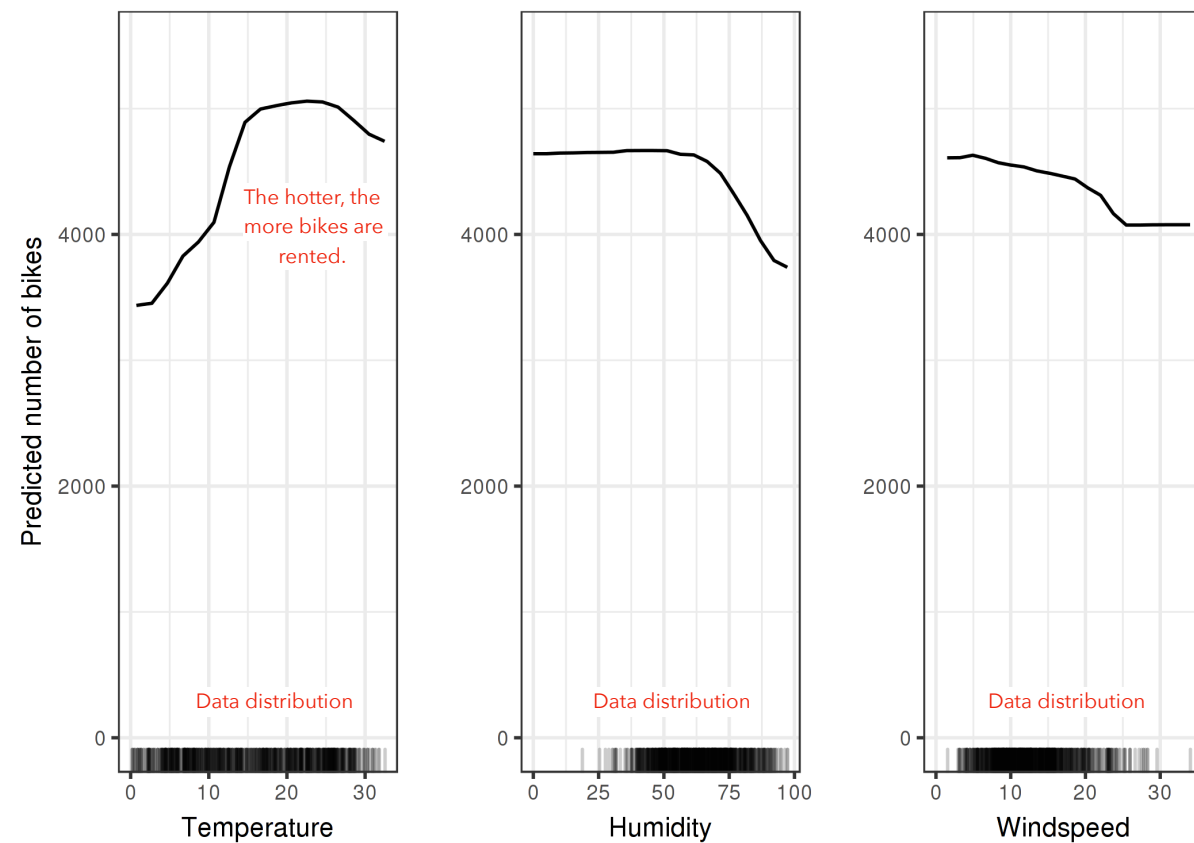
Finally, it plots out the average predictions.

X	A1	A2	A3
Y	yA1	yA2	yA3

This method can produce **unlikely data instances** when two or more features are correlated.

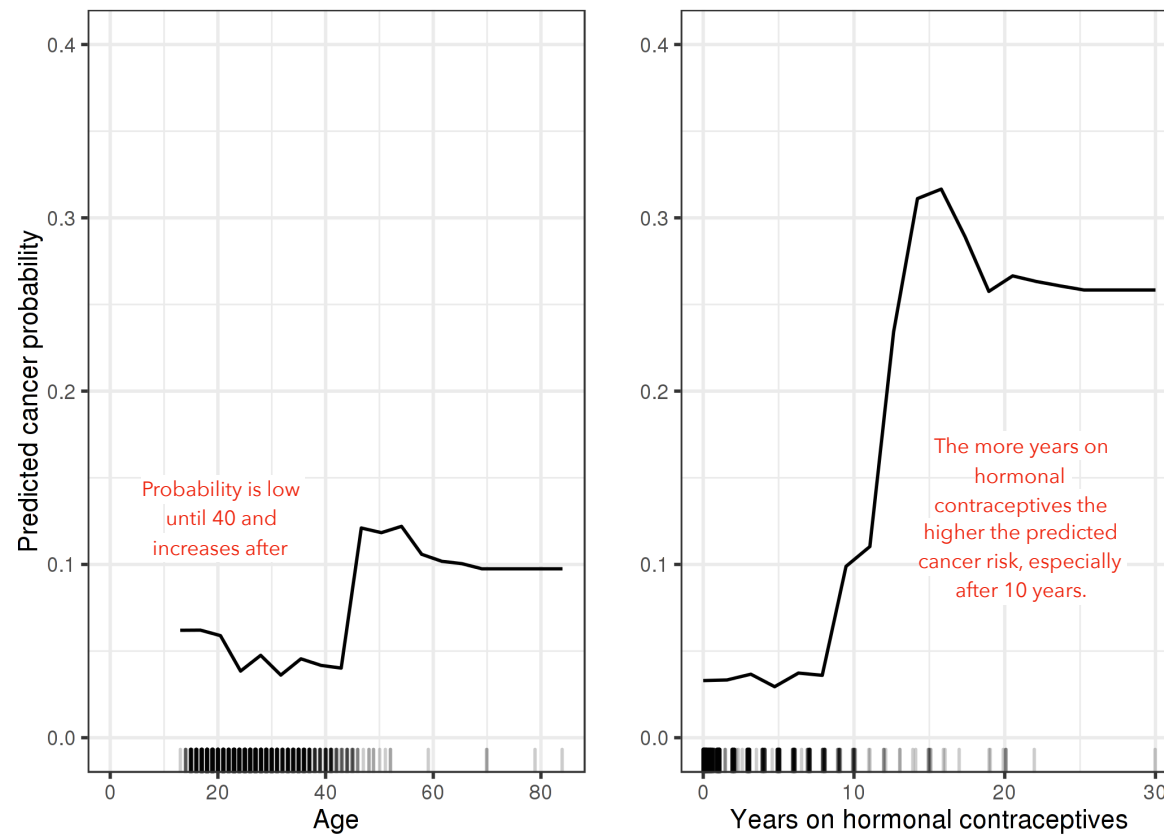
Model-Agnostic Models

Problem: predict the number of bikes that will be rented on a given day. The influence of the weather features on the predicted bike counts is visualized in the following figure.



Model-Agnostic Models

Problem: cervical cancer classification.



For both features not many data points with large values were available, so the PD estimates are less reliable in those regions.

Model-Agnostic Models

Permutation Test

The importance of a feature is the increase in the prediction error of the model after we permuted the feature's values, which breaks the relationship between the feature and the true outcome.

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.

The problem is the same as with partial dependence plots: The permutation of features produces **unlikely data instances** when two or more features are correlated.

Model-Agnostic Models

Permutation Test

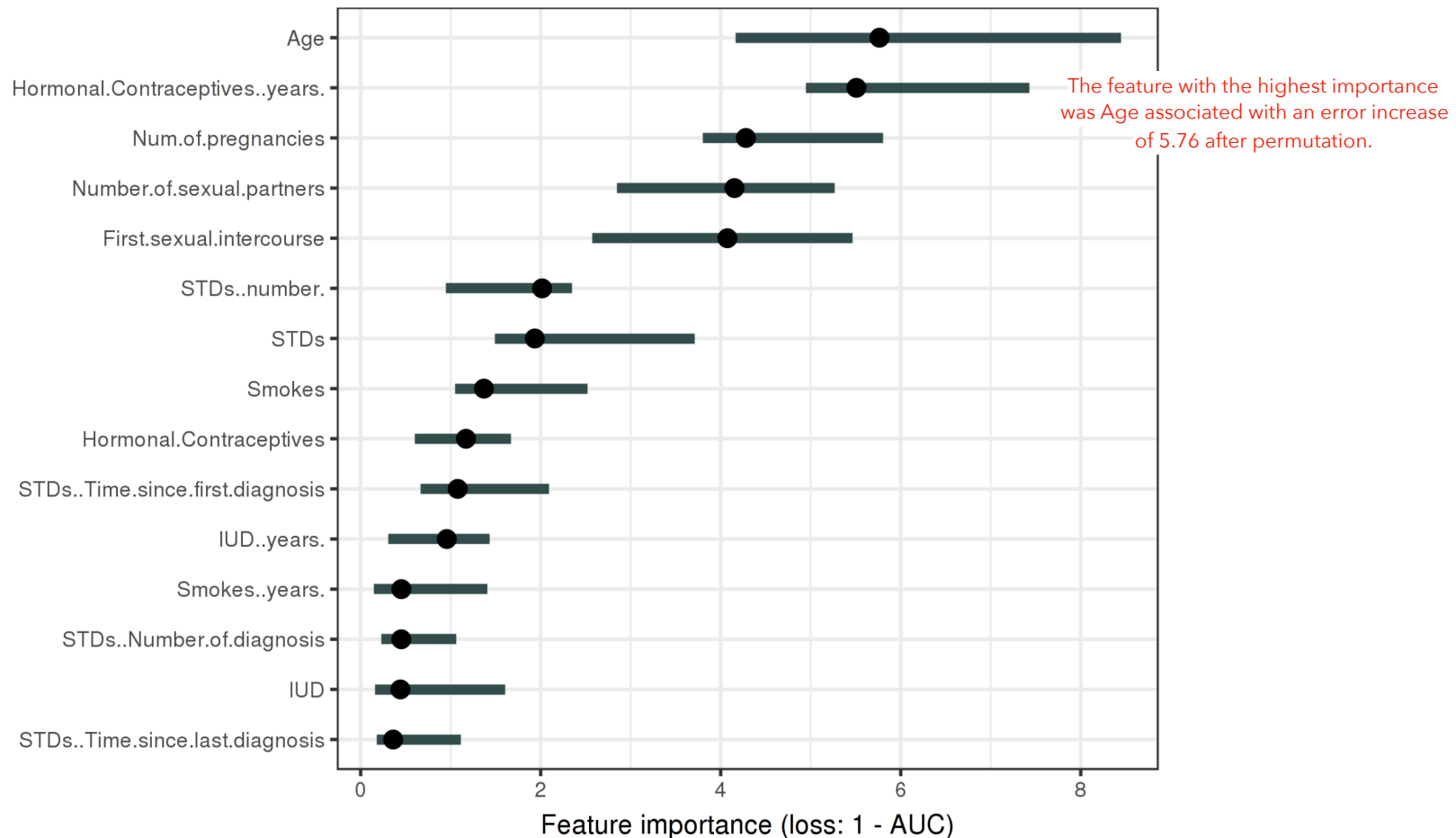
Input: Trained model f , feature matrix X , target vector y , error measure $L(y, f)$.

1. Estimate the original model error $e^{\text{orig}} = L(y, f(X))$ (e.g. mean squared error)
2. For each feature $j = 1, \dots, p$ do:
 - Generate feature matrix X^{perm} by permuting feature j in the data X . This breaks the association between feature j and true outcome y .
 - Estimate error $e^{\text{perm}} = L(Y, f(X^{\text{perm}}))$ based on the predictions of the permuted data.
 - Calculate permutation feature importance $FI^j = e^{\text{perm}} / e^{\text{orig}}$. Alternatively, the difference can be used: $FI^j = e^{\text{perm}} - e^{\text{orig}}$
3. Sort features by descending FI .

Model-Agnostic Models

Permutation Test

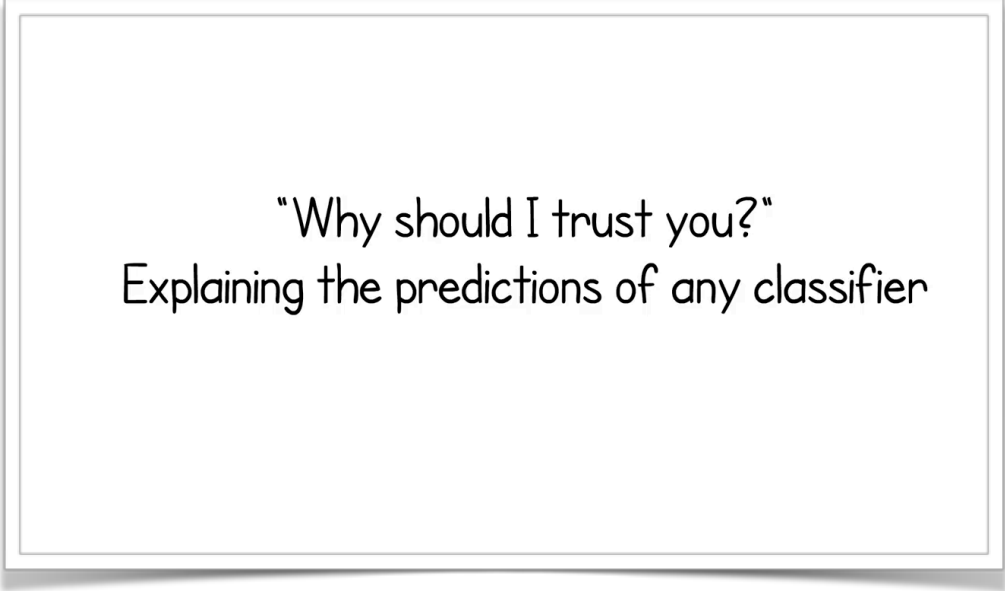
Problem: predict cervical cancer.



Model-Agnostic Models

LIME: Local interpretable model-agnostic explanations.

Local surrogate models are interpretable models that are used to explain individual predictions of black box machine learning models.



"Why should I trust you?"
Explaining the predictions of any classifier

Surrogate models are trained to approximate the predictions of the underlying black box model. Instead of training a global surrogate model, LIME focuses on training local surrogate models to explain individual predictions.

Model-Agnostic Models

LIME: Local interpretable model-agnostic explanations.

Imagine you can probe the box as often as you want.

LIME generates a new dataset consisting of permuted samples and the corresponding predictions of the black box model.

On this new dataset LIME then trains an interpretable model, which is weighted by the proximity of the sampled instances to the instance of interest.

Model-Agnostic Models

LIME: Local interpretable model-agnostic explanations.

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.

Model-Agnostic Models

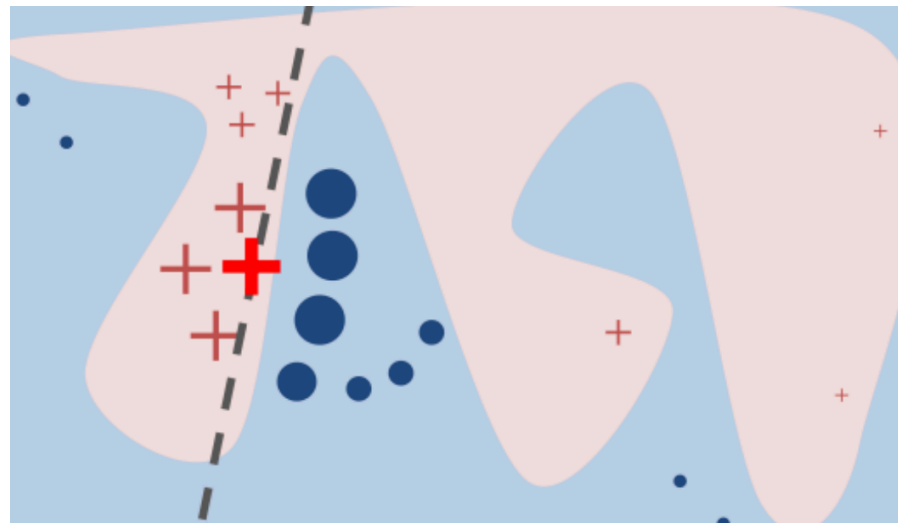
Local Surrogate (LIME)

The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background.

The bright bold red cross is the instance being explained.

LIME samples instances, gets predictions using f , and weighs them by the proximity to the instance being explained (represented here by size).

The dashed line is the learned explanation that is locally (but not globally) faithful.



Model-Agnostic Models

Local Surrogate (LIME) for text

Starting from the original text, new texts are created by randomly removing words from the original text. The dataset is represented with binary features for each word. A feature is 1 if the corresponding word is included and 0 if it has been removed.

Model-Agnostic Models

Local Surrogate (LIME) for images

Intuitively, it would not make much sense to perturb individual pixels, since many more than one pixel contribute to one class. Randomly changing individual pixels would probably not change the predictions by much. Therefore, variations of the images are created by segmenting the image into “superpixels” and turning superpixels off or on.



Original Image



Interpretable
Components

Model-Agnostic Models

Local Surrogate (LIME) for images

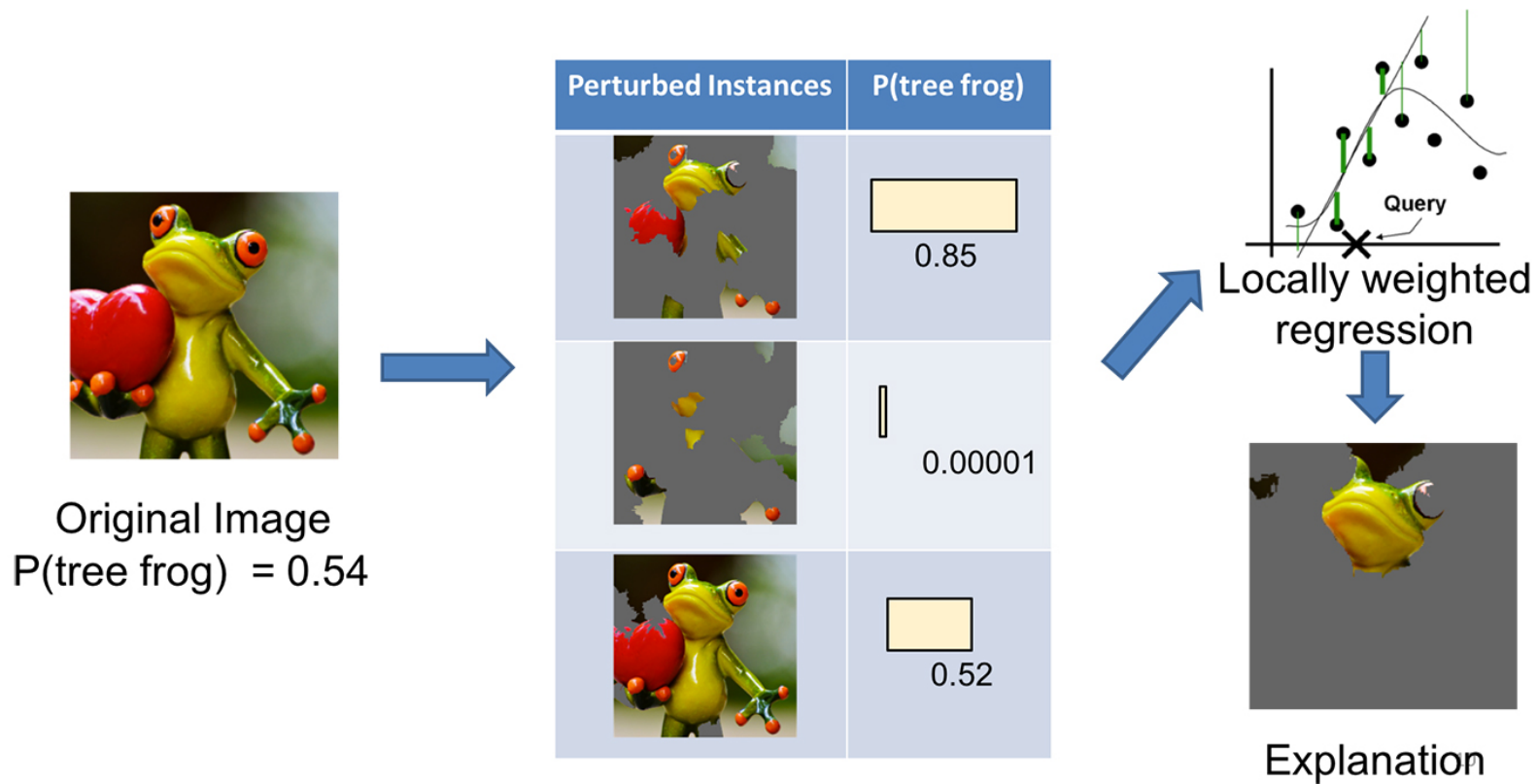
We then generate a data set of perturbed instances by turning some of the interpretable components “off” (in this case, making them gray).

For each perturbed instance, we get the probability that a tree frog is in the image according to the model. We then learn a simple (linear) model on this data set, which is locally weighted—that is, we care more about making mistakes in perturbed instances that are more similar to the original image.

In the end, we present the superpixels with highest positive weights as an explanation, graying out everything else.

Model-Agnostic Models

Local Surrogate (LIME) for images



Model-Agnostic Models

Shapley Values: Game Theory Attribution

In cooperative situations, something known as the Shapley value is used to fairly distribute credit or value to each individual player/participant.



Model-Agnostic Models

Shapley Values: Game Theory Attribution

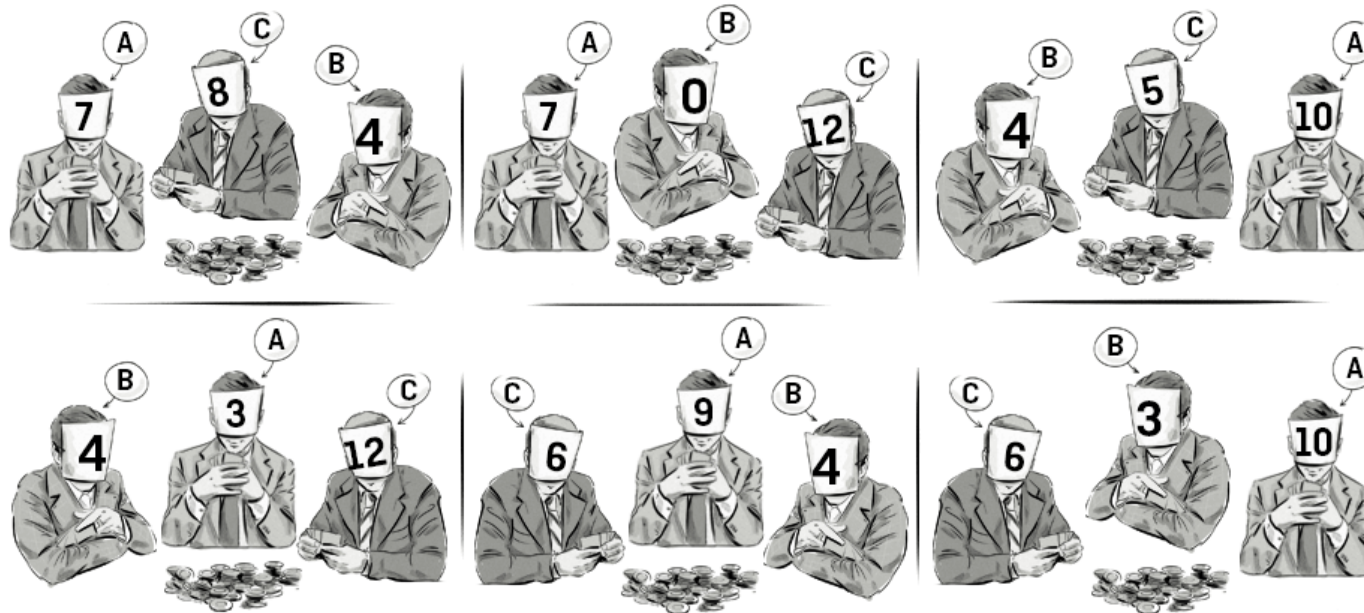
You first start by identifying each player's contribution when they play individually, when 2 play together, and when all 3 play together.



Model-Agnostic Models

Shapley Values: Game Theory Attribution

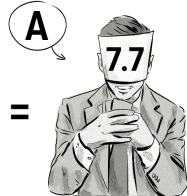
Then, you need to consider all possible orders and calculate their marginal value – e.g. what value does each player add when player A enters the game first, followed by player B, and then player C.

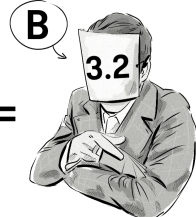


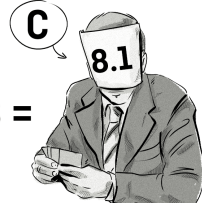
Model-Agnostic Models

Shapley Values: Game Theory Attribution

Now that we have calculated each player's marginal value across all 6 possible order combinations, we now need to add them up and work out the Shapley value (i.e. the average) for each player.

$$(7+7+10+3+9+10) / 6 =$$
A cartoon illustration of a man in a suit, labeled 'A' in a speech bubble. He is holding a card that displays the number '7.7', representing his Shapley value.

$$(4+0+4+4+4+3) / 6 =$$
A cartoon illustration of a man in a suit, labeled 'B' in a speech bubble. He is holding a card that displays the number '3.2', representing his Shapley value.

$$(8+12+5+12+6+6) / 6 =$$
A cartoon illustration of a man in a suit, labeled 'C' in a speech bubble. He is holding a card that displays the number '8.1', representing his Shapley value.

Model-Agnostic Models

Shapley Values: Game Theory Attribution

A prediction can be explained by assuming that each feature value of the instance is a “player” in a game where the prediction is the payout and the Shapley value is the average marginal contribution of a feature value across all possible coalitions.

All possible coalitions (sets) of feature values have to be evaluated with and without the j -th feature to calculate the exact Shapley value. For more than a few features, the exact solution to this problem becomes problematic as the number of possible coalitions exponentially increases as more features are added and several approximations have been proposed.

Model-Agnostic Models

Shapley Values: Game Theory Attribution

The Shapley value can be misinterpreted.

The Shapley value of a feature value is not the difference of the predicted value after removing the feature from the model training.

The interpretation of the Shapley value is: **Given the current set of feature values, the contribution of a feature value to the difference between the actual prediction and the mean prediction is the estimated Shapley value.**