

Explainable Machine Learning

Local Model-Agnostic Methods

Shim Jaewoong

jaewoong@seoultech.ac.kr

Model-Agnostic Methods

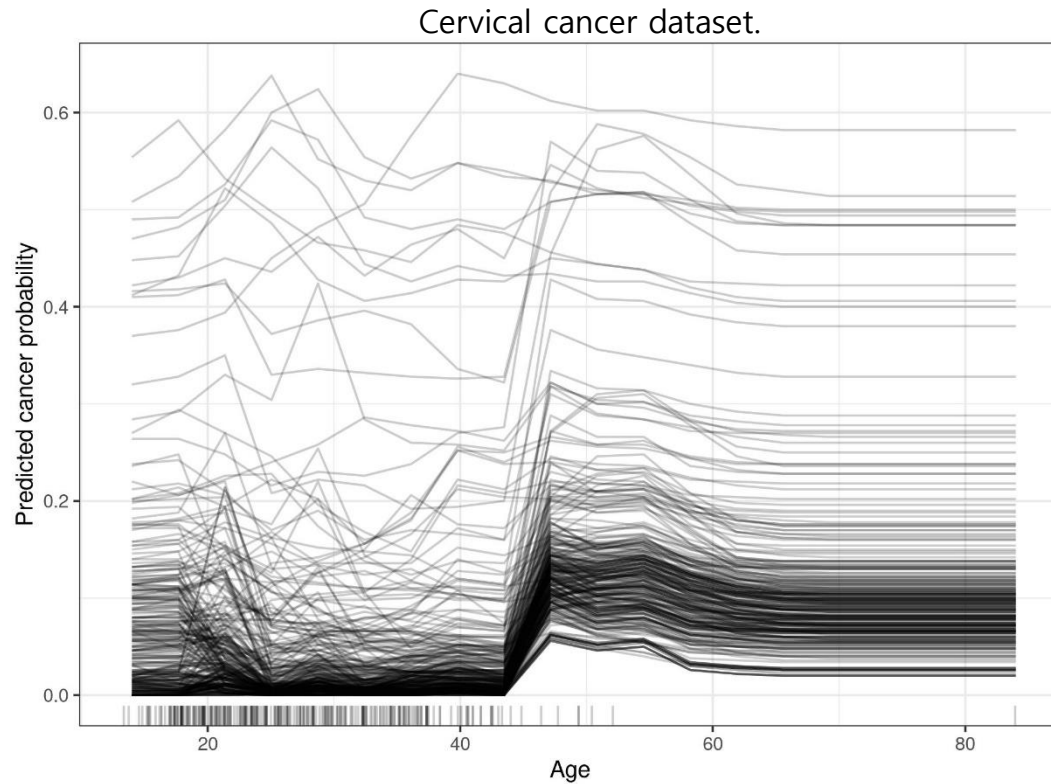
- Global model-agnostic
 - describe how features affect the prediction **on average**
 - Understand the general mechanisms
- Local model-agnostic
 - explain **individual predictions**

Individual Conditional Expectation (ICE)

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**
- A PDP for individual data instances
- **PDP** : average effect of a feature. *A PDP is the average of the lines of an ICE plot.*

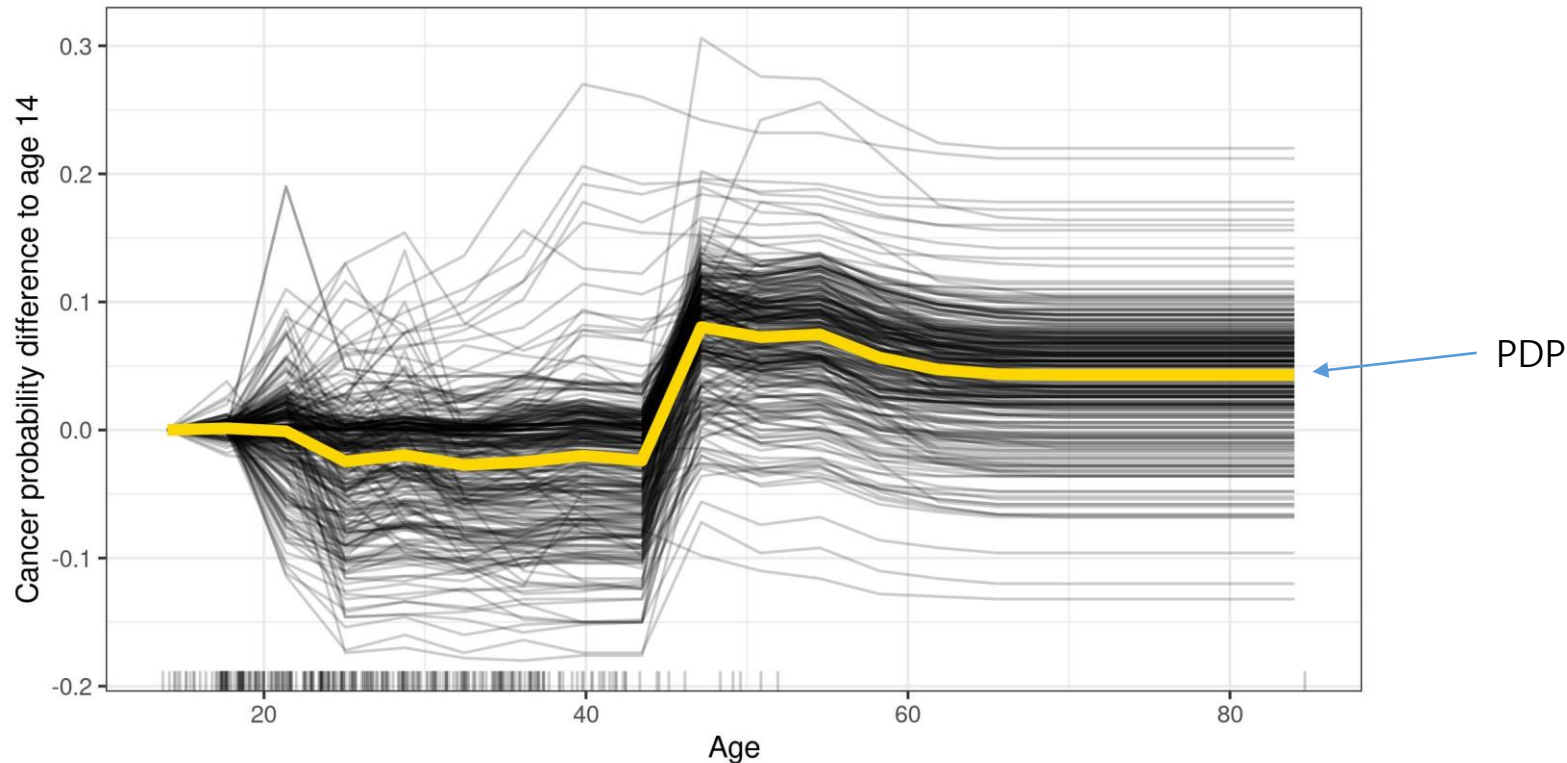


most women the age effect follows the average pattern of an increase at age 50, but there are some exceptions: For the few women that have a high predicted probability at a young age, the predicted cancer probability does not change much with age.

Individual Conditional Expectation (ICE)

- Centered ICE plot

- Sometimes it can be hard to tell whether the ICE curves differ between individuals because they start at different predictions.
- Anchoring the curves at the lower end of the feature
- It can be useful if we want to see the difference in the prediction compared to a fixed point.



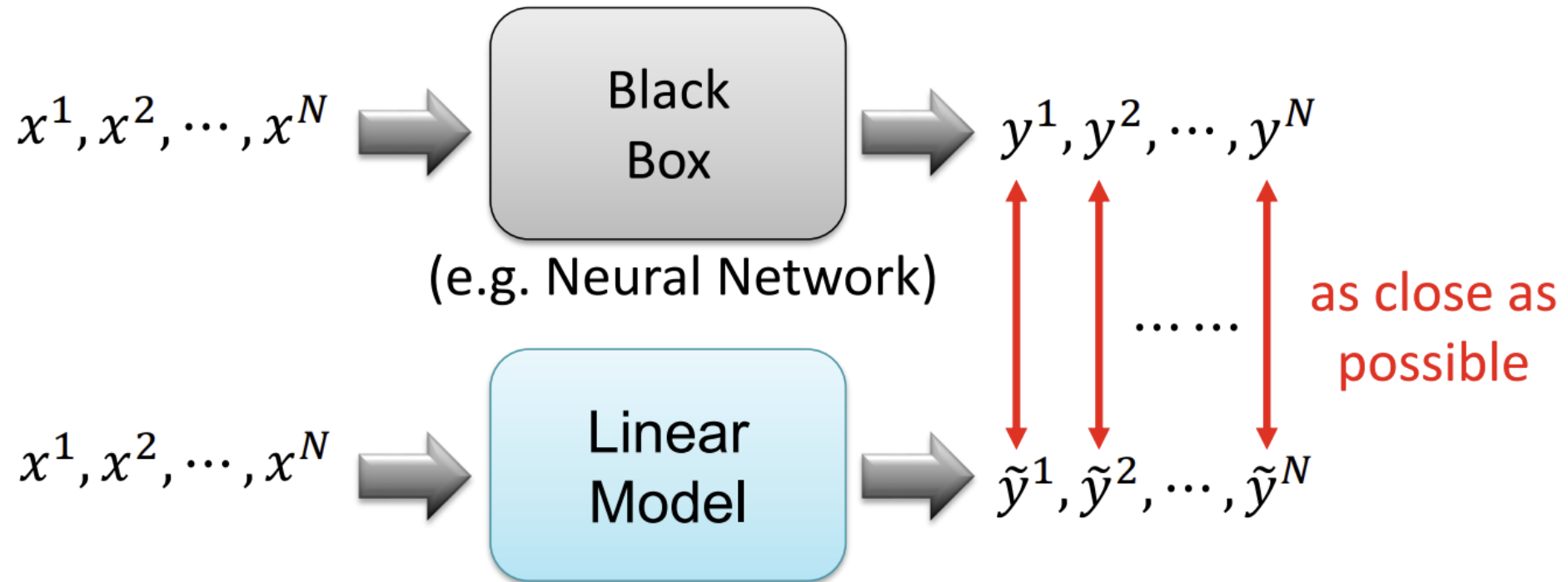
Individual Conditional Expectation (ICE)

- Advantages
 - **even more intuitive to understand** than partial dependence plots.
 - can **uncover heterogeneous relationships**.
- Disadvantages
 - **can only display one feature** meaningfully
 - If many ICE curves are drawn, the **plot can become overcrowded**

Local Surrogate

Local interpretable model-agnostic explanations (LIME)

Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016.



Can't do it globally of course, but locally ? Main Idea behind LIME

LIME

- Surrogate models
 - 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.

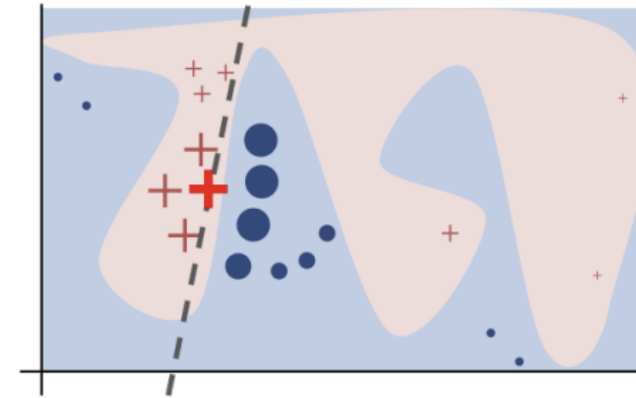


Figure 3: Toy example to present intuition for LIME. The black-box model's complex decision function f (unknown to LIME) is represented by the blue/pink background, which cannot be approximated well by a linear model. The bright bold red cross is the instance being explained. LIME samples instances, 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.

- **Local surrogate models**

- Mathematically,

Local surrogate model (explanation model for instance x)

$$\xi(x) = \operatorname{argmin}_{g \in G} \mathcal{L}(f, g, \Pi_x) + \Omega(g)$$

f : original model (black box)

G : is the family of possible explanations (e.g. all possible linear regression models)

Π_x : proximity measure

Ω : model complexity

- **Lasso** can be a good choice.
 - In practice, user determines the complexity, e.g. by selecting the maximum number of features **K** that the linear regression model may use.
 - Forward or backward selection can be used.

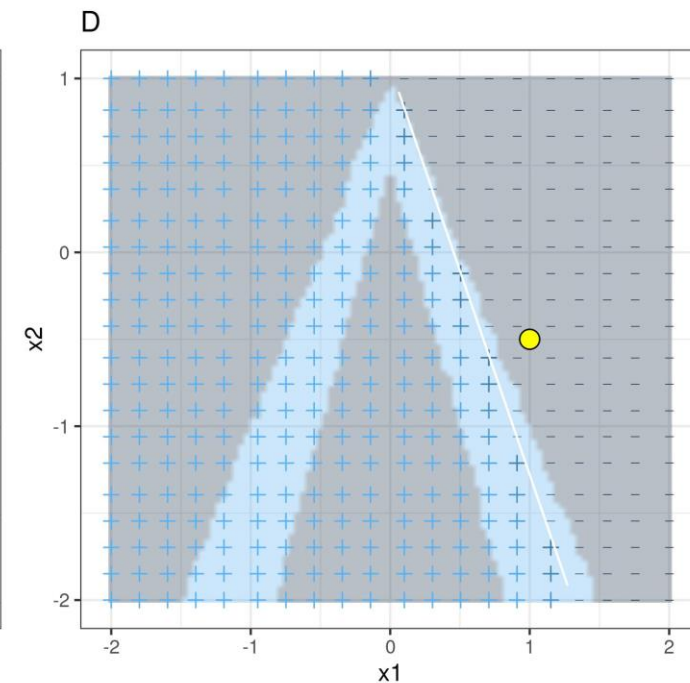
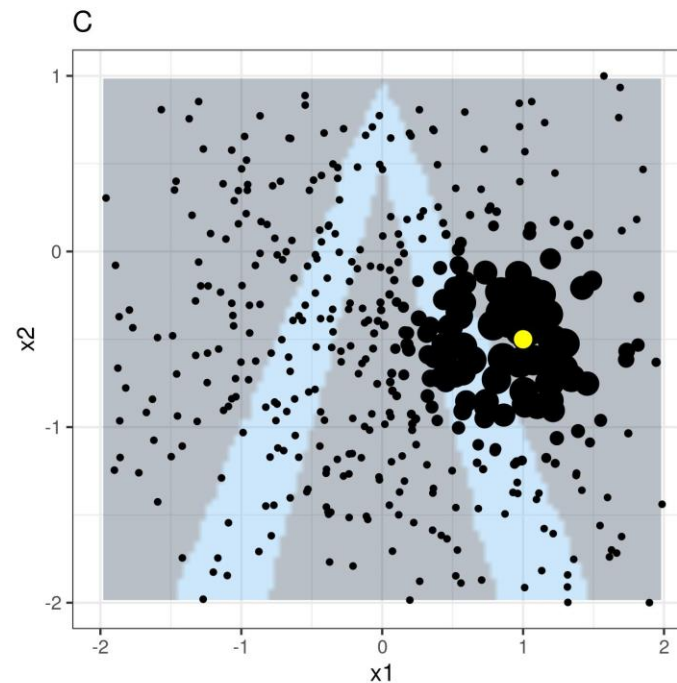
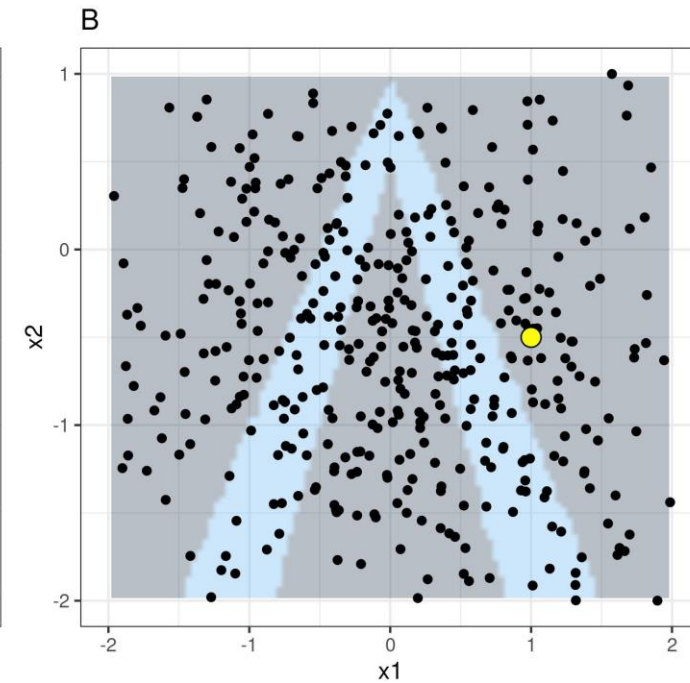
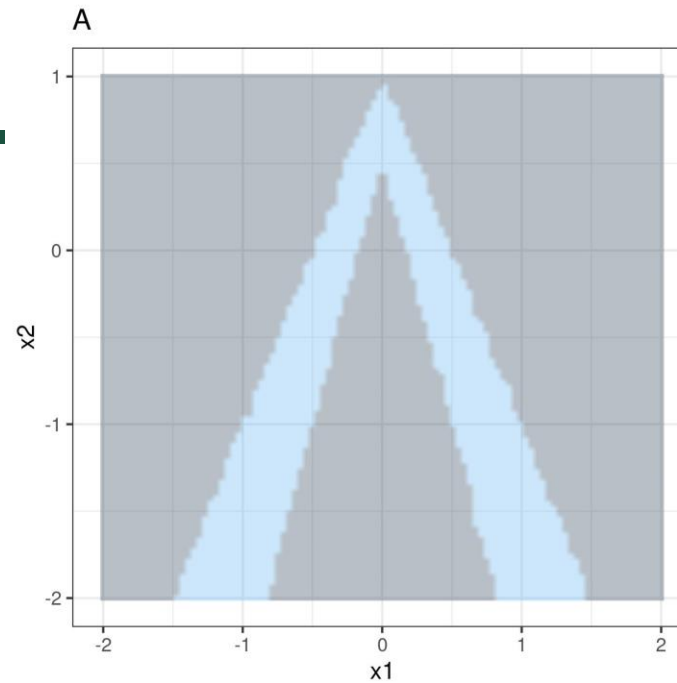
▪ Local surrogate models

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. **(Generate samples from neighbors)**
- 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.

LIME – tabular data

- A) Random forest predictions given features x_1 and x_2 . Predicted classes: 1 (dark) or 0 (light).
- B) Instance of interest (big dot) and data sampled from a normal distribution (small dots).
- C) Assign higher weight to points near the instance of interest.
- D) Signs of the grid show the classifications of the locally learned model from the weighted samples. The white line marks the decision boundary ($P(\text{class}=1) = 0.5$).



LIME – tabular data

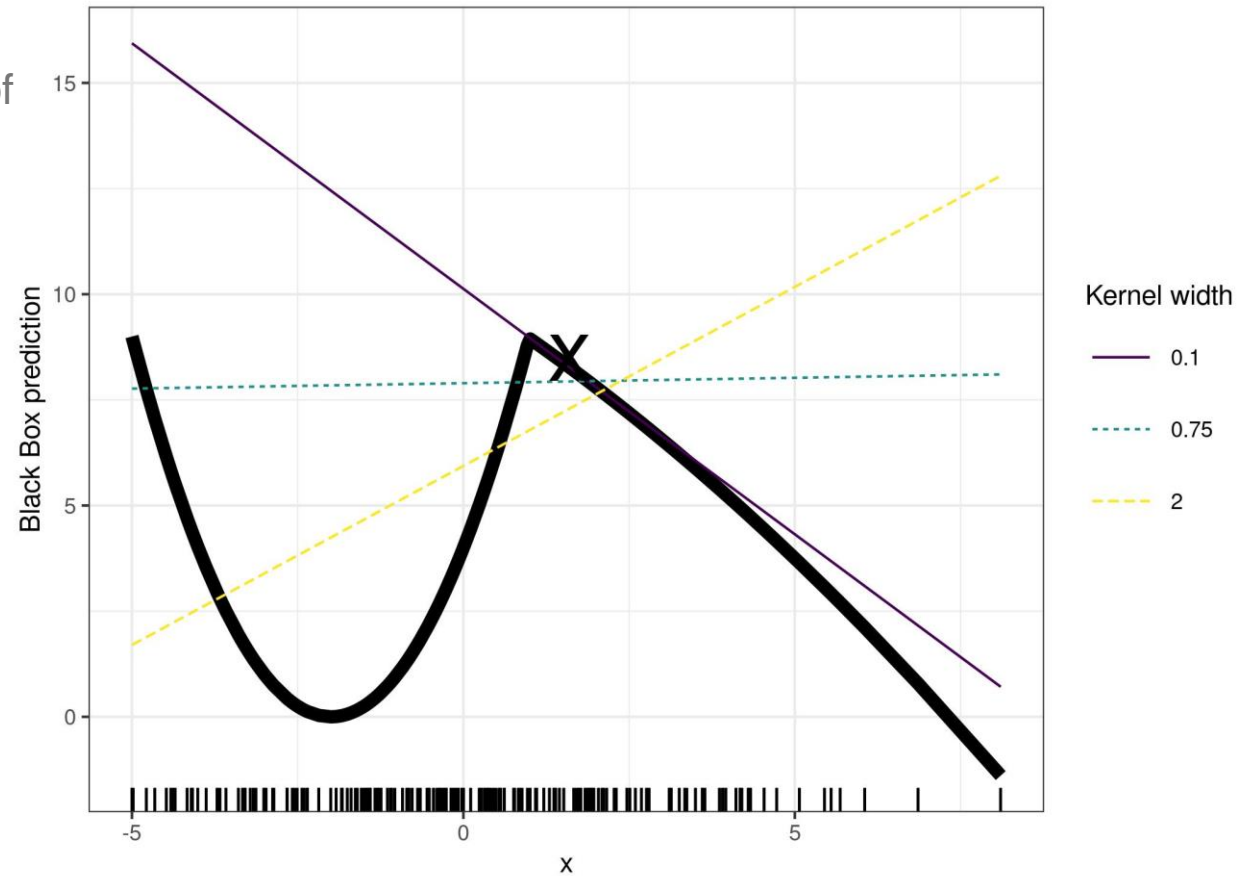
- Defining a meaningful neighborhood is difficult
 - Proximity measure
 - Exponential smoothing kernel

https://lime-ml.readthedocs.io/en/latest/lime.html#module-lime.lime_tabular

$$\Pi_x(z) = \exp(-D(x, z)^2 / \sigma^2)$$

Explanation of the prediction of instance $x = 1.6$. The predictions of the black box model depending on a single feature is shown as a thick line and the distribution of the data is shown with rugs. Three local surrogate models with different kernel widths are computed. The resulting linear regression model depends on the kernel width

It gets worse in high-dimensional feature spaces!



LIME – text

- How to perturb the data points?
 - 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.

LIME – text

■ Example

- Spam detection on YouTube comments
 - The black box model : deep decision tree trained on the term-frequency matrix.
 - Other approach is also available (recurrent neural network or SVM,..)

CONTENT		CLASS
267	PSY is a good guy	0
173	For Christmas Song visit my channel! ;)	1

create some variations

For	Christmas	Song	visit	my	channel!	;)	prob	weight
1	0	1	1	0	0	1	0.17	0.57
0	1	1	1	1	0	1	0.17	0.71
1	0	0	1	1	1	1	0.99	0.71
1	0	1	1	1	1	1	0.99	0.86
0	1	1	1	0	0	1	0.17	0.57

predicted probability from the black box model

proximity of the variation to the original sentence

LIME – text

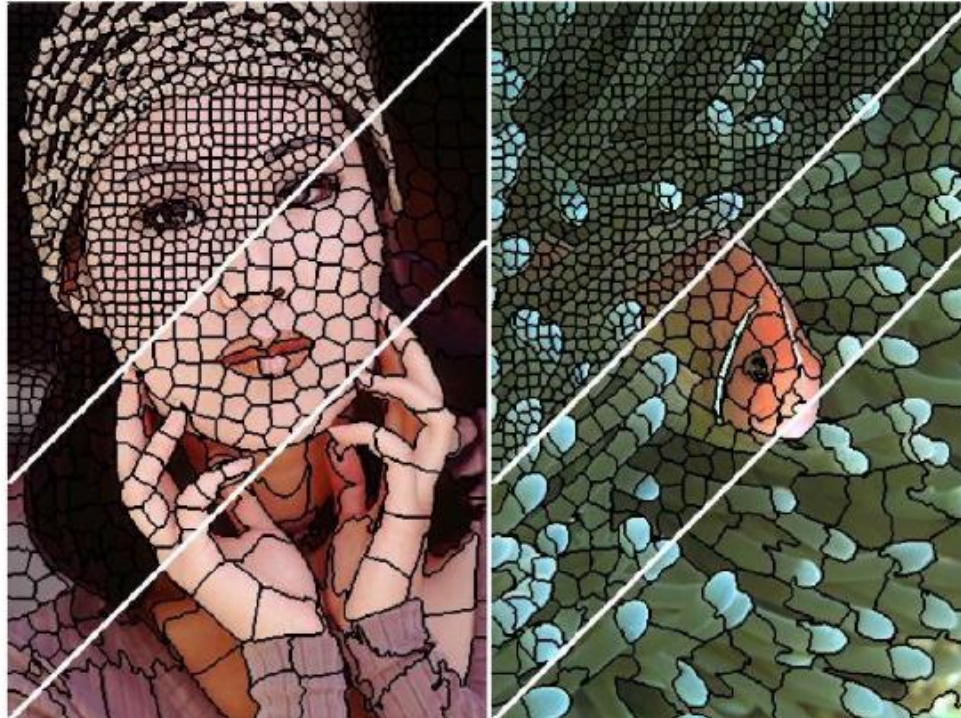
- Example
 - Spam detection on YouTube comments
 - Train lasso regression with weight.

case	label_prob	feature	feature_weight
2	0.9939024	channel!	6.180747
2	0.9939024	;)	0.000000
2	0.9939024	visit	0.000000

The word “channel” indicates a high probability of spam.

LIME – image

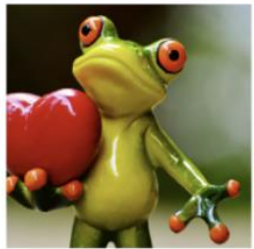
- How to perturb the data points?
 - many more than one pixel contribute to one class
 - not make much sense to perturb individual pixels
 - segmenting the image into “superpixels” and turning superpixels off or on.
 - “off” : replace each pixel with gray color.



Any superpixel algorithm
such as 'SLIC' could be used

LIME – image

1. Given a data point you want to explain
2. Sample at the nearby - Each image is represented as a set of superpixels (segments).



Original Image

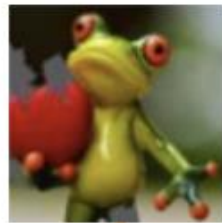


Interpretable
Components



Black

0.85



Black

0.52



Black

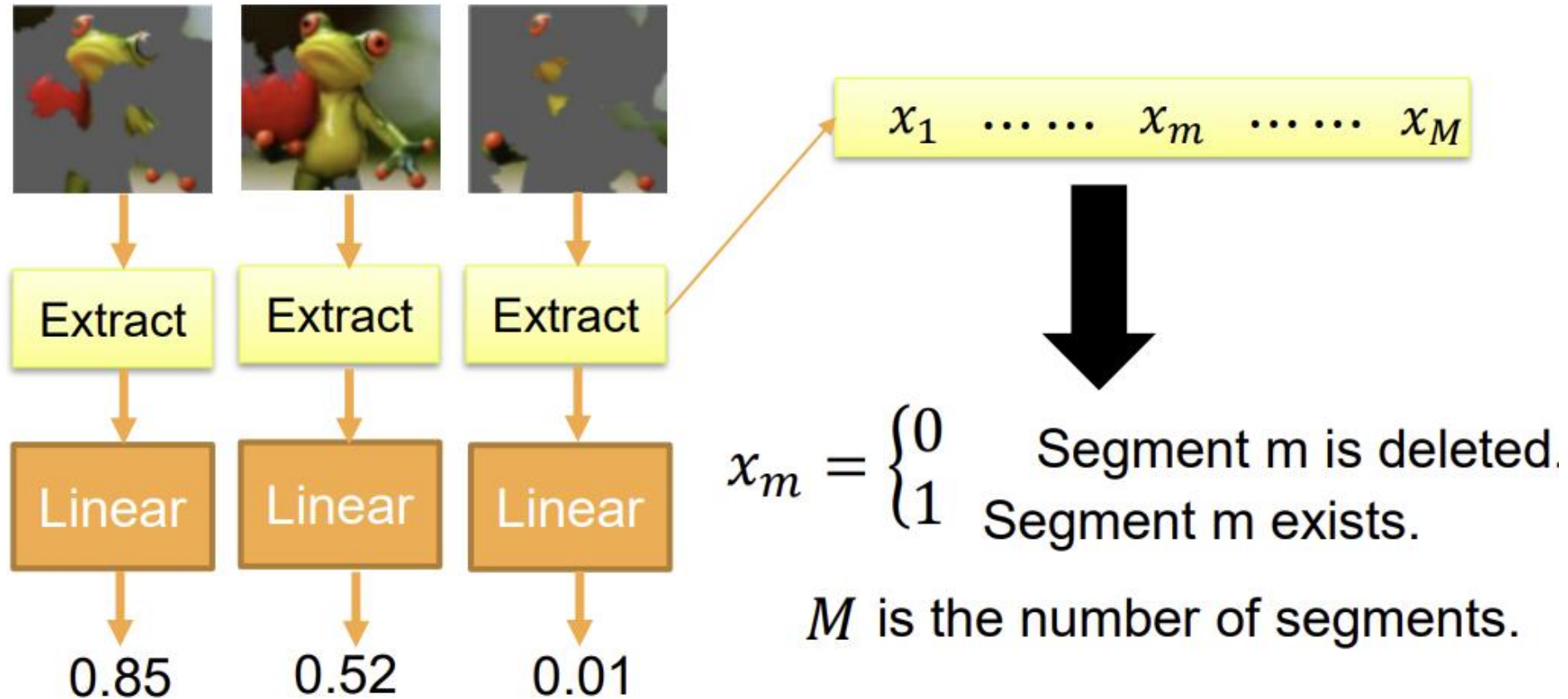
0.01

Randomly delete some
segments.

Compute the probability of “frog” by
black box

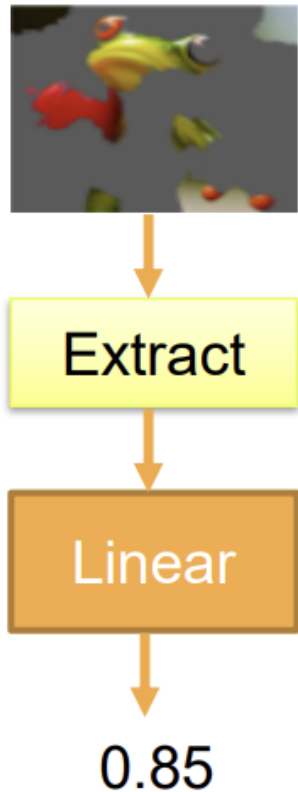
LIME – image

3. Fit with linear (or interpretable) model



LIME – image

4. Interpret the model you learned



$$y = w_1x_1 + \dots + w_mx_m + \dots + w_Mx_M$$

$$x_m = \begin{cases} 0 & \text{Segment } m \text{ is deleted.} \\ 1 & \text{Segment } m \text{ exists.} \end{cases}$$

M is the number of segments.

If $w_m \approx 0$ ➡ segment m is not related to “frog”

If w_m is positive ➡ segment m indicates the image is “frog”

If w_m is negative ➡ segment m indicates the image is not “frog”

LIME – image

- superpixel(segment) 단위의 weight

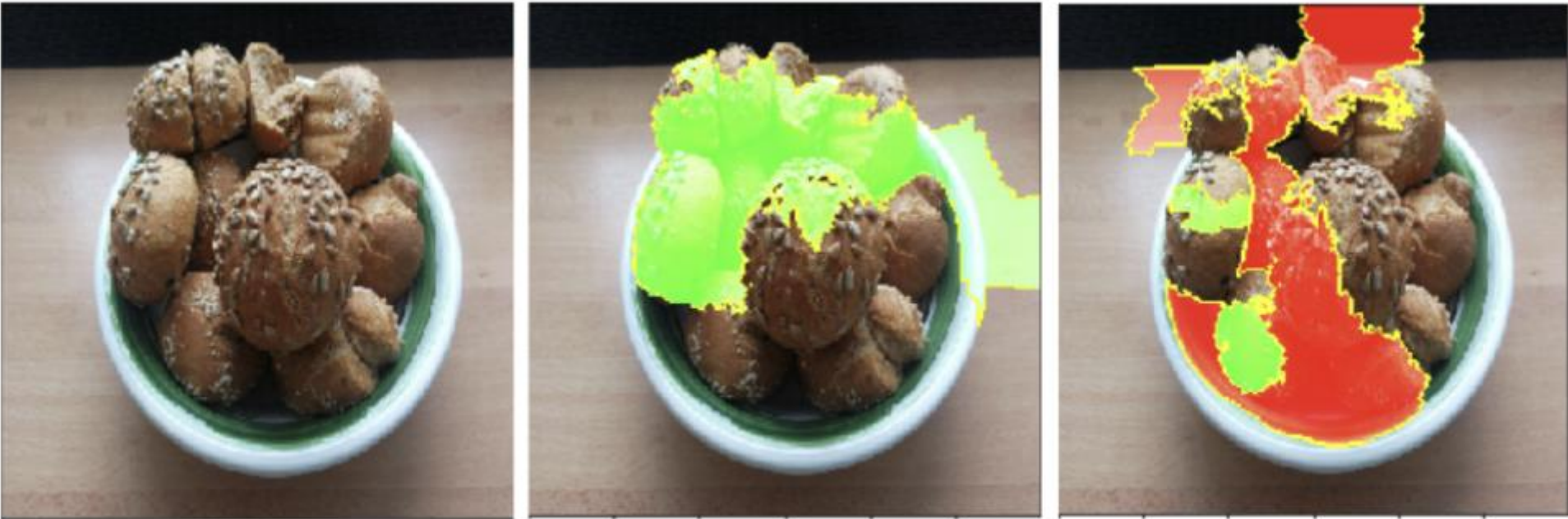


FIGURE 9.8: Left: Image of a bowl of bread. Middle and right: LIME explanations for the top 2 classes (bagel, strawberry) for image classification made by Google's Inception V3 neural network.

LIME

■ Advantages

- LIME is one of the few methods that **works for tabular data, text and images**.
- Very easy to use (<https://github.com/marcotcr/lime>)
- The explanations **can use other interpretable features than the original model was trained on**.
 - A text classifier can rely on abstract word embeddings as features, but the explanation can be based on the presence or absence of words in a sentence
 - The regression model could be trained on components of a principal component analysis (PCA), but LIME might be trained on the original features.

■ Disadvantages

- The correct definition of the neighborhood is a very big, unsolved problem, especially for tabular data.
- the instability of the explanations
 - repeating the sampling process(perturbation), then the explanations can be different.

SHAP

(SHapley Additive exPlanations)

Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." *Advances in neural information processing systems* 30 (2017).

Shapley values

- Shapely value
 - Coined by Shapley (1953)
 - A method from coalitional **game** theory.
 - assigning **payouts** to **players** depending on their contribution to the total payout.

Lloyd Shapley



2012 Nobel Prize
in Economics



SHAP

- Example
 - a machine learning model to predict apartment prices.

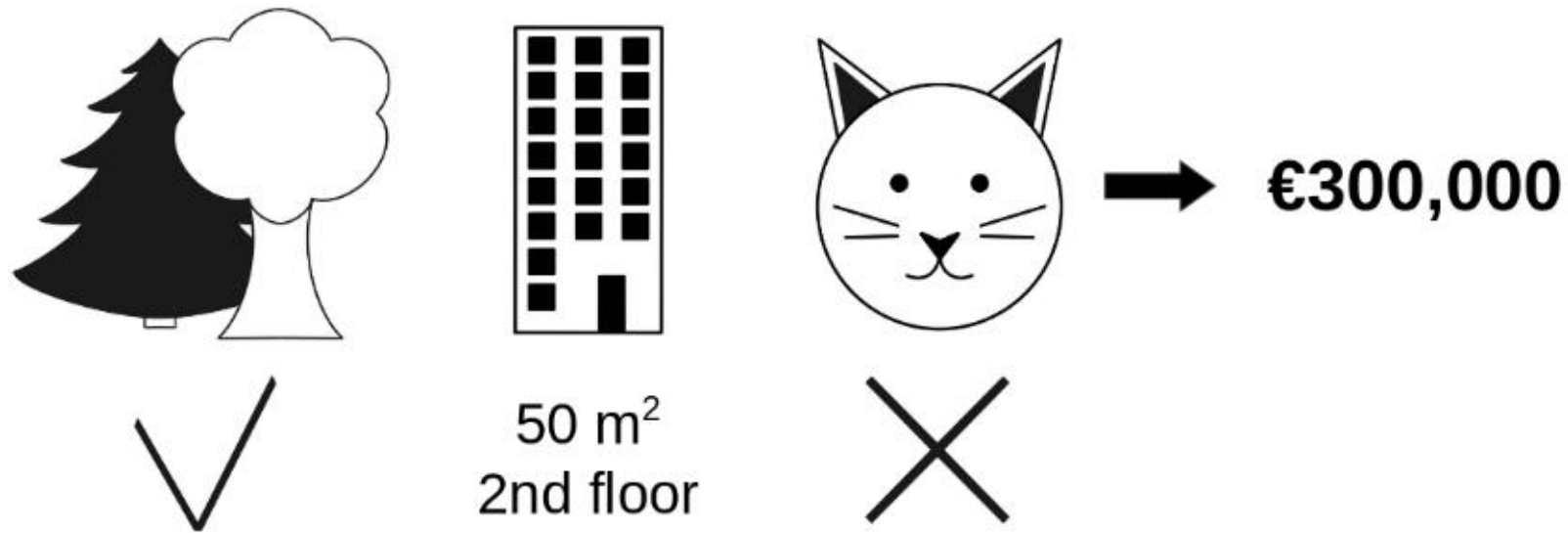


FIGURE 9.17: The predicted price for a 50 m^2 2nd floor apartment with a nearby park and cat ban is €300,000. Our goal is to explain how each of these feature values contributed to the prediction.

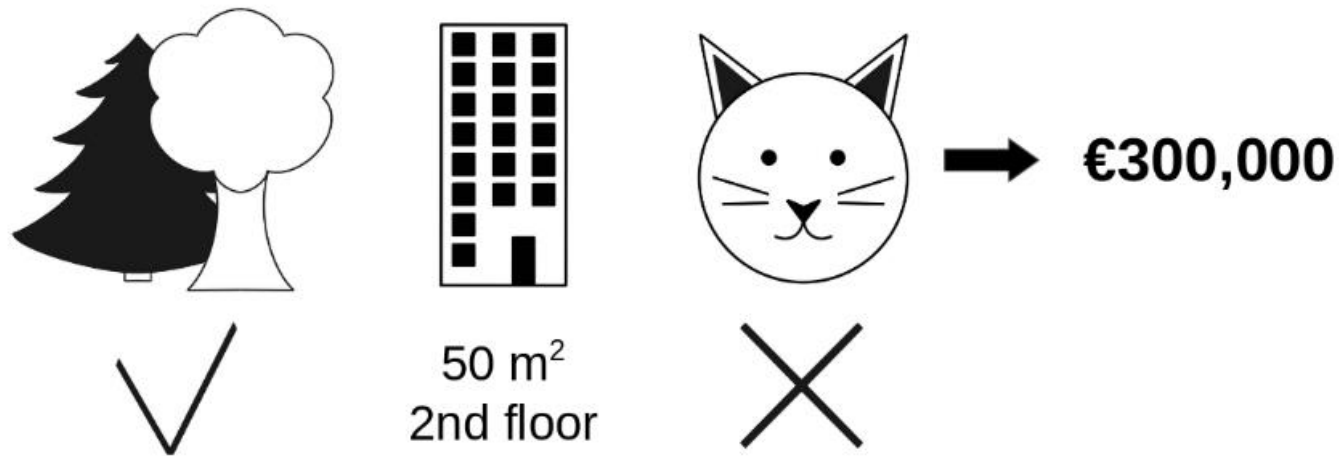
The average prediction for all apartments is €310,000.

How much has **each feature value contributed to the prediction compared to the average prediction?**

SHAP

■ Example

- “game”
 - the **prediction task** for a single instance
- “gain”
 - the actual prediction for this instance - the average prediction
- “players”
 - the **feature values** of the instance that collaborate to predict a certain value



Gain?
Player?

The average prediction for all apartments is €310,000.

FIGURE 9.17: The predicted price for a 50 m² 2nd floor apartment with a nearby park and cat ban is €300,000. Our goal is to explain how each of these feature values contributed to the prediction.

SHAP

- Example
 - results

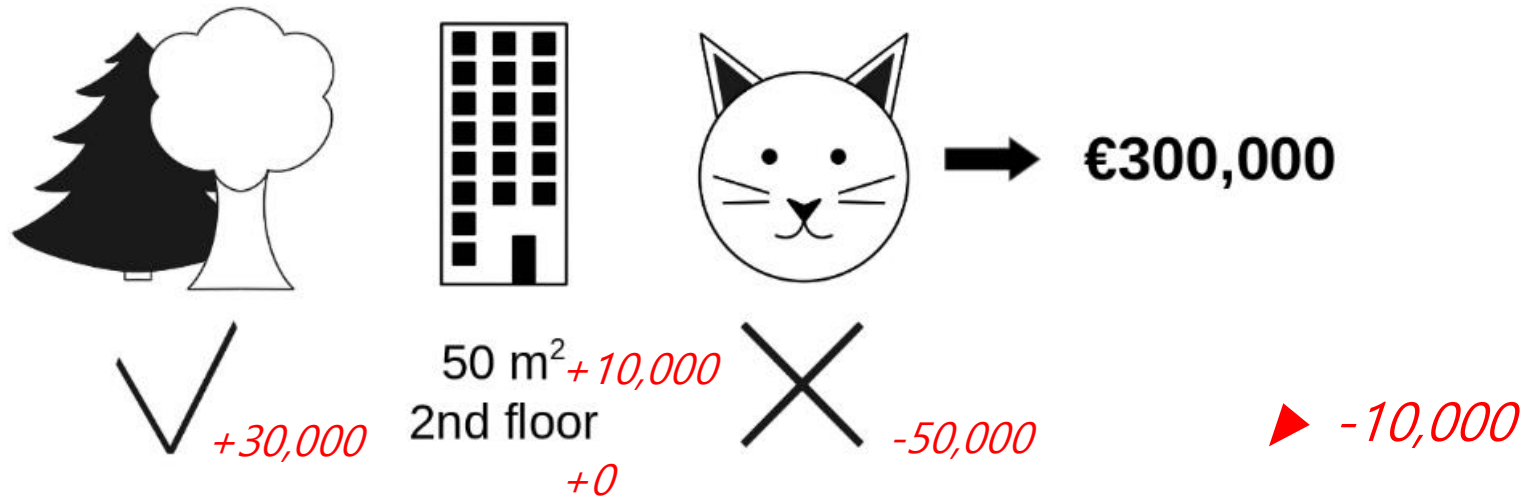


FIGURE 9.17: The predicted price for a 50 m² 2nd floor apartment with a nearby park and cat ban is €300,000. Our goal is to explain how each of these feature values contributed to the prediction.

SHAP

- Example
 - Random forest
 - cervical cancer (classification)
 - Summation to 0.54

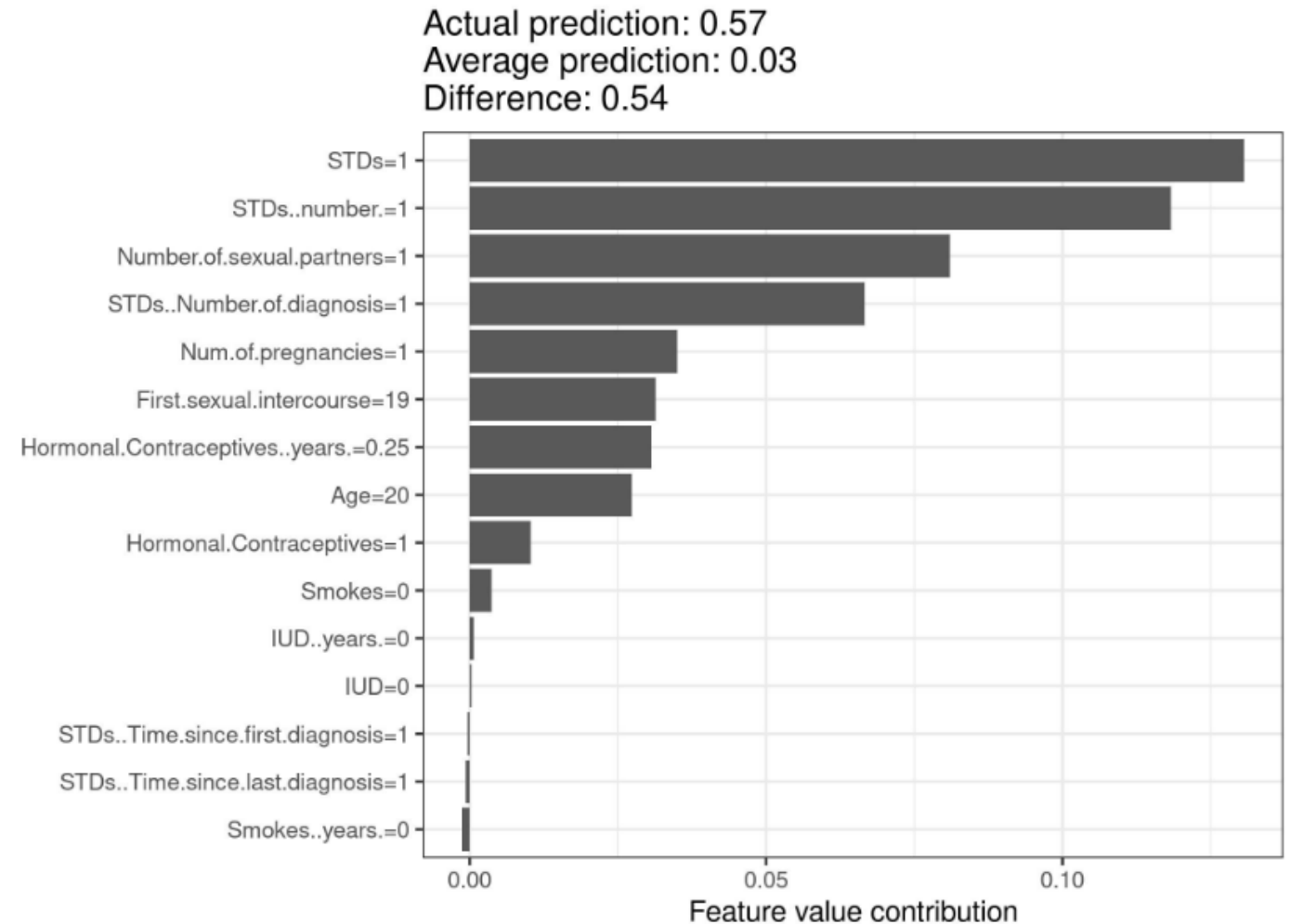


FIGURE 9.20: Shapley values for a woman in the cervical cancer dataset. With a prediction of 0.57, this woman's cancer probability is 0.54 above the average prediction of 0.03. The number of diagnosed STDs increased the probability the most. The sum of contributions yields the difference between actual and average prediction (0.54).

SHAP

- Example
 - Random forest
 - Bike rental (regression)
 - Summation to -2108

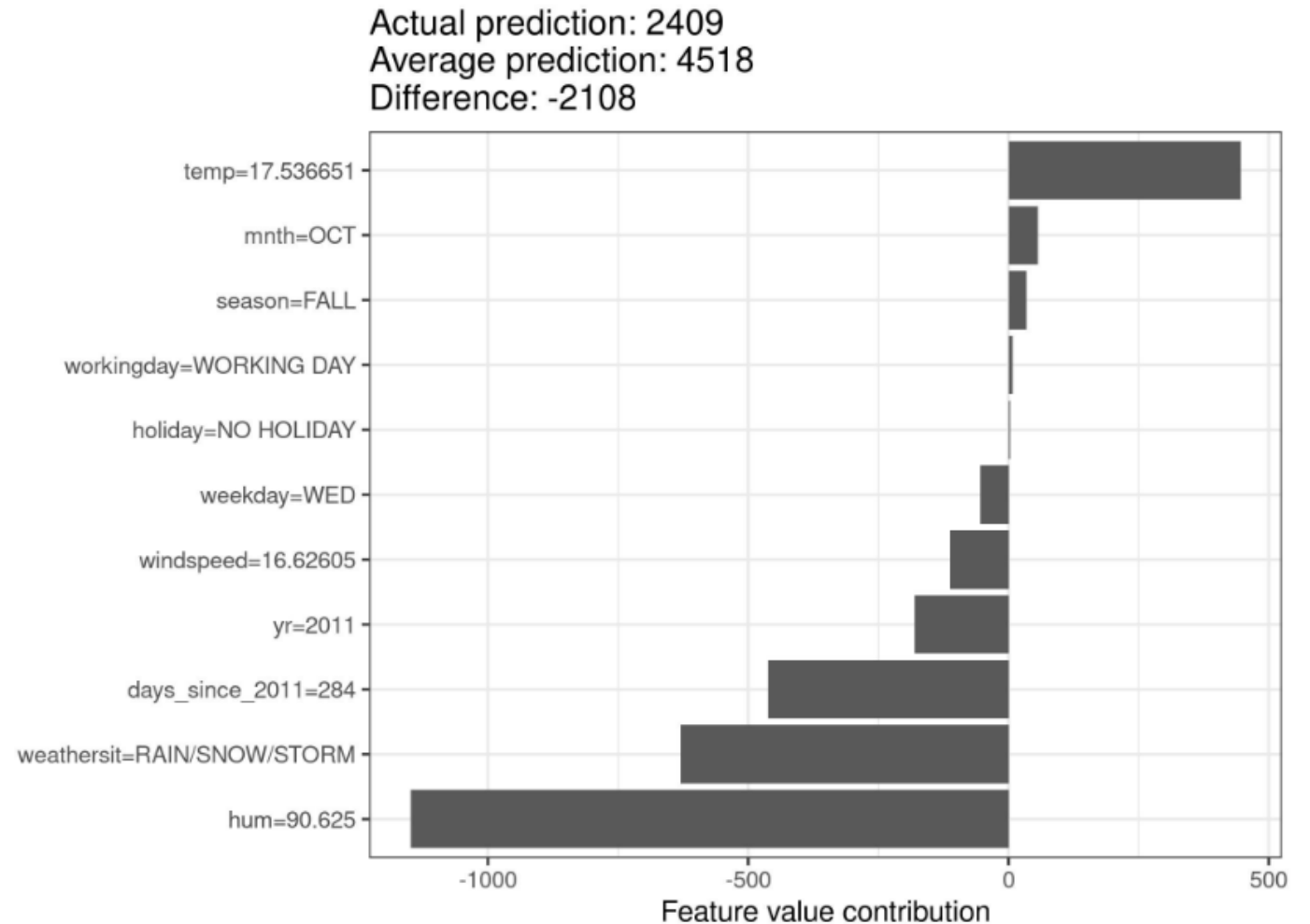


FIGURE 9.21: Shapley values for day 285. With a predicted 2409 rental bikes, this day is -2108 below the average prediction of 4518. The weather situation and humidity had the largest negative contributions. The temperature on this day had a positive contribution. The sum of Shapley values yields the difference of actual and average prediction (-2108).

■ SHAP

- Example: 변수 3개
 - X1의 Shapley value ?



	X1 사용	X2 사용	X3 사용	예측 값
Case ①	X	X	X	28
Case ②	O	X	X	32
Case ③	X	O	X	31
Case ④	X	X	O	30
Case ⑤	O	O	X	32
Case ⑥	O	X	O	33
Case ⑦	X	O	O	32
Case ⑧	O	O	O	35

학습 데이터에 대한 예측 평균

■ SHAP

- Example: 변수 3개
- X1의 Shapley value ?

	X1 사용	X2 사용	X3 사용	예측 값
Case ①	X	X	X	28
Case ②	O	X	X	32
Case ③	X	O	X	31
Case ④	X	X	O	30
Case ⑤	O	O	X	32
Case ⑥	O	X	O	33
Case ⑦	X	O	O	32
Case ⑧	O	O	O	35

$$\textcircled{2} - \textcircled{1} = 4$$

$$\textcircled{5} - \textcircled{3} = 1$$

$$\textcircled{6} - \textcircled{4} = 3$$

$$\textcircled{8} - \textcircled{7} = 3$$

변수 0개에서 1개가 되는 경우의 수: 3

→ Weight: 1/3

변수 1개에서 2개가 되는 경우의 수: 6

→ Weight: 1/6

변수 2개에서 3개가 되는 경우의 수: 3

→ Weight: 1/3

$$\text{X1에 대한 Shapley value}(\phi_1) = \frac{1}{3} \times 4 + \frac{1}{6} \times 1 + \frac{1}{6} \times 3 + \frac{1}{3} \times 3 = 3$$

■ SHAP

- Example: 변수 3개
 - 모든 변수에 대해서도 계산해보면,

	X1 사용	X2 사용	X3 사용	예측 값
Case ①	X	X	X	28
Case ②	O	X	X	32
Case ③	X	O	X	31
Case ④	X	X	O	30
Case ⑤	O	O	X	32
Case ⑥	O	X	O	33
Case ⑦	X	O	O	32
Case ⑧	O	O	O	35

$$\phi_1 = SHAP_{X1}(obs_1) = \frac{1}{3} \times 4 + \frac{1}{6} \times 1 + \frac{1}{6} \times 3 + \frac{1}{3} \times 3 = 3$$

$$\phi_2 = SHAP_{X2}(obs_1) = \frac{1}{3} \times 3 + \frac{1}{6} \times 0 + \frac{1}{6} \times 2 + \frac{1}{3} \times 2 = 2$$

$$\phi_3 = SHAP_{X3}(obs_1) = \frac{1}{3} \times 2 + \frac{1}{6} \times 1 + \frac{1}{6} \times 1 + \frac{1}{3} \times 3 = 2$$

The Shapley Value in Detail

- **Shapley value**

- The Shapley value of a feature value is its contribution to the payout, weighted and summed over **all possible feature value combinations**

$$\phi_j(val) = \sum_{S \subseteq \{1, \dots, p\} \setminus \{j\}} \frac{\overset{\text{Weight}}{|S|! (p - |S| - 1)!}}{p!} \underset{\text{Marginal contribution}}{(val(S \cup \{j\}) - val(S))}$$

The Shapley Value in Detail

- **value function** of players (val) :
 - the prediction for feature values in set S that are marginalized over features that are not included in set S

$$val_x(S) = \underbrace{\int \hat{f}(x_1, \dots, x_p) d\mathbb{P}_{x \notin S}}_{\text{Feature effect}} - \underbrace{E_X(\hat{f}(X))}_{\text{Average effect}}$$

- The machine learning model works with 4 features **x1, x2, x3 and x4**
- we evaluate the prediction for the coalition S consisting of feature values x1 and x3:

$$val_x(S) = val_x(\{1, 3\}) = \int_{\mathbb{R}} \int_{\mathbb{R}} \hat{f}(x_1, X_2, x_3, X_4) d\mathbb{P}_{X_2 X_4} - E_X(\hat{f}(X))$$

The Shapley Value in Detail

- Properties of the Shapley value (for a fair payout)

- **Efficiency**

- The feature contributions must add up to the difference of prediction for x and the average.

$$\sum_{j=1}^P \phi_j = \hat{f}(x) - E_X(\hat{f}(X))$$

- **Symmetry**

- The contributions of two feature values j and k should be the same if they contribute equally to all possible coalitions.

- **Dummy**

- A feature j that does not change the predicted value – regardless of which coalition of feature values it is added to – should have a Shapley value of 0.

- **Additivity**

For a game with combined payouts $val + val^+$ the respective Shapley values are as follows:

$$\phi_j + \phi_j^+$$

The Shapley Value in Detail

▪ Intuition

- The Shapley value of a feature value is the **average change** in the prediction that the coalition already in the room receives when the feature value joins them.

A player
(Feature value)



join



Average change in performance

■ Estimating the Shapley Value

- the exact solution problematic as **the number of possible coalitions exponentially increases** as more features are added.
- → Approximation with Monte-Carlo sampling:

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

Instance x
of interest

x1	x2	x4					xp
----	----	----	--	--	--	--	----

Randomly
selected
instance z

z1	z2	z3					zp
----	----	----	--	--	--	--	----

Randomly
assembled

x1	z2	x3		xj			zp
----	----	----	--	----	--	--	----

x1	z2	x3		zj			zp
----	----	----	--	----	--	--	----

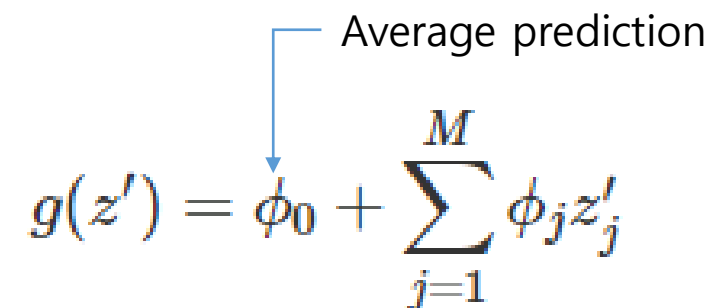
Approximate Shapley estimation for single feature value:

- Output: Shapley value for the value of the j -th feature
- Required: Number of iterations M , instance of interest x , feature index j , data matrix X , and machine learning model f
 - For all $m = 1, \dots, M$:
 - Draw random instance z from the data matrix X
 - Choose a random permutation o of the feature values
 - Order instance x : $x_o = (x_{(1)}, \dots, x_{(j)}, \dots, x_{(p)})$
 - Order instance z : $z_o = (z_{(1)}, \dots, z_{(j)}, \dots, z_{(p)})$
 - Construct two new instances
 - With j : $x_{+j} = (x_{(1)}, \dots, x_{(j-1)}, x_{(j)}, z_{(j+1)}, \dots, z_{(p)})$
 - Without j : $x_{-j} = (x_{(1)}, \dots, x_{(j-1)}, z_{(j)}, z_{(j+1)}, \dots, z_{(p)})$
 - Compute marginal contribution: $\phi_j^m = \hat{f}(x_{+j}) - \hat{f}(x_{-j})$
- Compute Shapley value as the average: $\phi_j(x) = \frac{1}{M} \sum_{m=1}^M \phi_j^m$

SHAP (SHapley Additive exPlanations)

■ SHAP

The representation as a linear model of coalitions


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

Average prediction

$z' \in \{0, 1\}^M$ is the coalition vector (Simplified feature) e.g. $(1, 0, 1, 1, 0, 0, 1, \dots, 0)$

KernelSHAP

- Approximation method

- Procedure
 - 1. Randomly generate **coalition vectors (simplified vectors)**
 - 2. Map to the original feature space
 - 3. Compute the weight with the **SHAP kernel**
 - 4. Fit weighted linear model
 - 5. Return the coefficients from the linear model.

KernelSHAP

- 1. Randomly generate **coalition vectors (simplified vectors)**
 - (1: feature present, 0: feature absent)
 - The K sampled coalitions become the dataset for the regression model.
- 2. Map to the original feature space
 - 0** <- value from the randomly selected instance
 - 1** <- value from the instance of interest

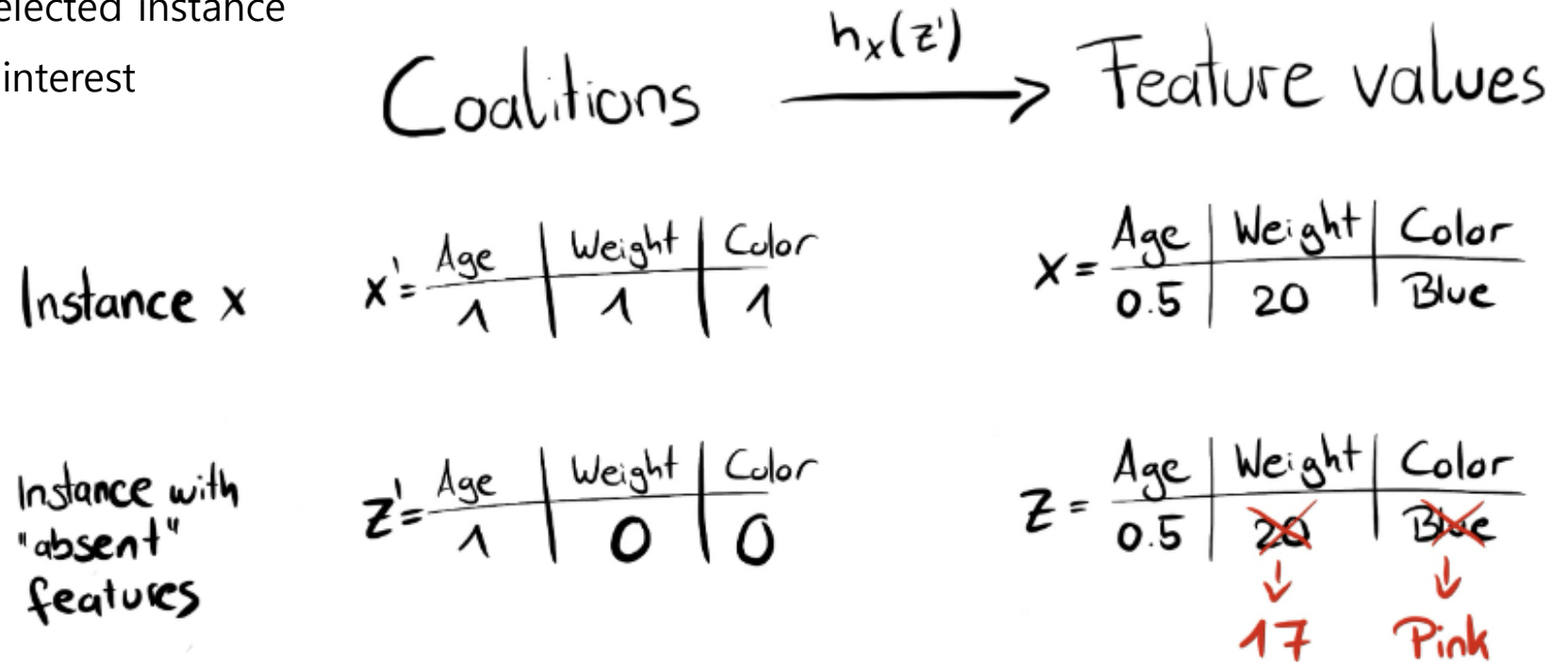


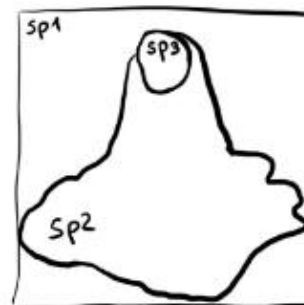
FIGURE 9.22: Function h_x maps a coalition to a valid instance. For present features (1), h_x maps to the feature values of x. For absent features (0), h_x maps to the values of a randomly sampled data instance.

KernelSHAP

- possible mapping function for image

Coalitions of superpixels $\xrightarrow{h_x(z')}$ Image

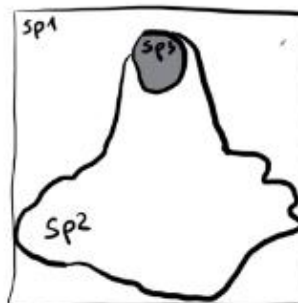
Instance x



sp1	sp2	sp3
1	1	1



Instance x
with absent
features



sp1	sp2	sp3
1	1	0



FIGURE 9.23: Function h_x maps coalitions of superpixels (sp) to images. Superpixels are groups of pixels. For present features (1), h_x returns the corresponding part of the original image. For absent features (0), h_x greys out the corresponding area. Assigning the average color of surrounding pixels or similar would also be an option.

KernelSHAP

- 3. Compute the weight with the **SHAP kernel**
 - **LIME**: weights the instances according to how close they are to the original instance.
 - The more 0's in the coalition vector, the smaller the weight in LIME.
 - **SHAP**: weights the sampled instances according to the weight in the Shapley value estimation
 - Small coalitions (few 1's) and large coalitions (i.e. many 1's) get the largest weights.

SHAP kernel:

$$\pi_x(z') = \frac{(M-1)}{\binom{M}{|z'|} |z'| (M - |z'|)}$$

M is the maximum coalition size and $|z'|$ the number of present features in instance z' .

It is shown that linear regression with this kernel weight yields Shapley values!

KernelSHAP

- 4. Fit weighted linear model
 - We train the linear model g by optimizing the following loss function L :

$$L(\hat{f}, g, \pi_x) = \sum_{z' \in Z} [\hat{f}(h_x(z')) - g(z')]^2 \pi_x(z')$$

blackbox model Linear model

← Weight by SHAP kernel

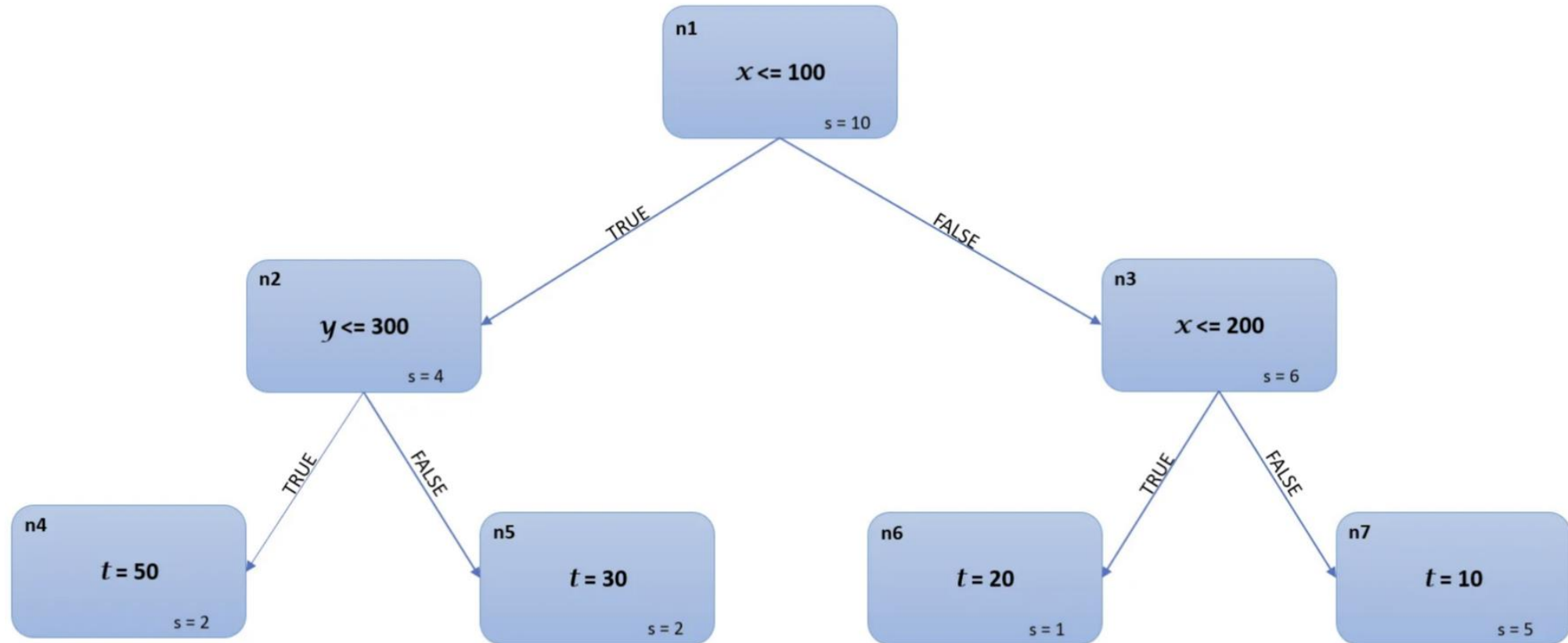
we can also add regularization terms to make the model sparse. (L1 norm)

- 5. Return the coefficients from the linear model.
 - The estimated coefficients of the model are the Shapely values.

TreeSHAP

- a variant of **SHAP for tree-based models** such as decision trees, random forests and gradient boosted trees.
- **Fast**, model-specific alternative to KernelSHAP
- Thanks to the Additivity property of Shapley values, the Shapley values of a tree ensemble is the average of the Shapley values of the individual trees.

- Example
 - 3 variables



Instance of interest : $[x=150, y=75, z=200]$

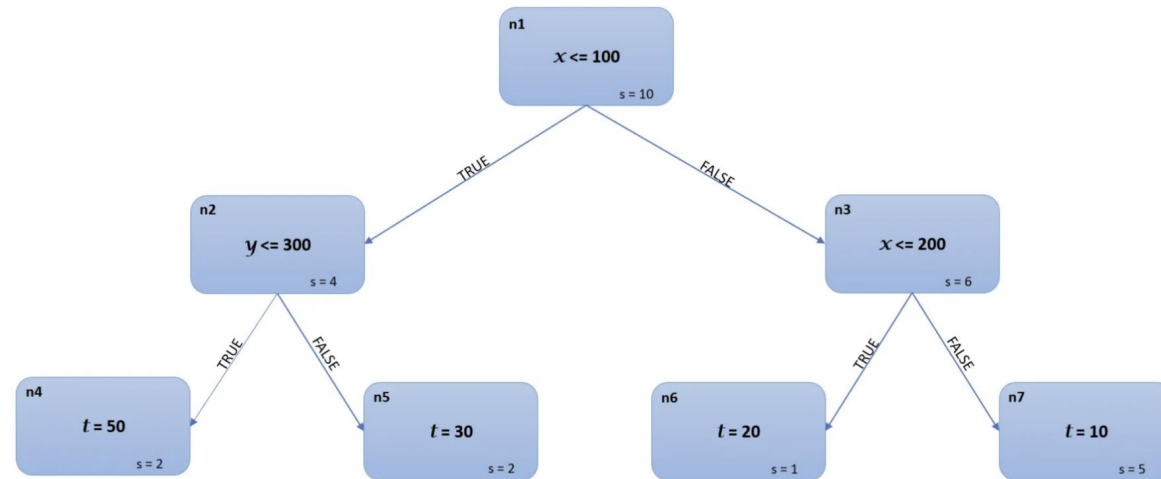
Prediction : $t = 20$

TreeSHAP

- Example

The prediction for the null model ϕ^0 = mean prediction for the training set
= $(50*2 + 30*2 + 20*1 + 10*5)/10 = 23$

– possible sequence $3!=6$



TreeSHAP

■ Example

- Consider the sequence: $x > y > z$:
 - x is added to the null model.

$$\phi^{x^1} = 20 - 23 = -3.$$

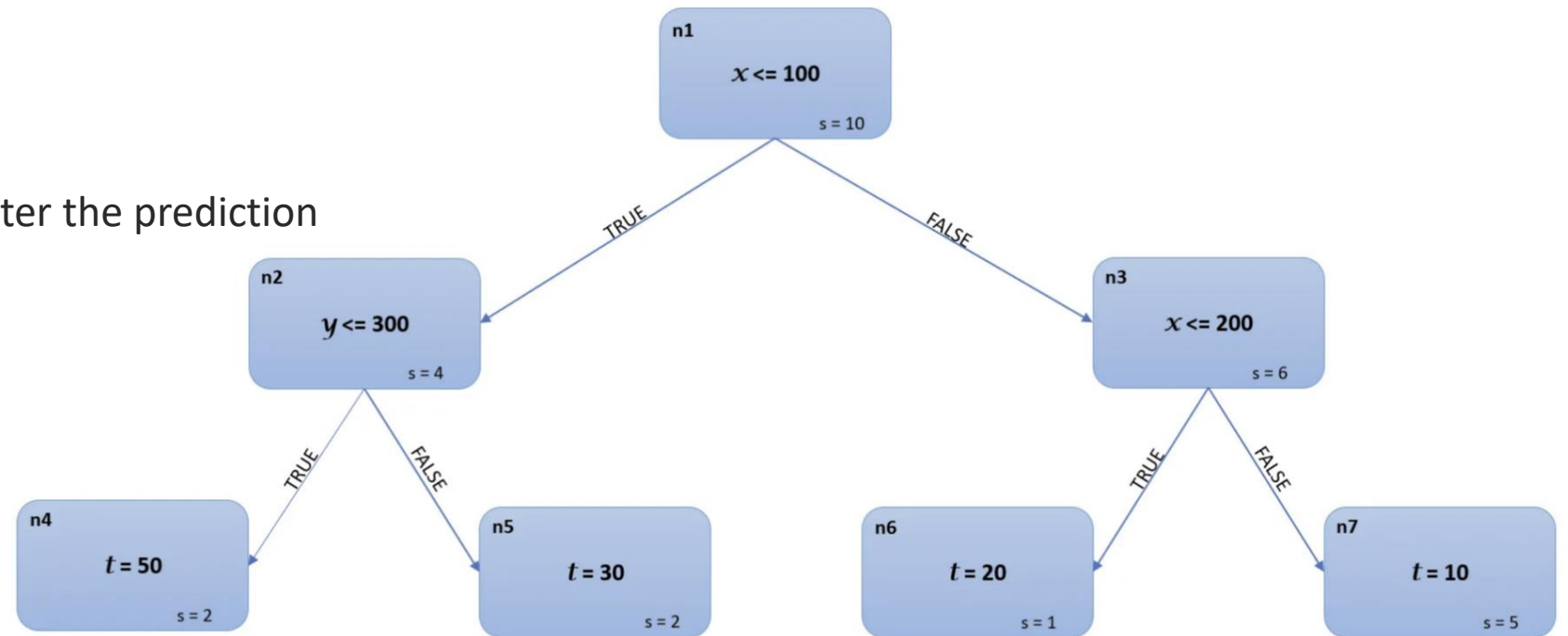
- y is added

Since adding y does not alter the prediction

$$\phi^{y^1} = 20 - 20 = 0$$

- z is added

$$\phi^{z^1} = 0$$



Instance of interest : $[x=150, y=75, z=200]$

TreeSHAP

■ Example

- Consider the sequence: $y > z > x$:
 - y is added to the null model.

$$\begin{aligned} & (4/10) * (\text{prediction from left child node } n2) + \\ & (6/10) * (\text{prediction from right child } n3) \\ & = (4/10) * 50 + (6/10) * \{(1/6) * 20 + (5/6) * 10\} \\ & = 27 \end{aligned}$$

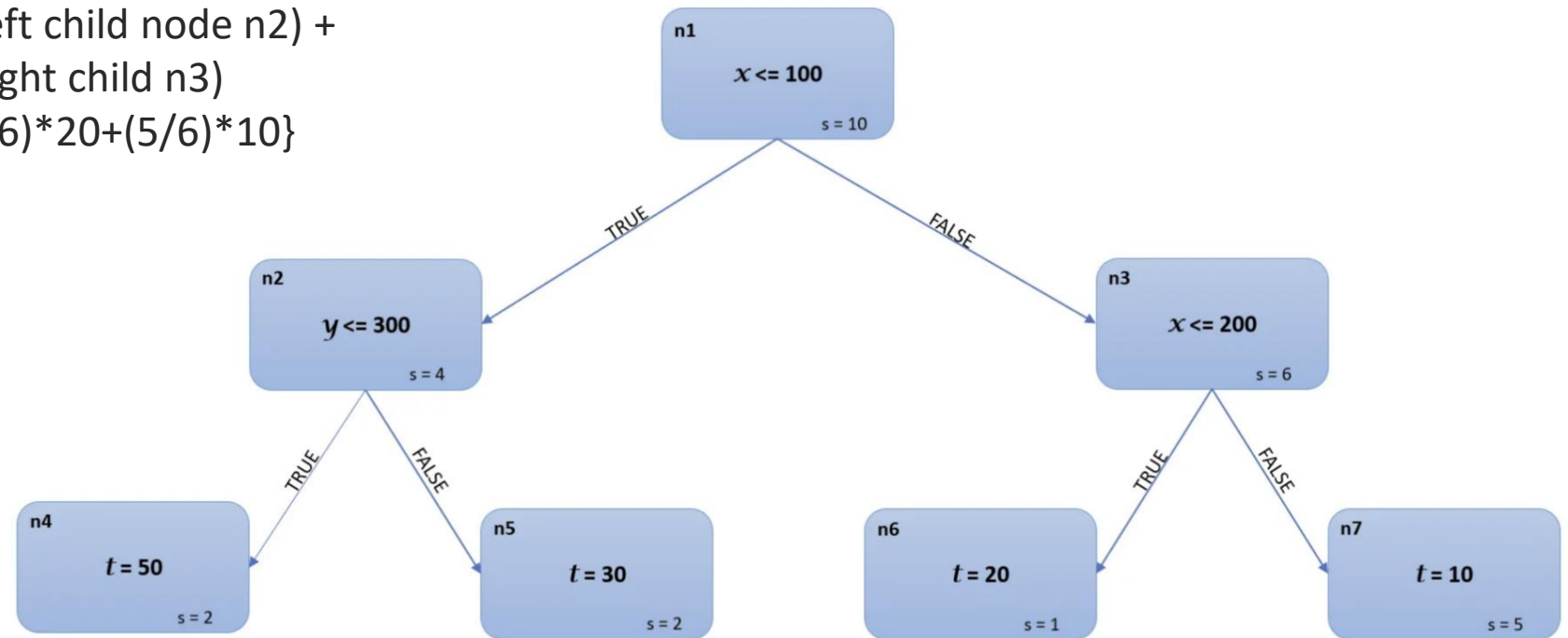
$$\phi^{y^2} = 27 - 23 = 4$$

- z is added

$$\phi^{z^2} = 0$$

- x is added

$$\phi^{x^2} = 20 - 27 = -7$$



Instance of interest : $[x=150, y=75, z=200]$

TreeSHAP

- Example

- Compute for all sequences...

Sequence $x > z > y$: $\phi^{x3} = -3, \phi^{y3} = 0, \phi^{z3} = 0$

Sequence $z > x > y$: $\phi^{x4} = -3, \phi^{y4} = 0, \phi^{z4} = 0$

Sequence $z > y > x$: $\phi^{x5} = -7, \phi^{y5} = 4, \phi^{z5} = 0$

Sequence $y > x > z$: $\phi^{x6} = -7, \phi^{y6} = 4, \phi^{z6} = 0$



Hence, SHAP values for the instance i are given by:

$$\phi^x = (\phi^{x1} + \phi^{x2} + \phi^{x3} + \phi^{x4} + \phi^{x5} + \phi^{x6})/6 = (-3-7-3-3-7-7)/6 = -5$$

$$\phi^y = (\phi^{y1} + \phi^{y2} + \phi^{y3} + \phi^{y4} + \phi^{y5} + \phi^{y6})/6 = (0+4+0+0+4+4)/6 = 2$$

$$\phi^z = (\phi^{z1} + \phi^{z2} + \phi^{z3} + \phi^{z4} + \phi^{z5} + \phi^{z6})/6 = (0+0+0+0+0+0)/6 = 0$$

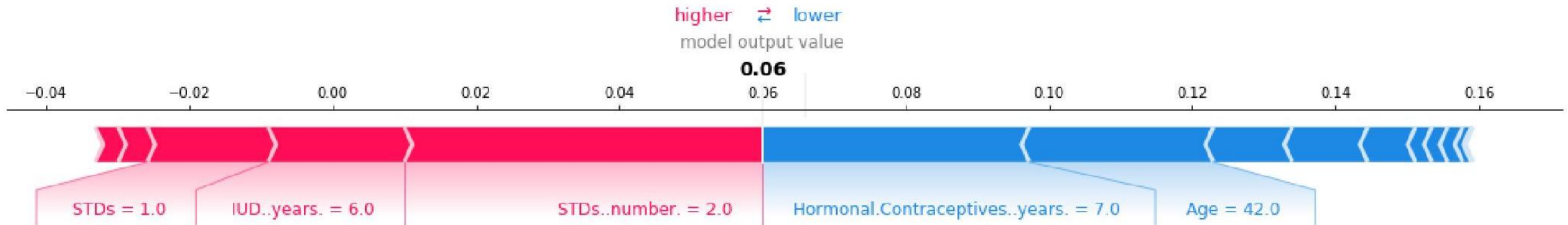
Explanation for the prediction for instance i

$$20 = \phi^0 + \phi^x + \phi^y + \phi^z = 23 + (-5) + 2 + 0$$

SHAP explanations

<https://github.com/slundberg/shap>

- Force plot
 - Risk for cervical cancer



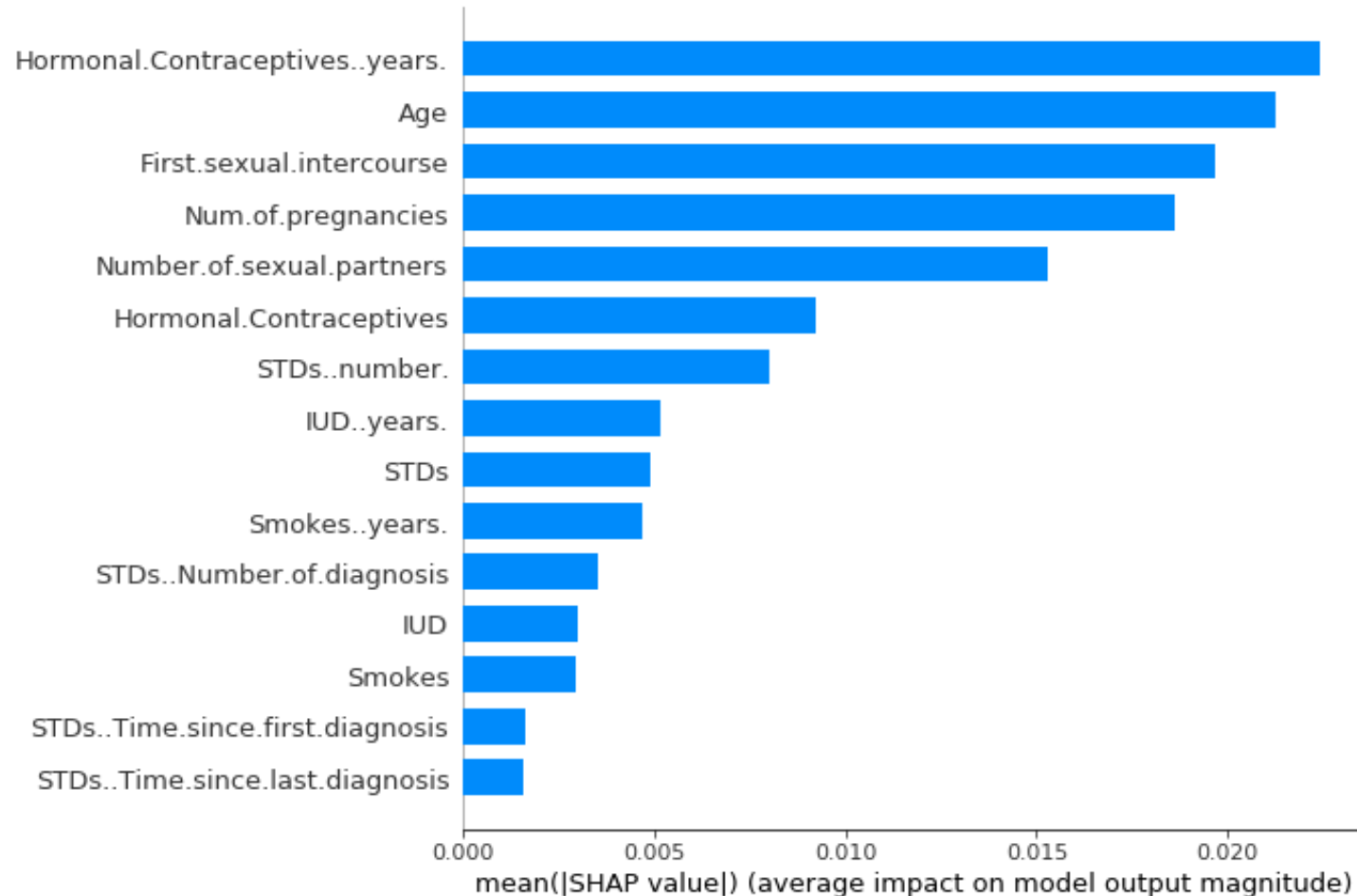
The baseline – the average predicted probability – is 0.066. The first woman has a low predicted risk of 0.06. Risk increasing effects such as STDs are offset by decreasing effects such as age.



The second woman has a high predicted risk of 0.71. Age of 51 and 34 years of smoking increase her predicted cancer risk.

SHAP explanations

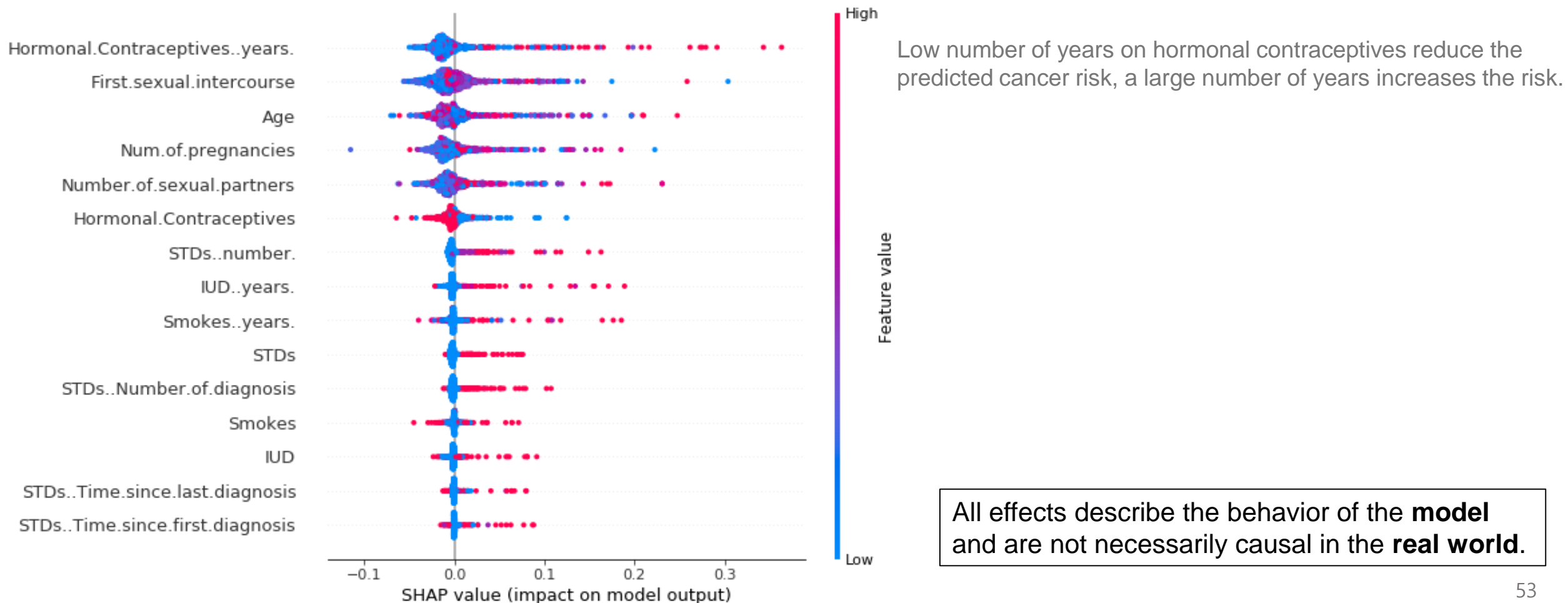
- SHAP Feature Importance (global explanations)
 - average the **absolute** Shapley values per feature across the data



SHAP explanations

■ SHAP Summary Plot

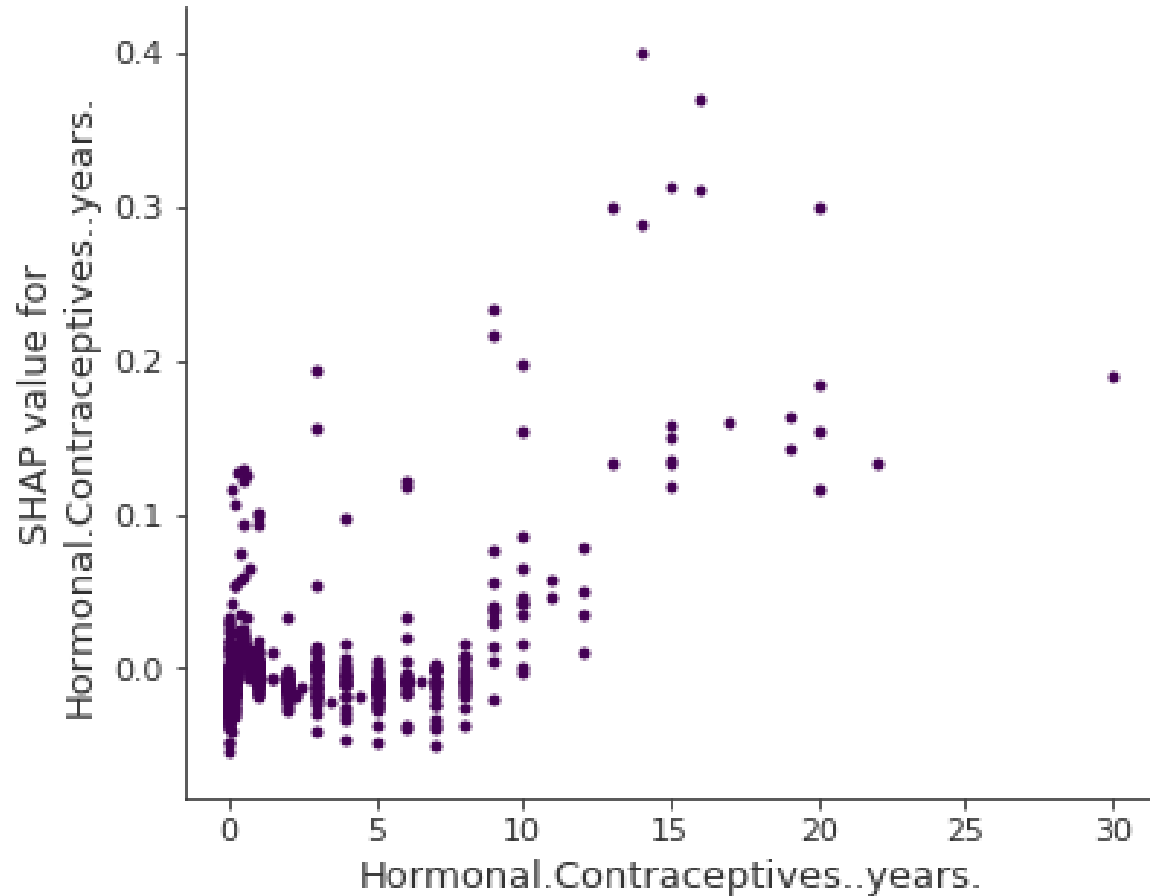
- Feature importance + feature effects
- Each point on the summary plot is a Shapley value for a feature and an instance



SHAP explanations

■ SHAP Dependence Plot

- For each feature, plot [SHAP value vs feature value]



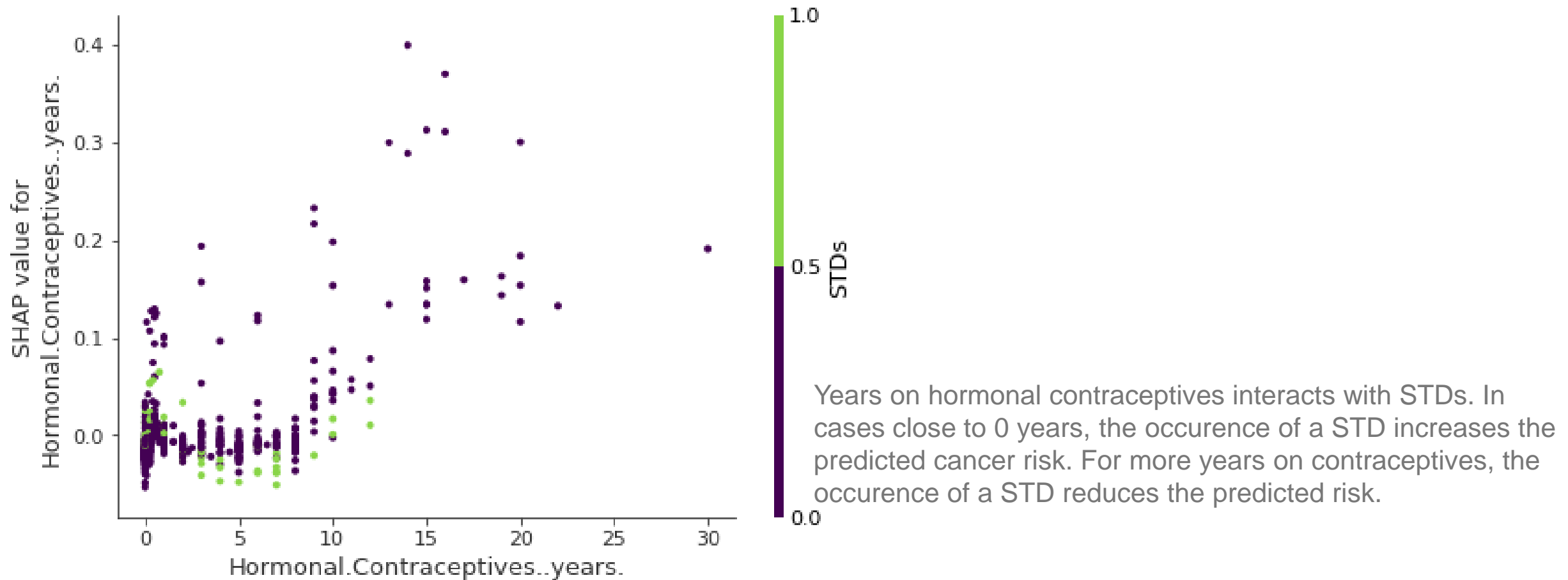
- alternative to partial dependence plots and accumulated local effects.
- While PDP and ALE plot show average effects, SHAP dependence also shows the variance on the y-axis

SHAP dependence plot for years on hormonal contraceptives. Compared to 0 years, a few years lower the predicted probability and a high number of years increases the predicted cancer probability.

SHAP explanations

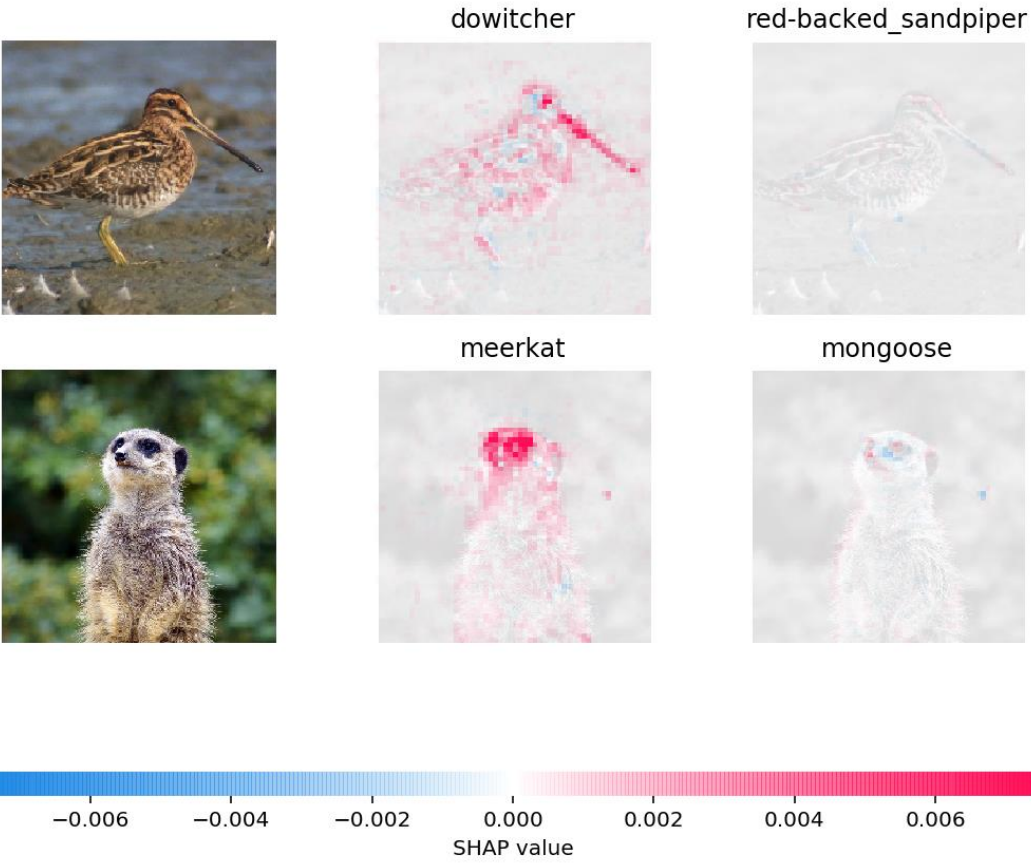
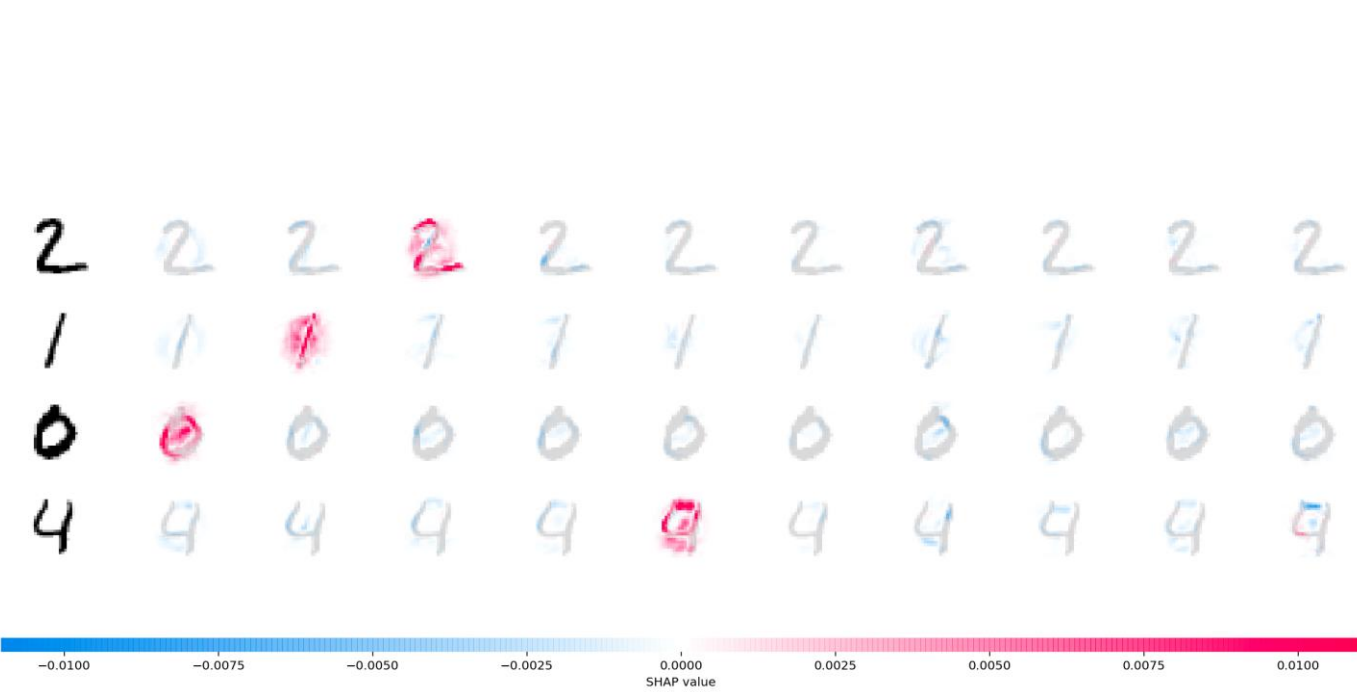
- SHAP Dependence Plot (with interaction visualization)

- For each feature, plot [SHAP value vs feature value]



SHAP explanations

- Others



SHAP

- Advantages

- The difference between the prediction and the average prediction is **fairly distributed** among the feature values of the instance – the **Efficiency property** of Shapley values.
 - LIME does not guarantee that the prediction is fairly distributed among the features.
- The Shapley value is the only explanation method with a **solid theory**.
 - LIME assume linear behavior of the machine learning model locally, but there is no theory as to why this should work.

- Disadvantages

- The Shapley value requires **a lot of computing time**.
 - only the approximate solution is feasible in most of real-world problems.
- Access to data is needed to compute them for new data
- KernelSHAP is slow.