An introduction to explainable artificial intelligence with LIME and SHAP

Aleix Nieto Juscafresa

Treball Final de Grau
Universitat de Barcelona
GRAU DE MATEMÀTIQUES

Dirigit per Dr. Albert Clapés i Dr. Sergio Escalera

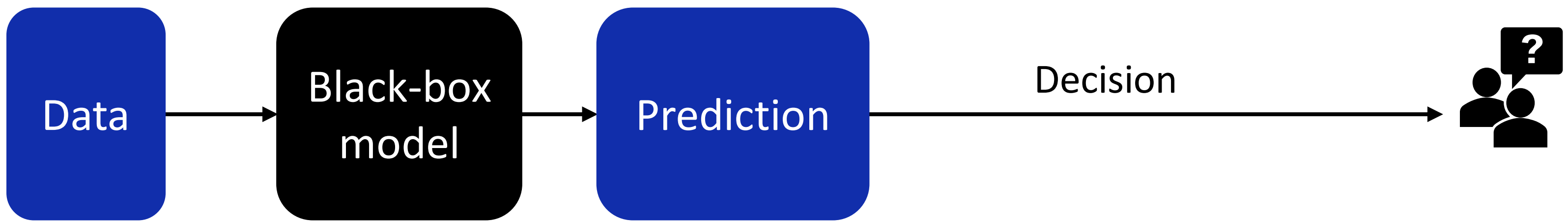
29 de juny de 2022

Universitat de Barcelona
Facultat de Matemàtiques i Informàtica

Contents

Introduction

Today



Data

Black-box model

Prediction

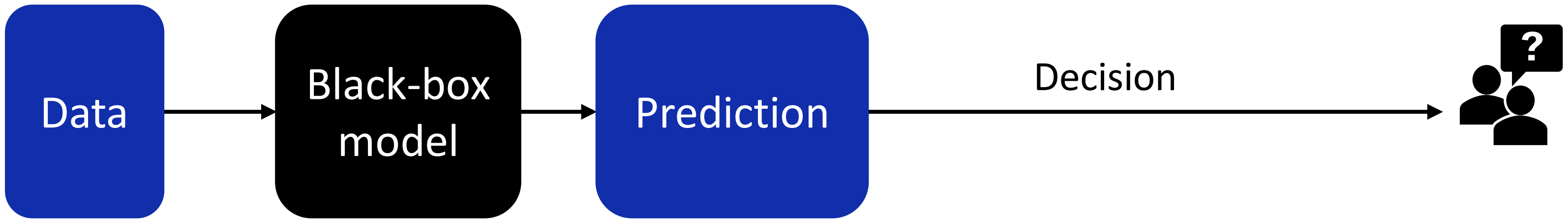
Decision

Confusion with Today's AI Black Box

- Why did you do that?
- Why did you not do that?
- When do you succed or fail?
- How do I correct an error?

Introduction

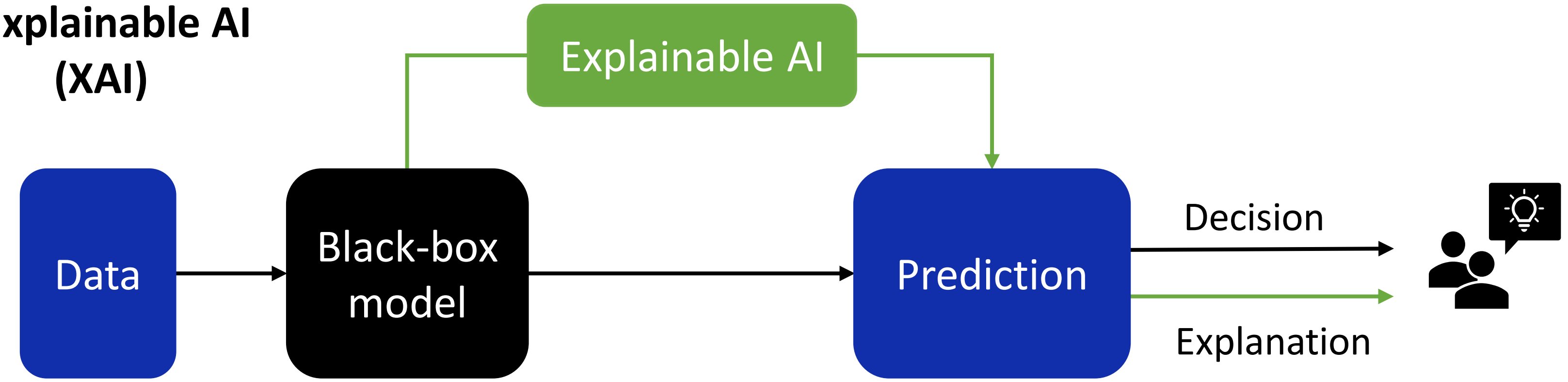
Today



Confusion with Today's AI Black Box

- Why did you do that?
- Why did you not do that?
- When do you succed or fail?
- How do I correct an error?

Explainable AI (XAI)



Clear & Transparent Decisions

- I understand why
- I understand why not
- I know why you suceed or fail
- I understand, I trust you more

Contents

Machine learning

Machine learning

Application of artificial intelligence dedicated to the creation of algorithms that allow systems to learn without human intervention.

SUPERVISED LEARNING

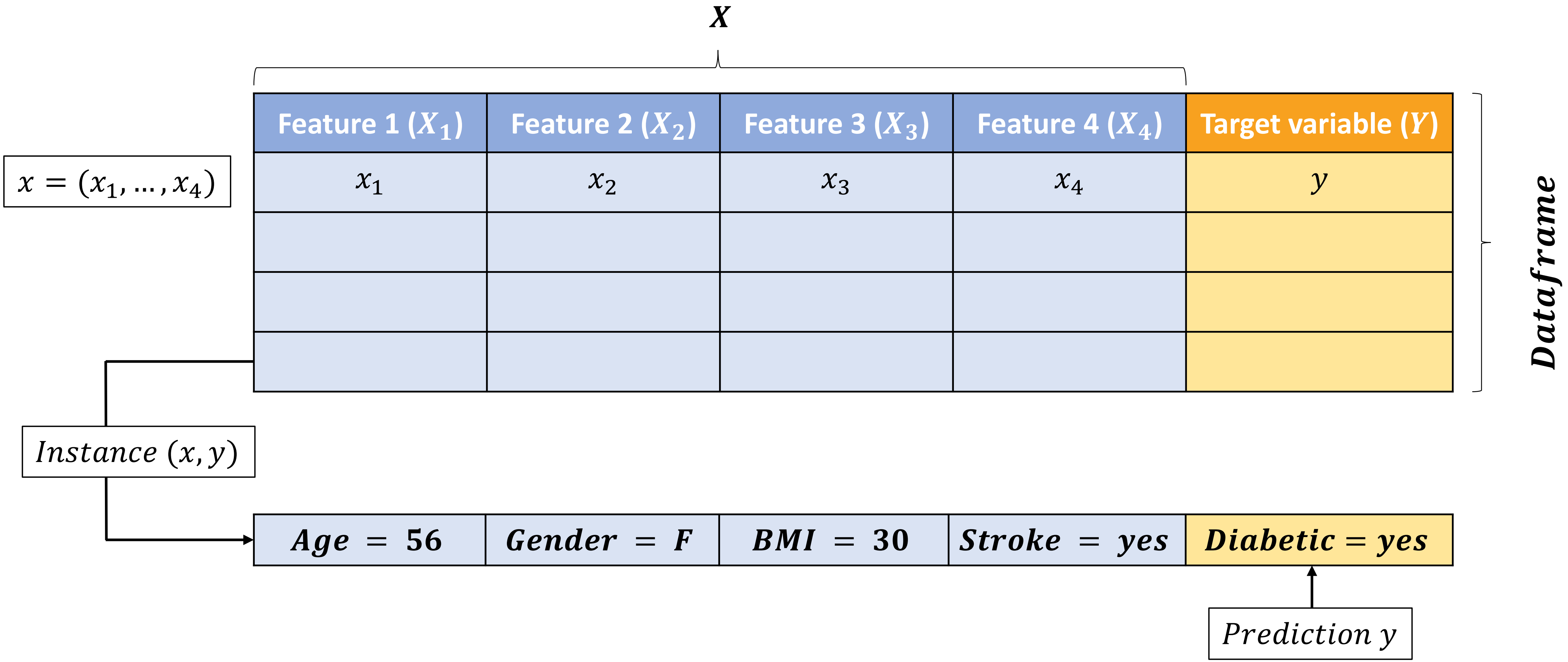
UNSUPERVISED LEARNING

SEMI-SUPERVISED LEARNING

REINFORCEMENT LEARNING

...

Terminology



$\boldsymbol{X}$

Feature 1 (X_1)	Feature 2 (X_2)	Feature 3 (X_3)	Feature 4 (X_4)	Target variable (Y)
x_1	x_2	x_3	x_4	y

Dataframe

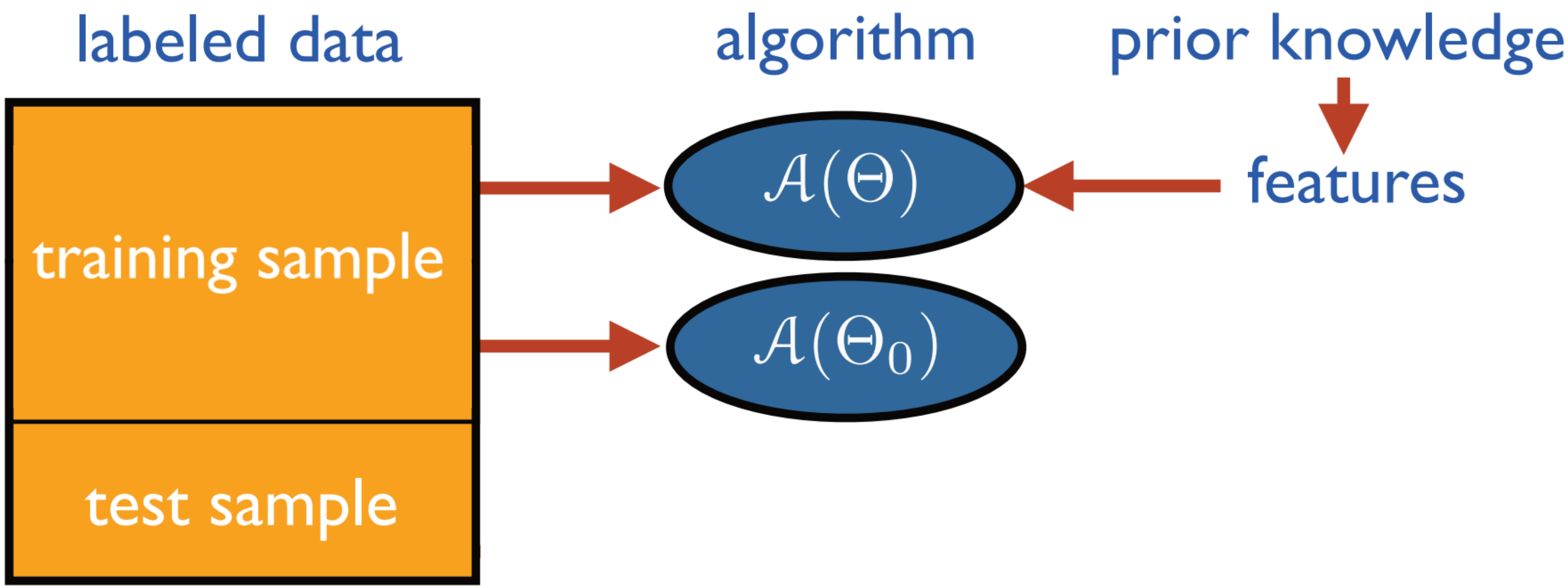
$x = (x_1, \dots, x_4)$

$Instance\ (x, y)$

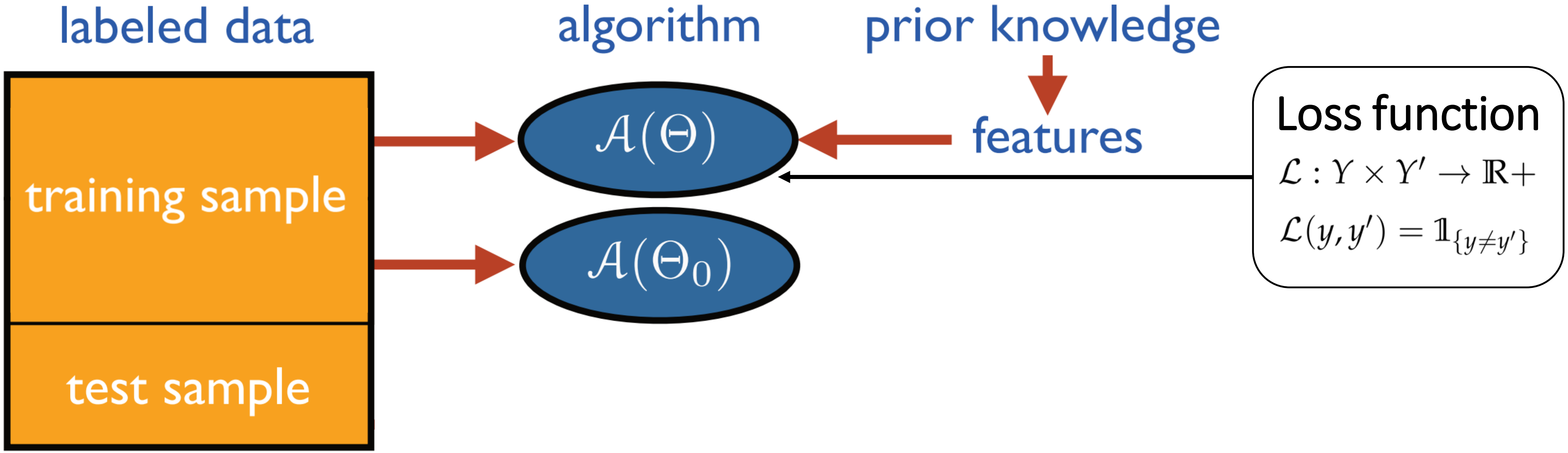
$\boldsymbol{Age\ =\ 56}$	$\boldsymbol{Gender\ =\ F}$	$\boldsymbol{BMI\ =\ 30}$	$\boldsymbol{Stroke\ =\ yes}$	$\boldsymbol{Diabetic = yes}$

$Prediction\ y$

Machine learning model learning stages



Machine learning model learning stages



labeled data

training sample

test sample

algorithm

$\mathcal{A}(\Theta)$

$\mathcal{A}(\Theta_0)$

prior knowledge

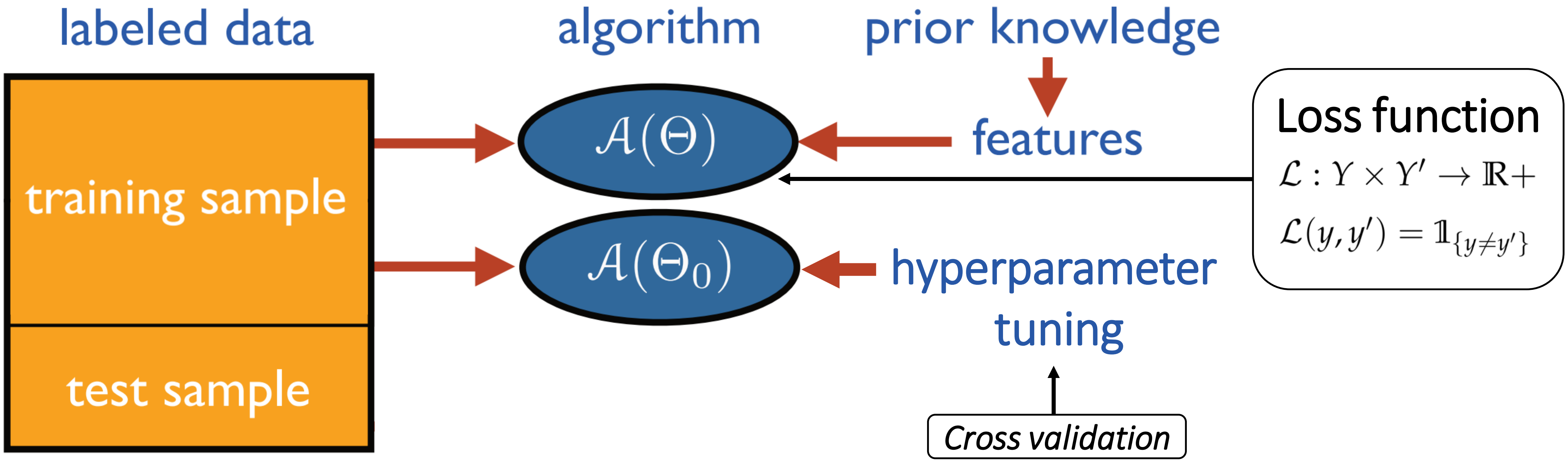
features

Loss function

$\mathcal{L} : Y \times Y' \to \mathbb{R}+$

$\mathcal{L}(y, y') = \mathbb{1}_{\{y \neq y'\}}$

Machine learning model learning stages



labeled data

training sample

test sample

algorithm

$\mathcal{A}(\Theta)$

$\mathcal{A}(\Theta_0)$

prior knowledge

features

hyperparameter tuning

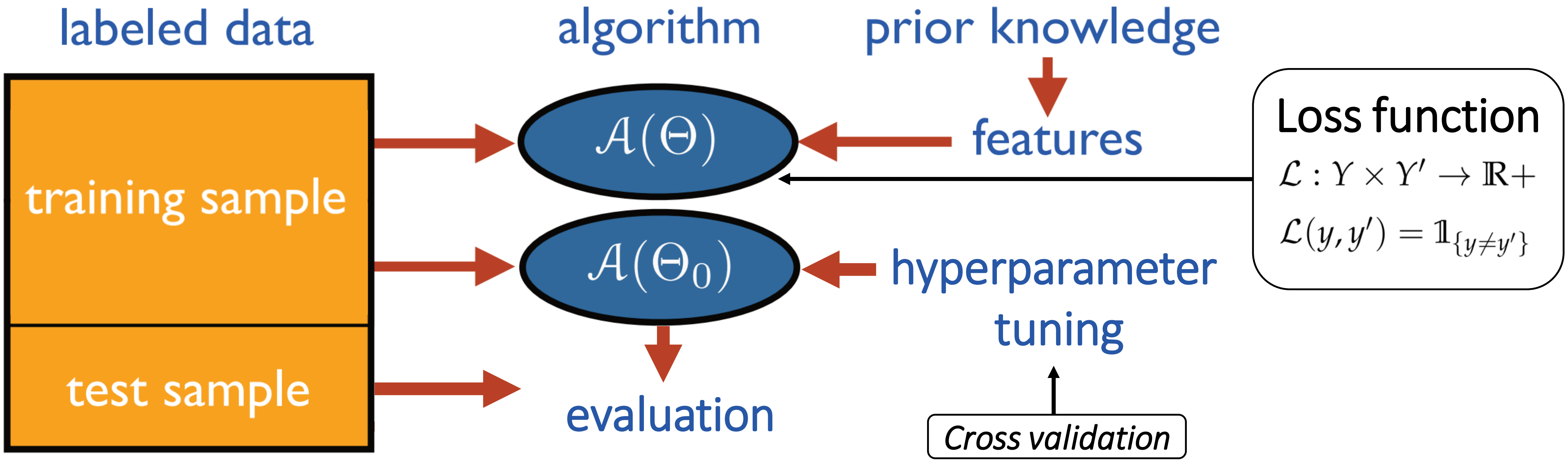
Cross validation

Loss function

$\mathcal{L} : Y \times Y' \to \mathbb{R}+$

$\mathcal{L}(y, y') = \mathbb{1}_{\{y \neq y'\}}$

Machine learning model learning stages



labeled data

training sample

test sample

algorithm

$\mathcal{A}(\Theta)$

$\mathcal{A}(\Theta_0)$

evaluation

prior knowledge

features

hyperparameter tuning

Cross validation

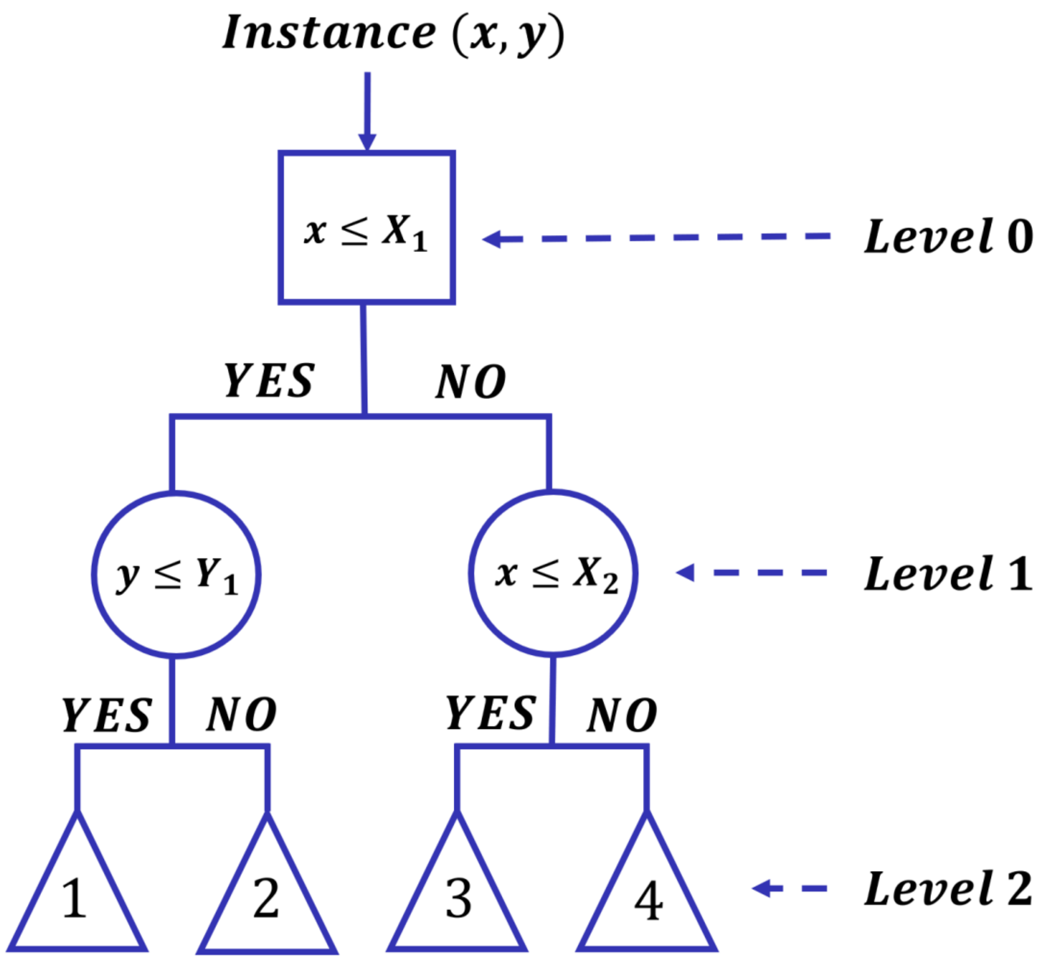
Loss function

$\mathcal{L} : Y \times Y' \to \mathbb{R}+$

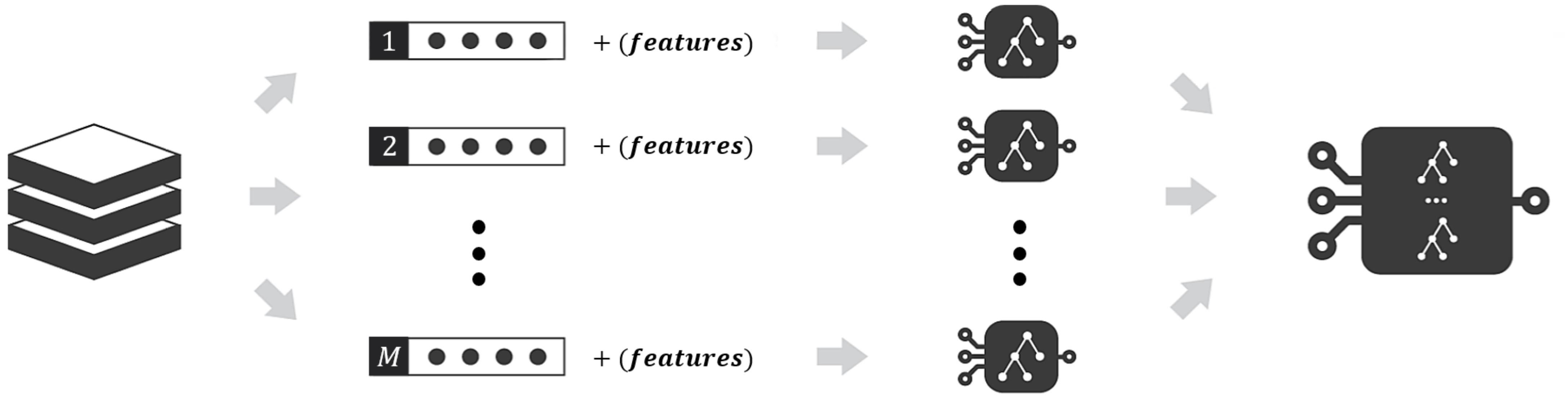
$\mathcal{L}(y, y') = \mathbb{1}_{\{y \neq y'\}}$

Contents

Decision tree



Random forest



Initial dataset

M bootstrap datasets

+

randomly selected features

Deep trees fitted on each bootstrap sample and considering only selected features

Random forest (ensemble model)

$$f(x) = \arg\max_{k\in\{1,\ldots,K\}} \left(\sum_{m=1}^{M} \mathbb{1}_{\{g_m(x)=k\}} \right)$$

Contents

Linear regression

$$\boxed{Y = f(X) + \epsilon = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \epsilon}$$

Linear regression

$$Y = f(X) + \epsilon = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \epsilon$$

Optimisation problem

$$\underset{\beta_0,\ldots,\beta_p}{\arg\min} \sum_{i=1}^{n} (y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}))^2 = \underset{\beta_0,\ldots,\beta_p}{\arg\min} \sum_{i=1}^{n} \epsilon_i^2$$

Linear regression

$$Y = f(X) + \epsilon = \beta_0 + \sum_{j=1}^{p} \beta_j X_j + \epsilon$$

Optimisation problem

$$\arg\min_{\beta_0,\dots,\beta_p} \sum_{i=1}^{n} (y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}))^2 = \arg\min_{\beta_0,\dots,\beta_p} \sum_{i=1}^{n} \epsilon_i^2$$

$$C = \mathbb{E}[\epsilon\epsilon^T] = \sigma^2 I$$

$$C = \begin{pmatrix} \sigma^2 & 0 & \dots & 0 \\ 0 & \sigma^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma^2 \end{pmatrix}$$

Model assumptions

- *Linearity*
- *Normality*
- *Independence*
- *Absence of multicollinearity*
- *Fixed features*
- *Homoscedasticity*

Weighted linear regression

$$Y = f(X) + \epsilon^* = \beta_0 + \sum_{j=1}^{p} X_j\beta_j + \epsilon^*$$

Weighted linear regression

$$Y = f(X) + \epsilon^* = \beta_0 + \sum_{j=1}^{p} X_j \beta_j + \epsilon^*$$

Optimisation problem

$$\arg\min_{\beta_0,\ldots,\beta_p} \sum_{i=1}^{n} \boldsymbol{w_i} \left(y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij})\right)^2 = \arg\min_{\beta_0,\ldots,\beta_p} \sum_{i=1}^{n} {\epsilon_i^*}^2$$

Weighted linear regression

$$Y = f(X) + \epsilon^* = \beta_0 + \sum_{j=1}^{p} X_j\beta_j + \epsilon^*$$

Optimisation problem

$$\underset{\beta_0,\dots,\beta_p}{\arg\min} \sum_{i=1}^{n} \boldsymbol{w_i}\,(y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}))^2 = \underset{\beta_0,\dots,\beta_p}{\arg\min} \sum_{i=1}^{n} {\epsilon_i^*}^2$$

$$C = \begin{pmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_n^2 \end{pmatrix}$$

$$W = \begin{pmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & w_n \end{pmatrix}$$

$$w_i = {}^{1}\!/\sigma_i^2$$

Violation of homoscedasticity

Weighted linear regression uses different weights for each observation based on their variance. A small error variance observation has a large weight since it includes more information than a large error variance observation, which has a small weight.

L1 and L2 regularisation

$$\text{RSS}(\beta) = \sum_{i=1}^{n}(y_i - (\beta_0 + \sum_{j=1}^{p} \beta_j x_{ij}))^2$$

Lasso regression (L1 regularisation)

$$\hat{\beta}_{lasso} = \underset{(\beta_0,\dots,\beta_p)\in\mathbb{R}^n}{\arg\min} \left\{ \text{RSS}(\beta) + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{\text{Penalty term}} \right\}$$

It tends far more to drive small weights to 0.

Ridge regression (L2 regularisation)

$$\hat{\beta}_{ridge} = \underset{(\beta_0,\dots,\beta_p)}{\arg\min} \left\{ \text{RSS}(\beta) + \underbrace{\lambda \sum_{j=1}^{p} \beta_j^2}_{\text{Penalty term}} \right\}$$

It pushes down big weights than tiny ones.

Contents

What is explainable AI?

Explainable artificial intelligence

Set of techniques that either produce more understandable models keeping high levels of performance or provide external tools to better understand the models that are inherently not interpretable.

XAI taxonomy

Agnosticity

Model-**agnostic**	Applicable to all model types
Model-**specific**	Only applicable to a specific model type

Scope

Global explanation	Explaining the whole model
Local explanation	Explaining individual predictions

Data Type

- Graph
- Image
- Text
- Tabular

Explanation Type

Visual

Data visualisation techniques may be used to understand the prediction or choice made over the input data.

Feature importance

After all possible combinations, we get the feature importance based on its average predicted marginal contribution to the model's decision.

Data points

This category includes all methods that return data points (already existent or newly created) to make a model interpretable.

Surrogate models

We can explain our complex model's prediction by using a simplified model (surrogate model) to approximate it around the prediction.

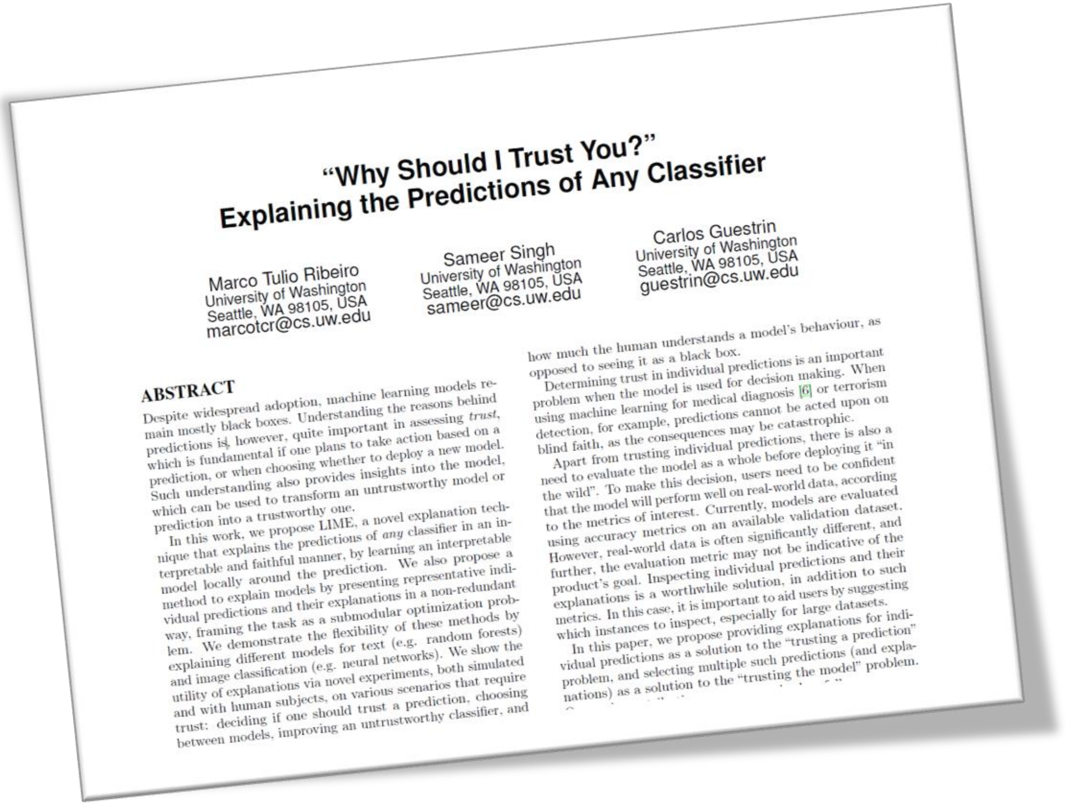
LIME

Local

Interpretable

Model-agnostic

Explanations



"Why Should I Trust You?"
Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

ABSTRACT

Despite widespread adoption, machine learning models remain mostly black boxes. Understanding the reasons behind predictions is, however, quite important in assessing *trust*, which is fundamental if one plans to take action based on a prediction, or when choosing whether to deploy a new model. Such understanding also provides insights into the model, which can be used to transform an untrustworthy model or prediction into a trustworthy one.

In this work, we propose LIME, a novel explanation technique that explains the predictions of *any* classifier in an interpretable and faithful manner, by learning an interpretable model locally around the prediction. We also propose a method to explain models by presenting representative individual predictions and their explanations in a non-redundant way, framing the task as a submodular optimization problem. We demonstrate the flexibility of these methods by explaining different models for text (e.g. random forests) and image classification (e.g. neural networks). We show the utility of explanations via novel experiments, both simulated and with human subjects, on various scenarios that require trust: deciding if one should trust a prediction, choosing between models, improving an untrustworthy classifier, and

how much the human understands a model's behaviour, as opposed to seeing it as a black box.

Determining trust in individual predictions is an important problem when the model is used for decision making. When using machine learning for medical diagnosis [6] or terrorism detection, for example, predictions cannot be acted upon on blind faith, as the consequences may be catastrophic.

Apart from trusting individual predictions, there is also a need to evaluate the model as a whole before deploying it "in the wild". To make this decision, users need to be confident that the model will perform well on real-world data, according to the metrics of interest. Currently, models are evaluated using accuracy metrics on an available validation dataset. However, real-world data is often significantly different, and further, the evaluation metric may not be indicative of the product's goal. Inspecting individual predictions and their explanations is a worthwhile solution, in addition to such metrics. In this case, it is important to aid users by suggesting which instances to inspect, especially for large datasets.

In this paper, we propose providing explanations for individual predictions as a solution to the "trusting a prediction" problem, and selecting multiple such predictions (and explanations) as a solution to the "trusting the model" problem.

LIME optimisation problem

$$\xi(x) = \underset{g \in G}{\arg\min}\, \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

LIME optimisation problem

$$\xi(x) = \underset{g\in G}{\arg\min}\, \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

$x \in \mathbb{R}^{d}$ → number of features

Age = 56	***Gender = F***	***BMI = 30***	***Stroke = yes***

LIME optimisation problem

$$\xi(x) = \underset{g\in G}{\arg\min}\,\mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Complex model

$f : \mathbb{R}^d \to \mathbb{R}$

Simple interpretable model

$x \in \mathbb{R}^d$ → number of features

Age = 56	***Gender = F***	***BMI = 30***	***Stroke = yes***

LIME optimisation problem

$$\xi(x) = \underset{g\in G}{\arg\min}\, \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Family of interpretable models

Complex model

$f : \mathbb{R}^d \rightarrow \mathbb{R}$

Simple interpretable model

$x \in \mathbb{R}^d \longrightarrow$ number of features

$Age = 56$	$Gender = F$	$BMI = 30$	$Stroke = yes$

LIME optimisation problem

$$\xi(x) = \underset{g\in G}{\arg\min}\, \mathcal{L}(f, g, \pi_x) + \Omega(g)$$

Family of interpretable models

Complex model

$f : \mathbb{R}^d \rightarrow \mathbb{R}$

Simple interpretable model

Neighbourhood of x

$x \in \mathbb{R}^d \longrightarrow$ number of features

Age = 56	*Gender* = *F*	*BMI* = 30	*Stroke* = *yes*

LIME optimisation problem

$$\xi(x) = \underset{g\in G}{\arg\min}\ \overset{(1)}{\boxed{\mathcal{L}(f, g, \pi_x)}} + \overset{(2)}{\boxed{\Omega(g)}}$$

Family of interpretable models

Complex model

$f : \mathbb{R}^d \rightarrow \mathbb{R}$

Simple interpretable model

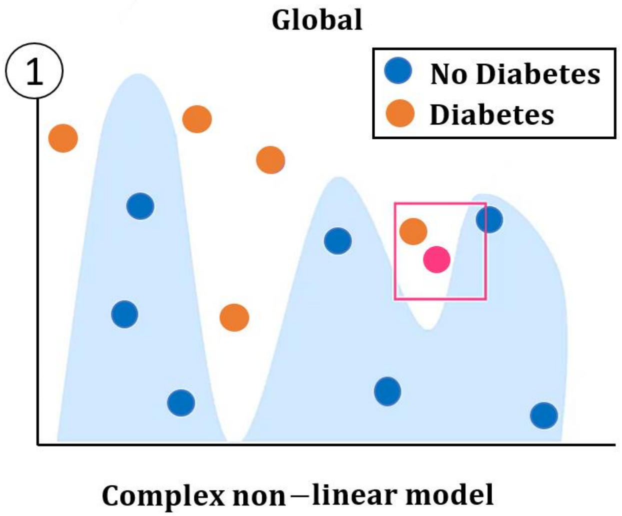
Neighbourhood of x

$x \in \mathbb{R}^d$ → number of features

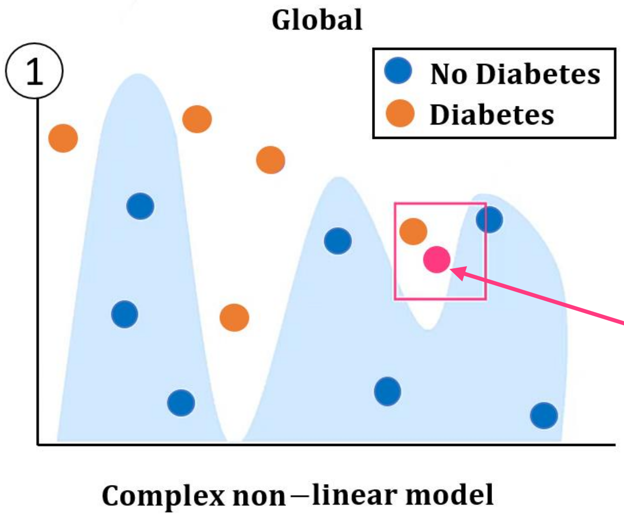
Age = 56	*Gender* = *F*	*BMI* = 30	*Stroke* = *yes*

2 LOSS TERMS

LIME step by step

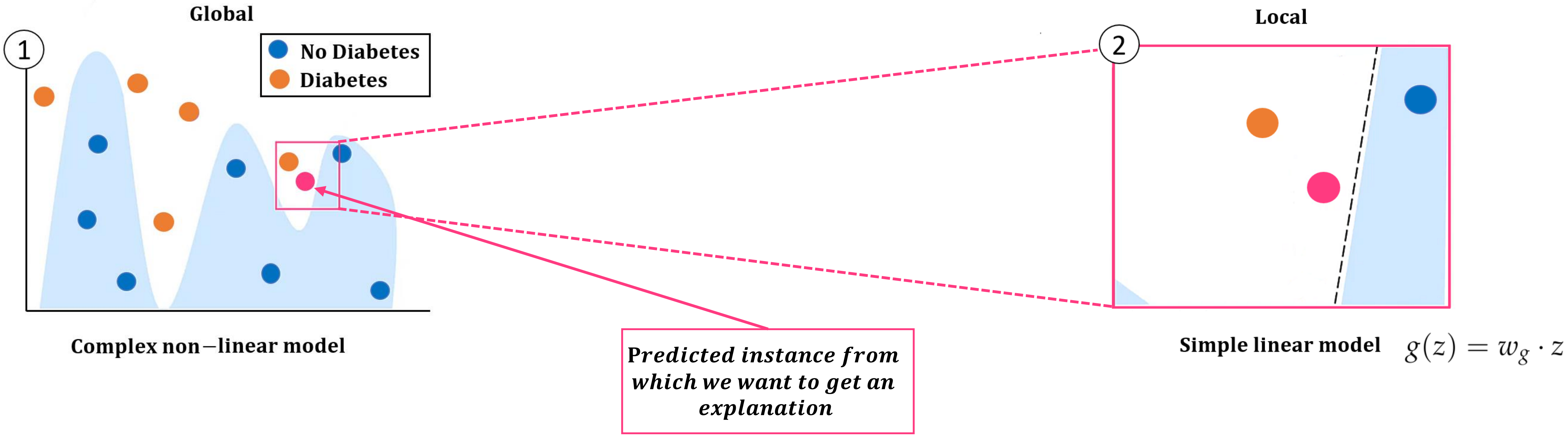


LIME step by step

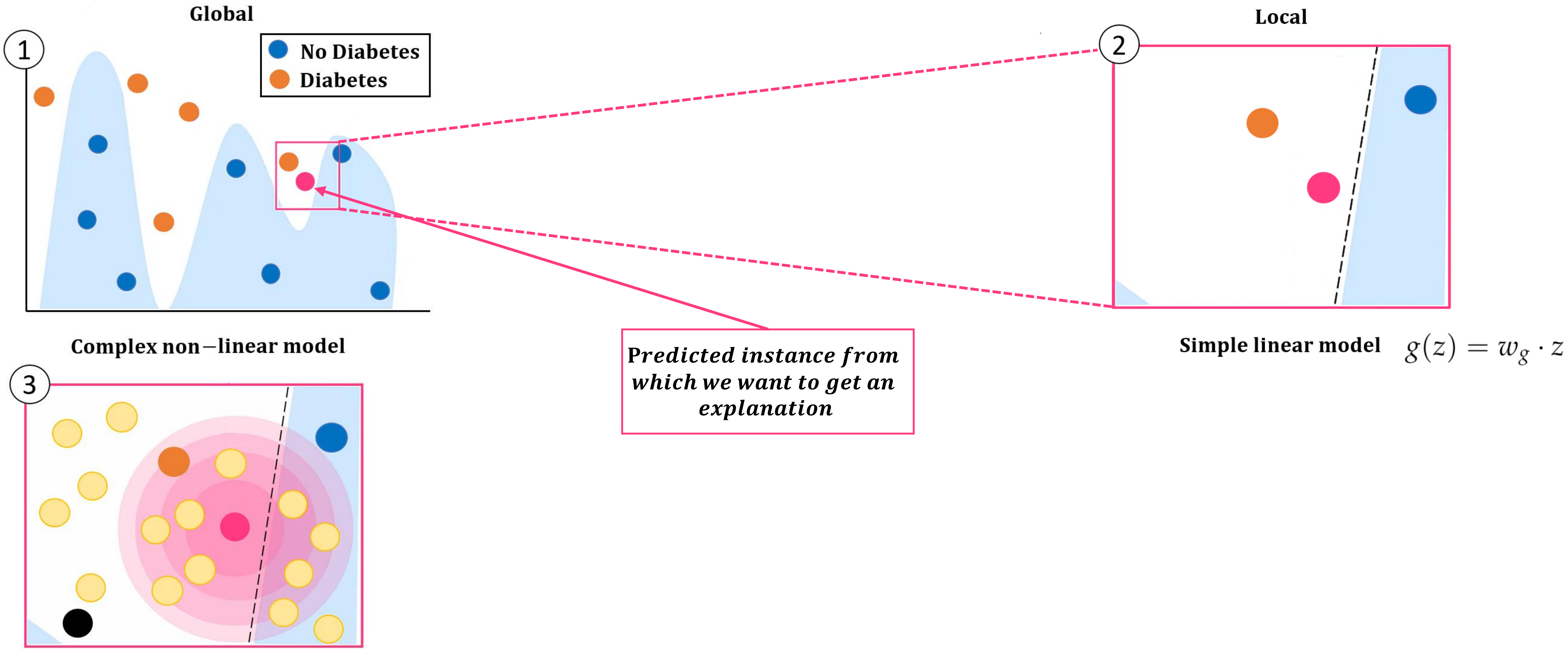


Predicted instance from which we want to get an explanation

LIME step by step



LIME step by step



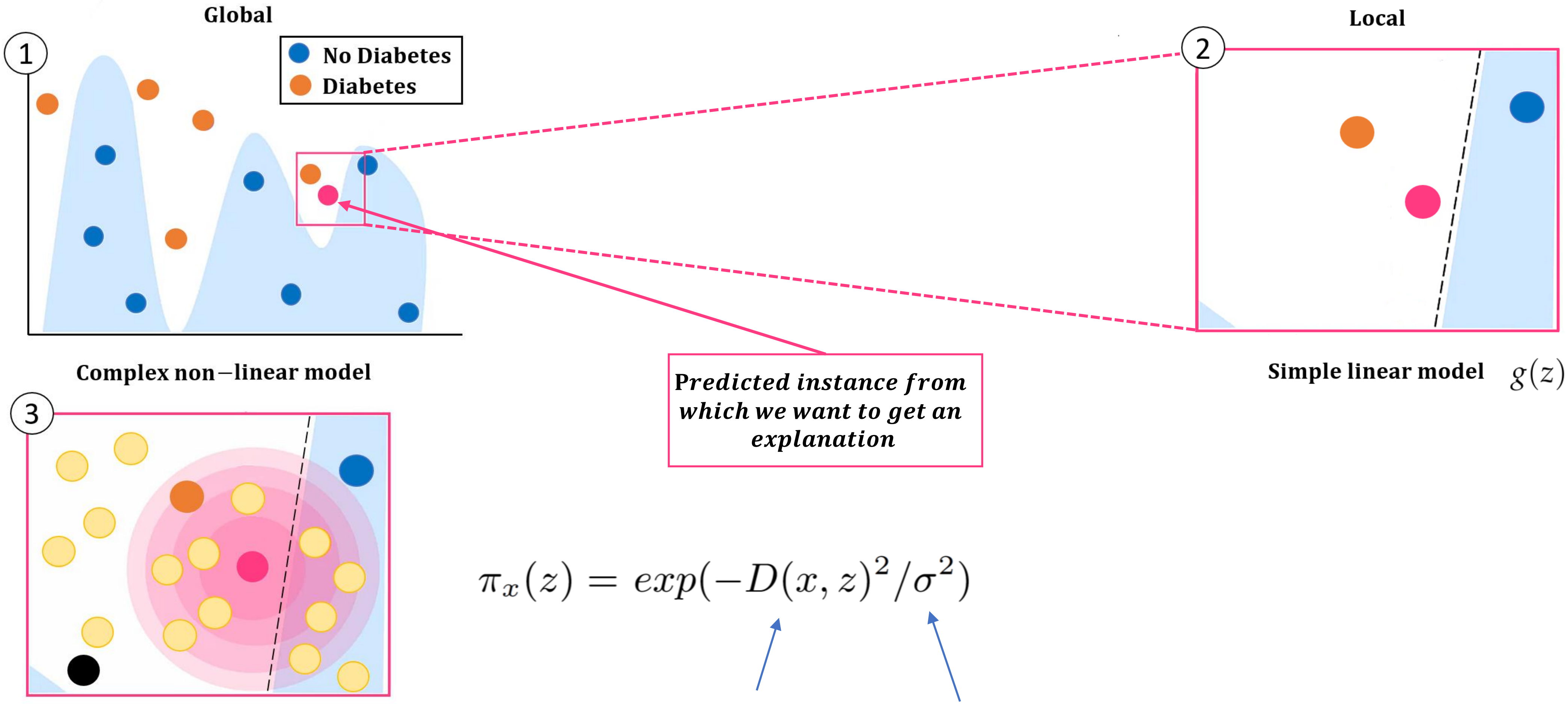
Complex non−linear model

Simple linear model $g(z) = w_g \cdot z$

Predicted instance from which we want to get an explanation

Perturbed data points, weighted according to the distance to our predicted instance

LIME step by step



Complex non−linear model

Simple linear model $g(z) = w_g \cdot z$

Predicted instance from which we want to get an explanation

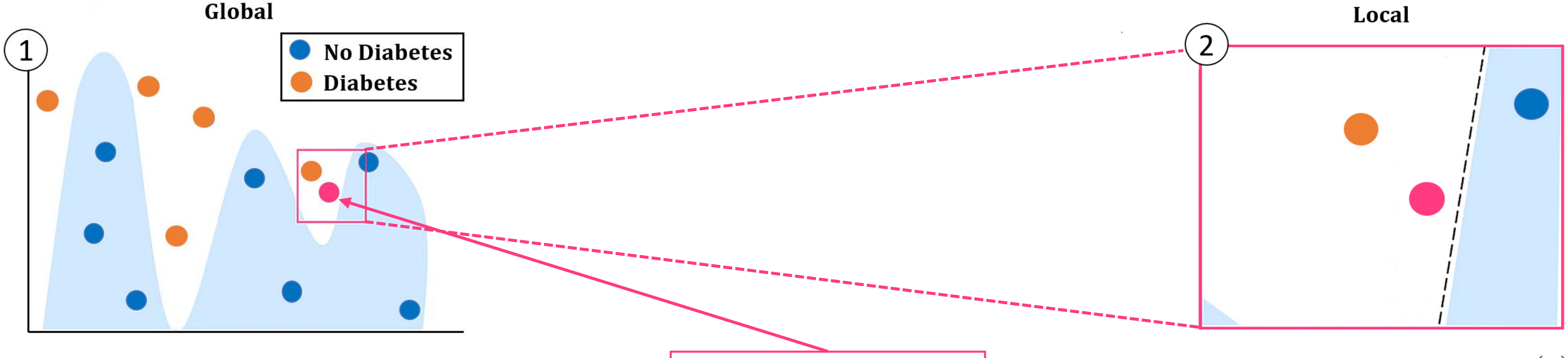
Perturbed data points, weighted according to the distance to our predicted instance

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

Some distance function D

Kernel width

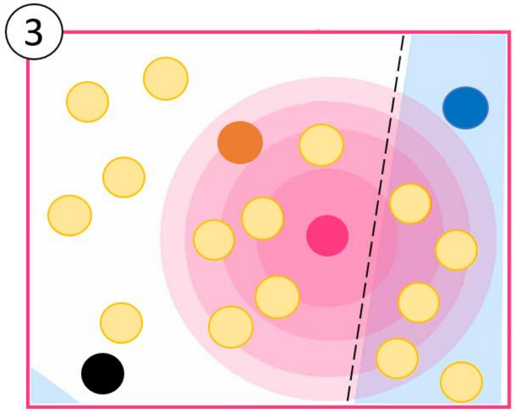
LIME step by step



Complex non−linear model

Simple linear model $g(z) = w_g \cdot z$

Predicted instance from which we want to get an explanation

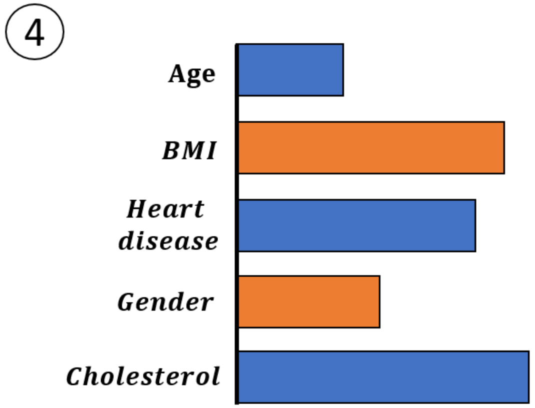


Perturbed data points, weighted according to the distance to our predicted instance

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

Some distance function D

Kernel width



Predicted instance relevant feature values contribution

Loss terms

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \; \boxed{\mathcal{L}(f, g, \pi_x)} + \boxed{\Omega(g)}$$

Train a weighted, interpretable model on the dataset with the perturbed instances

(1)

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z) \right)^2$$

Complex model prediction

Simple model prediction

Loss terms

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

Train a weighted, interpretable model on the dataset with the perturbed instances

(1) $$\mathcal{L}(f, g, \pi_x) = \sum_{z \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z) \right)^2$$

Complex model prediction

Simple model prediction

(2) $$\Omega(g)$$ **?**

Loss terms

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

Train a weighted, interpretable model on the dataset with the perturbed instances

(1)

$$\mathcal{L}(f, g, \pi_x) = \sum_{z \in \mathcal{Z}} \pi_x(z) \left(f(z) - g(z)\right)^2$$

Complex model prediction

Simple model prediction

(2) $\Omega(g)$ *LIME uses sparse linear models (K - LASSO)*

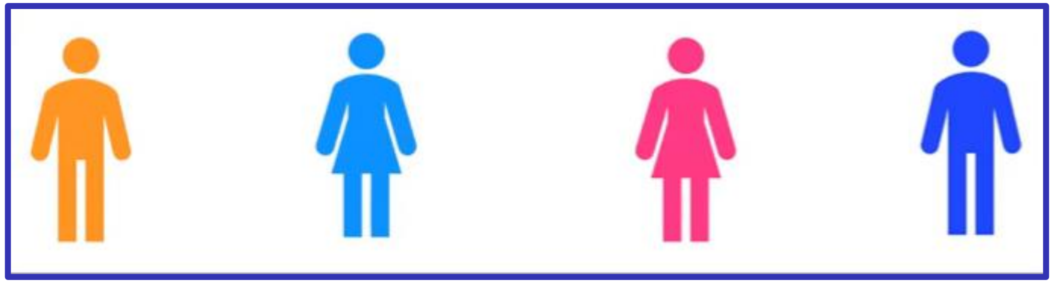
$$\hat{\beta}_{lasso} = \underset{(\beta_0,\ldots,\beta_p) \in \mathbb{R}^n}{\arg\min} \left\{ \text{RSS}(\beta) + \underbrace{\lambda \sum_{j=1}^{p} |\beta_j|}_{\text{Penalty term}} \right\}$$

LIME limitations

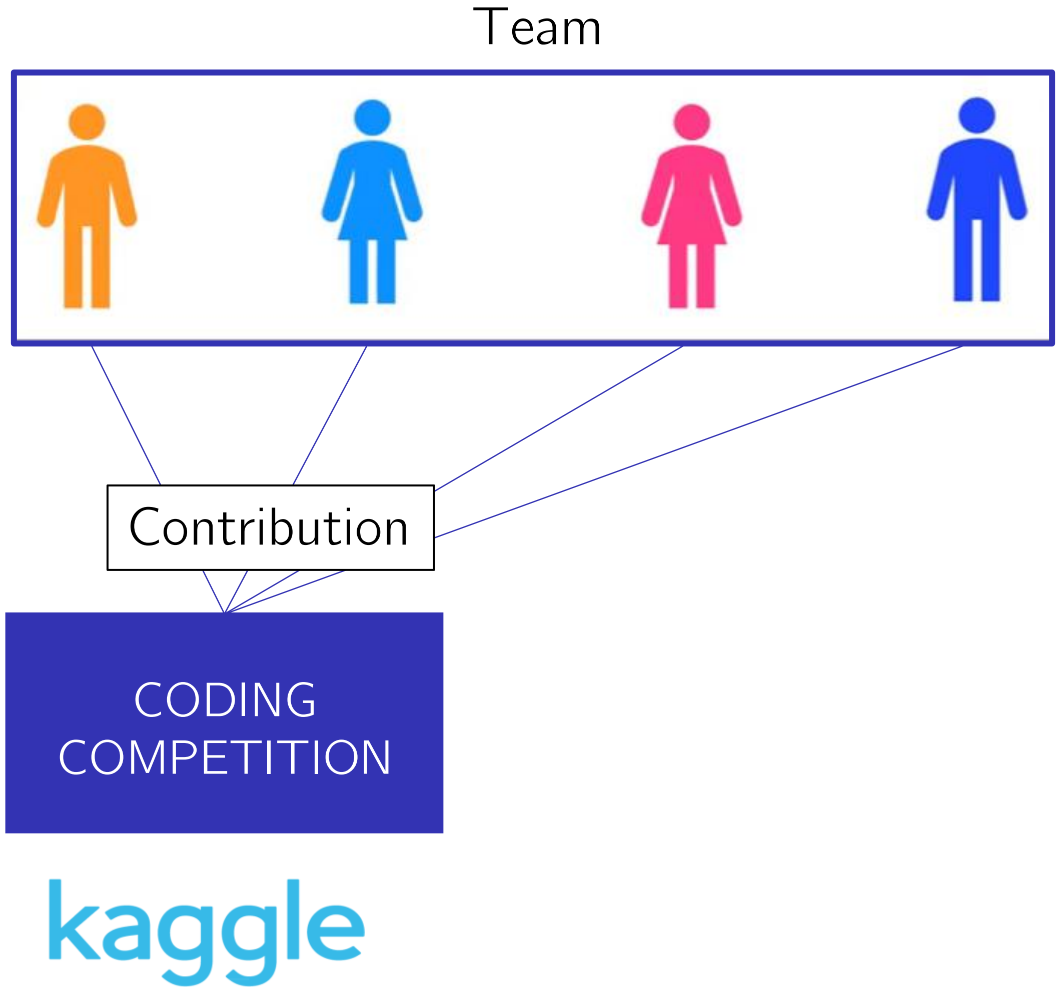
- Neighbourhood
- Non-linearity
- Improbable instances
- Instability

Shapley values

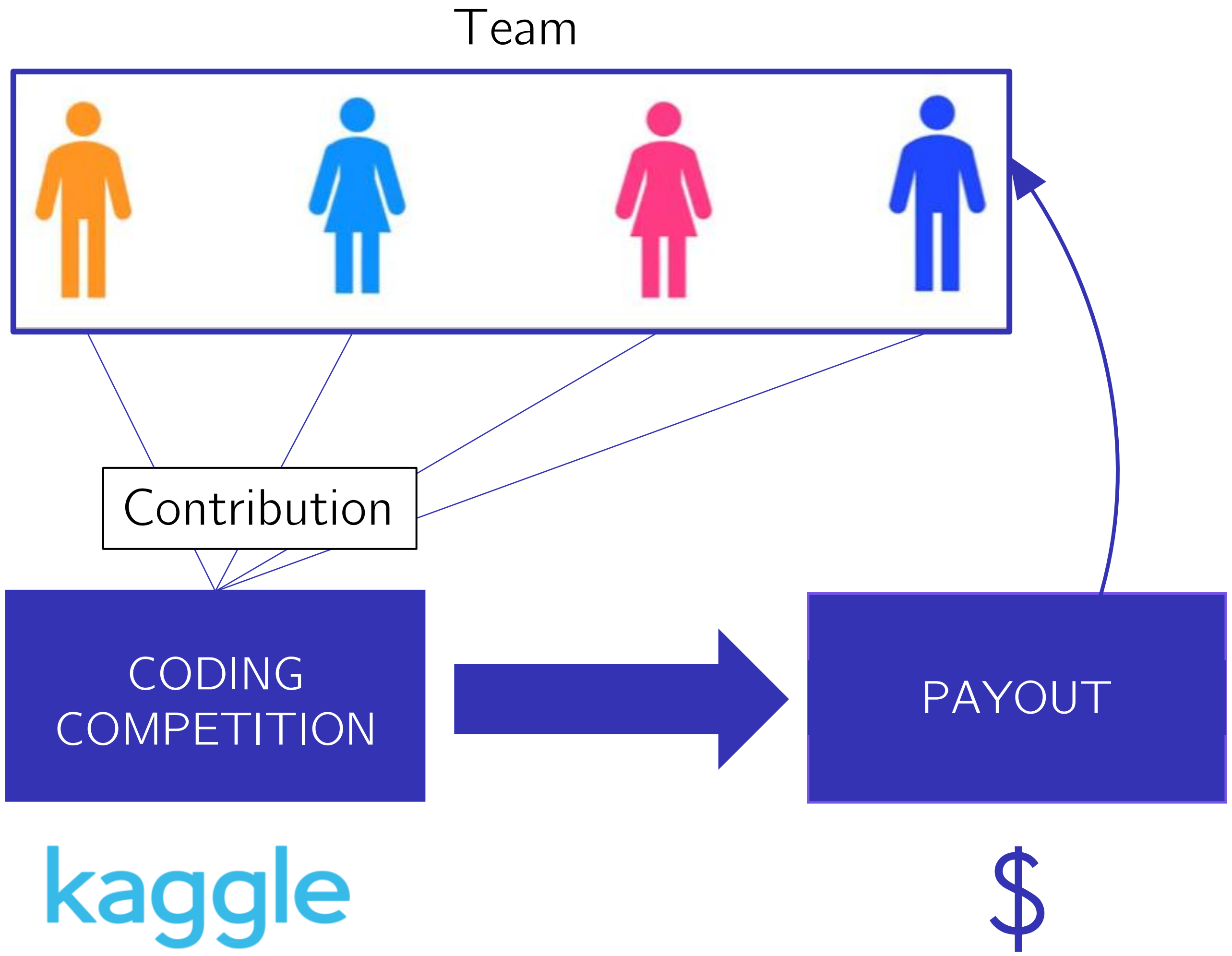
Team



Shapley values



Shapley values



Team

Contribution

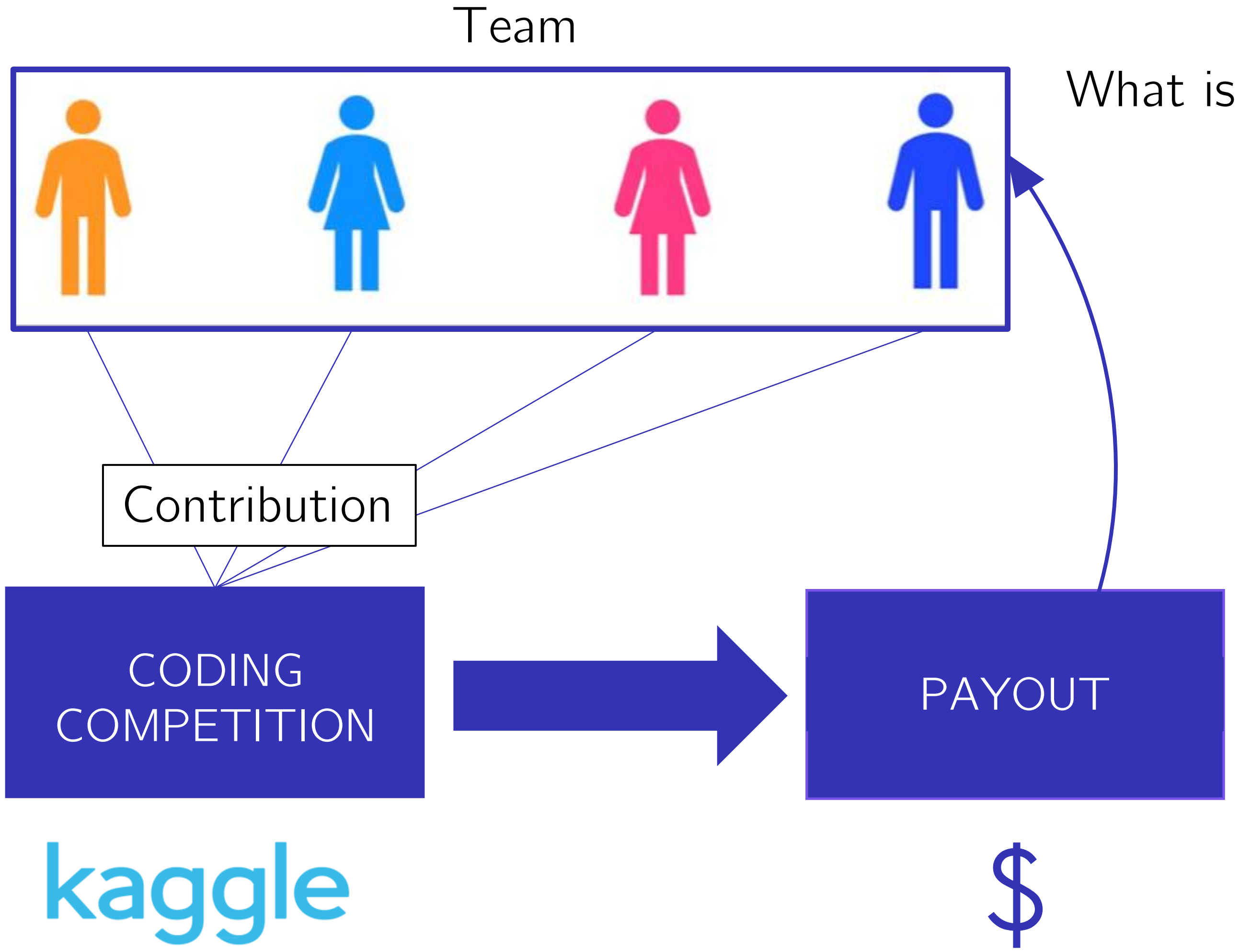
CODING COMPETITION

PAYOUT

kaggle

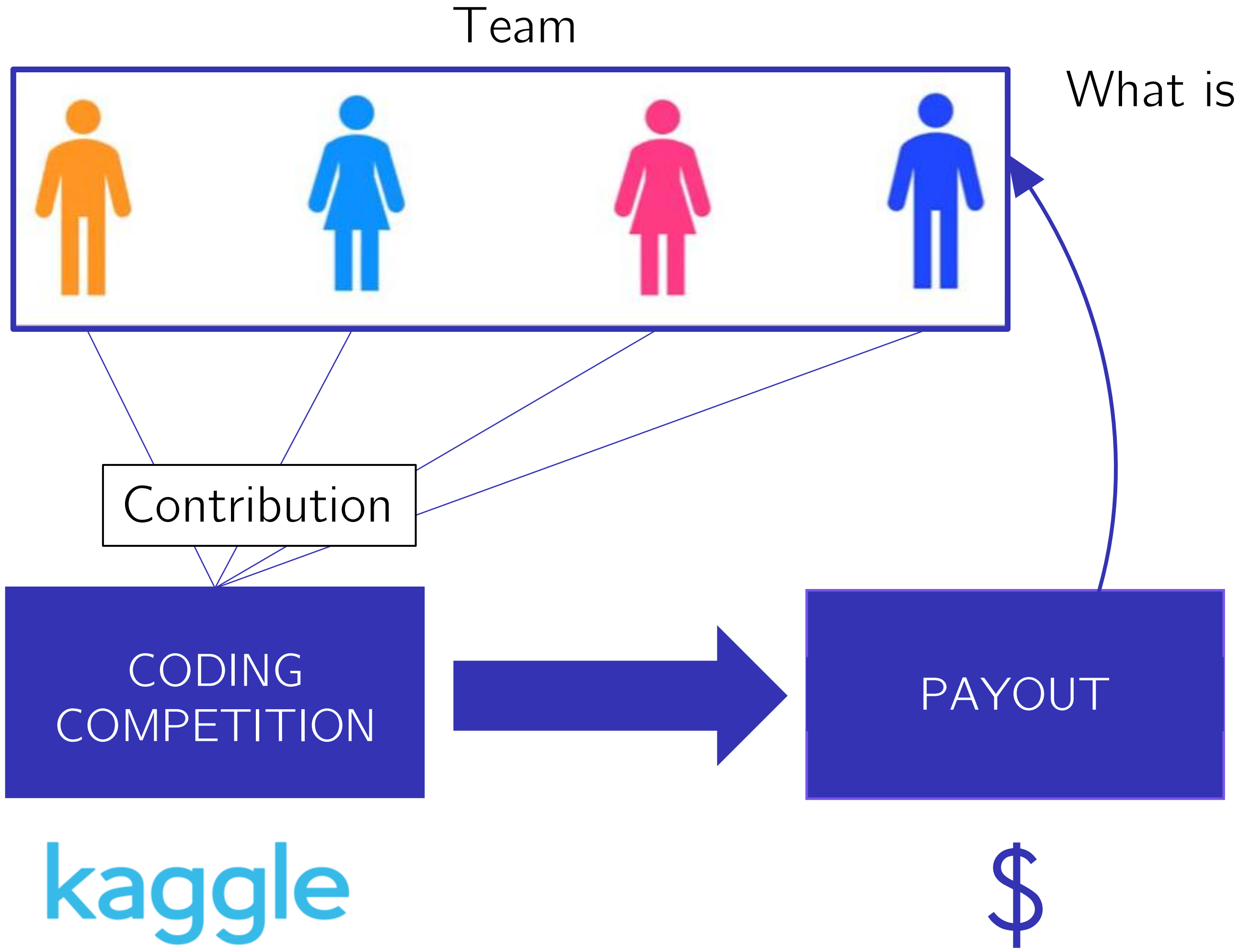
$

Shapley values



What is a fair distribution?

Shapley values

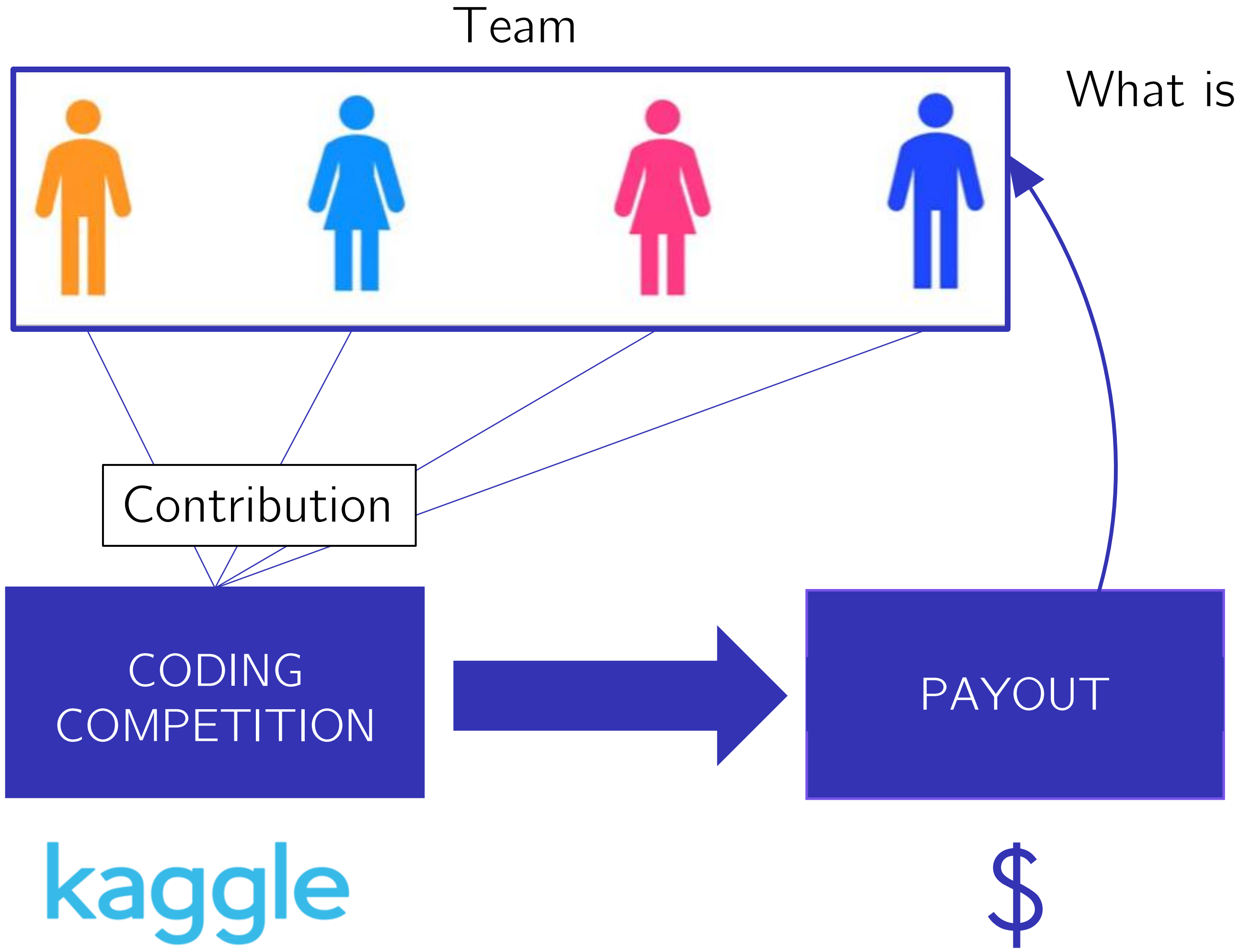


What is a fair distribution?

SHAPLEY VALUES

Lloyd Shapley
(1951)

Shapley values



What is a fair distribution?

SHAPLEY VALUES

Lloyd Shapley
(1951)

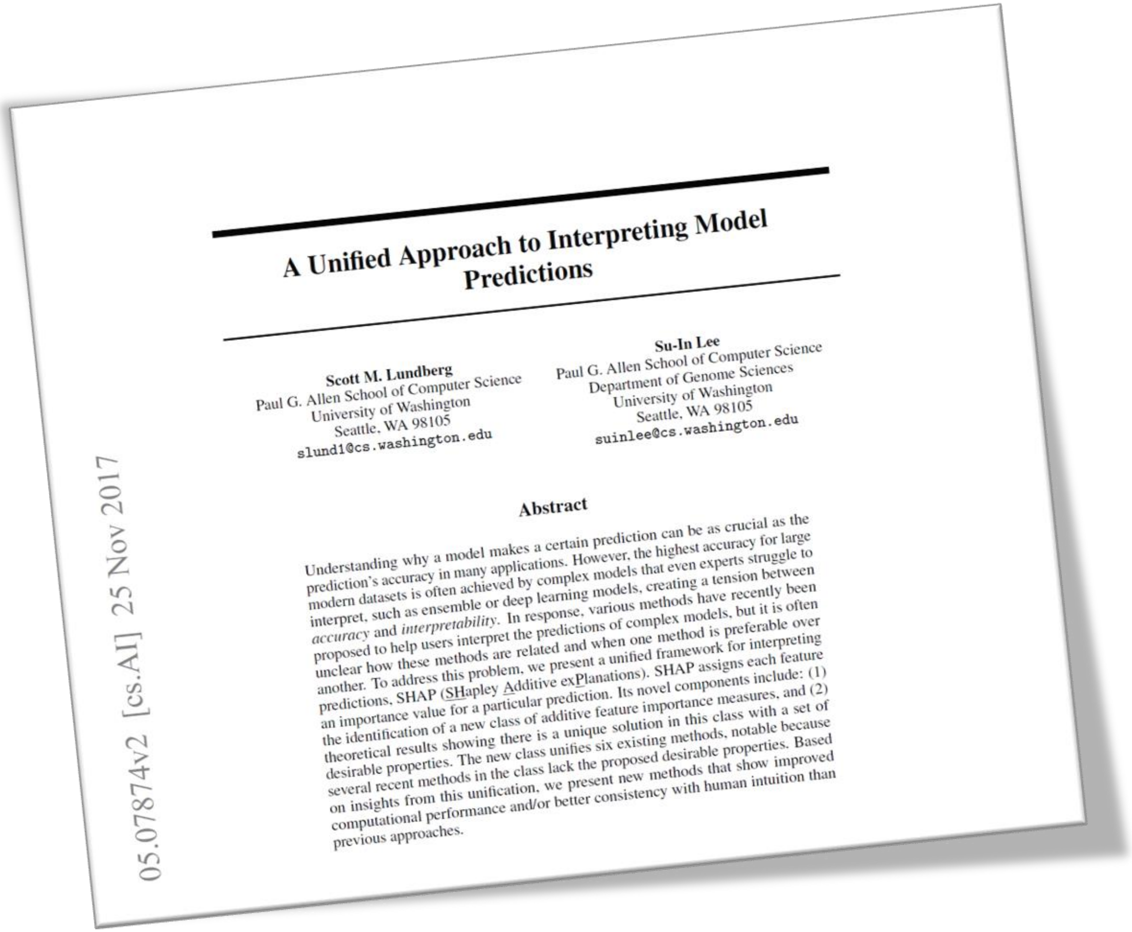
$$\phi_i(v) = \sum_{S\subseteq N\setminus\{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!}(v(S\cup\{i\})-v(S))$$

SHAP

SHappley

Additive

ex**P**lanations



05.07874v2 [cs.AI] 25 Nov 2017

A Unified Approach to Interpreting Model Predictions

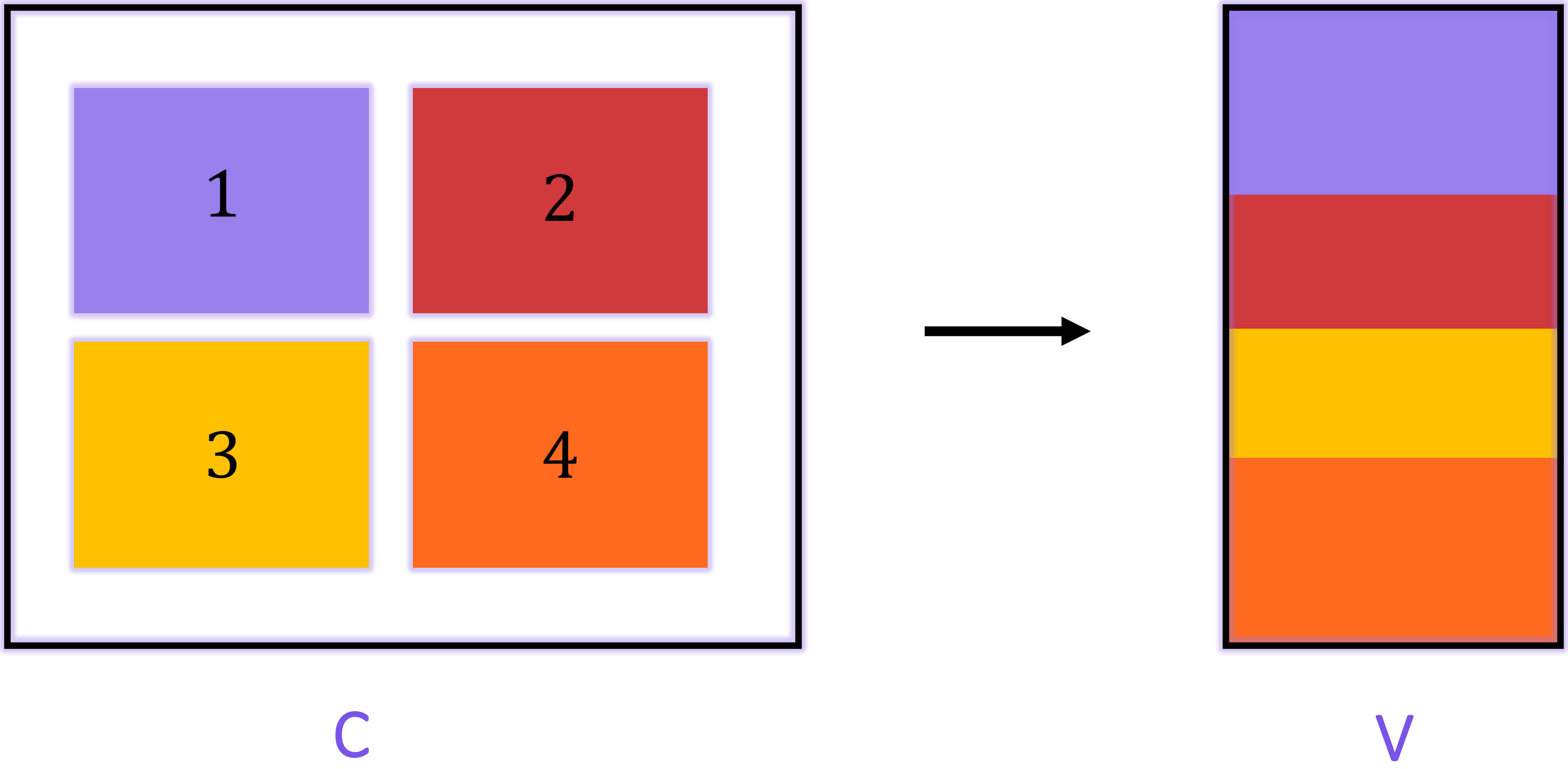
Scott M. Lundberg
Paul G. Allen School of Computer Science
University of Washington
Seattle, WA 98105
`slund1@cs.washington.edu`

Su-In Lee
Paul G. Allen School of Computer Science
Department of Genome Sciences
University of Washington
Seattle, WA 98105
`suinlee@cs.washington.edu`

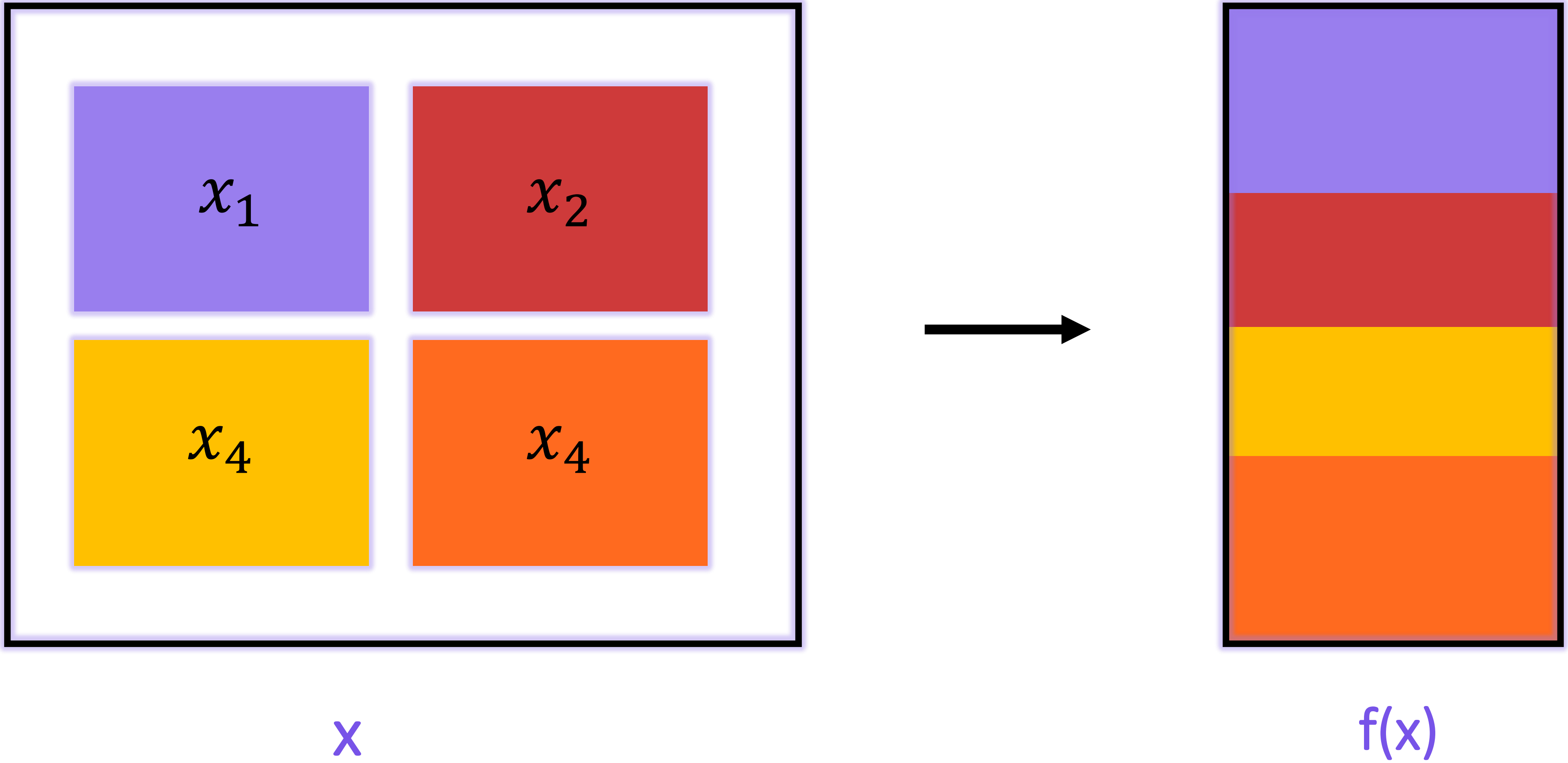
Abstract

Understanding why a model makes a certain prediction can be as crucial as the prediction's accuracy in many applications. However, the highest accuracy for large modern datasets is often achieved by complex models that even experts struggle to interpret, such as ensemble or deep learning models, creating a tension between *accuracy* and *interpretability*. In response, various methods have recently been proposed to help users interpret the predictions of complex models, but it is often unclear how these methods are related and when one method is preferable over another. To address this problem, we present a unified framework for interpreting predictions, SHAP (SHapley Additive exPlanations). SHAP assigns each feature an importance value for a particular prediction. Its novel components include: (1) the identification of a new class of additive feature importance measures, and (2) theoretical results showing there is a unique solution in this class with a set of desirable properties. The new class unifies six existing methods, notable because several recent methods in the class lack the proposed desirable properties. Based on insights from this unification, we present new methods that show improved computational performance and/or better consistency with human intuition than previous approaches.

SHAP



SHAP



SHAP

SHappley

Additive

ex**P**lanations

Additive feature attribution methods

1. $if \ x \approx x' \ then \ f(x) \approx g(x')$

Additive feature attribution methods

①
$$if\ \ x \approx x'\ \ then\ \ f(x) \approx g(x')$$

②
$$g(x') = \phi_0 + \sum_{i=1}^{p} \phi_i x'_i$$

Additive feature attribution methods properties

Properties

Additive feature attribution methods properties

Properties

1

Local accuracy

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{p} \phi_i x'_i$$

Additive feature attribution methods properties

Properties

2

Missingness

$$x'_i = 0 \implies \phi_i = 0$$

Additive feature attribution methods properties

Properties

3

Consistency

Let $f_x(z') = f(h_x(z'))$ and $z'\backslash i$ denote setting $z'_i = 0$. For any two models f and f':

$$\forall\ z' \in \{0,1\}^p,\ f'_x(z') - f'_x(z'\backslash i) \geq f_x(z') - f_x(z'\backslash i) \implies \phi_i(f',x) \geq \phi_i(f,x)$$

Attribution methos satisfying properties 1, 2, 3

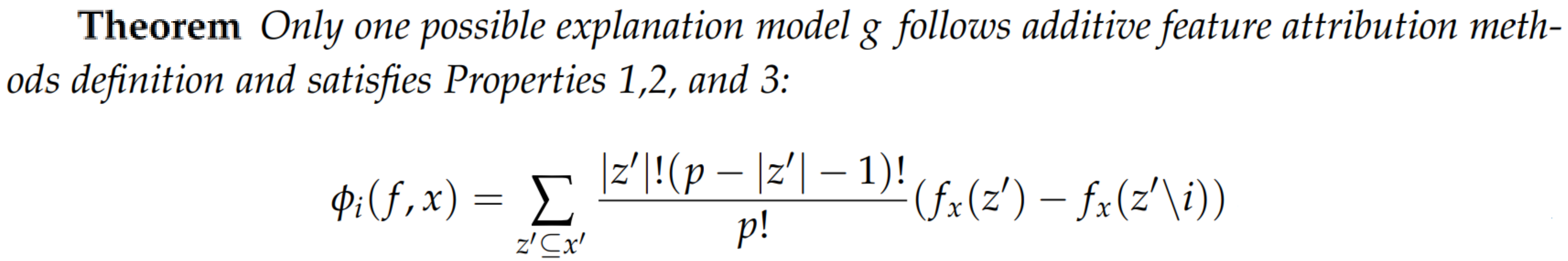
$$g(x') = \phi_0 + \sum_{i=1}^{p} \phi_i x'_i$$

Attribution methos satisfying properties 1, 2, 3

$$g(x') = \phi_0 + \sum_{i=1}^{p} \phi_i x'_i$$

Attribution methos satisfying properties 1, 2, 3

$$g(x') = \phi_0 + \sum_{i=1}^{p} \phi_i x'_i$$



Theorem *Only one possible explanation model g follows additive feature attribution methods definition and satisfies Properties 1,2, and 3:*

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(p - |z'| - 1)!}{p!} (f_x(z') - f_x(z' \backslash i))$$

Calculating Shapley values

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Calculating Shapley values

$$\underset{\uparrow}{\phi_i}(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} \left[f_x(z') - f_x(z' \setminus i) \right]$$

Shapley value
for feature i

Calculating Shapley values

AGE

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} \left[f_x(z') - f_x(z' \setminus i) \right]$$

Shapley value
for feature i

Calculating Shapley values

Black Box model

AGE

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Shapley value
for feature i

Calculating Shapley values

Black Box model

Input data point

AGE

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Shapley value
for feature i

Calculating Shapley values

Black Box model

Input data point

AGE

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Shapley value
for feature i

$x =$

$\boldsymbol{Age\ =\ 56}$	$\boldsymbol{Gender\ =\ F}$	$\boldsymbol{BMI\ =\ 30}$	$\boldsymbol{Stroke\ =\ yes}$	…

Calculating Shapley values

Black Box model

Input data point

AGE

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Shapley value
for feature i

$$x = \boxed{Age = 56}\boxed{Gender = F}\boxed{BMI = 30}\boxed{Stroke = yes}\boxed{\ldots}$$

Calculating Shapley values

Black Box model

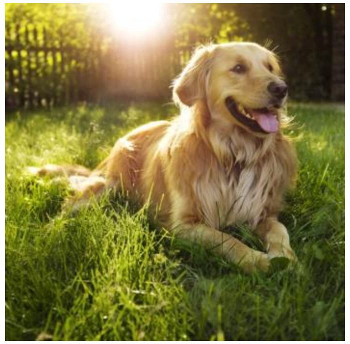
Input data point

AGE

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Shapley value for feature i

Simplified data input



$x =$

***Age* = 56**	***Gender* = *F***	***BMI* = 30**	***Stroke* = *yes***	…

Calculating Shapley values

Black Box model

Input data point

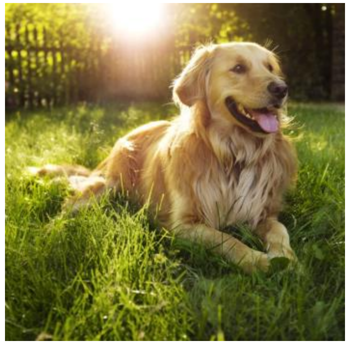
AGE

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Shapley value for feature i

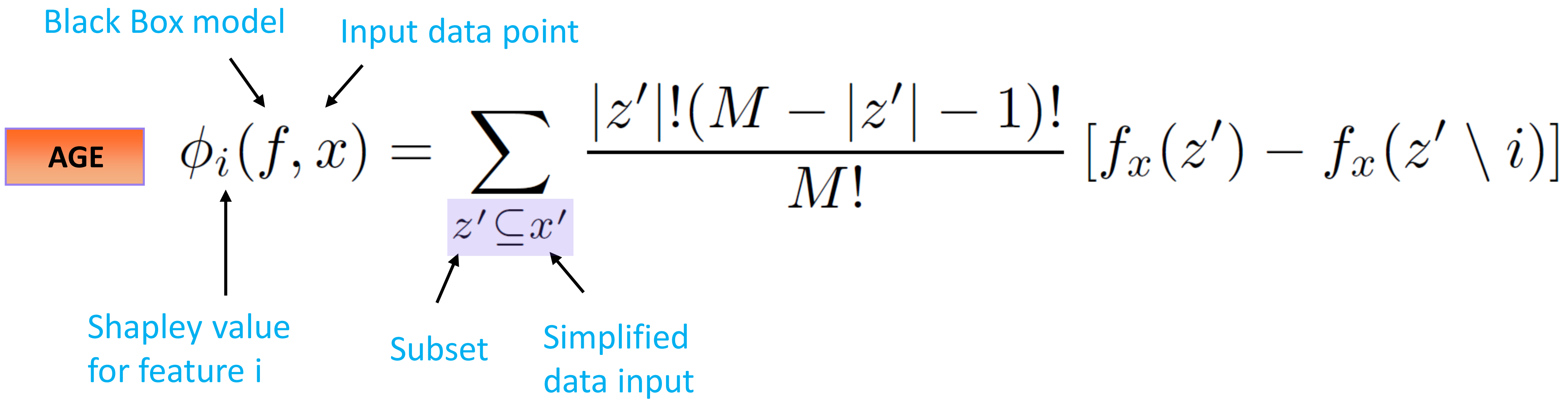
Subset

Simplified data input



$x =$	***Age = 56***	***Gender = F***	***BMI = 30***	***Stroke = yes***	…

Calculating Shapley values



Black Box model

Input data point

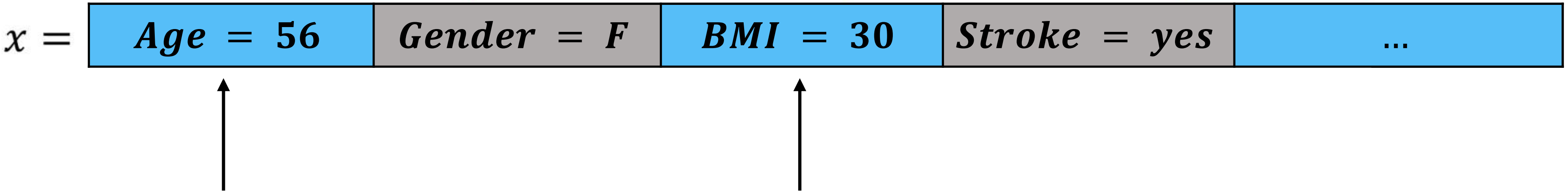
AGE

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Shapley value for feature i

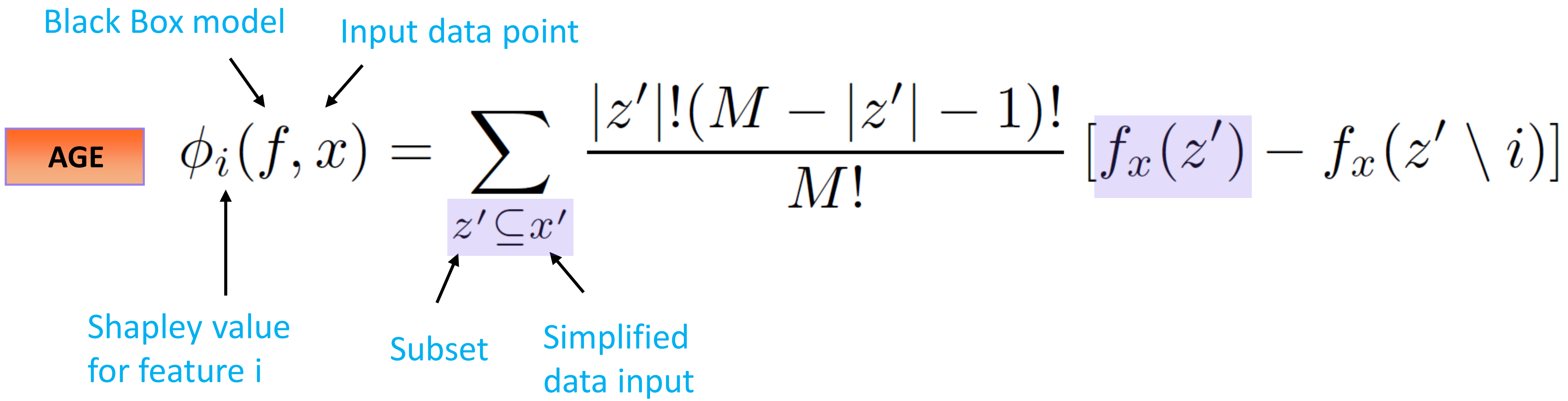
Subset

Simplified data input



$x =$ | ***Age = 56*** | ***Gender = F*** | ***BMI = 30*** | ***Stroke = yes*** | …

Calculating Shapley values



Black Box model

Input data point

AGE

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Shapley value for feature i

Subset

Simplified data input

$x =$

$Age = 56$	$Gender = F$	$BMI = 30$	$Stroke = yes$	…

Calculating Shapley values

Black Box model

Input data point

AGE

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(z' \setminus i)]$$

Shapley value for feature i

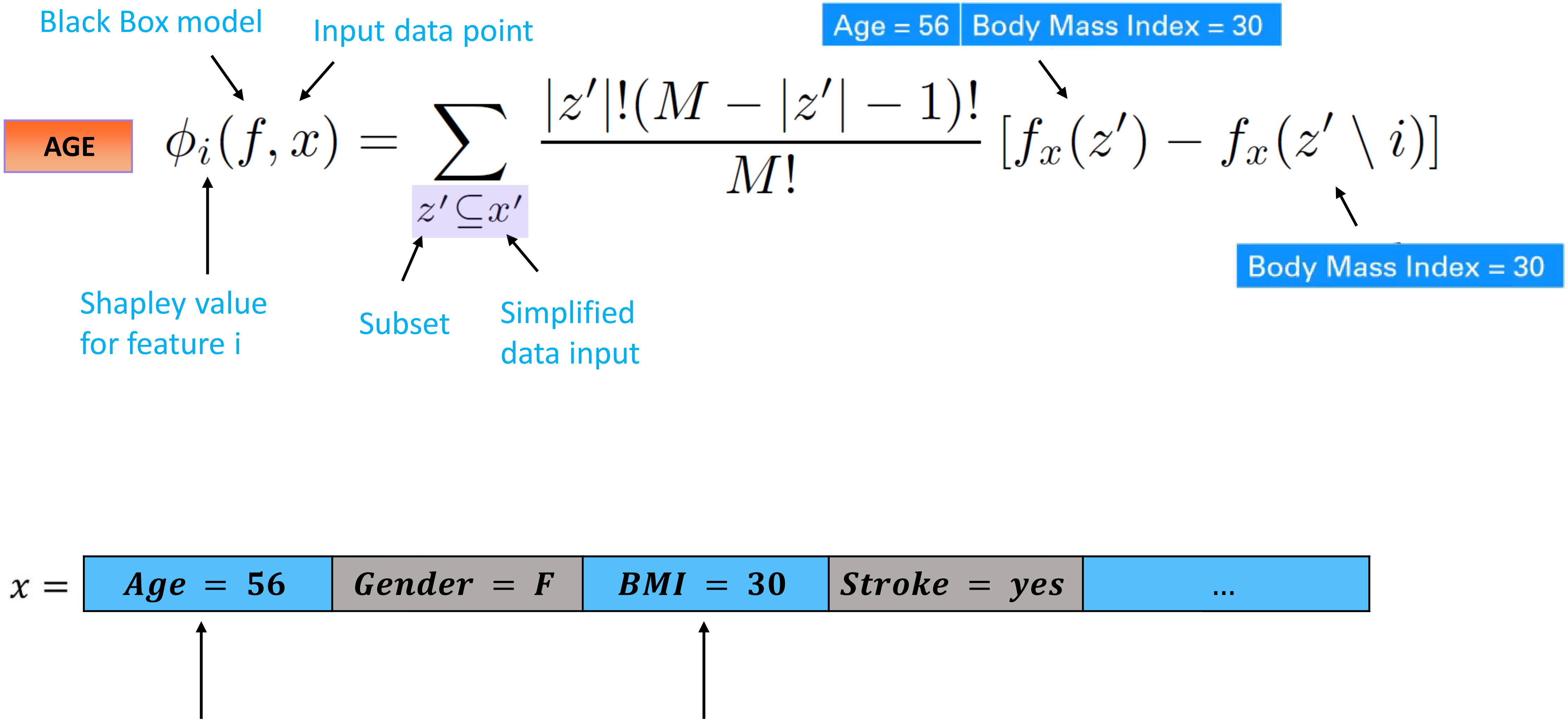
Subset

Simplified data input

$x =$

Age = 56	*Gender* = *F*	*BMI* = 30	*Stroke* = *yes*	…

Calculating Shapley values



Black Box model

Input data point

Age = 56 | Body Mass Index = 30

AGE

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!}\left[f_x(z') - f_x(z' \setminus i)\right]$$

Body Mass Index = 30

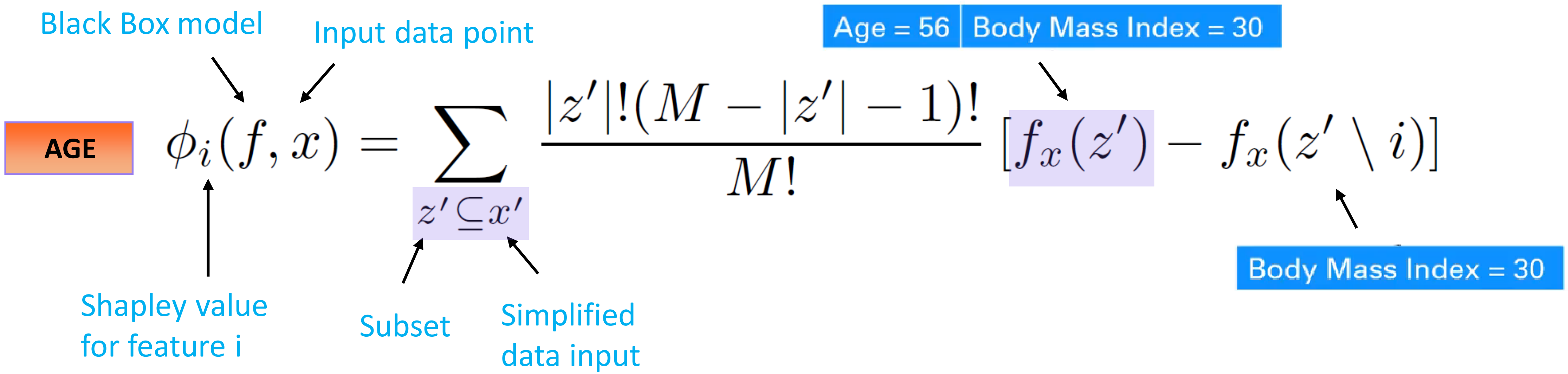
Shapley value for feature i

Subset

Simplified data input

$$x = \boxed{\boldsymbol{Age\ =\ 56}}\boxed{\boldsymbol{Gender\ =\ F}}\boxed{\boldsymbol{BMI\ =\ 30}}\boxed{\boldsymbol{Stroke\ =\ yes}}\boxed{\ldots}$$

Calculating Shapley values



Black Box model

Input data point

Age = 56 | Body Mass Index = 30

AGE

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!}\left[f_x(z') - f_x(z' \setminus i)\right]$$

Body Mass Index = 30

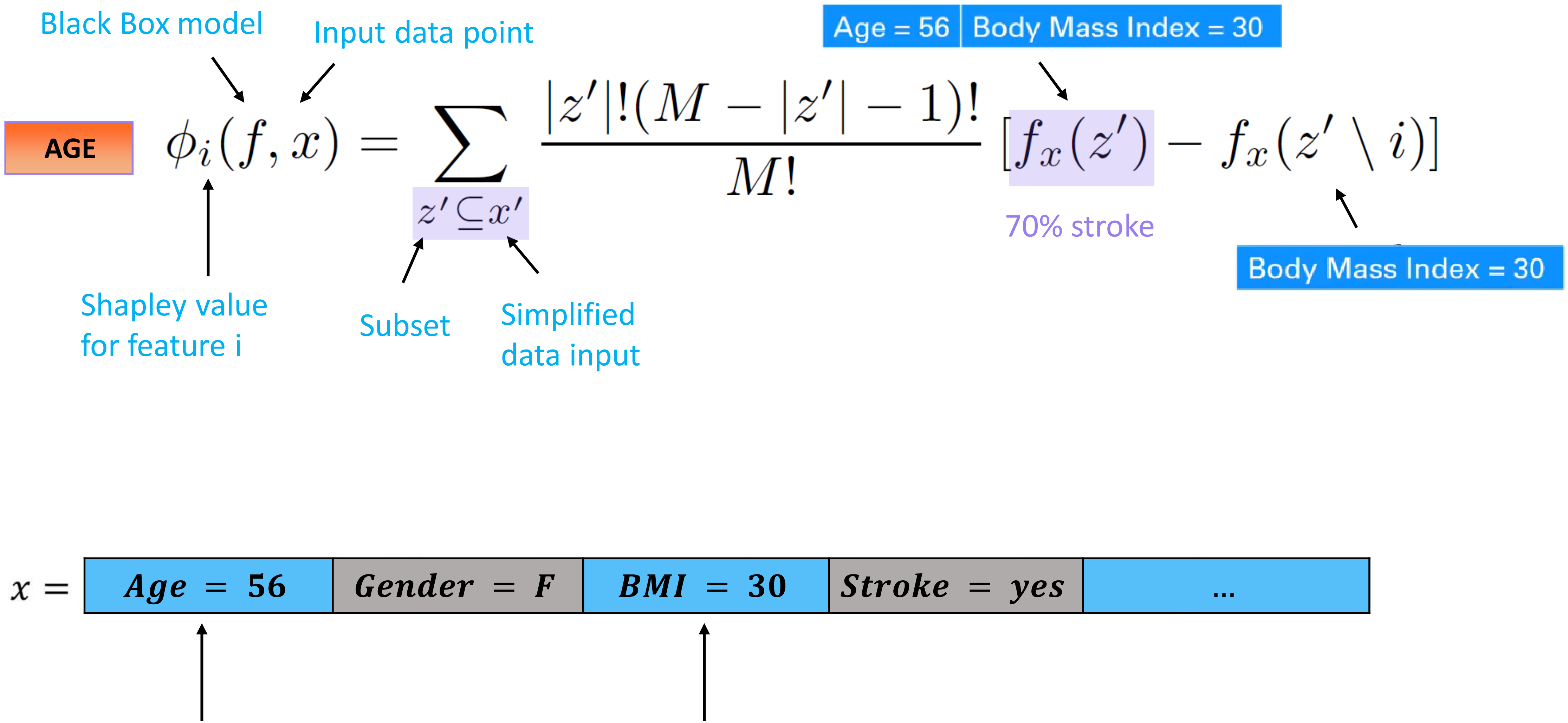
Shapley value for feature i

Subset

Simplified data input

$x =$ *Age* = 56 | *Gender* = *F* | *BMI* = 30 | *Stroke* = *yes* | …

Calculating Shapley values



AGE

Black Box model

Input data point

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!}[f_x(z') - f_x(z' \setminus i)]$$

Age = 56 Body Mass Index = 30

70% stroke

Body Mass Index = 30

Shapley value for feature i

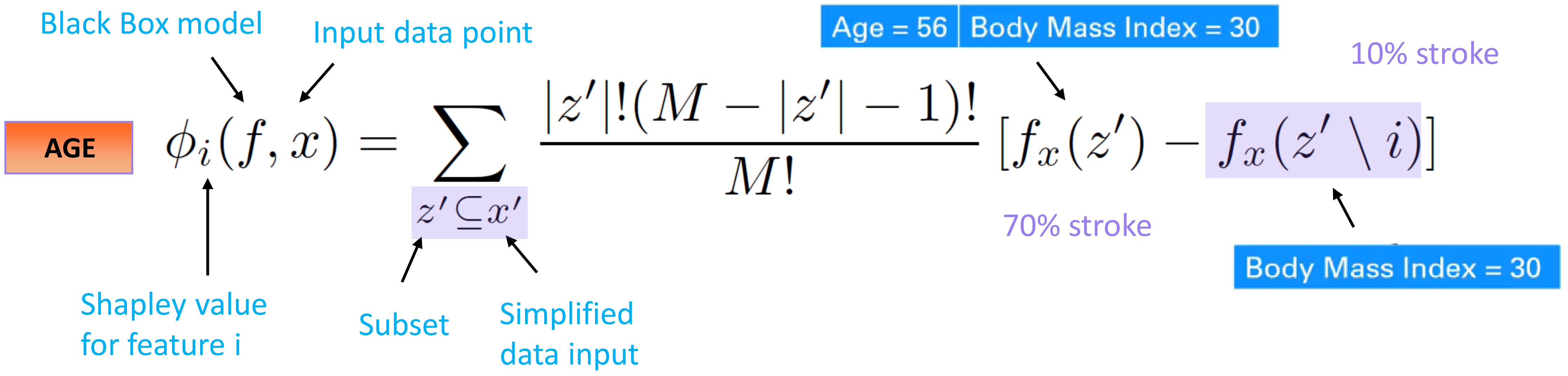
Subset

Simplified data input

$x =$

Age = 56	*Gender* = *F*	*BMI* = 30	*Stroke* = *yes*	...

Calculating Shapley values



Black Box model

Input data point

Age = 56 Body Mass Index = 30

10% stroke

AGE

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M-|z'|-1)!}{M!}[f_x(z') - f_x(z' \setminus i)]$$

70% stroke

Body Mass Index = 30

Shapley value
for feature i

Subset

Simplified
data input

$x =$

$Age = 56$	$Gender = F$	$BMI = 30$	$Stroke = yes$	…

Calculating Shapley values

UNIVERSITAT DE BARCELONA

Black Box model

Input data point

AGE

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \underbrace{\frac{|z'|!(M-|z'|-1)!}{M!}}_{\text{Weighting}} [f_x(z') - f_x(z' \setminus i)]$$

Shapley value for feature i

Subset

Simplified data input

$x =$

Age = 56	***Gender = F***	***BMI = 30***	***Stroke = yes***	…

Calculating Shapley values

Black Box model

Input data point

AGE

$$\phi_i(f, x) = \sum_{z' \subseteq x'} \underbrace{\frac{|z'|!(M - |z'| - 1)!}{M!}}_{\text{Weighting}} \underbrace{[f_x(z') - f_x(z' \setminus i)]}_{\text{Contribution}}$$

Shapley value for feature i

Subset

Simplified data input

$x =$	$Age = 56$	$Gender = F$	$BMI = 30$	$Stroke = yes$	…

Calculating Shapley values

Black Box model

Input data point

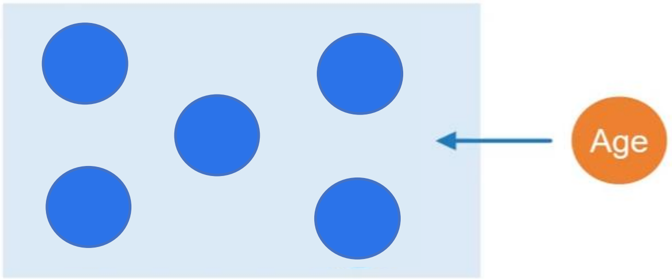
AGE

$$\phi_i(f,x) = \sum_{z' \subseteq x'} \underbrace{\frac{|z'|!(M-|z'|-1)!}{M!}}_{\text{Weighting}} \underbrace{[f_x(z') - f_x(z' \setminus i)]}_{\text{Contribution}}$$

Shapley value for feature i

Subset

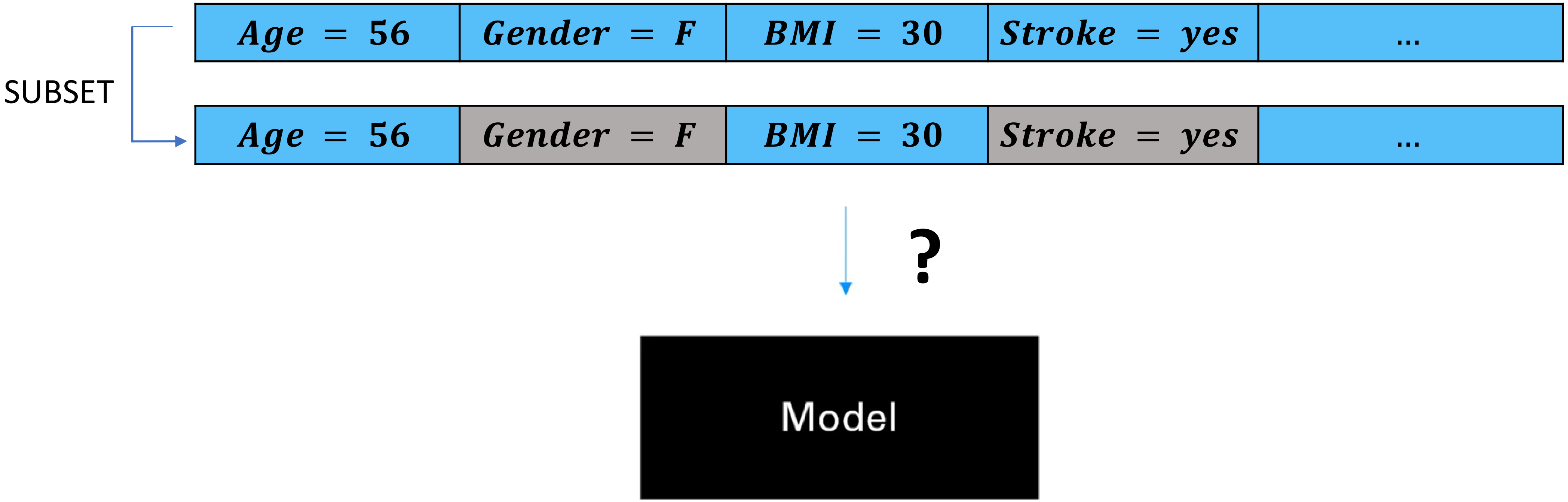
Simplified data input



$x =$

Age = 56	*Gender = F*	*BMI = 30*	*Stroke = yes*	...

Calculating Shapley values



$Age = 56$ | $Gender = F$ | $BMI = 30$ | $Stroke = yes$ | …

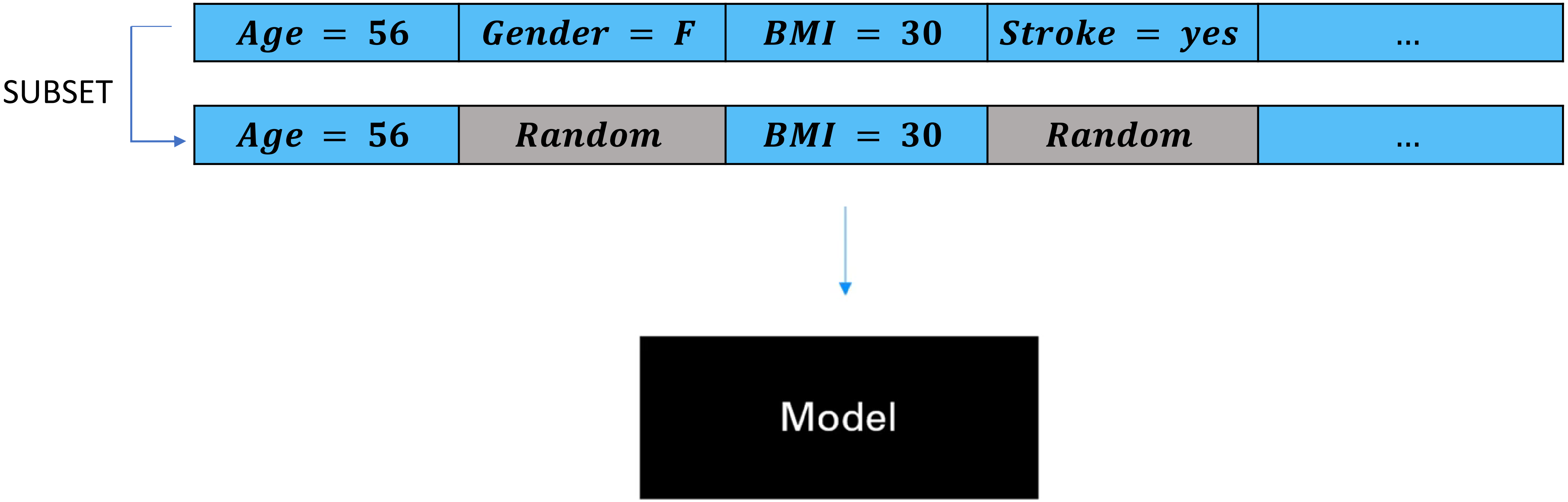
SUBSET

$Age = 56$ | $Gender = F$ | $BMI = 30$ | $Stroke = yes$ | …

?

Model

Calculating Shapley values



SUBSET

$Age = 56$	$Gender = F$	$BMI = 30$	$Stroke = yes$	…
$Age = 56$	$Random$	$BMI = 30$	$Random$	…

Model

Calculating Shapley values

$$2^n = total\ number\ of\ subsets\ of\ a\ set\ of\ size\ n$$

Calculating Shapley values

$$2^n = total\ number\ of\ subsets\ of\ a\ set\ of\ size\ n$$

$$4\ features\colon\ 64\ total\ coalitions\ to\ sample$$

Calculating Shapley values

$$2^n = total\ number\ of\ subsets\ of\ a\ set\ of\ size\ n$$

$$4\ features\colon\ 64\ total\ coalitions\ to\ sample$$

$$32\ features\colon\ 17.1\ billion$$

Shapley kernel

Shapley kernel

Shapley kernel theorem

$$\left.\begin{aligned}\Omega(g) &= 0,\\ \pi_{x'}(z') &= \frac{M-1}{\binom{M}{|z'|}|z'|(M-|z'|)},\\ \mathcal{L}(f,g,\pi_{x'}) &= \sum_{z'\in\mathcal{Z}}\left[f(h_x^{-1}(z'))-g(z')\right]^2\pi_{x'}(z')\end{aligned}\right\} \quad \xi(x) = \underset{g\in G}{\arg\min}\,\mathcal{L}(f,g,\pi_x)+\Omega(g)$$

Kernel SHAP = LIME + SHAPLEY VALUES

SHAP limitations

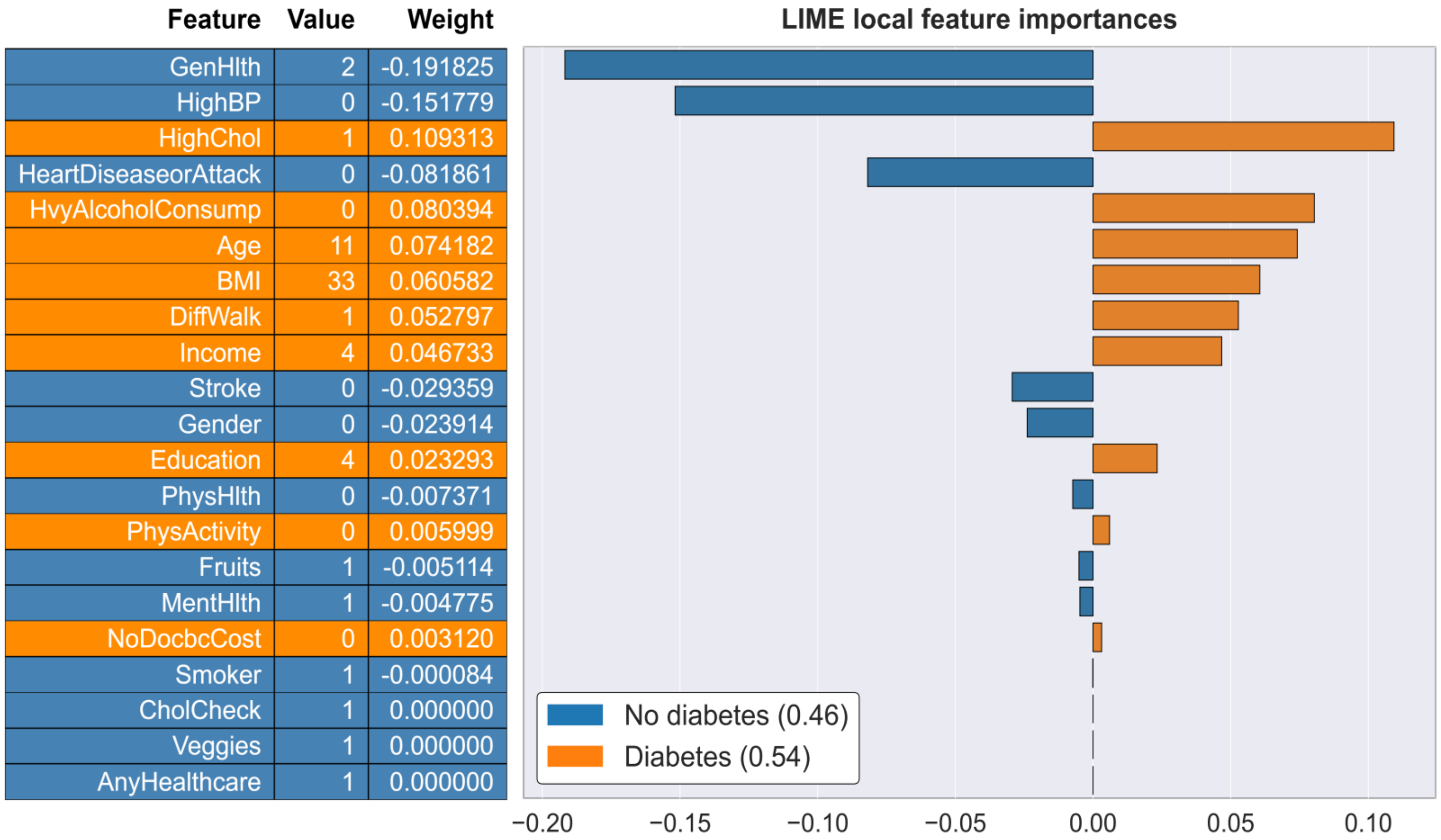
- Computational cost
- Access to data
- Feature dependencies

LIME and SHAP application

Step by step

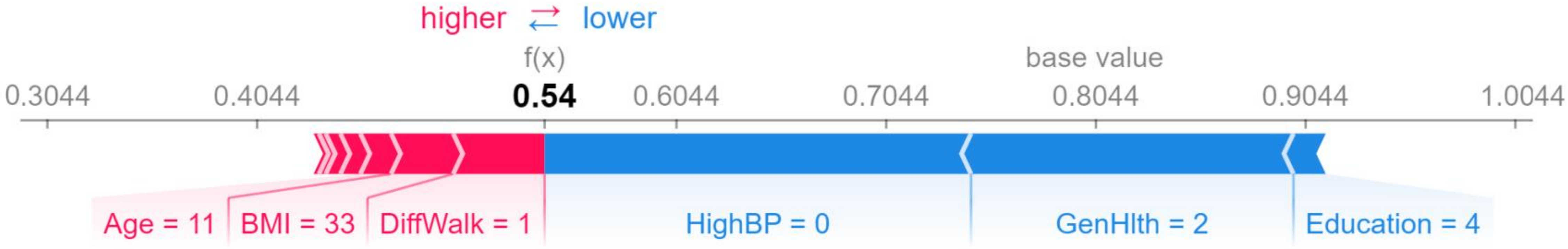
- *Diabetes public tabular database*
- *Random forest fit with this database*
- *LIME and SHAP explanations*

LIME explanation



Feature	Value	Weight
GenHlth	2	-0.191825
HighBP	0	-0.151779
HighChol	1	0.109313
HeartDiseaseorAttack	0	-0.081861
HvyAlcoholConsump	0	0.080394
Age	11	0.074182
BMI	33	0.060582
DiffWalk	1	0.052797
Income	4	0.046733
Stroke	0	-0.029359
Gender	0	-0.023914
Education	4	0.023293
PhysHlth	0	-0.007371
PhysActivity	0	0.005999
Fruits	1	-0.005114
MentHlth	1	-0.004775
NoDocbcCost	0	0.003120
Smoker	1	-0.000084
CholCheck	1	0.000000
Veggies	1	0.000000
AnyHealthcare	1	0.000000

SHAP explanation



Contents

Conclusion: LIME vs SHAP

	LIME	SHAP
Theory driven	Fails at being consistent. ✗	Supported by the Shapley values theory properties and consistency property. ✓
Time expensive	Time affordable. ✓	Computation of marginal contributions for all possible coalitions makes it time expensive. ✗
Require training data	Does not require the training set for fitting the surrogate model. ✓	Requires the training set for generating the background set that will be used to train the surrogate model. ✗
What-if explanations	Can provide what-if explanations. ✓	Cannot provide what-if explanations. ✗
Improbable instances	Improbable instances may be generated when obtaining perturbed instances. ✗	When imputing omitted features, improbable instances may be generated. ✗
Instability	Kernel width can make it unstable. ✗	Its strong theoretical properties makes it stable. ✓

Conclusion: Summary

- Detailed insight into the theory behind random forest
- Formalise and unify the theory behind LIME and SHAP
- Healthcare application for LIME and SHAP

Future work

- See how LIME explanations vary depending on the kernel width
- Expand LIME and SHAP theory and application to images
- Compare LIME and SHAP explanations