



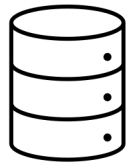
Explaining by removing

Explainable and Trustworthy AI

Eliana Pastor

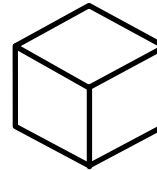
Stages of Explainability

- Explainability involves the entire AI development pipeline



Pre-modelling explainability

- Before building the model
- Data exploration
 - Data selection
 - Feature engineering



Explainable modeling

- Build inherently interpretable models
- Manage the accuracy and interpretability trade-off

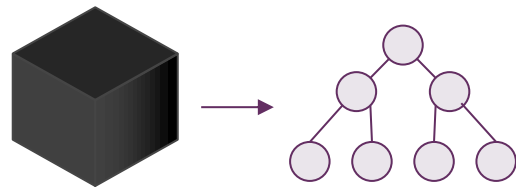


Post-modelling explainability

- After model development
- Explaining predictions and behavior of trained models

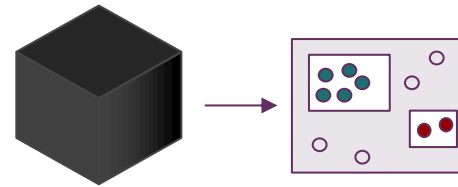
Scope of Explainability

- *What do we explain?*



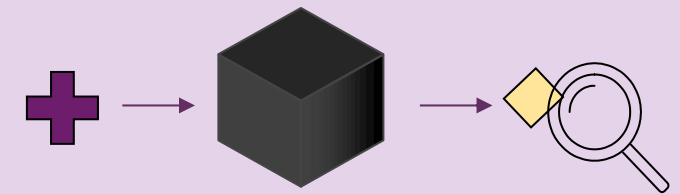
Global

How the model globally works



Subgroup

How the model behaves in data subgroups



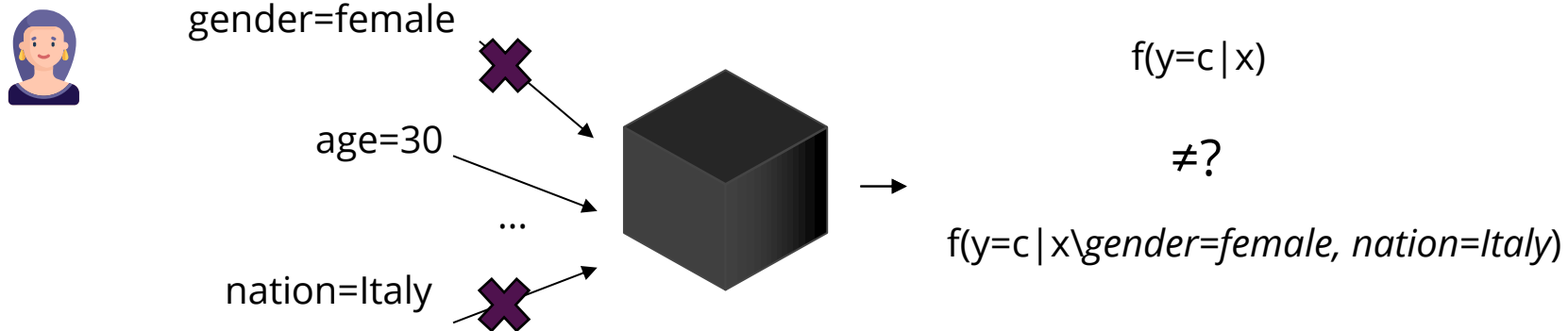
Individual/local

Explaining the reasons behind individual predictions

Explaining by removing

Also known as *occlusion-based* or *perturbation-based* since often the occlusion is performed via perturbations

The idea is to remove one or more input feature (or simulate the removal) to quantify the feature influence



Explaining by removing - Approaches

- PredDiff – base approach
- IME – Explaining with Shapley Values
- SHAP
- LIME*

*LIME indeed use perturbations!

PredDiff – Prediction Difference

Let f the model and let be x the instance to explain.

The importance of attribute A_i is:

$$\text{predDiff}_i(x) = f(x) - f(x \setminus A_i)$$

Where $x \setminus A_i$ is the instance x without the information of A_i .

PredDiff – Prediction Difference

Evaluation of the difference → Solution a) Difference in probability

A possibility is to directly evaluate the **difference between probabilities** for class c

$$\mathit{predDiff}_i(x) = p(y = c|x) - p(y = c|x \setminus A_i)$$

Where c is typically the predicted class, as we want to explain why the model made a particular decision

PredDiff – Prediction Difference

Evaluation of the difference → Solution b) Information difference

Another possibility is to compute the information difference

$$\text{infoDiff}_i(x) = \log_2 p(y = c|x) - \log_2 p(y = c|x \setminus A_i)$$

Where c is typically the predicted class, as we want to explain why the model made a particular decision

PredDiff

How to model the removal of information $f(x \setminus A_i)$?

The approach was proposed for tabular data.

Simulate it with an 'average value'

$$p(y|x \setminus A_i) = \sum_{s=1}^{m_i} p(A_i = a_s) p(y|x \leftarrow A_i = a_s)$$

Categorical attributes.

Replace the value $A_i = a_k$ in x with all possible values of A_i (m_i is the number of values of the attribute A_i) and weight each prediction by the prior probability of the value.

The $p(y|x \leftarrow A_i = a_s)$ represents the probability for y when in x we replace the value of A_i with a_s

PredDiff

How to model the removal of information $f(x \setminus A_i)$?

The approach was proposed for tabular data.

Simulate it with an 'average value'

$$p(y|x \setminus A_i) = \sum_{s=i}^{s=m_i} p(A_i = a_s) p(y|x \leftarrow A_i = a_s)$$

Numerical attributes.

First discretize and split the values of A_i into sub-intervals. The middle points of these sub-intervals are the representative values for a_s

Use probabilities of the sub-intervals for weighting the predictions

PredDiff

We obtain a **feature importance** for each feature

The higher the magnitude \rightarrow the more the attribute values impact the prediction

Positive contribution \rightarrow the attribute values pushes toward the prediction for class c

Negative contribution \rightarrow the attribute values pushes against the prediction for class c

Advantages of PredDiff

- Model agnostic
- Local explanations
- Provides feature attributions
- Direct interpretation of the result

Limitations of PredDiff

- Defined for structured data
- Perturbations/info removals may create unrealistic data
- Need access data to compute $x \setminus A_i$
 - Replace the attribute with values from (training) instances $p(y|x \leftarrow A_i = a_s)$
- Does not consider interaction between features for the removal

Explaining by removing – consider interactions

Consider the **contribution of multiple features** at the time,
i.e., removing feature A_i and removing features A_i and A_j

Research question.

How can we **aggregate** such importance scores into a single attributions?

- i.e., the contribution of A_i , considering the interactions with other attributes

A possibility → Shapley values

Shapley value

Concept for game theory to assign a **relevance score to each player of a team**, assuming that they collaborate

Team



Player 1



Player 2



Player 3



Player 4

Total score of Team

Goal. Compute the contribution of each player ϕ_i ?



Shapley value

- N players
- v is a function maps subsets of players to the real numbers: $v: 2^N \rightarrow \mathbb{R}$, with $v(\emptyset) = 0$
- S = coalition of players
- $v(S)$ is the total expected sum of payoffs the player in S can obtain by cooperating

Contribution of player i $\phi_i(v)$

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

Marginal contribution

Shapley value

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

- Players A, B, C
- $\phi_A(v)$?

Terms to sum:

- $\frac{2}{6} (v(A) - v(\emptyset))$
- $\frac{1}{6} ((v(\{A, B\}) - v(B))$
- $\frac{1}{6} (v(\{A, C\}) - v(C))$
- $\frac{2}{6} (v(\{A, B, C\}) - v(\{B, C\}))$
- Coalition considered: 2^3

Properties of Shapley Value (1/2)

- **Efficiency**

The sum of the Shapley values of all players is equal to the value of the entire team

$$\sum_{i \in N} \phi_i(v) = v(N)$$

- **Simmetry**

Players with the same marginal contributions have the same contribution ϕ

If $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $S \subseteq N \setminus \{i, j\}$, then $\phi_i(v) = \phi_j(v)$

Properties of Shapley Value (2/2)

- **Linearity**

If two games described by functions v and w are combined, then the distributed contribution correspond to sum of the contribution from v and the contribution from w

- $\phi_i(v + w) = \phi_i(v) + \phi_i(w)$
- $\phi_i(\alpha \cdot v) = \alpha \cdot \phi_i(v)$

- **Null player**

A player with marginal contribution equal to 0 will have 0 as contribution ϕ

$$v(S \cup i) = v(S) \quad \forall S \rightarrow \phi_i(v) = 0$$

Shapley value is the **only** assignment that satisfy the Efficiency, Symmetry, Linearity and Null player properties

Use Shapley value for XAI

Players \rightarrow

Total score $v(N) \rightarrow$

$S \rightarrow$

$v(S) \rightarrow$

$\phi \rightarrow$

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

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

Use Shapley value for XAI

Players \rightarrow feature values

Total score $v(N)$ \rightarrow difference in probability with the average prediction (prior)

S \rightarrow instance when feature values $N \setminus S$ are omitted

$v(S)$ \rightarrow (a) prediction probability for instance S with only feature values S are available compared to average prediction / (b) prediction

ϕ \rightarrow feature attributions

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

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

Function v

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

$$v(S) = f(S) - E[f(X)]$$

where $f(S)$ is the model prediction marginalizing over the feature not in S

This interpretation is in line with the interpretation of some interpretable model such as linear regression

- Contribution of the feature value respect to the average/prior behavior

Following this line, $v(N)$ (i.e., for the instance to explain x , all players) is the prediction probability for an instance minus the average prediction for all instances ,

$$v(N) = v(x) = f(x) - E[f(X)]$$

Computation of $v(S)$ and $v(S \setminus \{i\})$

How to model the removal of information?

- As for PredDiff: simulate it with an 'average value' substituting with possible values
 - Less adopted
- **Random**
 - Replace the feature values of features that are not in a coalition with random feature values from the dataset
 - Consider multiple replacement and provide the average value v

Example of coalitions to evaluate

Example. Gender=F, age=30, nationality=IT, income=20k

- To compute the contribution of gender=F, we need to evaluate the following coalitions (2^4)
 - $\{\}$ | gender=F
 - age=30 | gender=F
 - nationality=IT | gender=F
 - income=20k | gender=F
 - age=30, nationality=IT | gender=F
 - age=30, income=20k | gender=F
 - nationality=IT, income=20k | gender=F
 - age=30, nationality=IT, income=20k | gender=F

Shapley value

- Contribution of player i $\phi_i(v)$

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

- Alternative form

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

Advantages of using the Shapley Values for XAI

- Model agnostic & Local explanations & Provides feature attributions & Direct interpretation of the result
- Consider interaction among features
- Efficiency Property of Shapley Values
 - The difference between prediction for instance x and average prediction is distributed among feature values.
 - $\sum_{i=1}^p \phi_i(f) = f(x) - E[f(X)]$
Where p is the number of attributes and f is the model to explain
- Solid Theoretical Foundation
 - Axioms: efficiency, symmetry, linearity, null player

Limitations of using the Shapley Values for XAI

- Computation time is exponential in the number of players/feature values
 - Exact computation is computationally expensive.
 - For N players, 2^K coalitions of the feature values
- Need access data to compute $v(S)$ and $v(S \setminus \{i\})$
- Perturbations/info removals may create unrealistic data
- The absence of a feature to compute $v(S)$ and $v(S \setminus \{i\})$ is simulated by drawing random instances \rightarrow variance for the estimate of the Shapley value

Problem of exact computation

Exact computation → **exponential** in the number of players/feature and feature values

- To compute $\phi_i(v)$ we have to evaluate all possible coalitions of feature values with and without the feature A_i

The **exact solution** to this problem becomes **problematic** as the number of possible coalitions exponentially increases as more features are considered

Proposed solution: approximation with Monte-Carlo sampling

Shapley Approximation - Random coalition

- Take a random instance z from the dataset
- x_{+j} is an instance where a **random** number of **feature values** are **replaced by feature values from the random data point z** . The value for feature j is as in the original feature x
 - The random features that are replaced are the features not in the coalition
- x_{-j} is as x_{+j} where **also the value of j is replaced** by the value of the sampled z .

Example of random coalition

- Input instance x : Gender=F, age=30, nationality=IT, income=20k
- Random instance z : Gender=M, age=35, nationality=IT, income=40k

- Random features values: **age=30, income=20k**

To compute $\phi_{j=gender=Female}(v)$ we consider

- x_{+j} - Gender=F, **age=35**, nationality=IT, **income=40k**
- x_{-j} - Gender=M, **age=35**, nationality=IT, **income=40k**

Note that the considered coalition S in this case is nationality=IT

Approximated Shapley estimation

M = number of iterations

- For m in $1, \dots, M$:
 - Sample a random instance z from the dataset
 - Randomly select a permutation of the feature values O
 - Compute x_{+j} and x_{-j}
 - x_{+j} = take values from x preceding j -th in O + j + take values from z succeeding j -th in O
 - x_{-j} = take values from x preceding j -th in O + take values from z succeeding j -th-1 in O (i.e., j -th included)
 - Compute the marginal contribution $\phi_j^m = f(x_{+j}) - f(x_{-j})$
- Compute the Shapley values as the average over the M iterations $\phi_j(x) = \frac{1}{M} \sum_{m=1}^M \phi_j^m$

Example of random coalition (2) – use permutations

- Input instance x : Gender=F, age=30, nationality=IT, income=20k
- Random instance z : Gender=M, age=35, nationality=IT, income=40k

- Random permutation at iteration m : **nationality**, gender, age, income

To compute $\phi_{j=gender=Female}^m(v)$ we consider

- x_{+j} - **nationality=IT**, Gender=F, age=35, income=40k
- x_{-j} - **nationality=IT**, Gender=M, age=35, income=40k

Note that the considered coalition S in this case is nationality=IT

Advantages and limitations of the approximation

Advantages

- Same as exact Shapley
- +
- Reduced computational complexity

Limitations

- Same as exact Shapley except for exponential computation
- +
- It is an approximation
- Variance for the estimate of the Shapley value depends on the number of iterations M
 - Decreasing M reduces computation time but increases the variance of the Shapley value

SHAP

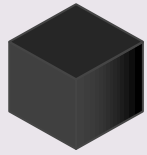
- SHAP = SHapley Additive exPlanations
- Local, model agnostic explanation method based on Shapley values
- SHAP proposes two approaches for estimating the Shapley values
 - KernelSHAP - kernel-based estimation approach
 - Model agnostic
 - TreeSHAP - efficient estimation approach for tree-based models
 - Not model agnostic
- It also propose to aggregate local explanation to provide global insights

SHAP

- Idea of SHAP brings is to represent the Shapley value explanation as an additive feature attribution method, a linear model
- Use the notion of surrogate model

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

SHAP



f : the original model to explain

x : instance to explain. We want to explain $f(x)$



e.g.,
 $x' = [1, 1, \dots, 1]$

x' : simplified input of x , such that $x = h_x(x')$ where h_x is a mapping function

As the interpretable representation of LIME

SHAP

e.g., #patches, #tokens

$z' = [0, 1, \dots, 1]$

$h_x(z')$



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

M : number of simplified features

$h_x(\cdot)$: mapping function that maps from simplified inputs to the original one

g : simpler explanation model, **interpretable approximation** of the original model

SHAP target defining a g such that: $g(x') \approx f(h_x(x'))$

SHAP - SHapley Additive exPlanations

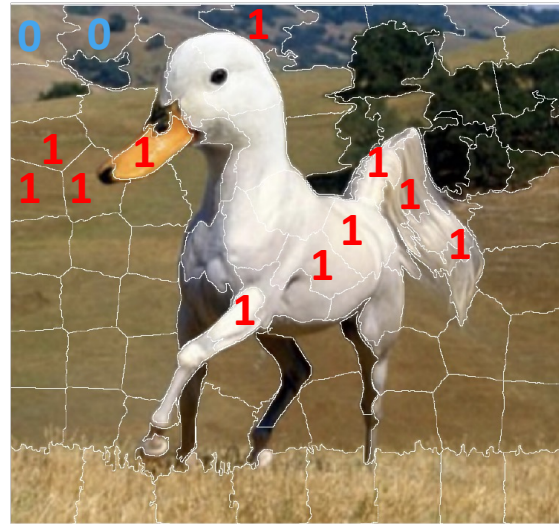
- M : number of simplified features
- $x' \in \{0,1\}^M$: defined as binary feature
 - As the interpretable representation of LIME
- Definition of **additive feature attribution method**

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

Example for image data



Original input



$$z' = [0, 0, 0, 1, 0, \dots, 0, 1, \dots]$$

Simplified input for SHAP computation



$$h_x(z')$$

Remapped input

SHAP - SHapley Additive exPlanations

- Recall the interpretable representation of LIME for text data

Welcome	to	the	Explainable	and	Trustworthy	AI	Course
1	1	0	0	1	0	0	1

- We use $x' \in \{0,1\}^M$ to model the coalitions of players!
 - An entry of 1 means that the corresponding feature value is “present” and 0 that it is “absent”
 - In the example, the player are the tokens/words ‘Welcome’, ‘to’, ‘and’ and ‘Course’. The other words are omitted/removed
 - Recall that to compute Shapley values, we simulate that only some feature values are playing (“present”) and some are not (“absent”).

Properties of additive feature attributions

Local accuracy

- The explanation model $g(x')$ matches the original model $f(x)$ when $x = h_x(x')$

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

If we define $\phi_0 = E[f(X)]$

$$f(x) = \phi_0 + \sum_{i=1}^M \phi_i x'_i = E[f(X)] + \sum_{i=1}^M \phi_i x'_i$$

Properties of additive feature attributions

Missingness

- Simplified input x model the presence and absence of feature
- Missingness requires that features missing have no impact, i.e., get an attribution 0
- $x'_i = 0 \implies \phi_i = 0$

Properties of additive feature attributions

Consistency

- If a model changes so that the **marginal contribution of a simplified input increases or stays the same regardless of the other feature, the attribution of that feature should not decrease** (hence, it should increase or stay the same as well)

Let $f_x(z') = f(h_x(z'))$ and $z' \setminus i$ indicate that $z'_i = 0$ (i.e., feature i is absent/removed).

For any two models f and f' , if

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i)$$

For all $z' \in \{0,1\}^M$, then

$$\phi_i(f', x) \geq \phi_i(f, x)$$

SHAP

- The **Shapley value** is the only one possible explanation model g that follows the definition of 'Additive feature attribution methods' and the three Properties

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

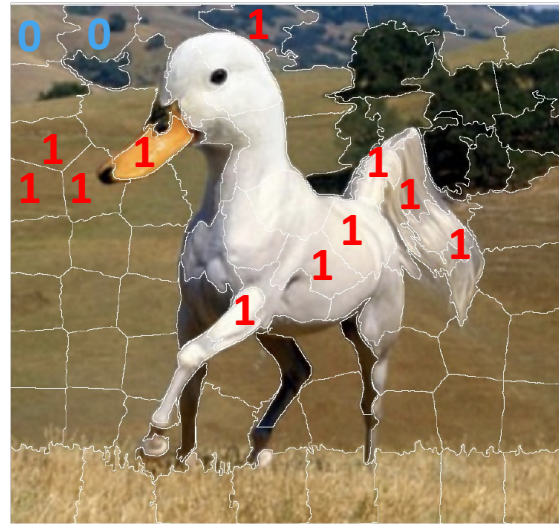
Where $|z'|$ is the number of non-zero values in z' and $z' \subseteq x'$ indicates all z' where non-zero entries are a subset of the non-zero entries in x'

SHAP - Components

- **Definition of z'**
 - Coalition sampling
 - Depends on the type of data

- **Definition of $f_x(z')$**

Example of coalition sampling – image data



$$z'_k = [0, 0, 0, 1, 0, \dots, 0, 1, \dots]$$



$$h_x(z')$$

Example of coalition sampling - tabular

- x : Gender=F, age=30, nationality=IT, income=20k
- $z' = [1, 0, 1, 0]$
- $h_x(z') = \text{Gender=F, age=55, nationality=IT, income=20k}$
 - $f([z_S, E[z_{\bar{S}}]])$
 - marginalizing, substituting the feature not in S (\bar{S}) with values for a random instance (actually done by marginalizing over multiple random instances, less variance)

Example of coalition sampling – text data

- x : Welcome to the Explainable and Trustworthy AI Course
- $z' = [1, 1, 0, 0, 1, 0, 0, 1]$

Welcome	to	the	Explainable	and	Trustworthy	AI	Course
1	1	0	0	1	0	0	1

- $h_x(z')$ = Welcome to [UNK] [UNK] and [UNK] [UNK] Course

SHAP - $f_x(z')$

- $f_x(z') = f(h_x(z')) = E[f(z)|z_S]$

where

- S : set of non-zero values in z'
- z_S : instance that has **missing** values for features **not in S**
 - i.e., non-zero just the ones in S
- We approximate $f(z_S)$ with $E[f(z)|z_S]$ by marginalizing the features not in S

$$S = \{a_1, a_3\}$$

$$z_S' = [0, 1, \dots, 1]$$

$h_x(z')$

$f($



)

SHAP - $f_x(z')$ - Estimation

$$f_x(z') = f(h_x(z')) = E[f(z)|z_S] =$$

$$= E_{z_{\bar{S}}|z_S}[f(z)]$$

expectation over $z_{\bar{S}}|z_S$

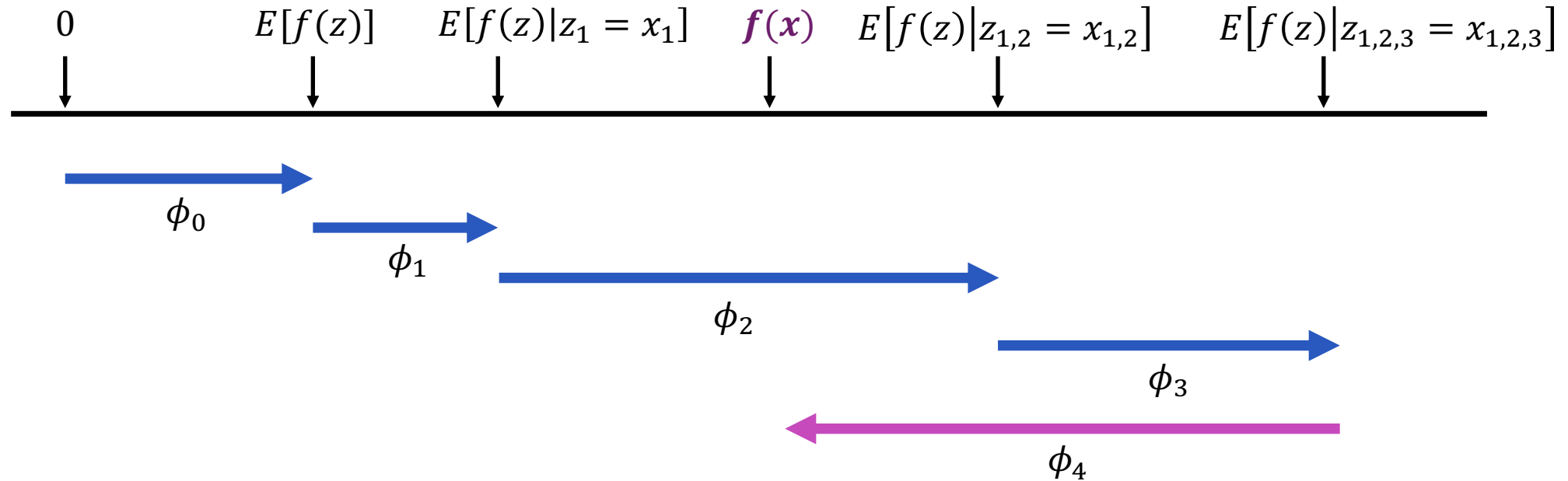
where \bar{S} is the complement of S (i.e., features not in S)

$$\approx E_{z_{\bar{S}}}[f(z)]$$

assume feature independence

$$\approx f([z_S, E[z_{\bar{S}}]])$$

SHAP - ϕ_i interpretation



- $E[f(z)]$ is the base value, when we did not know any features
- The diagram shows a single ordering (1 then 2 then 3)
 - The weights takes into account the multiple ordering --> weighted average t

Kernel SHAP

- The exact computation of the Shapley values is expensive
 - Exponential in the number of players

KernelSHAP is a technique to estimate for an instance x the contributions of each feature value to the prediction as an approximation of the Shapley value - ϕ_i

It uses the intuition of **local interpretable models**

From LIME to Kernel SHAP

Recall the objective of LIME

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

The explanation for instance x is the model g that minimizes loss L :

- $L(f, g, \pi_x)$ – how unfaithful g is to f in the locality given by π_x - **Local**

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

Ridge regression - Linear least squares with l2 regularization in the code

- $\Omega(g)$ – i.e., the complexity of g - how the model is interpretable is kept low – **Interpretable**

Shapley Kernel

- Interpretability term

$$\Omega(g) = 0$$

- Locality term

$$L(f, g, \pi_x) = \sum_{z' \in \mathcal{Z}} \pi_{x'}(z') (f(h_x(z')) - g(z'))^2$$

- Where $\pi_{x'}(z')$ is computed as follows

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

$|z'|$ is the number of non-zero elements in z and M is the maximum coalition size (i.e., the number of interpretable features)

- $h_x(z')$ is the function to go from the interpretable feature space of z' to the original feature space to compute $f(h_x(z'))$ as model f operates on the original space

Kernel SHAP - Pseudocode

- Sample K coalitions $z'_k \in \{0,1\}^M$, $k \in \{1, \dots, K\}$
 - $z'_{k,j}$ has value 1 if the feature j is present in the coalition, 0 if it is absent
- Compute $f(h_x(z'_k))$, i.e., the prediction of model f first converting the instance z'_k to the original feature space with function h_x

- Compute the weight for each coalition z'_k with SHAP kernel

$$\pi_{x'}(z'_k) = \frac{M - 1}{\binom{M}{|z'_k|} |z'_k| (M - |z'_k|)}$$

- Fit weighted linear model

$$L(f, g, \pi_x) = \sum_{z' \in \mathcal{Z}} \pi_{x'}(z') (f(h_x(z')) - g(z'))^2$$

- The coefficient of the linear model ϕ_i are the (approximated) Shapley values

Global insights – SHAP Feature importance

Global feature importance as the average of absolute Shapley values per feature across the data

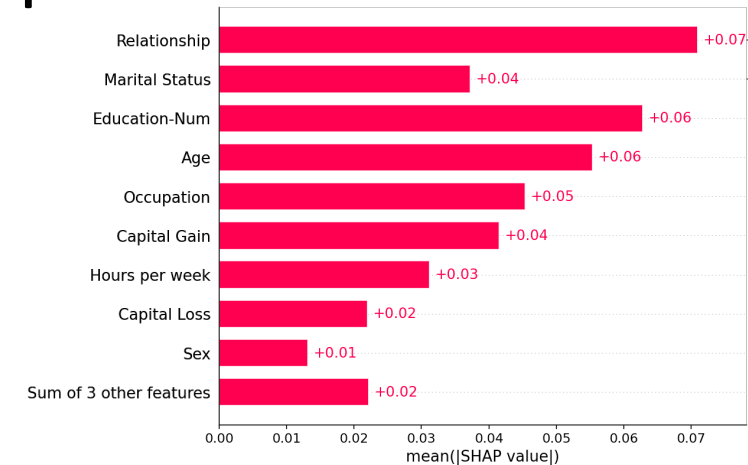
Steps

- Compute SHAP for each instance of the datasets
 - For each attribute, we have the importance of its value for explaining a specific instance
- Compute the average the absolute Shapley values per feature across the data:

$$I_j = \frac{1}{n} \sum_{i=1}^n |\phi_j^{(i)}|$$

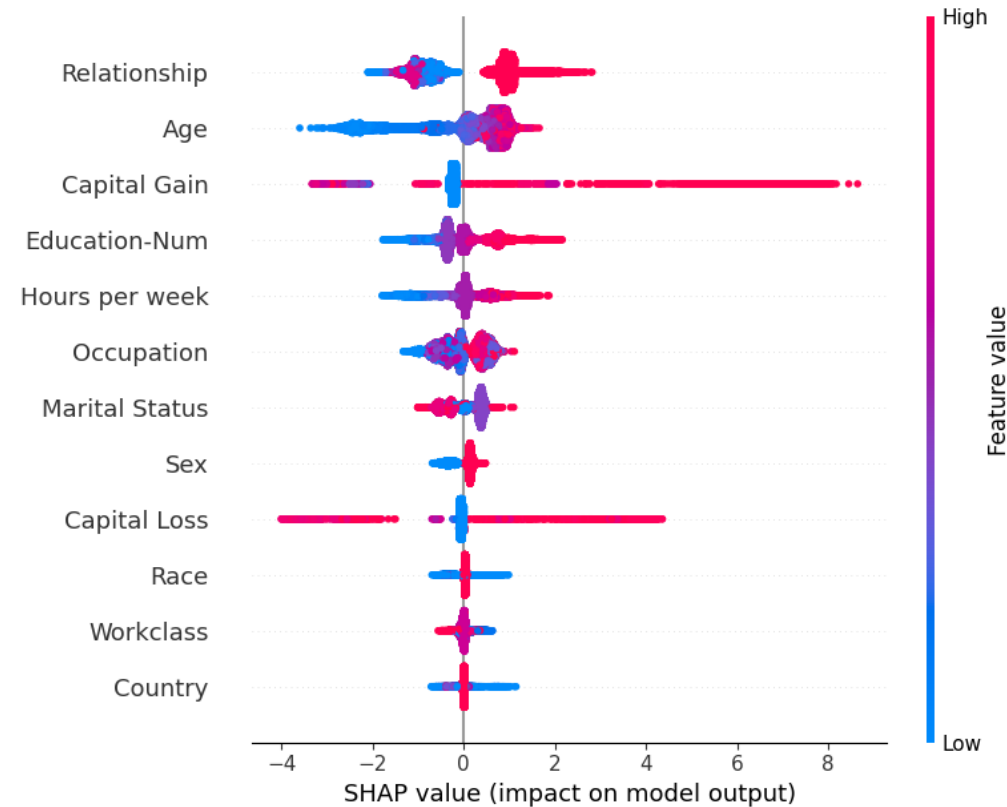
whew $\phi_j^{(i)}$ is the ϕ of the attribute j for the i -th instance and n is the number of instances

The higher the value of a feature, the more the features is important for model f



Global insights – SHAP Summary plot

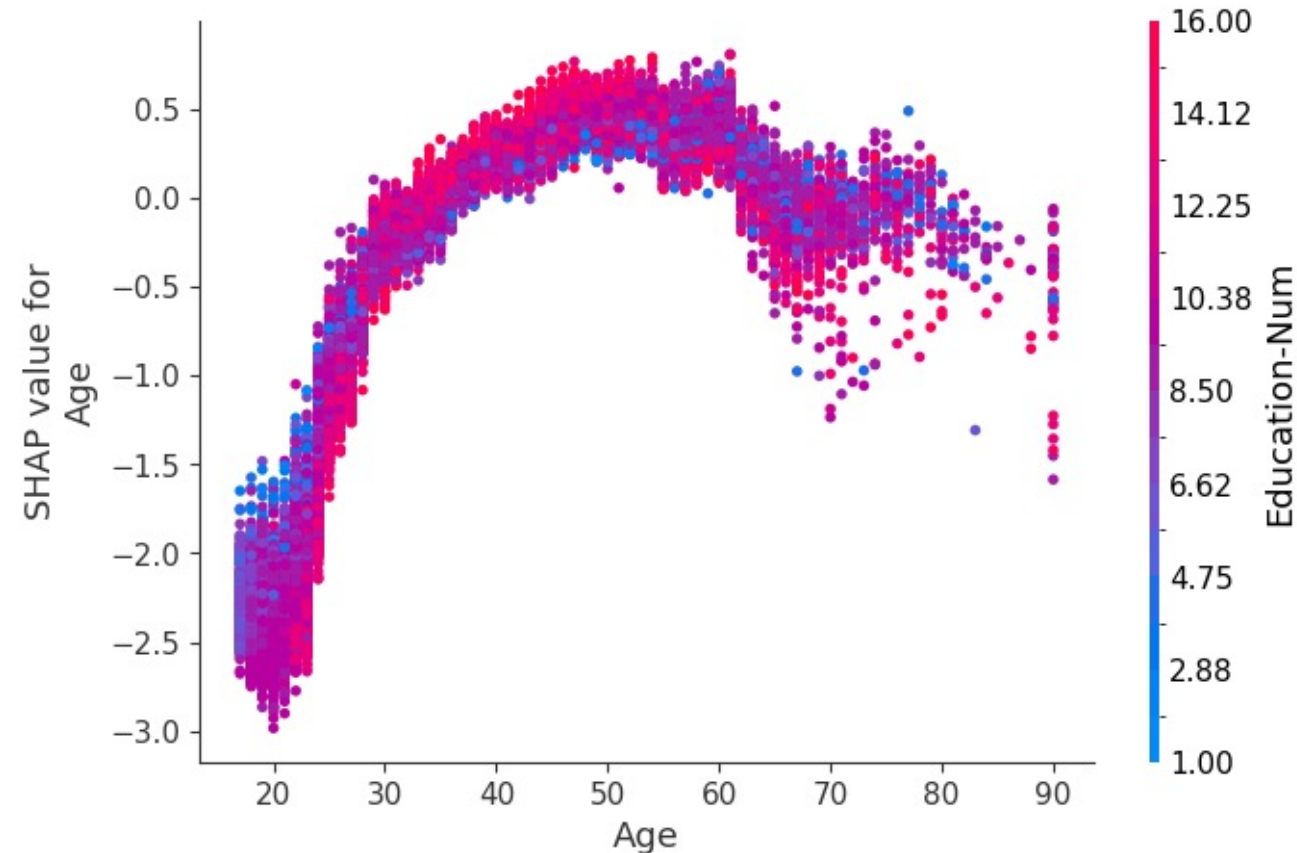
- **Density scatter plot**
- Each point is a Shapley value for a feature and an instance
 - We compute SHAP for all instances first
- x-axis - the Shapley value
- y-axis - the feature
- Features are sorted by the sum of the SHAP value magnitudes across all samples.



Global insights – SHAP Dependence Plot

Dependence Plot

- Given a feature j , for each data instance i we plot a $\{x_j^{(i)}, \phi_j^{(i)}\}$ with
 - the feature value on the x-axis
 - the corresponding Shapley value on the y-axis
- We can also color the point based on another feature k to highlight possible interactions with other features



Advantages of SHAP

- Same as exact Shapley
 - Model agnostic & Local explanations & Provides feature attributions & Direct interpretation
- Consider interaction among features
- Efficiency Property of Shapley Values
 - The difference between prediction for instance x and average prediction is distributed among feature values
- Solid Theoretical Foundation
 - Axioms: efficiency, symmetry, linearity, null player
- +
- **Reduced computational complexity**
- **Global model interpretations**
- **Nice and well documented library**

Limitations of SHAP

- Need access data to compute $f(h_x(z'))$
- Perturbations/info removals may create unrealistic data
 - Assumption feature independence
- The absence of a feature to compute $f(h_x(z'))$ is simulated by drawing random instances
→ variance on the estimation
- It is an approximation of Shapley values
- KernelSHAP is slow
 - But other faster approximation have been proposed and are available in the library

Unification of removal-based explanations

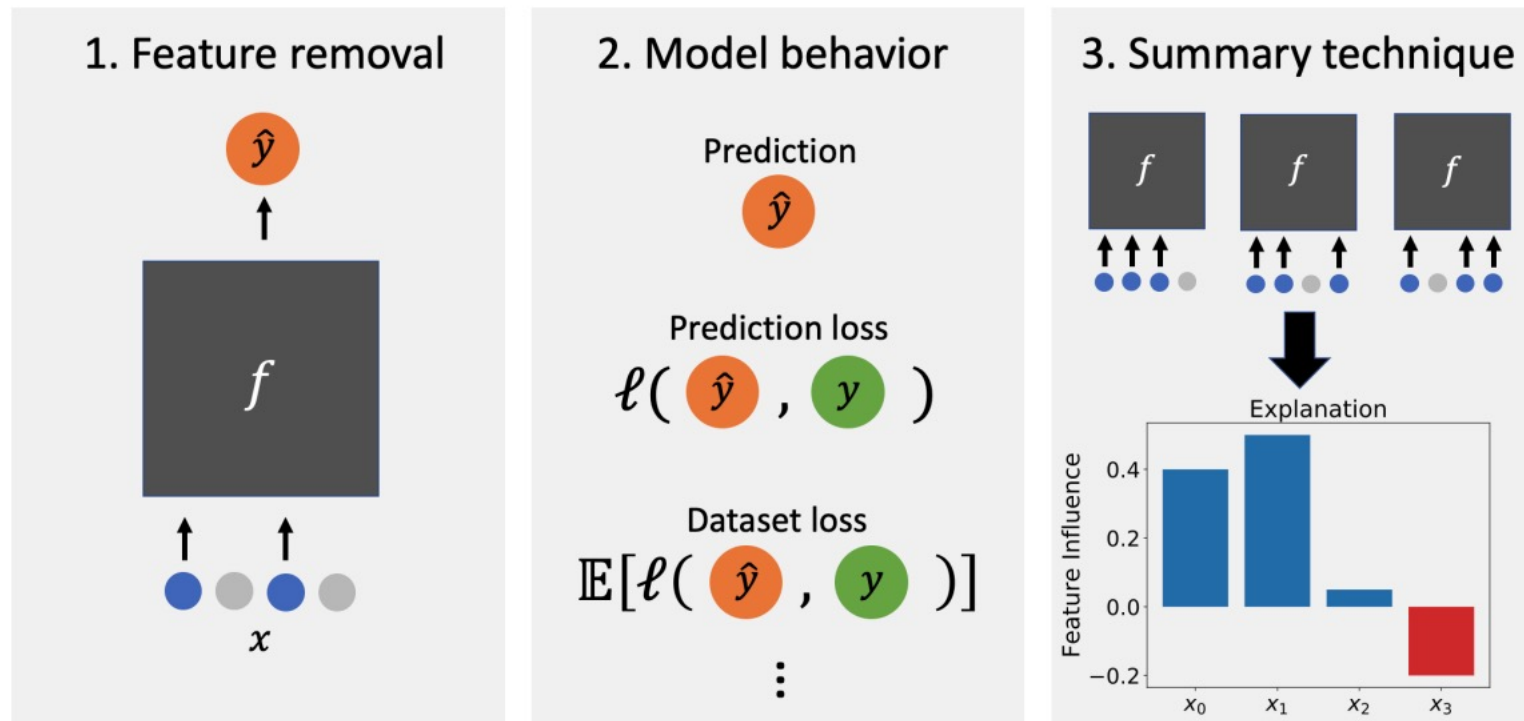
- Removal-based explanations are based on the principle of simulating feature removal to quantify the influence of each feature to the prediction
- We can characterize them according three dimensions
 - 1) Feature removal**

How the method removes features (e.g. by setting them to default values, or by marginalizing over a distribution of values)
 - 2) Model behavior**

What model behavior the method explains (e.g., the probability of the true class, or the model loss)
 - 3) Summary technique**

How the method summarizes each feature's influence (e.g., by removing a feature individually, or by calculating Shapley values)

Unification of removal-based explanations



1) Feature removal $f(x \setminus \bar{S})$ - examples

- **Zeroing**

- Remove features simply by setting them to zero:

$$f(x \setminus \bar{S}) = f(x_S, 0)$$

- **Default values**

- Remove features by setting them to user-defined default values such as gray pixels for images or a mean value (e.g., in LIME for image). Given default values $r \in X$, calculate:

$$f(x \setminus \bar{S}) = f(x_S, r_{\bar{S}})$$



1) Feature removal $f(x \setminus \bar{S})$ - examples

- **Blurring**

- Remove features from images by blurring them with a Gaussian kernel

blur



- **Marginalize with marginal**

- Removes features by marginalizing them out using their joint marginal distribution (e.g., KernelSHAP):

$$f([z_S, E[z_{\bar{S}}]])$$

2) Model behavior - examples

- **Prediction probability**

$$f(\mathbf{x}) = p(y = c|\mathbf{x})$$

- **Prediction loss**

Consider the true label y for an input x and calculate the prediction loss using a loss function

- **Dataset loss**

Consider the expected loss across the entire dataset, i.e., quantify how much the model's performance degrades when different features are removed

3) Summarizing feature influence - examples

- **Remove individual**

Calculate the impact of removing a single feature from the model (e.g., PredDiff)

Note that this can be applied for interpretable representations (e.g., super-pixels)

- **Additive model**

Fit a regularized additive linear model to a dataset of perturbed examples (e.g., LIME). The learned coefficients represent the incremental value associated with including each feature.

- **Shapley value**

Calculate feature attributions using the Shapley value

Recap of the limitations of removal-based techniques

- Perturbations/info removals may create unrealistic data
- The absence of a feature is often simulated by drawing random instances → variance on the estimation
- The removal is often based on marginalization/mean values, thus requiring accessing to the data

References

- Robnik-Šikonja, Marko, and Igor Kononenko. "Explaining classifications for individual instances." TKDE (2008)
- Shapley, Lloyd S. "A value for n-person games." Contributions to the Theory of Games, 1953
- Štrumbelj, Erik, and Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions." Knowledge and information systems, 2014
- Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." Neurips 2017
- Covert, Ian, Scott Lundberg, and Su-In Lee. "Explaining by removing: A unified framework for model explanation." *Journal of Machine Learning Research*, 2021