Prediction explanation with Shapley values

(Prediction explanation = Local model explanation)

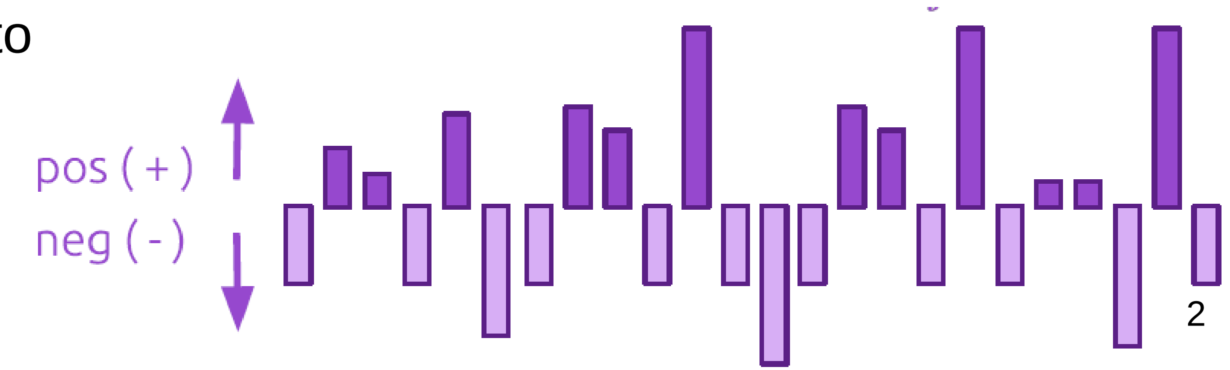
Martin Jullum (jullum@nr.no),

Skatteetaten, 29.10.2021

Prediction explanation – by example

- Car insurance
  - Response y: The insured crashes
  - Features $\boldsymbol{x} = (x_1, \dots, x_M)$: Data about the insured, his/her car and crashing history
  - Predictive model f: Model trained to predict probability of crash: $f(\boldsymbol{x}) \approx \Pr(y = yes|\boldsymbol{x})$

- Prediction explanation
  - Why did a guy with features $\boldsymbol{x}^*$ get a predicted probability of crashing equal to $f(\boldsymbol{x}^*)$= 0.3?



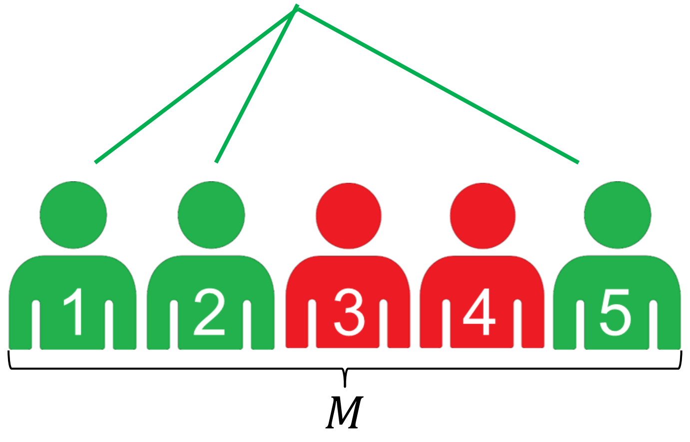
Shapley values

- Concept from (cooperative) game theory in the 1950s
- Used to distribute the total payoff to the players
- Explicit formula for the “fair” payment to every player j:

$$\phi_j = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|!\,(|M| - |S| - 1)}{|M|!} \big(v(S \cup \{j\}) - v(S)\big)$$

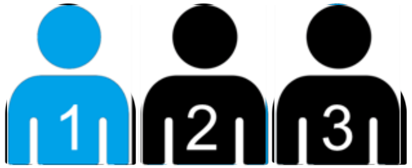
$v(S)$ is the payoff with only players in subset S

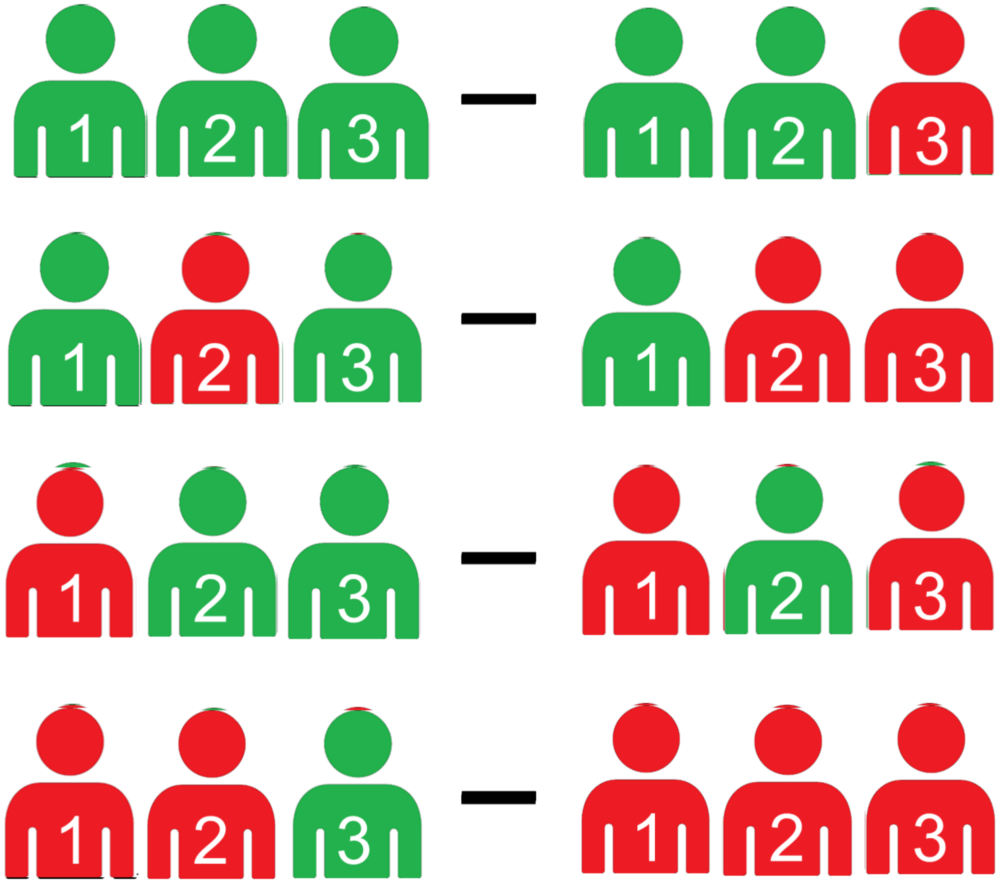
- Several mathematical optimality properties



Intuition behind the Shapley formula

Game with 3 players

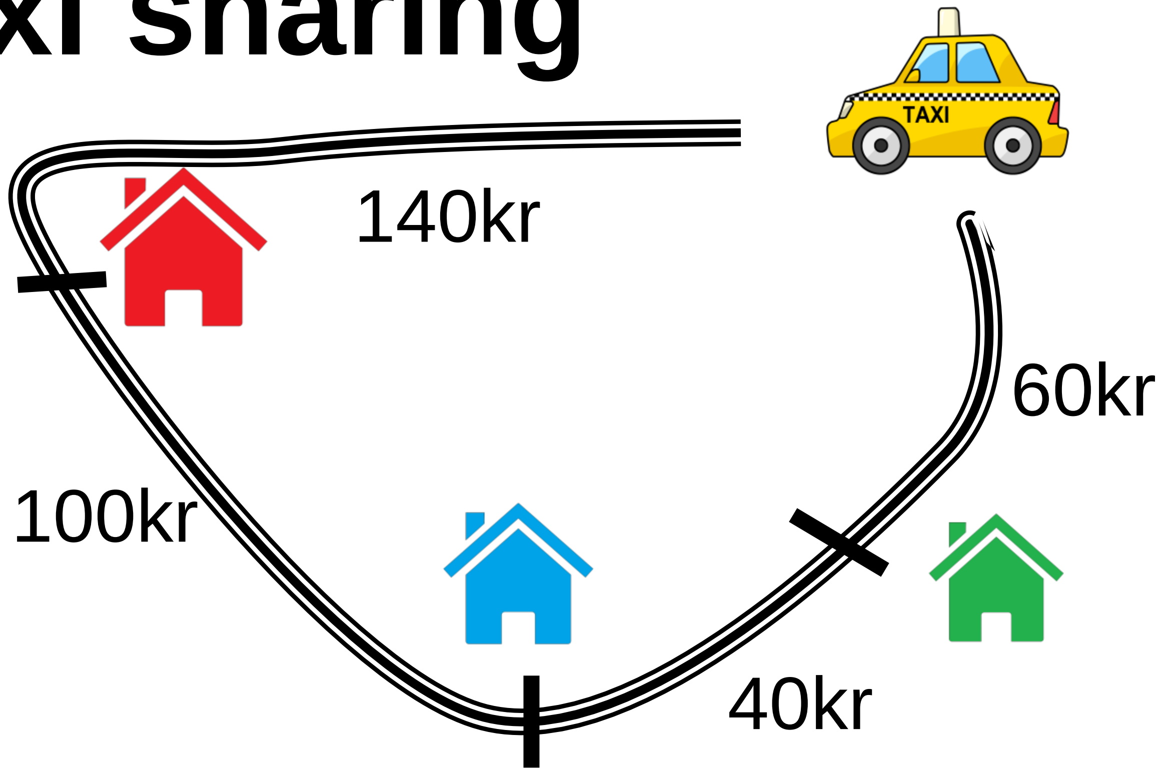




Shapley values for taxi sharing

$v(\{R, B, G\}) = 60 + 40 + 100 = 200kr$
$v(\{\ \}) = 0$

$v(\{R\}) = 140kr$
$v(\{B\}) = 60 + 40 = 100kr$
$v(\{G\}) = 60kr$
$v(\{R, B\}) = 60 + 40 + 100 = 200kr$
$v(\{R, G\}) = 60 + 40 + 100 = 200kr$
$v(\{B, G\}) = 60 + 40 = 100kr$



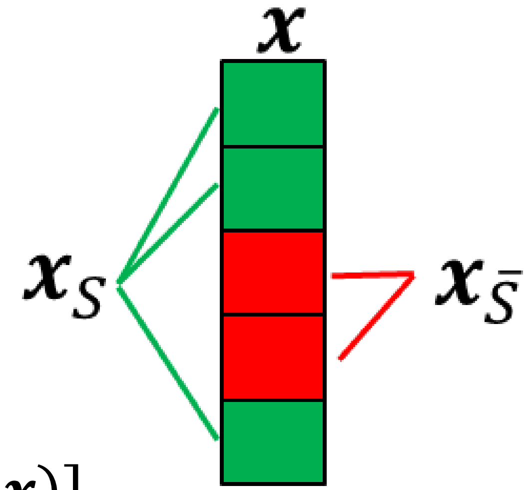
$$\phi_R = \frac{1}{3}\left(v(\{R,B,G\}) - v(\{B,G\})\right) + \frac{1}{6}\left(v(\{R,B\}) - v(\{B\})\right) + \frac{1}{6}\left(v(\{R,G\}) - v(\{G\})\right) + \frac{1}{3}\left(v(\{R\}) - v(\{\ \})\right) = 120\text{kr}$$

$$\phi_B = \frac{1}{3}\left(v(\{R,B,G\}) - v(\{R,G\})\right) + \frac{1}{6}\left(v(\{R,B\}) - v(\{R\})\right) + \frac{1}{6}\left(v(\{B,G\}) - v(\{G\})\right) + \frac{1}{3}\left(v(\{B\}) - v(\{\ \})\right) = 50\text{kr}$$

$$\phi_G = \frac{1}{3}\left(v(\{R,B,G\}) - v(\{R,B\})\right) + \frac{1}{6}\left(v(\{R,G\}) - v(\{R\})\right) + \frac{1}{6}\left(v(\{B,G\}) - v(\{B\})\right) + \frac{1}{3}\left(v(\{G\}) - v(\{\ \})\right) = 30\text{kr}$$

Shapley values for prediction explanation

- Approach popularised by Lundberg & Lee (2017)
  - Players = features ($x_1, \dots, x_M$)
  - Payoff = prediction ($f(\boldsymbol{x}^*)$)
  - Contribution function: $v(S) = E[f(\boldsymbol{x})|\boldsymbol{x}_S = \boldsymbol{x}_S^*]$
  - Properties



$$\sum_{j=1}^{M} \phi_j = f(\boldsymbol{x}^*) - \phi_0 \qquad\qquad \phi_0 = E[f(\boldsymbol{x})]$$

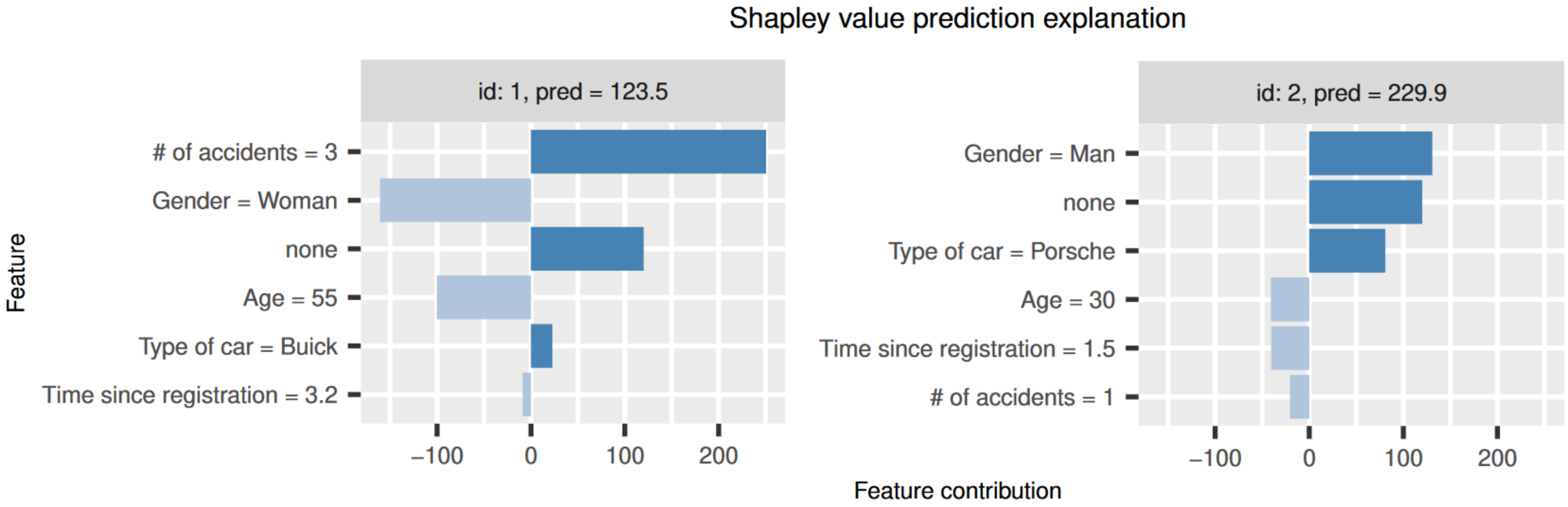
$f(\boldsymbol{x}) \perp\!\!\!\perp x_j$ implies $\phi_j = 0$

x_i, x_j same contribution implies $\phi_i = \phi_j$

- Rough interpretation of ϕ_j: **The prediction change when you don't know the value of** x_j – averaged over all features

Example of Shapley value explanation

- Consider a model $f(x)$ trained to predict the price of a car insurance based on the following features x:
  - Owner's age, owner's gender, type of car, time since the car was registered, number of accidents the last 5 years

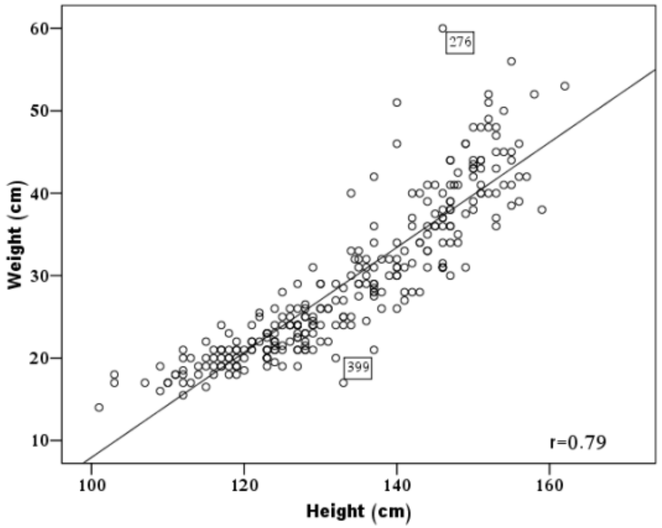


Linear models $f(x) = \beta_0 + \sum_{j=1} \beta_j x_j$

- Linear model with independent covariates:

$$\phi_j = \beta_j(x_j^* - E[x_j]), \quad \phi_0 = \beta_0 + \textstyle\sum_j \beta_j E[x_j]$$

- Explanation not simple with dependent covariates!
  - Example
    - x_1 = height (cm)
    - x_2 = weight (kg)
    - Y = PB in high jump (cm)
  - Model 1: $Y = 100 + 2x_1 - 2x_2$
  - Model 2: $Y = 100 - 2x_1 + 2x_2$



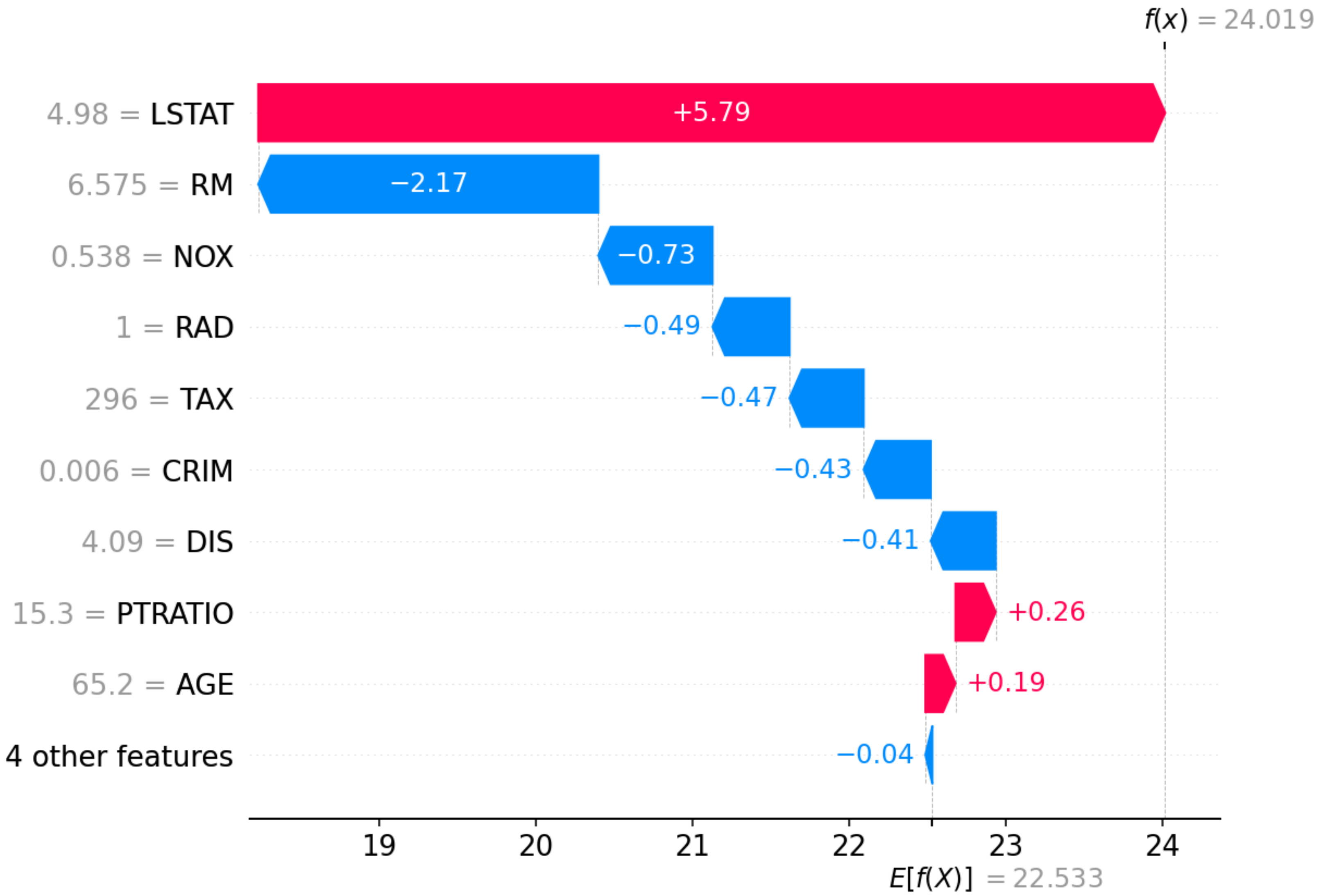
- Shapley values gives $\phi_1 \approx \phi_2$ in such a setting

Visualization/summary of Shapley value explantaions

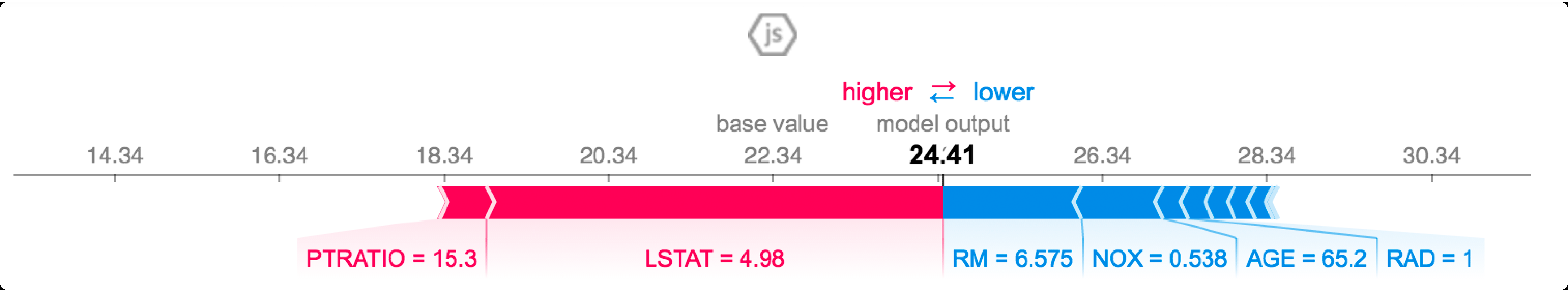
- Consider $f(\boldsymbol{x})$ trained to predict housing prices in Boston based on 16 features $\boldsymbol{x}$, including
  - LSTAT - % lower status of the population
  - RM - average number of rooms per dwelling
  - NOX - nitric oxides concentration (parts per 10 million)
  - RAD - index of accessibility to radial highways
  - TAX - full-value property-tax rate per $10,000
  - CRIM - per capita crime rate by town

- Next slides shows visualizations from the *shap* Python package

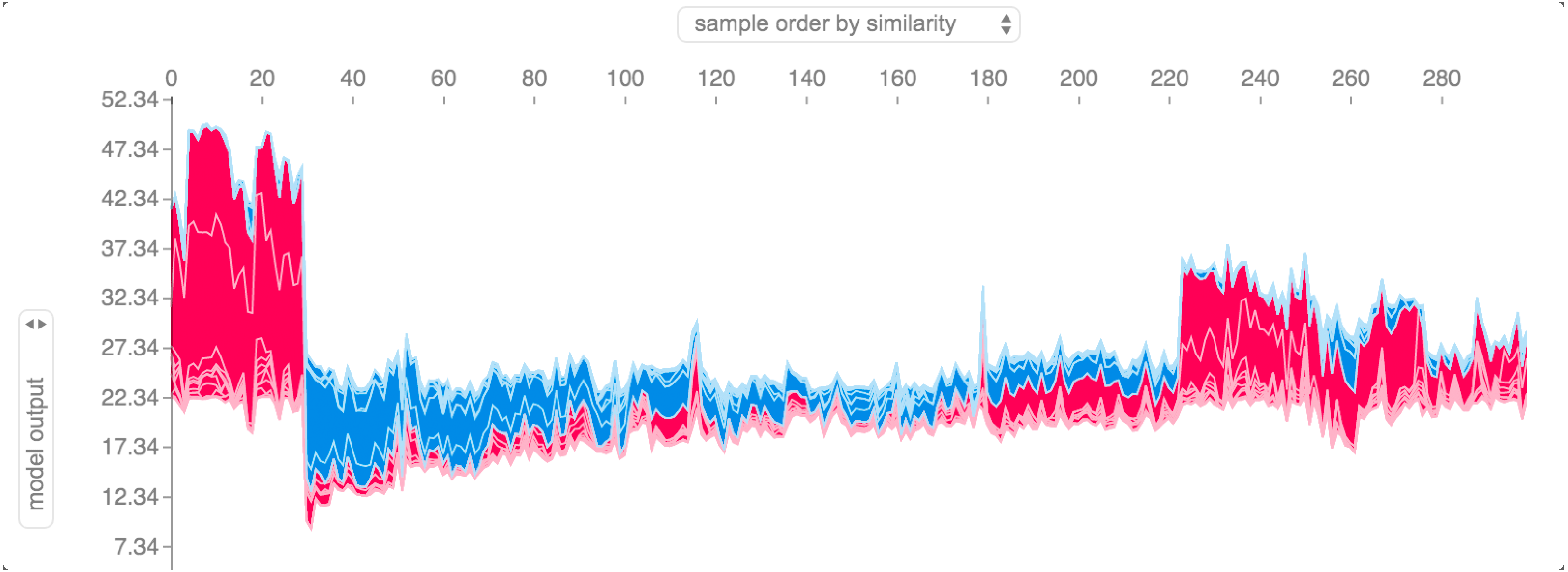
Visualization/summary of Shapley value explantaions



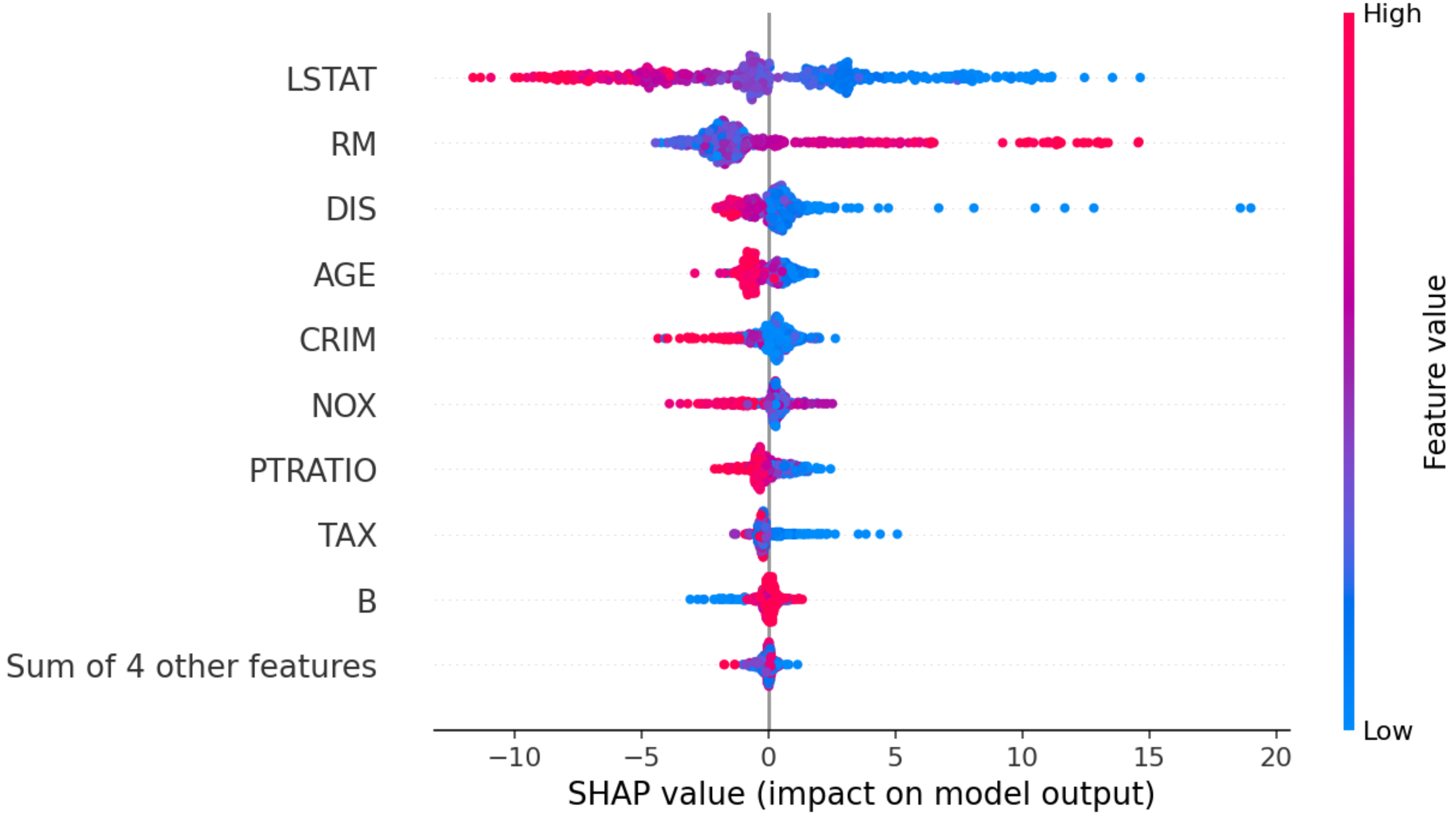
Visualization/summary of Shapley value explantaions



Visualization/summary of Shapley value explantaions



Visualization/summary of Shapley value explantaions



Two challenges with Shapley values for prediction explanation

1. The exponentially growing computational complexity in the Shapley formula

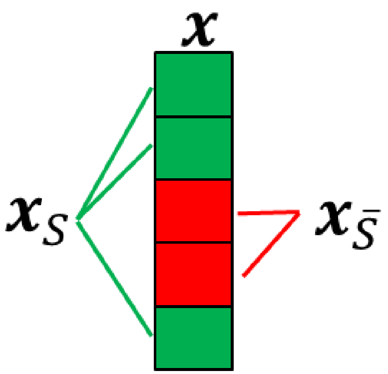
$$\phi_j = \sum_{S \subseteq M \setminus \{j\}} \frac{|S|!\,(|M| - |S| - 1)}{|M|!} \big(v(S \cup \{j\}) - v(S)\big)$$

- Approximate solutions may be obtained by cleverly reducing the sum by subset sampling (KernelSHAP; Lundberg & Lee, 2017)

2. Estimating the contribution function

$$v(S) = E[f(\boldsymbol{x}) | \boldsymbol{x}_S = \boldsymbol{x}_S^*] = \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S) p(\boldsymbol{x}_{\bar{S}} | \boldsymbol{x}_S = \boldsymbol{x}_S^*) \mathrm{d}\boldsymbol{x}_{\bar{S}}$$

Recall



- Lundberg & Lee (2017), Python shap package, uses the approximation

$$v(S) \approx \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S^*) p(\boldsymbol{x}_{\bar{S}}) \mathrm{d}\boldsymbol{x}_{\bar{S}}$$

This implicitly assumes the features are **independent**!

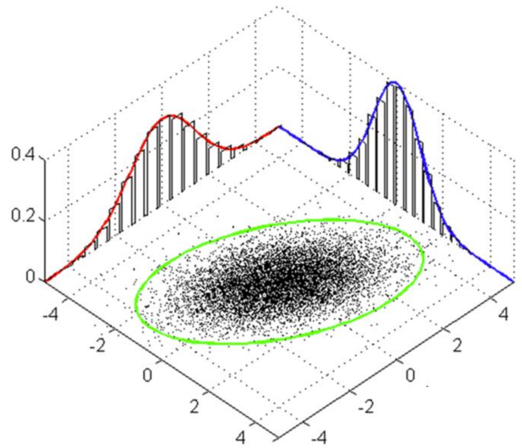
Consequences of the independence assumption

- Requires evaluating $f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S)$ at potentially **unlikely or illegal** combinations of $\boldsymbol{x}_{\bar{S}}$ and $\boldsymbol{x}_S$

- Example 1
  - Number of transactions to Switzerland: 0
  - Average transaction amount to Switzerland: 100 €

- Example 2
  - Age: 17
  - Marital status: Widow
  - Profession: Professor

NR/Big Insight work on Shapley values



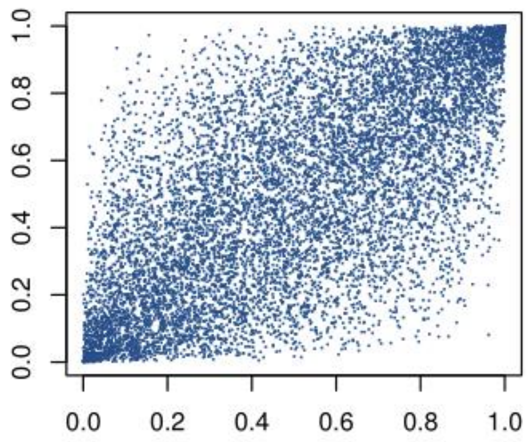
- *Dependence-aware* approaches to estimate

  $v(S) = E[f(\boldsymbol{x})|\boldsymbol{x}_S = \boldsymbol{x}_S^*]$ properly

- We do this by estimating $p(\boldsymbol{x}_{\bar{S}}|\boldsymbol{x}_S = \boldsymbol{x}_S^*)$ properly

- Several alternative methods
  - Gaussian distribution
  - Empirical nonparametric method
  - Empirical margins + vine copulas to estimate dependence structure
  - Conditional inference trees (ctree)
  - Variational autoencoders with arbitrary conditioning (VAEAC)

- Methods implemented in the ***shapr*** R-package



Nice to know

- Independence approach (most common)
  - There are different “explainers” in the **shap** Python package
    - General purpose, tree based models, deep learning, NLP
  - If you are using Shapley values produced directly by the GBM libraries *xgboost, lightgbm, catboost*, you are using the tree based approach in **shap**
- Independence vs dependence-aware approaches in practice
  - Consider $f(x_1, x_2) = x_1, \quad cor(x_1, x_2) = \rho \neq 0$
  - Independence approach will give $\phi_2 = 0$
  - Dependence-aware approach will give $\phi_2 \neq 0$
- Dependence aware approaches
  - Comes at a higher computational cost
  - May give different results depending on what dependence-estimation method you use

Nice to know II

- Be careful when using and interpreting Shapley values from the independence approach
  - May be useful for pure debugging/investigation of how $f(\cdot)$ behaves
- Dependence-aware approach should be used in practical applications, as explanations of individual predictions (where feature dependence needs to be obeyed)
- Some authors have claimed the independence approach is the right one referring to causal inference, but this has recently been rejected by a more general causal inference perspective (Heskes et al., 2020)

BigInsight