WHY AI NEEDS MATHS AND STATS - lessons from working in a CS field

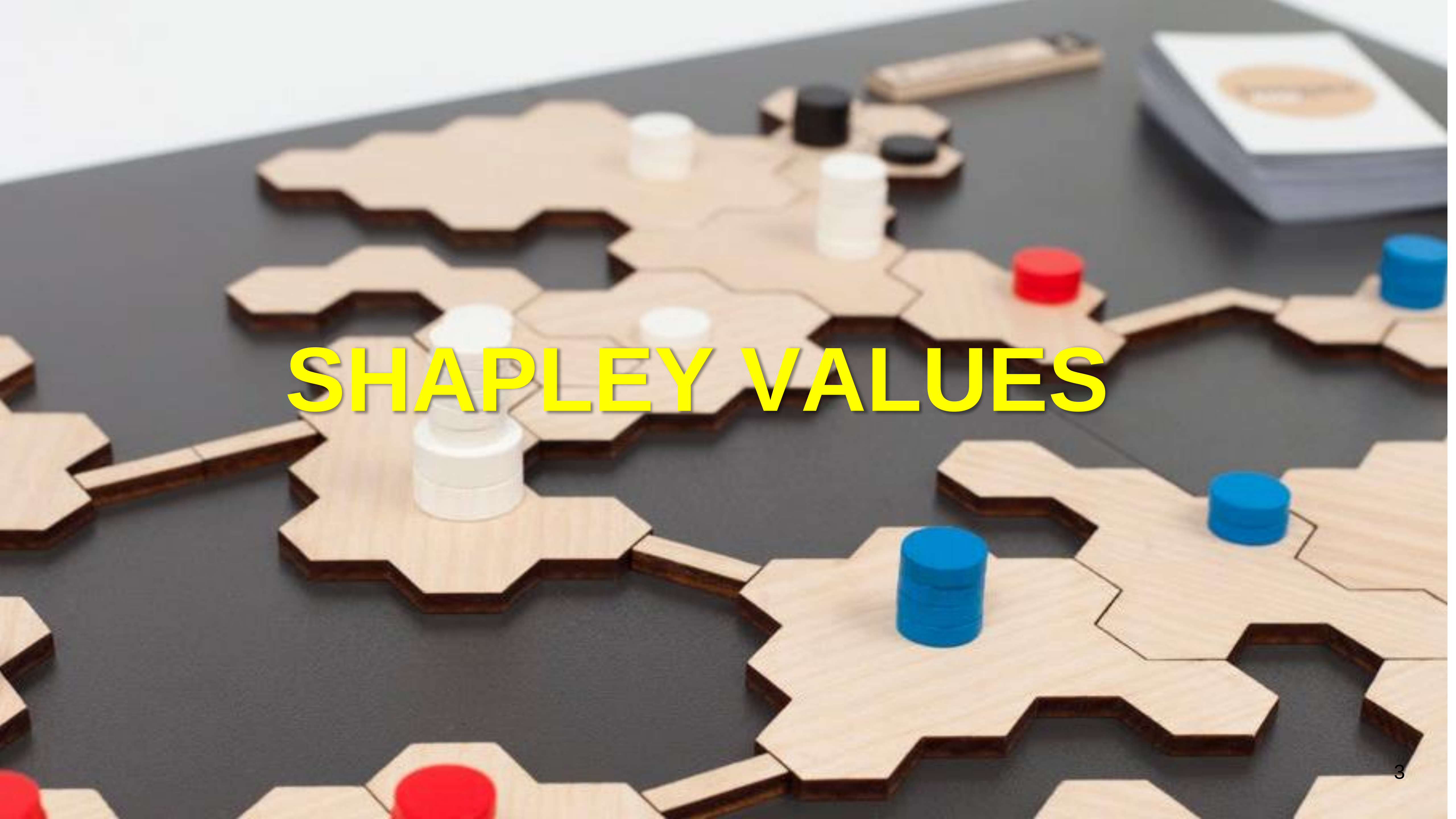
Martin Jullum

Godt Hjort, Dec 5th 2023

Explainable AI (XAI)

- Understanding what black box models do
- A field largely driven by computer scientists

- Me and colleagues
  - Work in the subfield restricted to tabular data – i.e. regression: y = f(x1,x2,x3)
  - Try to use our statistical mindset to improve/repair the methodology in the field



SHAPLEY VALUES

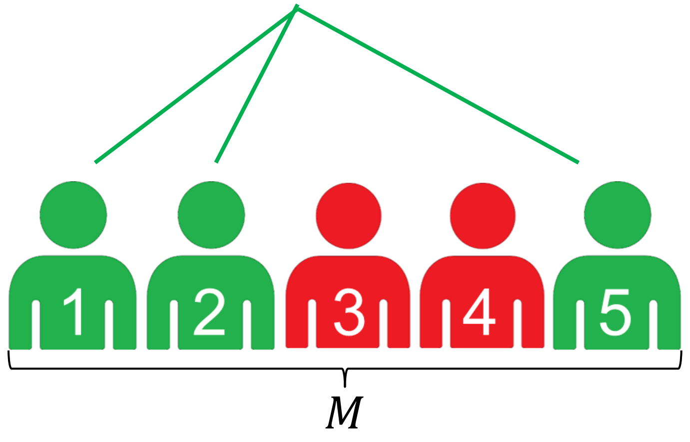
Shapley values

- 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:

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

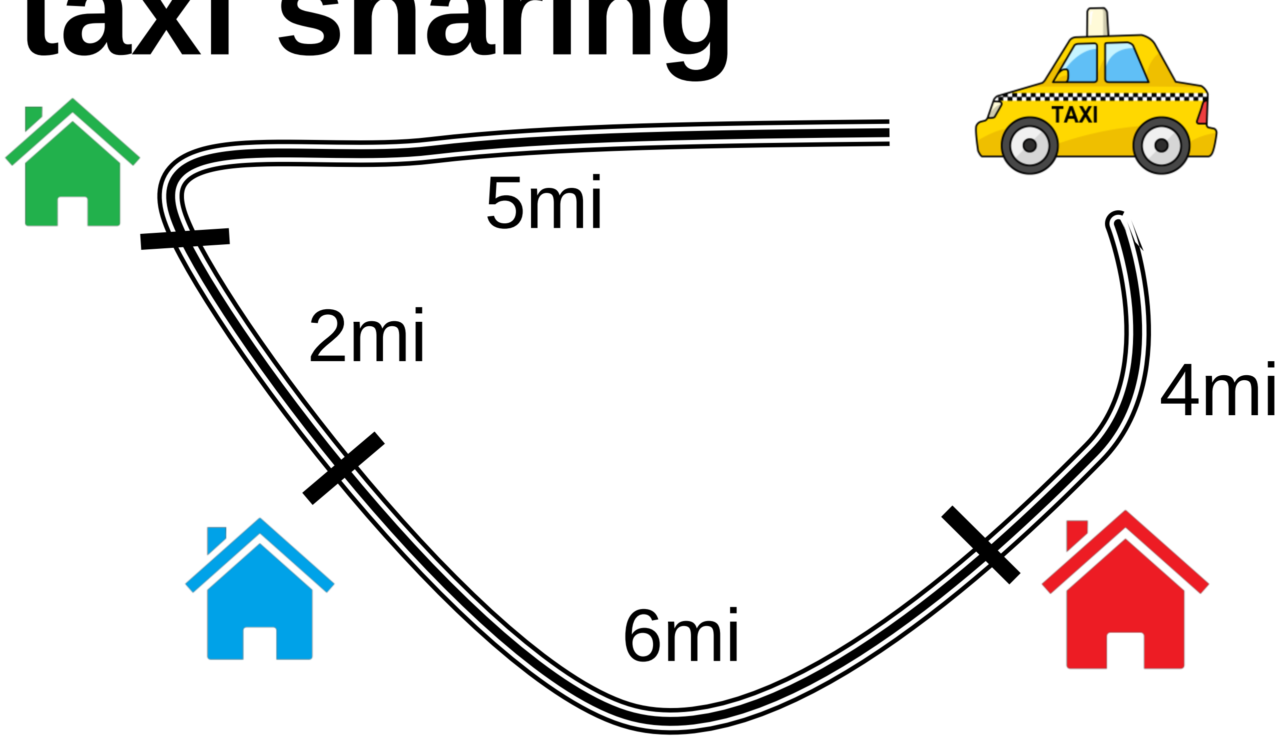
$v(S)$ is the payoff with only players in subset S

- Several mathematical optimality properties



Shapley values for taxi sharing

Costs: $3/mi



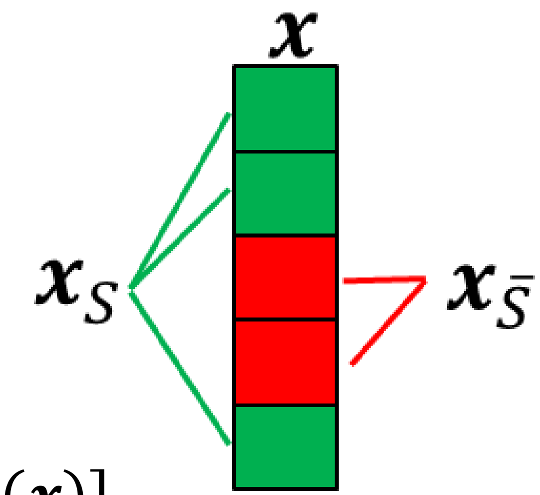
$v(\{R,B,G\}) = (4+6+2)mi * \$3 = \$36$
$v(\{\ \}) = \$0$

$v(\{R\}) = 4mi * \$3 = \$12$
$v(\{B\}) = (5+2)mi * \$3 = \$21$
$v(\{G\}) = 5mi * \$3 = \$15$
$v(\{R,B\}) = (4+6)mi * \$3 = \$30$
$v(\{R,G\}) = (4+6+2)mi * \$3 = \$36$
$v(\{B,G\}) = (5+2)mi * \$3 = \$21$

$$\phi_R = \frac{1}{3}\big(v(\{R,B,G\}) - v(\{B,G\})\big) + \frac{1}{6}\big(v(\{R,B\}) - v(\{B\})\big) + \frac{1}{6}\big(v(\{R,G\}) - v(\{G\})\big) + \frac{1}{3}\big(v(\{R\}) - v(\{\ \})\big) = \$14$$
$$\phi_B = \frac{1}{3}\big(v(\{R,B,G\}) - v(\{R,G\})\big) + \frac{1}{6}\big(v(\{R,B\}) - v(\{R\})\big) + \frac{1}{6}\big(v(\{B,G\}) - v(\{G\})\big) + \frac{1}{3}\big(v(\{B\}) - v(\{\ \})\big) = \$11$$
$$\phi_G = \frac{1}{3}\big(v(\{R,B,G\}) - v(\{R,B\})\big) + \frac{1}{6}\big(v(\{R,G\}) - v(\{R\})\big) + \frac{1}{6}\big(v(\{B,G\}) - v(\{B\})\big) + \frac{1}{3}\big(v(\{G\}) - v(\{\ \})\big) = \$11$$

Shapley values for prediction explanation

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



$$\phi_0 + \sum_{j=1}^{M} \phi_j = f(\boldsymbol{x}^*)$$

$$\phi_0 = E[f(\boldsymbol{x})]$$

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

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

- Interpretation of ϕ_j: **The prediction change caused by observing the value of** x_j – averaged over whether the other covariates were observed or not

Two main challenges

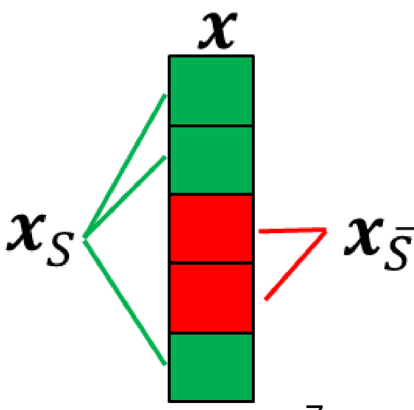
1. The computational complexity in the Shapley formula is of size 2^M

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

2. Estimating the contribution function

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

- Lundberg & Lee (2017)
  - Approximates $v(S) \approx \int f(\boldsymbol{x}_{\bar{S}}, \boldsymbol{x}_S^*) p(\boldsymbol{x}_{\bar{S}}) \mathrm{d}\boldsymbol{x}_{\bar{S}}$,
  - Estimates $p(\boldsymbol{x}_{\bar{S}})$ using the empirical distribution of the training data
  - Monte Carlo integration to solve the integral



This assumes the covariates are independent!

Our contribution

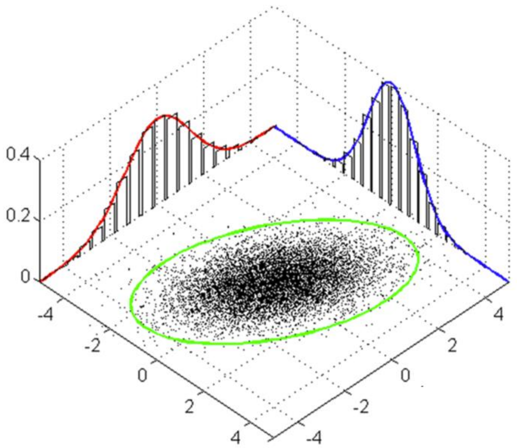
- *Dependence-aware* approaches to estimate

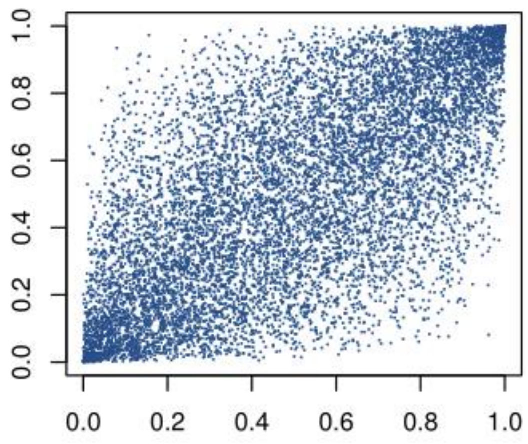
  $v(S) = E[f(\boldsymbol{x}) | \boldsymbol{x}_S = \boldsymbol{x}_S^*]$ properly

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

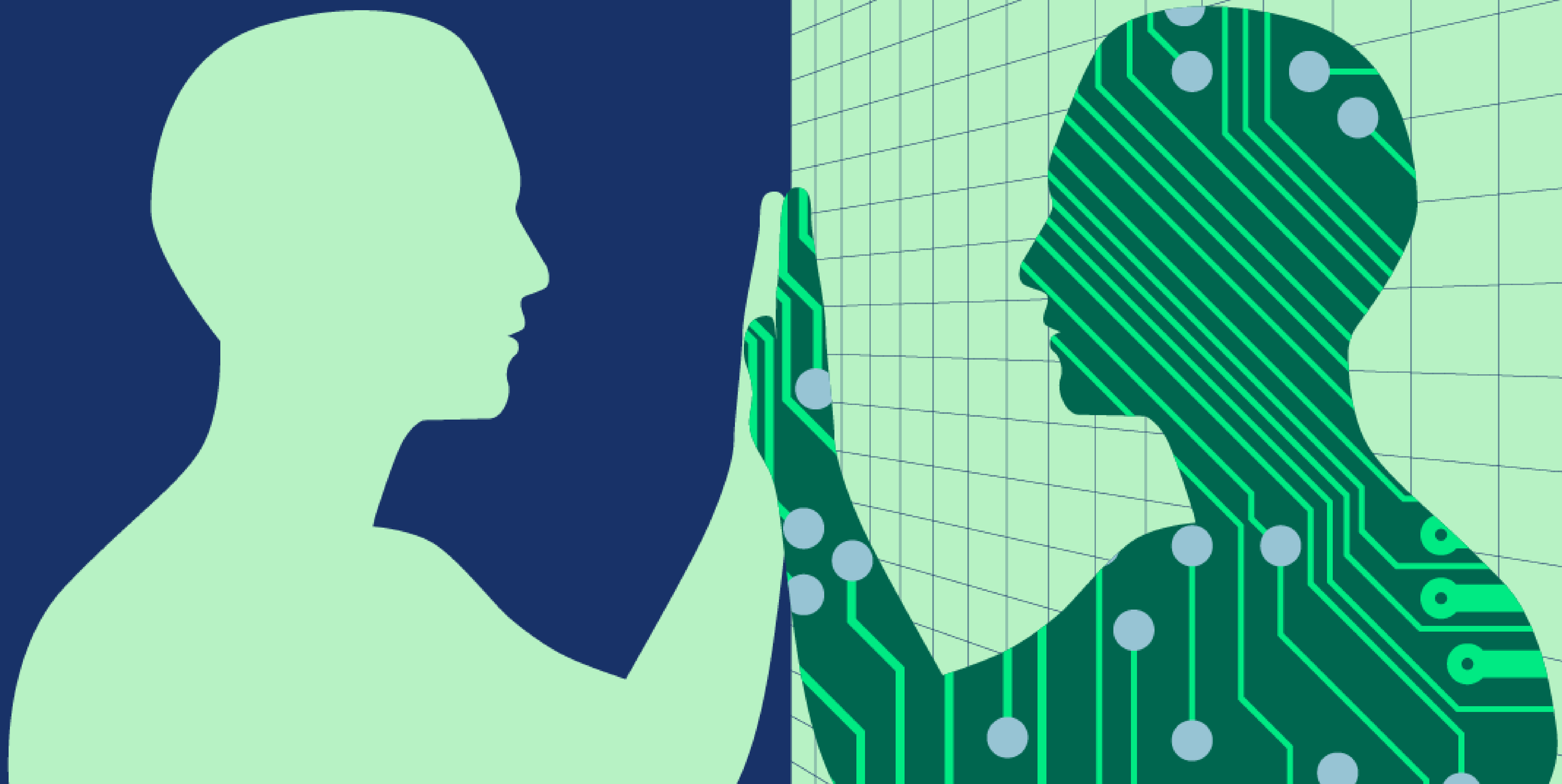
- Several alternative methods
  - Gaussian distribution
  - Empirical nonparametric method
  - Empirical margins + vine copulas to estimate dependence structure
  - Conditional inference trees (ctree)
  - Variational autoencoders with arbitrary conditioning (VAEAC)

  - Direct regression on $v(S) = E[f(\boldsymbol{x}) | \boldsymbol{x}_S = \boldsymbol{x}_S^*] \sim \boldsymbol{x}_S$
  - Common regression model for any $v(S)$ using masking trick





COUNTERFACTAL EXPLANATIONS



Example case

Automatic processing of loaning applications based on default prediction model

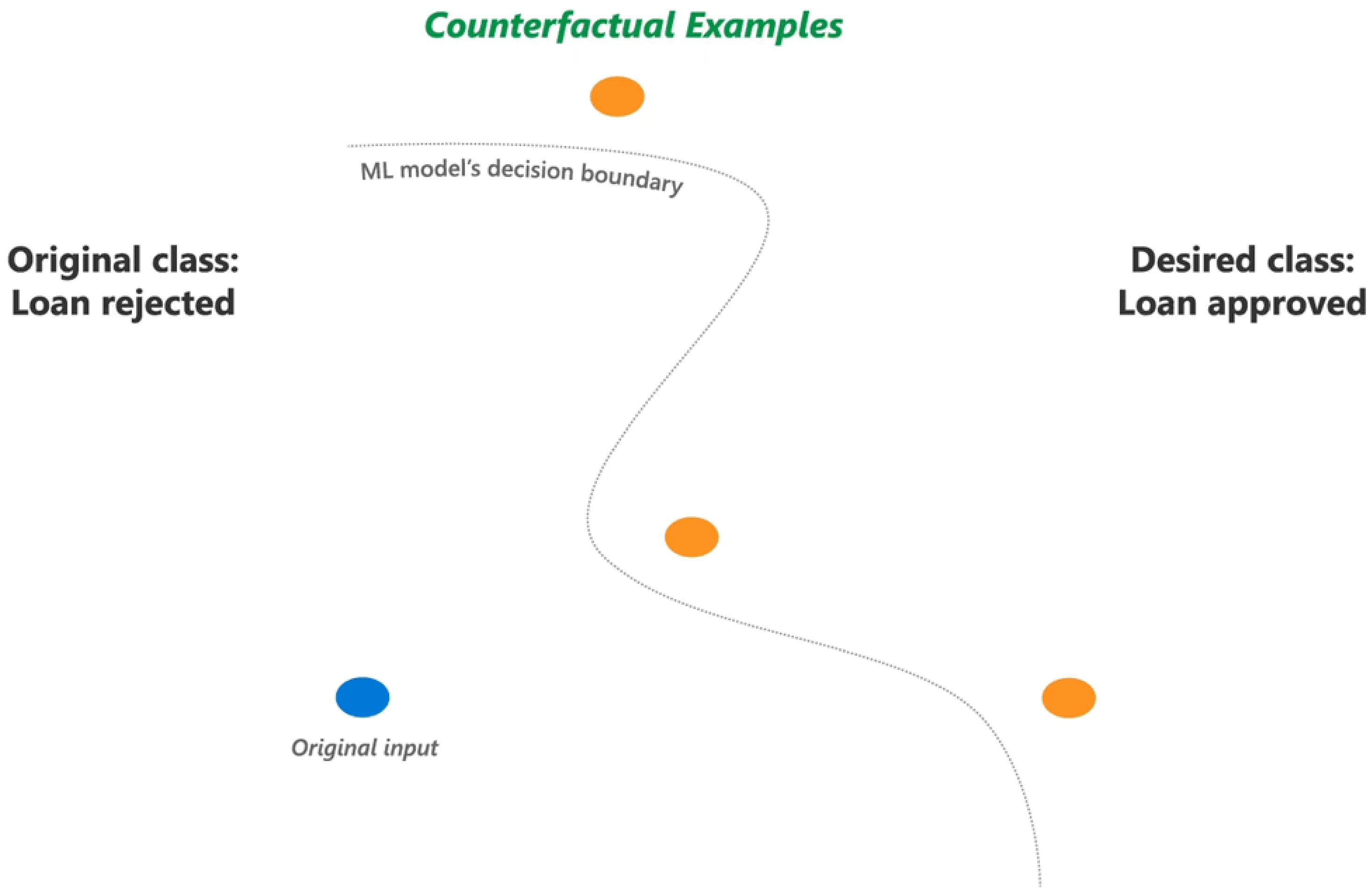
- Response y: Loan defaulted or not
- Covariates $\boldsymbol{x} = (x_1, \ldots, x_p)$: Info about the applicant, salary, previous defaults, transactions history, etc
- Fit regression model f: Model trained to predict probability of default:
$$f(\boldsymbol{x}) \approx \Pr(y = \text{default}|\boldsymbol{x})$$
- *Loan approved if* $f(\boldsymbol{x}) < c = 0.1$

CASE: Peter has features $\boldsymbol{x}^*$, and got his loan application rejected as $f(\boldsymbol{x}^*) = 0.2 > c$

Question: What can Peter do to receive a loan?

The idea

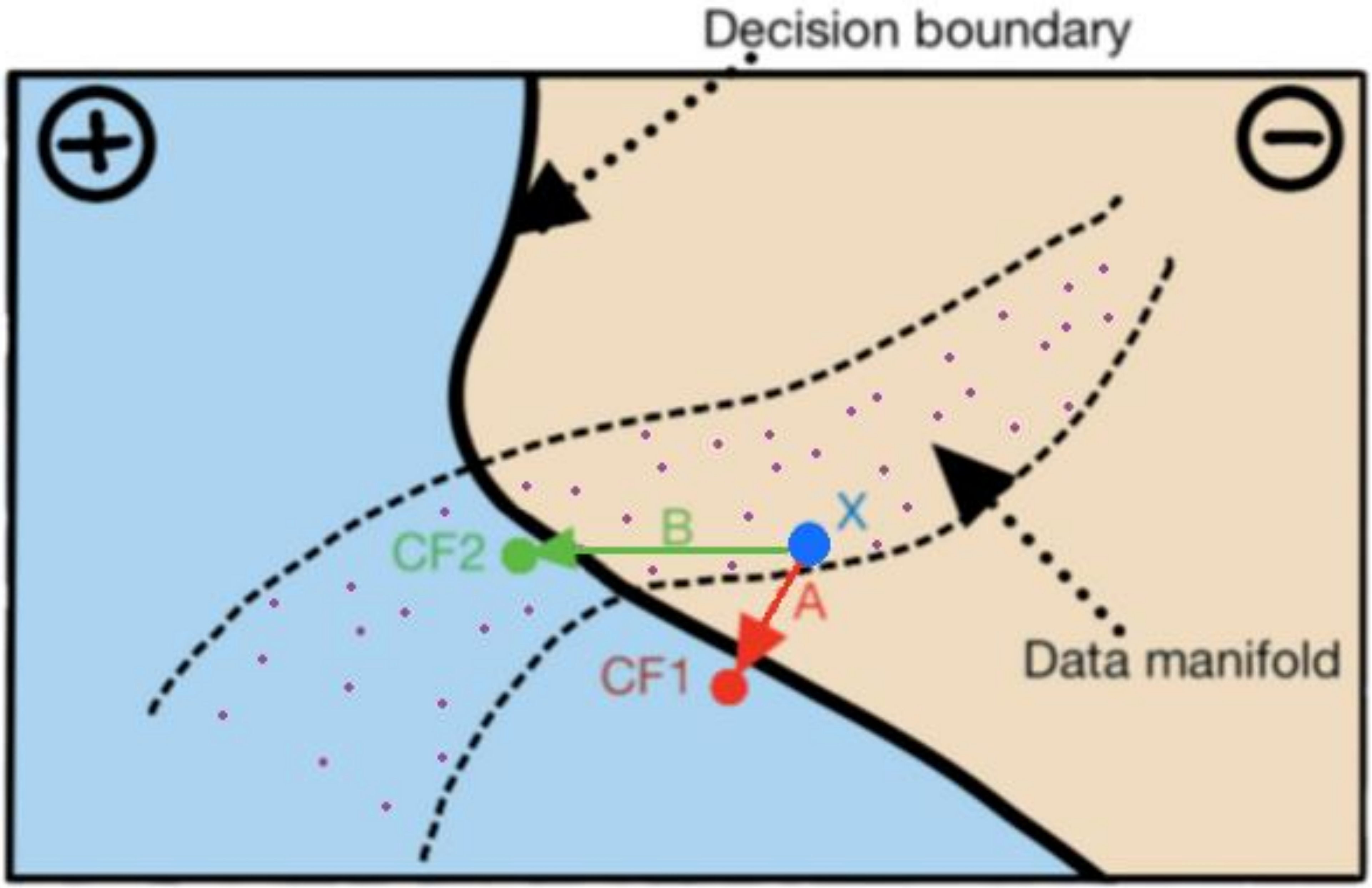
CE solution: Examples of (minimal) changes in covariates which approves the application



Criteria

Desired properties

1. On-manifold
2. Actionable
3. Valid
4. Low cost



Types of CE methods

Guidotti (2022)

Optimization based methods

- Minimize loss (wrt example ***e***) of the type $L_{\boldsymbol{x}^*}(\boldsymbol{e}) = \mathrm{dist}_1(f(\boldsymbol{e}), c) + \lambda \cdot \mathrm{dist}_2(\boldsymbol{x}^*, \boldsymbol{e})$
  - Often require differentiable *f*
  - Not necessarily on-manifold
  - Categorical covariates more troublesome

Heuristic search-based methods

- Optimization with heuristic search strategies

Instance-based methods

- Finds counterfactuals by searching for instances in a reference distribution/dataset

Our simple method: MCCE

MCCE: Monte Carlo sampling of valid and realistic counterfactual explanations

3-step procedure to produce a counterfactual example $\boldsymbol{e}$

1. **Model**: Model the joint distribution of mutable covariates, given the fixed covariates and *the decision*
2. **Generate**: Generate a large number K of samples from the modelled distribution with the specified fixed covariates $\boldsymbol{x}^{*f}$ and desired decision
3. **Post-process**: Discard the invalid samples, and choose the one "nearest" to $\boldsymbol{x}^{*}$

Step 1: Model

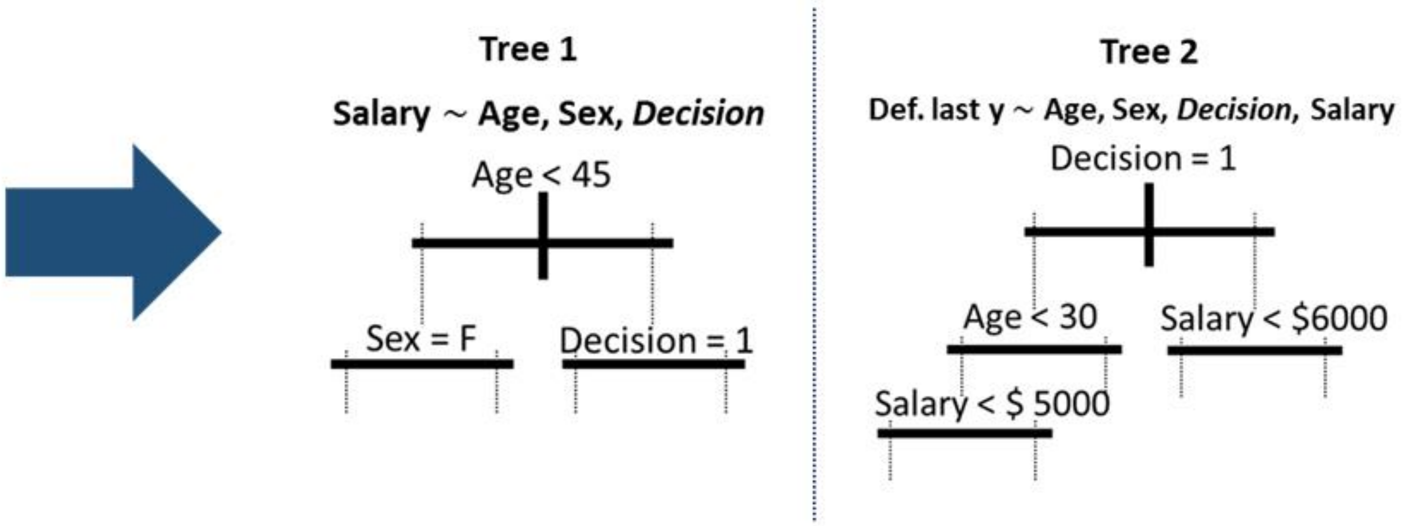
- Utilize the standard probability property:

$$p(\boldsymbol{X}^m \mid \boldsymbol{X}^f, Y') = p(X_1^m \mid \boldsymbol{X}^f, Y') \prod_{i=2}^{q} p(X_i^m \mid \boldsymbol{X}^f, Y', X_1^m, \ldots, X_{i-1}^m)$$

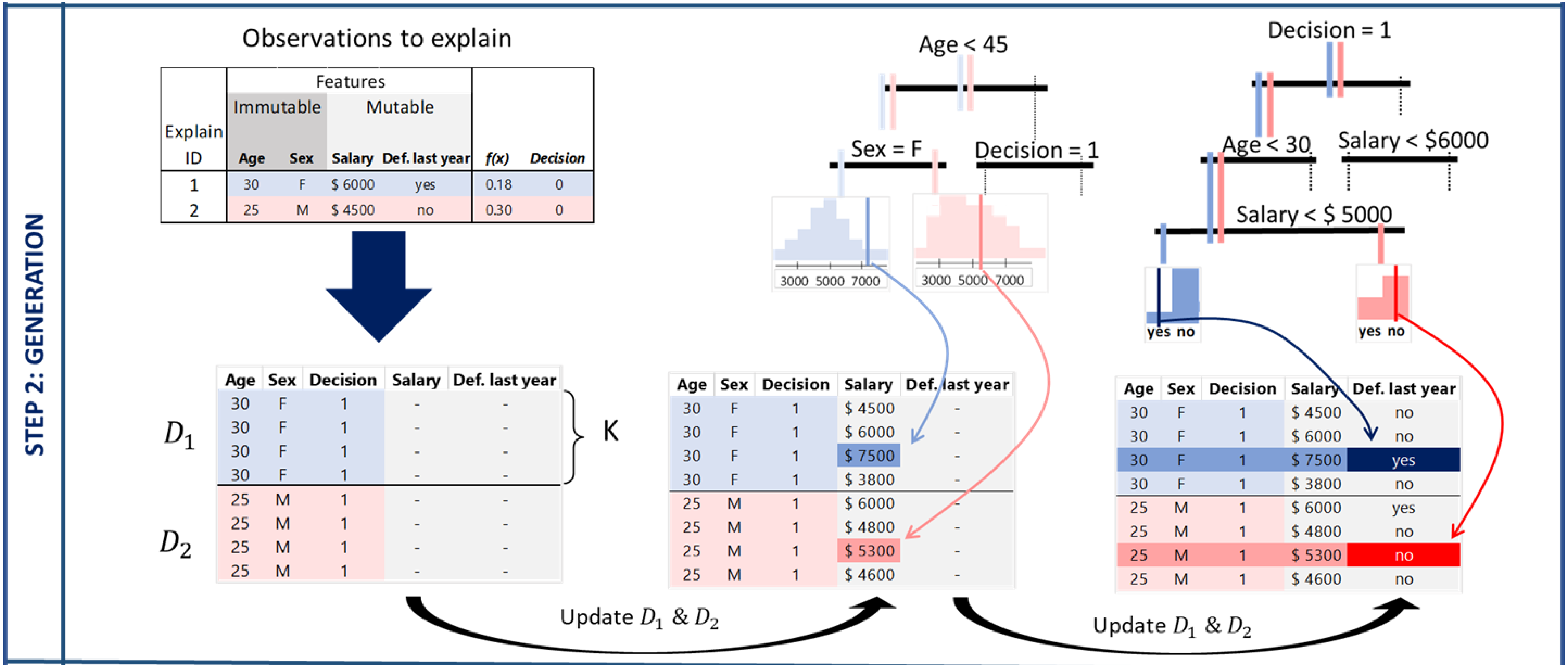
STEP 1: MODEL

Training data

Features					
Immutable		Mutable			
Age	Sex	Salary	Def. last year	f(x)	Decision
30	M	$ 3500	yes	0.24	0
28	F	$ 8000	no	0.12	0
42	M	$ 7500	no	0.04	1
27	F	$ 9500	yes	0.21	0
39	M	$ 5000	no	0.09	1
28	F	$ 4000	no	0.08	1
32	F	$ 7300	no	0.12	0
⋮	⋮	⋮	⋮	⋮	⋮
23	M	$ 4300	yes	0.31	0



Step 2: Generation



Step 3: Post-processing

STEP 3: POST-PROCESSING

Populated D_1 & D_2

Age	Sex	Decision	Salary	Def. last year
30	F	1	$ 4500	no
30	**F**	**1**	**$ 6000**	**no**
30	F	1	$ 7500	yes
30	F	1	$ 3800	no
25	M	1	$ 6000	yes
25	**M**	**1**	**$ 4800**	**no**
25	M	1	$ 5300	no
25	M	1	$ 4600	no

Compute performance metrics

Sequentially discard samples

f(x)	valid	L0	Gower
0.08	1	2	*0.6*
0.07	**1**	**1**	**0.5**
0.09	1	1	0.8
0.07	1	2	*0.7*
0.05	1	2	*0.8*
0.08	**1**	**1**	**0.3**
0.05	1	1	0.5
0.12	0	*1*	*0.1*

Final counterfactual explanations

Explain ID	Age	Sex	Decision	Salary	Def. last year
1	**30**	**F**	**1**	**$ 6000**	**no**
2	**25**	**M**	**1**	**$ 4800**	**no**

Discarding reasons

INVALID DECISION

LARGER L0

LARGER GOWER

Our contribution

MCCE

- Simple, yet effective
- Flexible
- Scalable and easy to implement
- Outperforms competing methods in terms of both accuracy and speed


TAKE HOME

To think differently

To simplify

DARE!

To take a role in an unfamiliar field

To use your statistical mindset