Introduction to Explainable Machine Learning Using Stata

Aramayis Dallakyan

Senior Statistician and Software Developer

Oceania Stata 2026

Outline

Section 1

New in Stata 19

- `h2o` and `_h2oframe` suite of commands for working with H2O cluster and frames.
- `h2oml` suite of commands for end-to-end machine learning analysis:
  - Methods
  - Performance
  - Prediction
  - **Explainability**

Stata's h2oml suite of commands: Explainability

- Local methods:
  - SHAP values: `h2omlgraph shapvalues`
  - Individual conditional expectation plot: `h2omlgraph ice`
- Global methods:
  - SHAP beeswarm plot: `h2omlgraph shapsummary`
  - Partial dependence plot: `h2omlgraph pdp`
  - Variable importance: `h2omlgraph varimp`
  - Permutation importance: `h2omlgraph permimp`
- Save decision tree DOT file and display rule set: `h2omltree`

H2O setup

For details, see https://tinyurl.com/mr3mj67z

1. Go to https://h2o.ai/resources/download/.
2. Click on the tab **H2O Open Source Platform**.
3. Go to **Latest Stable Release** or **Prior Releases**. Stata's H2OML documentation is written using **Version 3.46.0.6**.
4. Click on **Download H2O**.
5. After downloading the file (for example, `h2o-3.46.0.6.zip`), unzip it and look for the `h2o.jar` file. This is the only file from within the zip file that you will need.

Running example: Predicting employee attrition

- Data presents an employee survey from IBM. **Response** `attrition` indicates whether there is attrition or not.
- The data set contains approximately **1500 observations** and **18 predictors**.
- **Predictors** can be categorized into:
  - Demographic/ Personal characteristics: `age, gender, education, maritalstatus`.
  - Job related characteristics: `jobrole, businesstravel, monthlyincome, totalworkingyears, numcompaniesworked, yearsatcompany, yearsincurrentrole, yearswithcurrmanager, performance`.
  - Satisfaction metrics: `workinglifebalance, environmentsat, jobinvolvement, jobsatisfaction, relationshipsat`

Running example

- Load dataset into Stata's memory

```
. use https://www.stata.com/users/assaad_dallakyan/attrition
(IBM HR analytics employee attrition)
```

- Initialize a cluster

```
. h2o init
(output omitted)
```

```
. _h2oframe put, into(attrition)
```

```
Progress (%): 0 100
```

```
. _h2oframe change attrition
```

Running example

```
. _h2oframe describe

          Rows:      1470
          Cols:        20
```

Column	Type	Missing	Zeros	+Inf	-Inf	Cardinality
age	int	0	0	0	0	
education	enum	0	572	0	0	5
employeenumber	int	0	0	0	0	
environmentsat	enum	0	453	0	0	4
jobinvolvement	enum	0	868	0	0	4
jobsatisfaction	enum	0	442	0	0	4
monthlyincome	int	0	0	0	0	
numcompaniesw~d	int	0	197	0	0	
performance	enum	0	1244	0	0	2
relationshipsat	enum	0	459	0	0	4
totalworkingy~s	int	0	11	0	0	
worklifebalance	enum	0	893	0	0	4
yearsatcompany	int	0	44	0	0	
yearsincurren~e	int	0	244	0	0	
yearswithcurr~r	int	0	263	0	0	
attrition	enum	0	1233	0	0	2
businesstravel	enum	0	150	0	0	3
gender	enum	0	588	0	0	2
jobrole	enum	0	131	0	0	9
maritalstatus	enum	0	327	0	0	3

Our goal

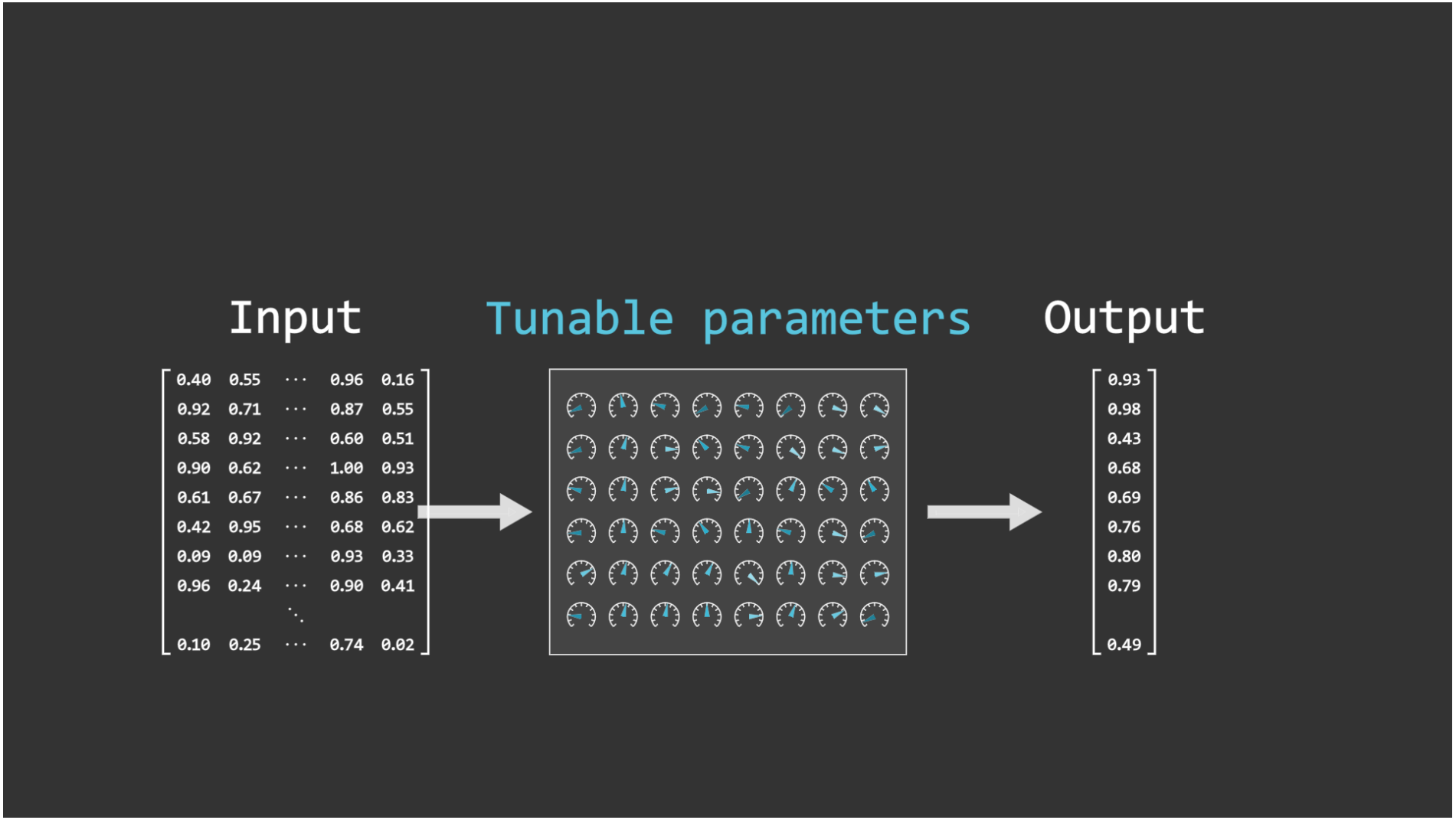
Our goal is threefold:

- **Develop an accurate prediction model for employee attrition**. Reliable predictions improve HR's ability to intervene on time and address potential issues.
- **Identify which predictors most influence attrition**, offering actionable insights for organizational decision-making.
- **Understand and explain how specific predictors contribute to the attrition risk for individual employees**, enabling personalized interventions.

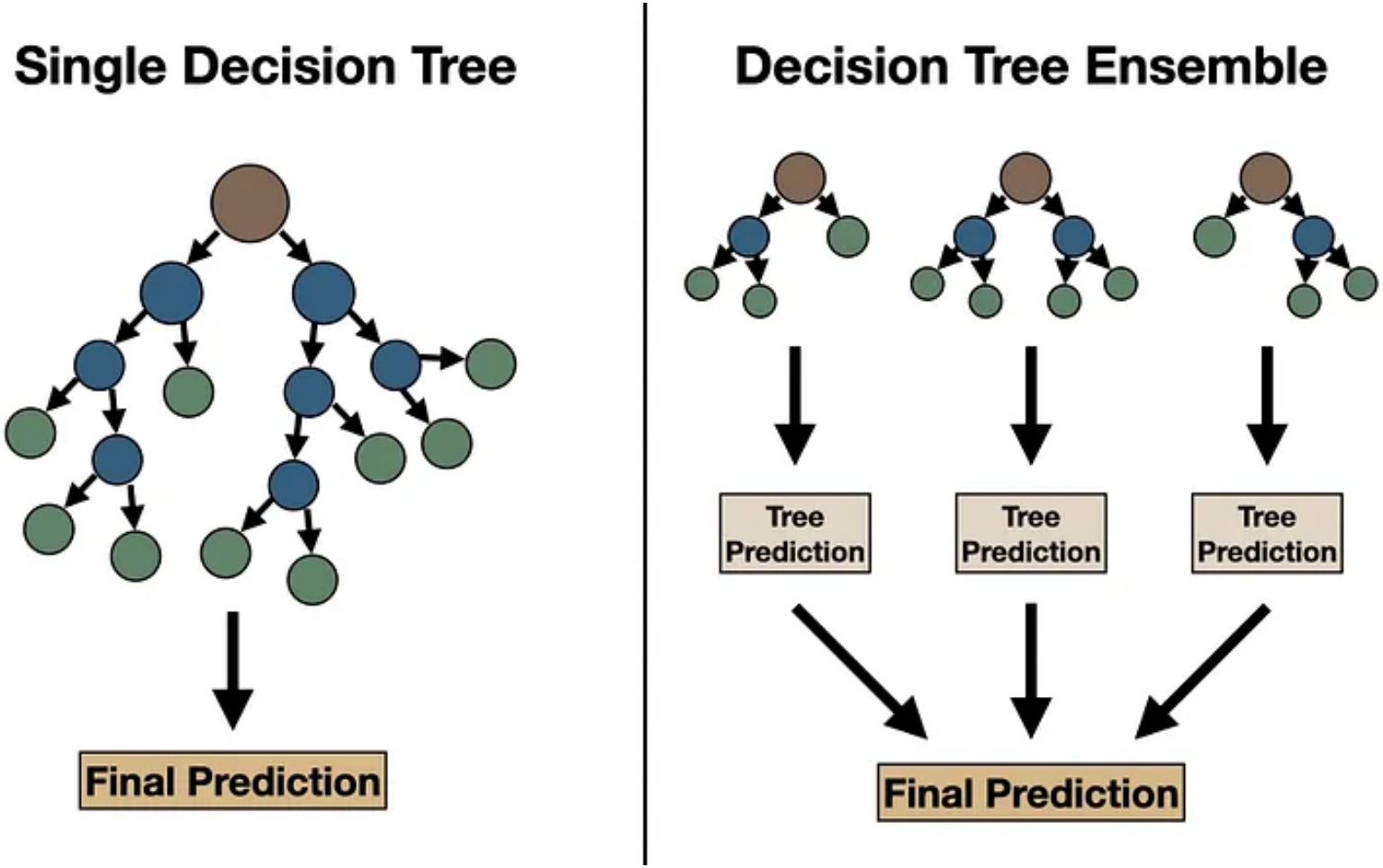
Two mindsets of model development

- In a seminal article Breiman (2001) identified two cultures of statistical modeling
  - **Explanatory modeling**, where models are applied for inferential purposes, i.e., hypothesis testing, consistency, etc.
  - **Predictive modeling**, where models are used for the purpose of predicting the value of a new, unseen observation.

What is Machine Learning?



Ensemble decision trees

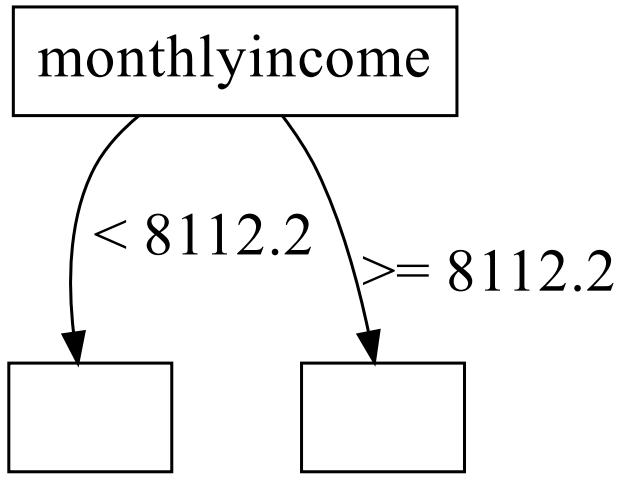


Decision Trees

- **Decision trees** can be used for both regression and classification.

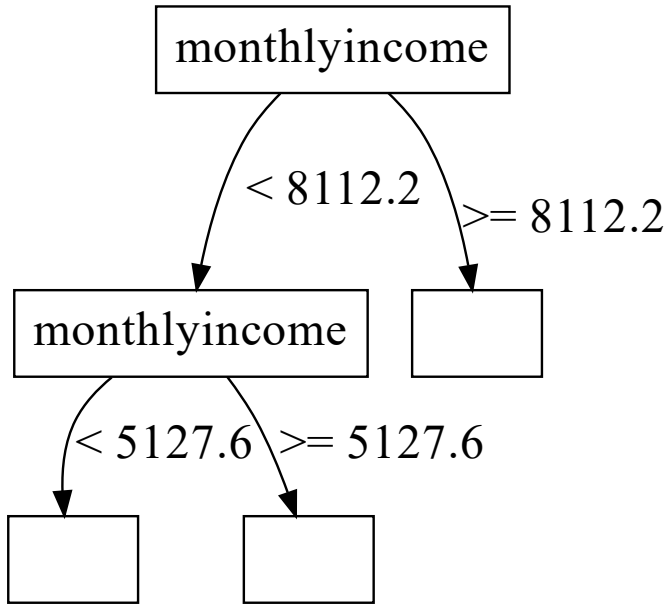
Decision Trees

Monthly income	Performance	Attrition
4236	Very high	No
5036	High	Yes
5098	High	No
5136	High	No
…	…	…
…	…	…
7096	Very high	No
8236	High	Yes
…	…	…



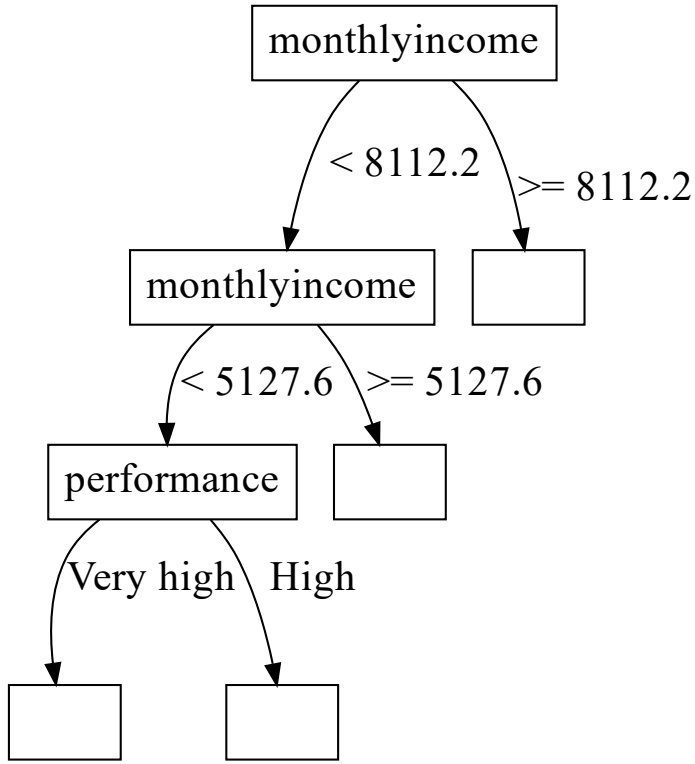
Decision Trees

Monthly income	Performance	Attrition
4236	Very high	No
5036	High	Yes
5098	High	No
5136	High	Yes
…	…	…
…	…	…
7096	Very high	Yes
8236	High	Yes
…	…	…



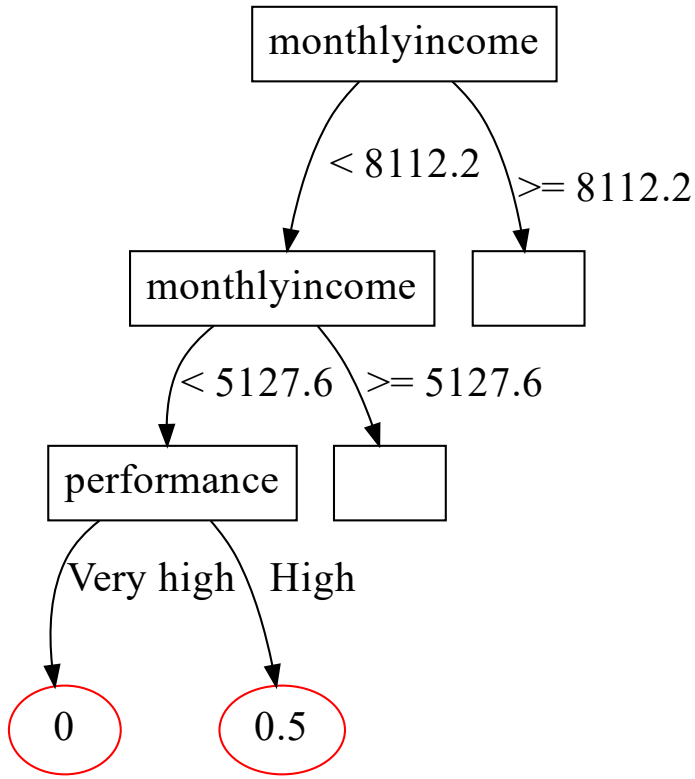
Decision Trees

Monthly income	Performance	Attrition
4236	Very high	No
5036	High	Yes
5098	High	No
5136	High	Yes
…	…	…
…	…	…
7096	Very high	Yes
8236	High	Yes
…	…	…



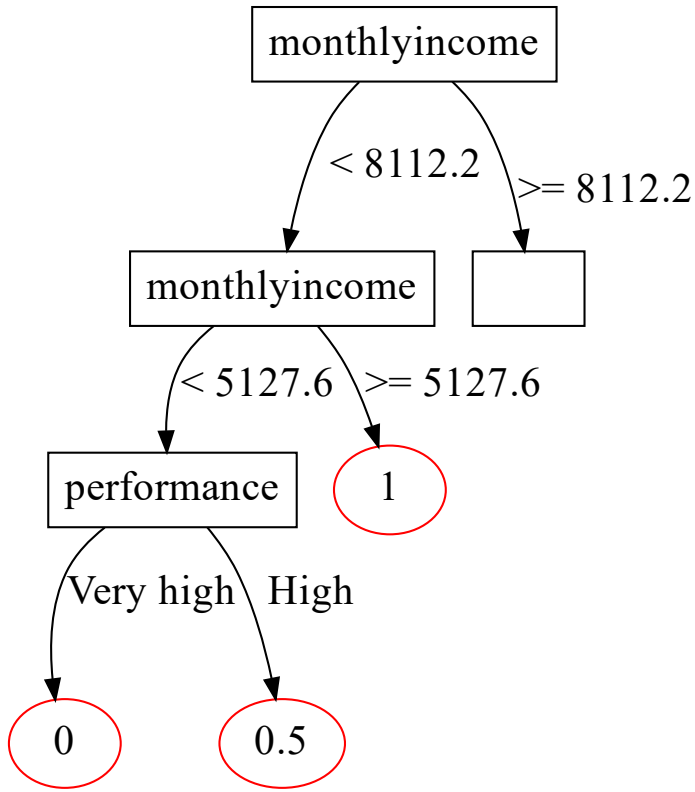
Decision Trees

Monthly income	Performance	Attrition
4236	Very high	No
5036	High	Yes
5098	High	No
5136	High	Yes
…	…	…
…	…	…
7096	Very high	Yes
8236	High	Yes
…	…	…



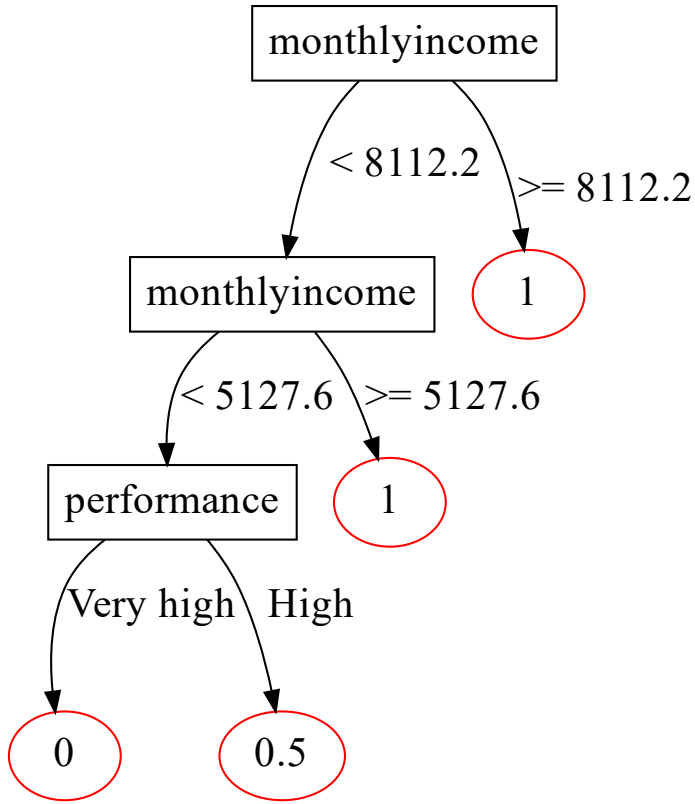
Decision Trees

Monthly income	Performance	Attrition
4236	Very high	No
5036	High	Yes
5098	High	No
5136	High	Yes
…	…	…
…	…	…
7096	Very high	Yes
8236	High	Yes
…	…	…

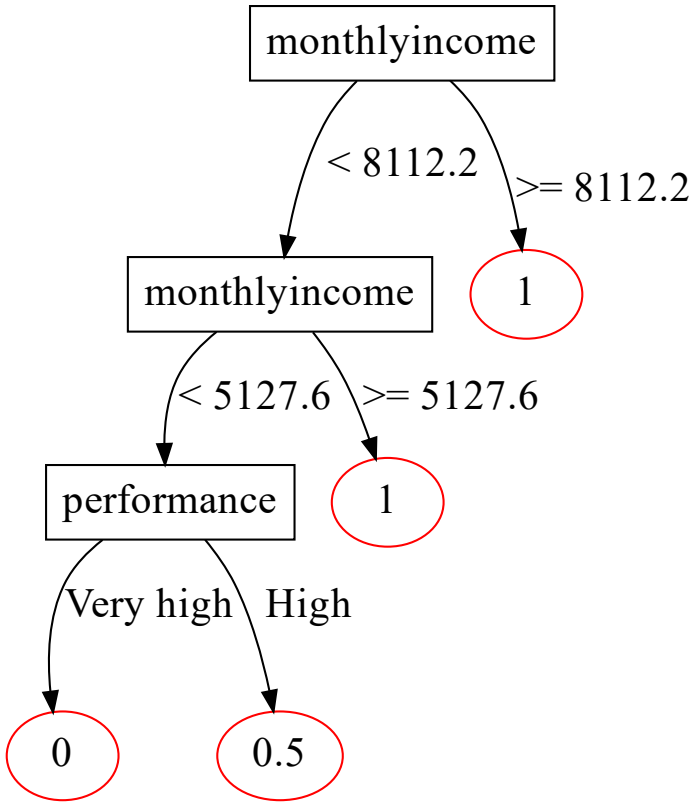


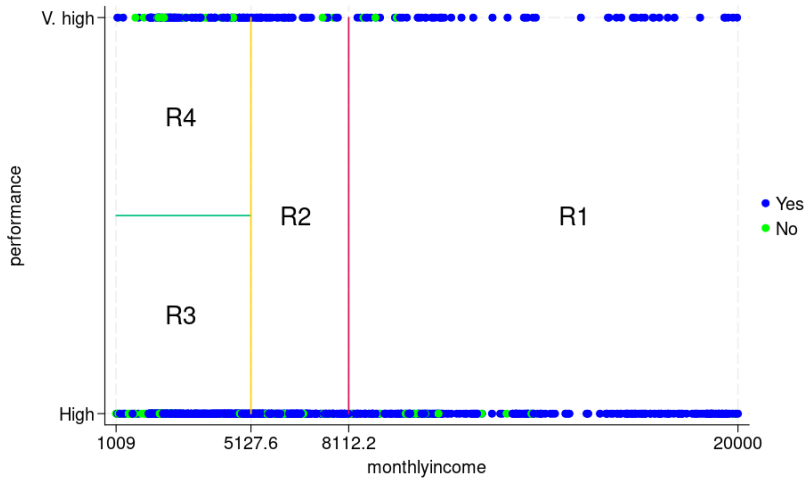
Decision Trees

Monthly income	Performance	Attrition
4236	Very high	No
5036	High	Yes
5098	High	No
5136	High	Yes
…	…	…
…	…	…
7096	Very high	Yes
8236	High	Yes
…	…	…



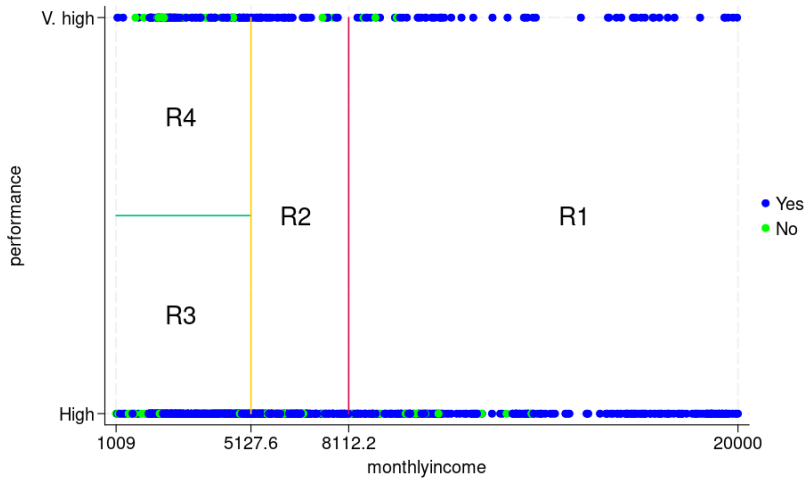
Decision Trees





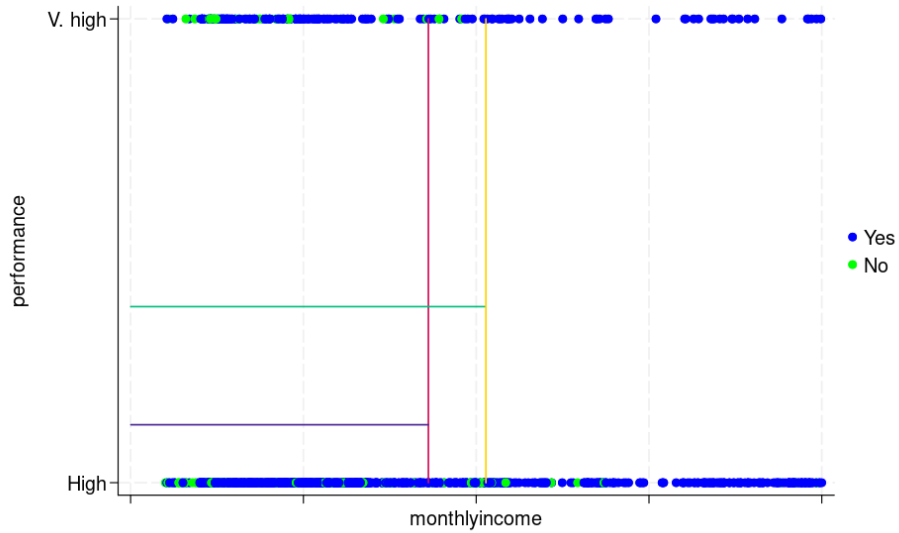
Decision Trees

- How to partition the predictor space?
  - Which predictor to choose?
  - Which value to use for splitting?



Partitioning the predictor space

- Ideally, we want to split by considering **all possible** splits of all predictors.
- Finding the best binary partition is computationally infeasible.
- We need a **greedy approach**.



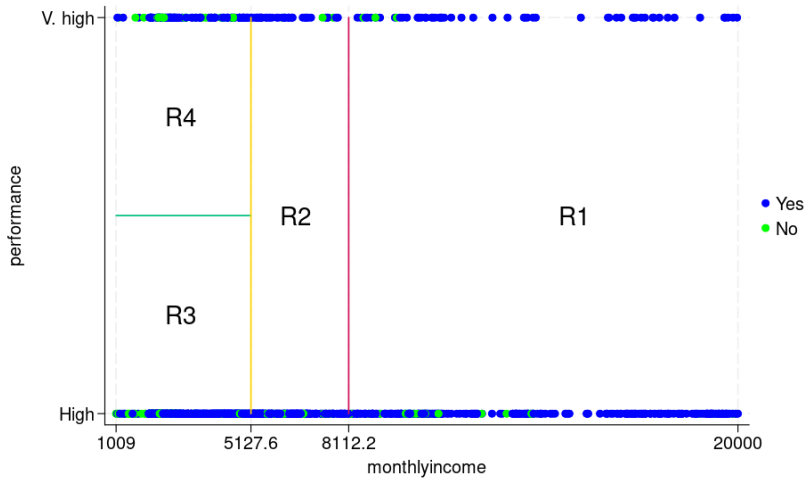
Partitioning the predictor space

- We seek the splitting variable X_j and split point t that minimize **impurity measure**.
  - Classification: cross-entropy

$$i(X_j, t) = -p\ln(p) - (1-p)\ln(1-p)$$

  - Regression: MSE

$$i(X_j, t) = \frac{1}{N_m}\sum_{x_i \in R_m}(y_i - \widehat{c}_m)^2;\ \widehat{c}_m = \frac{1}{N_m}\sum_{x_i \in R_m} y_i;\ N_m = \#\{x_i \in R_m\}$$



Decision trees

- Pros
  - Represent information in an intuitive and easy- to-visualize way.
  - Predictors can be of any type: numeric, binary, categorical, etc.
  - Monotone transformation or scaling do not change the model outcome.
  - Graciously handle missing data
- Cons
  - Notoriously unstable and have a high variance.
  - A small change in the data can lead to a completely different set of splits.
  - Have difficulties with modeling simple smooth functions.

Ensemble methods

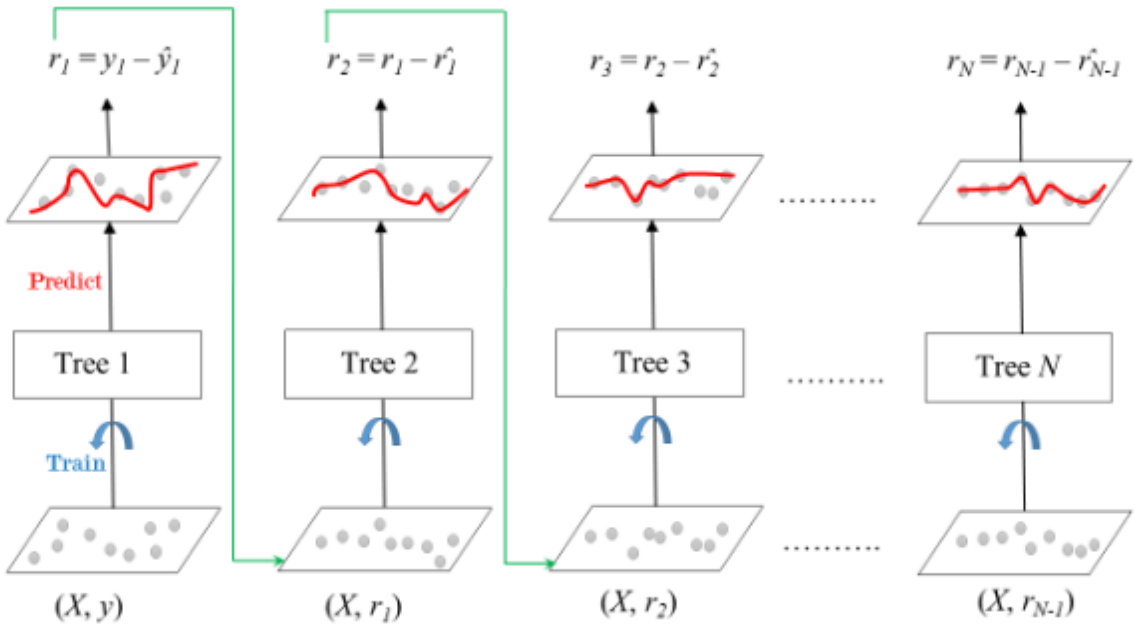
- **Ensemble methods**: a mechanism that forms a smart committee of incompetent but carefully selected members(trees) to solve a machine learning problem.
- Ensemble methods use **unstable learners** (trees) to provide more variable outcomes than stable learners, which aid in generalization to unseen data.

Gradient boosting machine (GBM)

- GBM builds a series of trees **sequentially**.
- Each tree tries to **correct the errors** made by the previous one.
- The final ensemble is obtained

$$\widehat{f}_{\text{GBM}} = \sum_{m=1}^{M} \widehat{\alpha}_m f^m(\boldsymbol{x})$$

- α_m measures the importance of the classifier $f_m(\cdot)$.



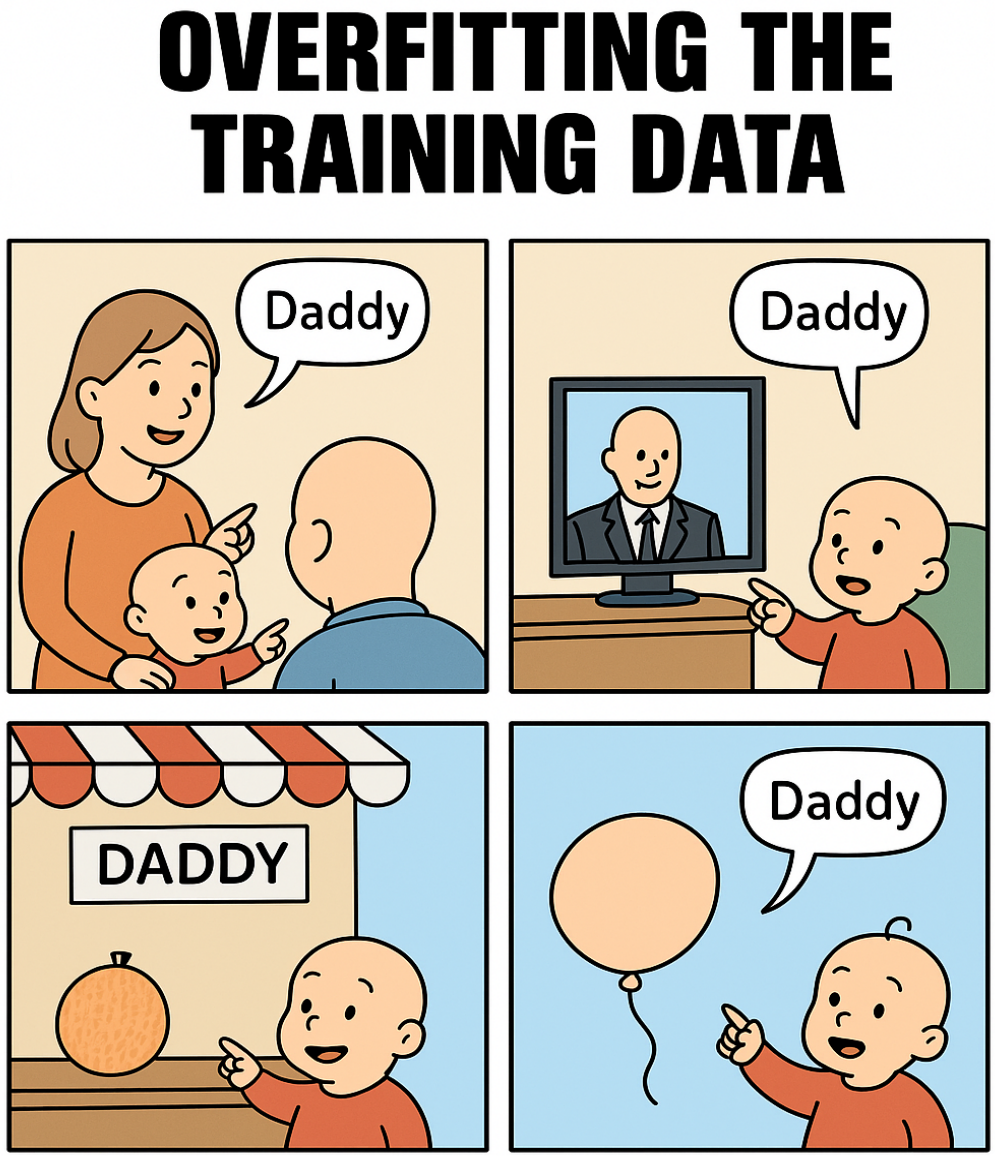
Borrowed from Ozkurt (2024)

GBM: “Walking" knowledge

- Imagine you’re studying for a big exam. You take a practice test and get some answers wrong.
- Instead of starting over from scratch, your tutor looks at your mistakes and gives you **targeted feedback** on what you got wrong.
- Next time, you take another test and your **focus is on the areas** you messed up before.
- Again, you make mistakes, and again your tutor gives you feedback on those specific errors.
- Over several rounds, you get better and better, not by relearning everything, but by **focusing only** on your previous mistakes.

Hyperparameter tuning

- Our goal is to make accurate predictions on future data.
- Using the same training data for estimation and tuning leads to **overfitting and poor generalization** to unseen data.
- We want to minimize **the generalization error**.



Hyperparameter tuning

- Two common approaches to minimize the generalization error
  - Three-way houldout method
    - Split data into training, validation, and testing
  - Two way-houldout method with K-fold cross-validation
    - Split data into training and testing. Use cross-validation for hyperparameter tuning.

Hands-on hyperparameter tuning

- We split the data into a training and a testing datasets.
- Use 5-fold cross-validation for tuning.

```
. _h2oframe split attrition, into(train test) ///
>         ratio(0.7 0.3) rseed(19)

. _h2oframe change train

. global predictors age education environmentsat   ///
     jobinvolvement jobinvolvement jobsatisfaction ///
     monthlyincome numcompaniesworked performance  ///
     relationshipsat totalworkingyears worklifebalance  ///
     yearsatcompany yearsincurrentrole yearswithcurrmanager
     businesstravel gender jobrole maritalstatus
```

Hands-on hyperparameter tuning

```
. h2oml gbbinclass attrition $predictors, h2orseed(19) cv(5, modulo) ///
>         maxdepth(1(1)4) predsamprate(0.3(0.1)0.6) ///
>         ntrees(100(50)250) tune(metric(aucpr))

Progress (%): 0 100

Gradient boosting binary classification using H2O

Response: attrition
Loss:     Bernoulli
Frame:                                  Number of observations:
  Training: train                                 Training =  1,039
                                     Cross-validation =  1,039
Cross-validation: Modulo                Number of folds     =      5

Tuning information for hyperparameters

Method: Cartesian
Metric: AUCPR
```

Hyperparameters	Grid values Minimum	Maximum	Selected
Number of trees	100	250	150
Max. tree depth	1	4	1
Pred. sampling rate	.3	.6	.4

Hands-on performance on testing dataset

- After choosing the best model, we would like to compare the performance of different methods(e.g. random forest, GBM) on the testing dataset.

```
. h2omlest store gbm

. h2omlpostestframe test
(testing frame test is now active for h2oml postestimation)
```

- We run a hypothetical random forest model and store the results.

```
. h2oml rfbinclass attrition $predictors, h2orseed(19) cv(5, modulo) ///
>         ntrees(90) maxdepth(4)
(Output omitted)

. h2omlest store rf

. h2omlpostestframe test
(testing frame test is now active for h2oml postestimation)
```

Hands-on performance on testing dataset

```
. h2omlgof gbm rf

Performance metrics for model comparison using H2O
Testing frame: test
```

	gbm	rf
Testing		
No. of observations	431	431
Log loss	.3884012	.4193272
Mean class error	.2845085	.2863426
AUC	.7850783	.7626068
AUCPR	.5676777	.4882438
Gini coefficient	.5701567	.5252137
MSE	.1188699	.1310679
RMSE	.3447751	.362033

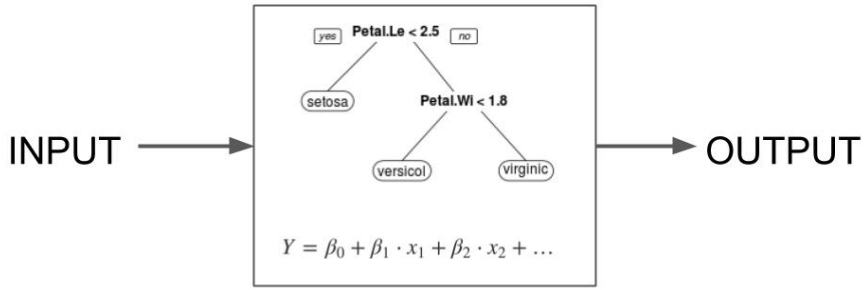
Section 2

Interpretable vs explainable methods

- Machine learning methods often treated as **black boxes** that do not explain their predictions in a way that practitioners can understand.
- Traditionally, machine learning models are evaluated by comparing performance metrics using validation data.
- This may be unreliable because validation data may not always be fully representative of real-world data.
- The use of explainable methods might shed light on model performance.

Interpretable vs explainable methods

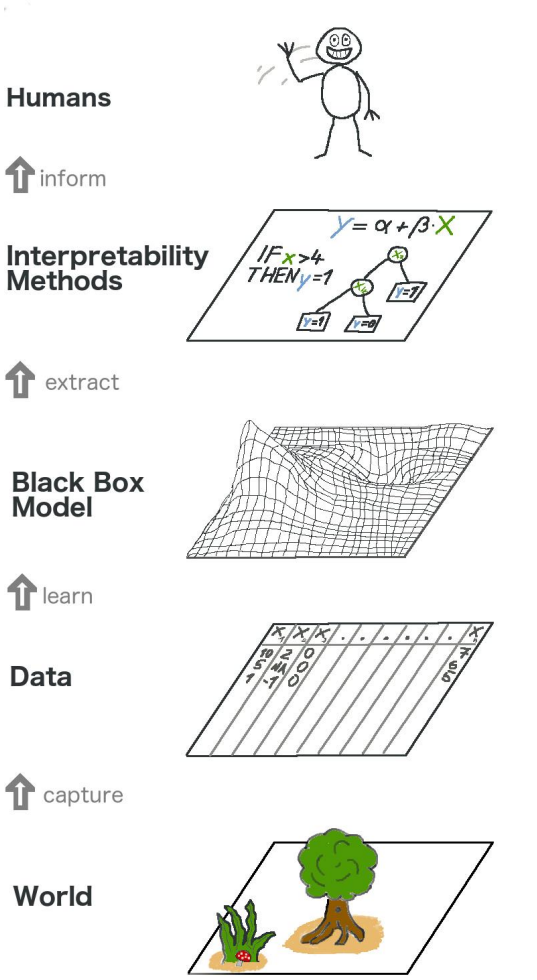
- **Interpretable methods:** a human can understand how it works internally and how it arrives at predictions, without external tools.
- We can look at the model's structure (like coefficients) and understand the result directly.
- Commonly used:
  - Linear and logistic regressions
  - Decision trees
  - Decision-set and rule-based methods
- Typically have lower prediction accuracy than machine learning models.



Borrowed from Molnar (2025)

Interpretable vs explainable methods

- **Explainable methods** rely on external tools to make their predictions understandable to a human.
- Provide **post hoc** models that **explain the prediction** of the original black-box models.



Borrowed from Molnar (2025)

A word of caution

- Explainable methods are **not recommended** for high-stakes decisions.
- Such as in medicine, criminal justice, social bias, etc.
- We recommend using those techniques as a tool for analysis and algorithmic audit.

Some more discussion

- The partitioning of Interpretable vs Explainable methods can be harmful.
- It creates an erroneous illusion that interpretable models are natively correct and free of bias.
- Two cultures of researchers (Biecek and Samek, 2024):
  - **Human values oriented explanations**: revolves around concepts such as trust, causality, fairness, and ethical decision making (Lipton, 2018).
  - **Model validation oriented explanations**: used to diagnose, audit, debug and validate the model (Ribeiro et al., 2016).

Global and local explainable models

- Explainable methods can be further divided into:
  - Local: aim to explain **individual predictions**
  - Global: describe how predictors affect prediction **on average**.

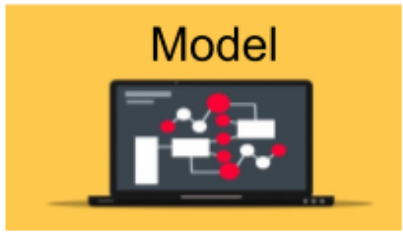
Global and local explainable methods

- Local methods:
  - SHAP values: `h2omlgraph shapvalues`
  - Individual Conditional Expectation: `h2omlgraph ice`
- Global methods:
  - SHAP summary: `h2omlgraph shapsummary`
  - Partial Dependence Plot (PDP): `h2omlgraph pdp`
  - Variable importance: `h2omlgraph varimp` and `h2omlgraph permimp`
  - Surrogate models

Local methods: SHAP values

- Shapley Additive Explanations (SHAP), introduced in Lundberg and Lee (2017), try to explain contribution of each predictor for a given observation.

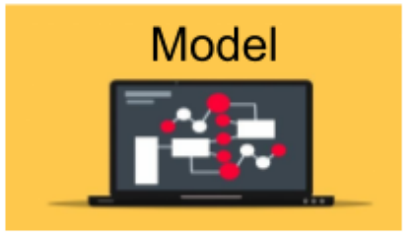
Local methods: SHAP values



Motivated from Scott Lundberg presentation
(https://www.youtube.com/watch?v=ngOBhhINWb8)

Local methods: SHAP values

John, an employee



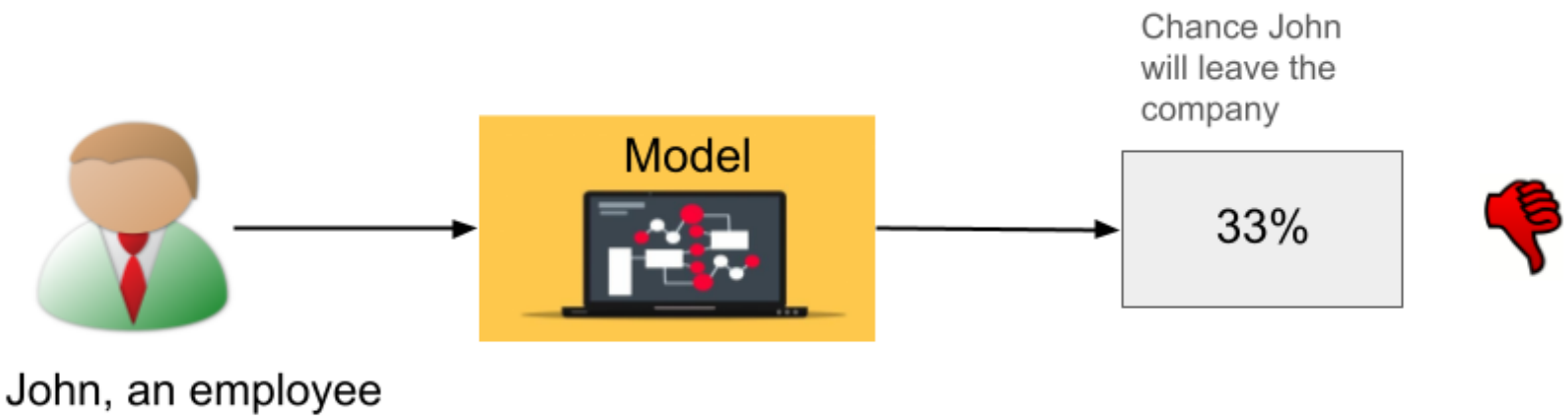
Motivated from Scott Lundberg presentation
(https://www.youtube.com/watch?v=ngOBhhINWb8)

Local methods: SHAP values

John, an employee

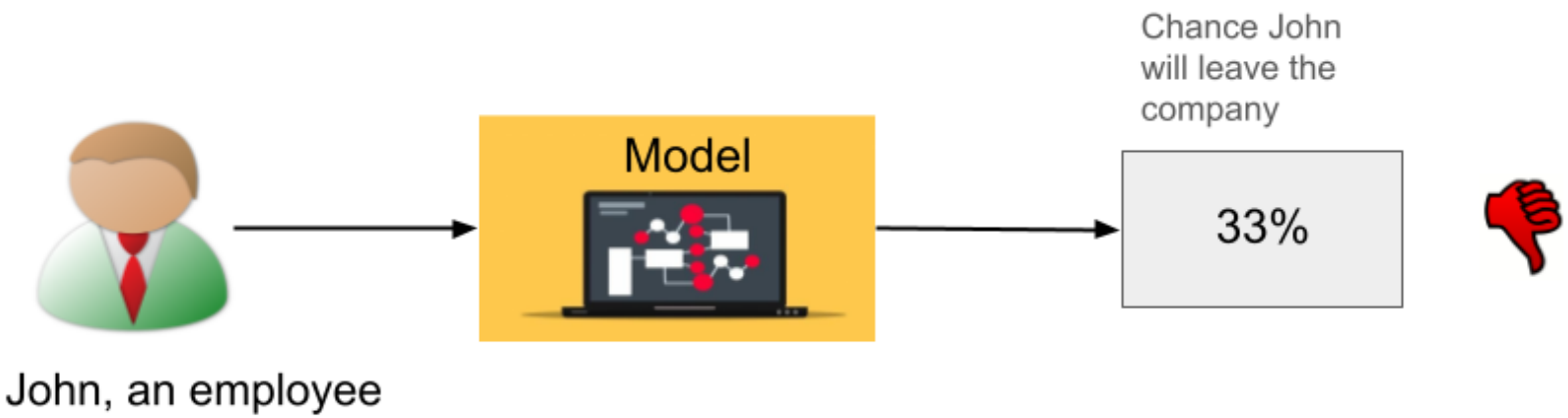
Motivated from Scott Lundberg presentation
(https://www.youtube.com/watch?v=ngOBhhINWb8)

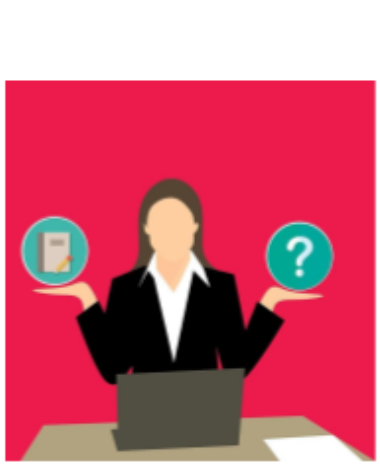
Local methods: SHAP values



Motivated from Scott Lundberg presentation (https://www.youtube.com/watch?v=ngOBhhINWb8)

Local methods: SHAP values

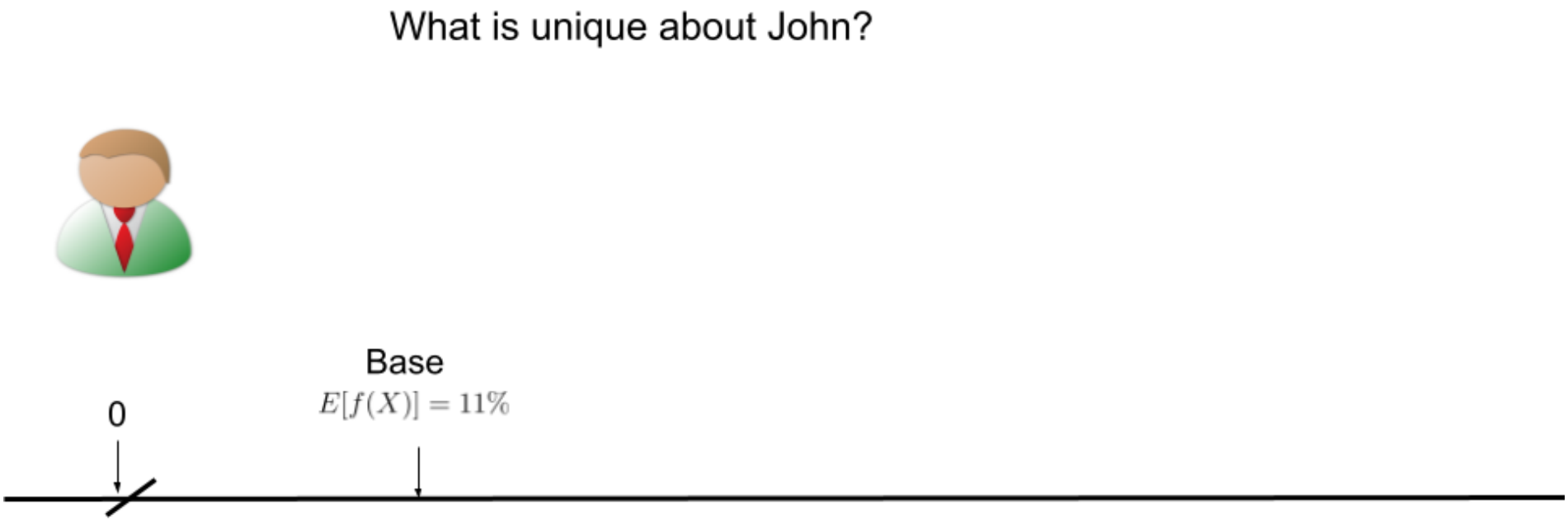




Local methods: SHAP values

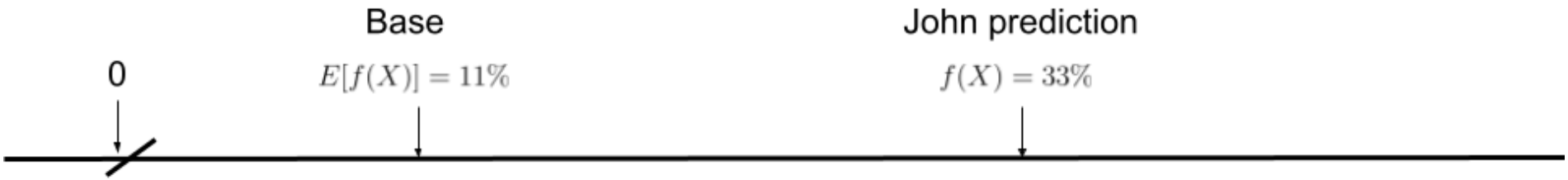
What is unique about John?

Local methods: SHAP values

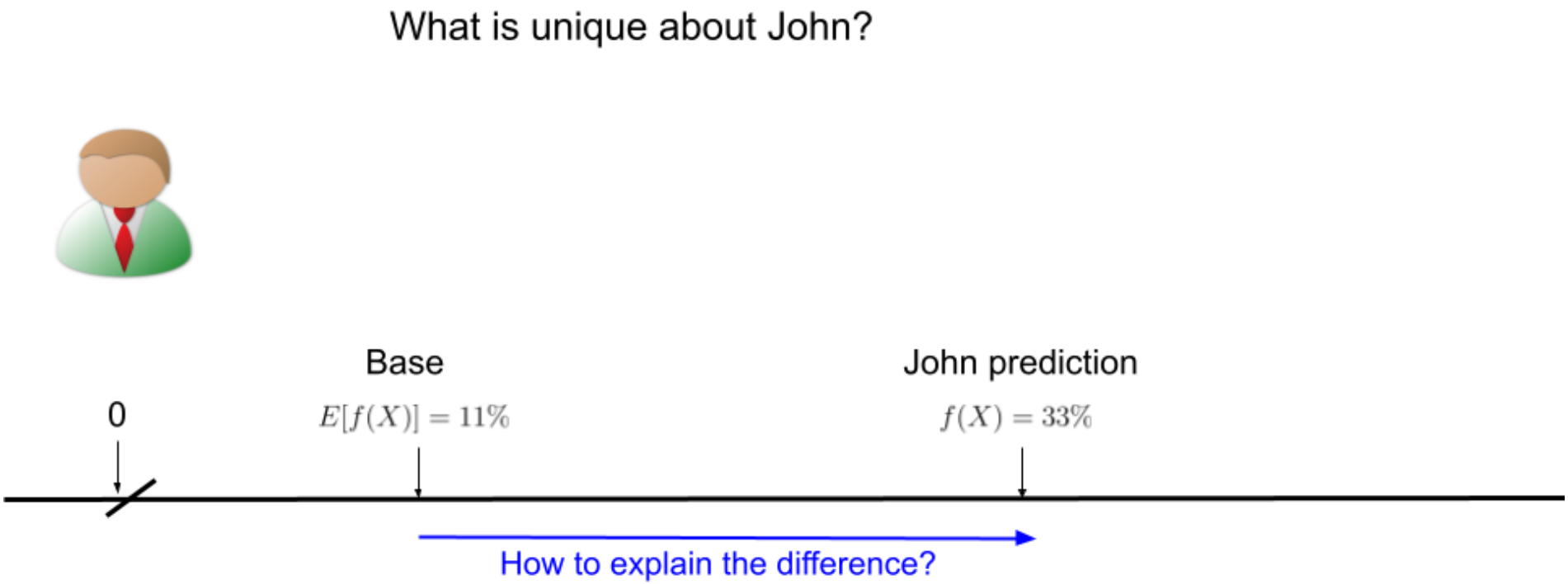


Local methods: SHAP values

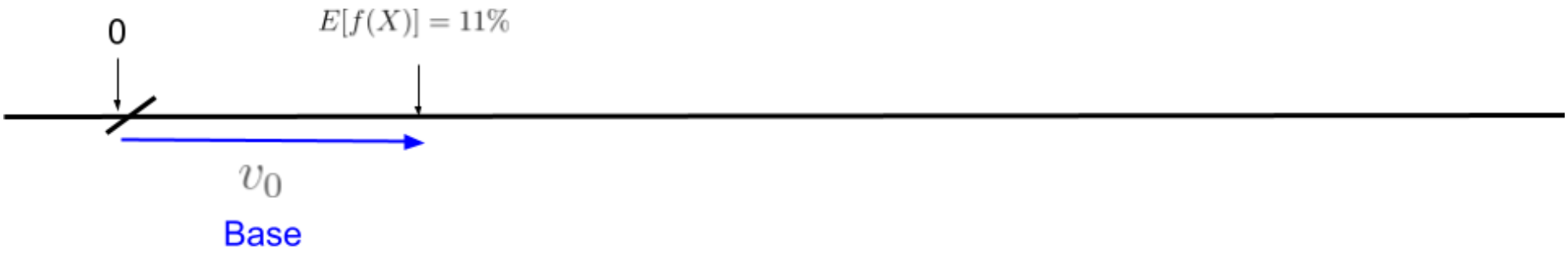
What is unique about John?



Local methods: SHAP values

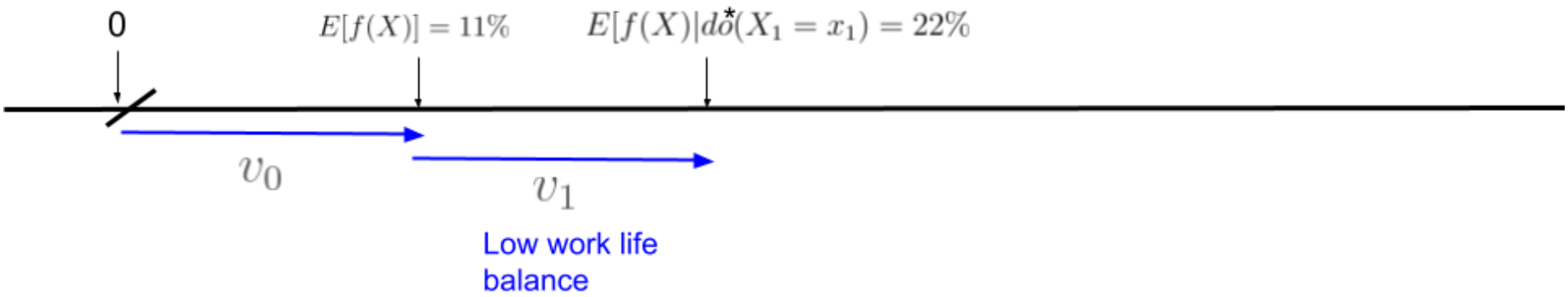


Local methods: SHAP values



Local methods: SHAP values

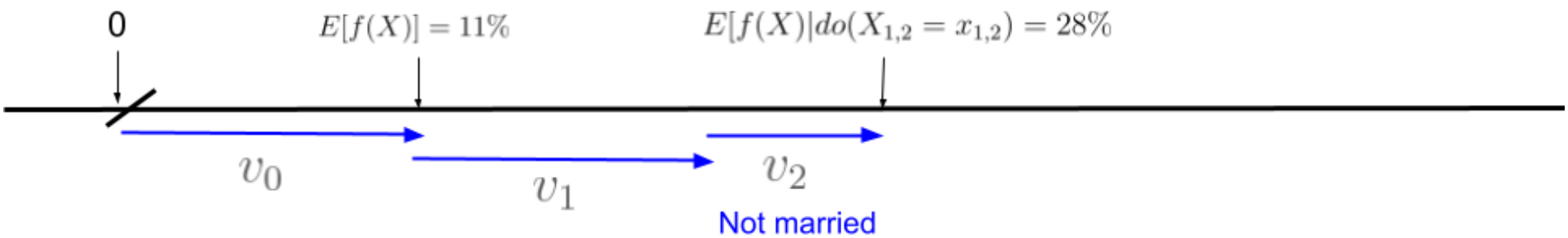
What is unique about John?



*Pearl (2009) and Janzing et. al (2019)

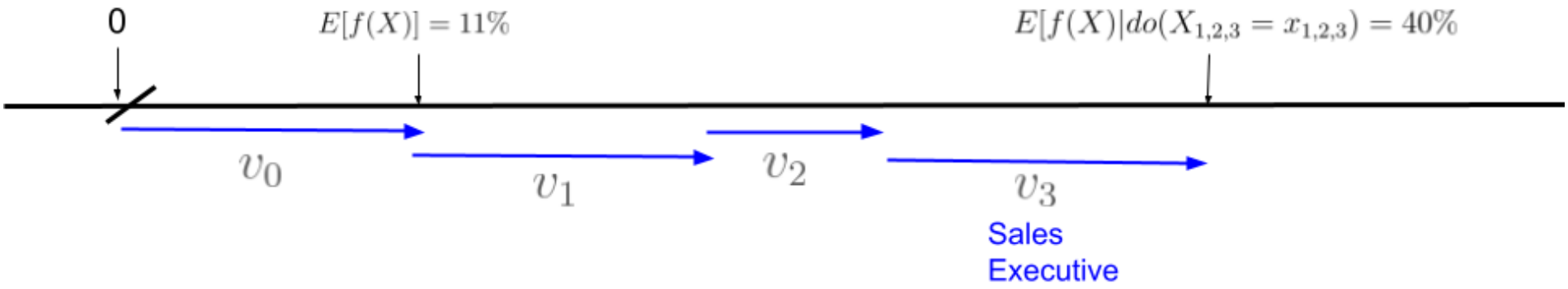
Local methods: SHAP values

What is unique about John?



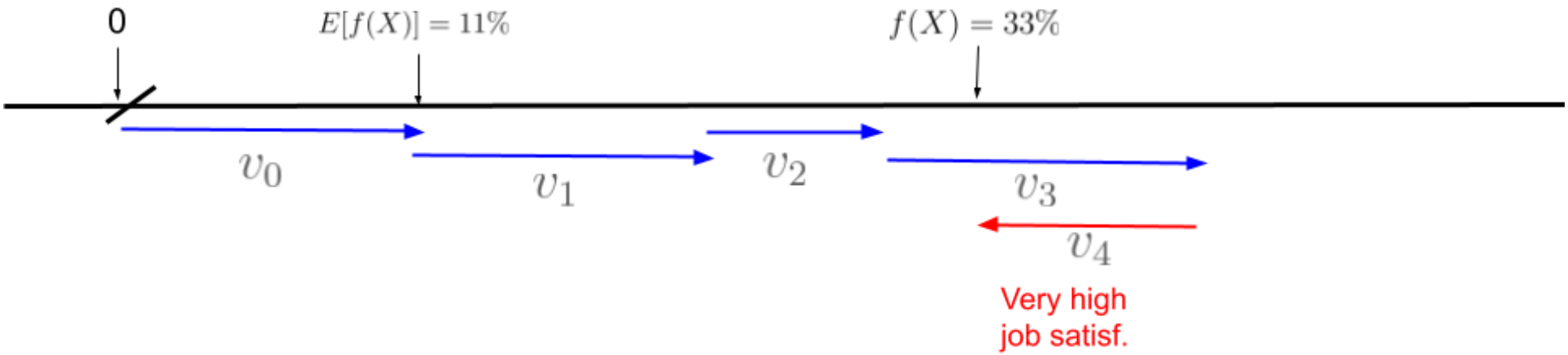
Local methods: SHAP values

What is unique about John?



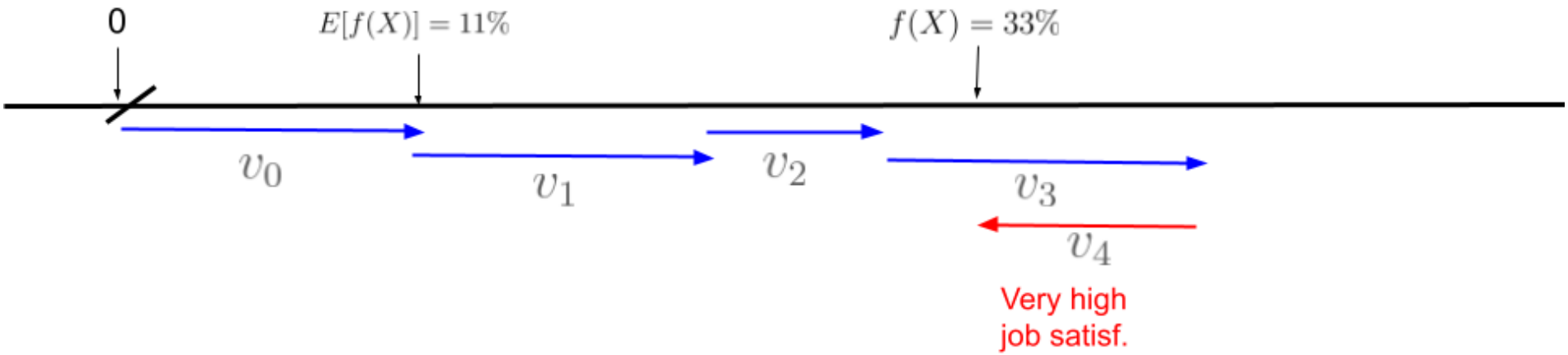
Local methods: SHAP values

What is unique about John?



Local methods: SHAP values

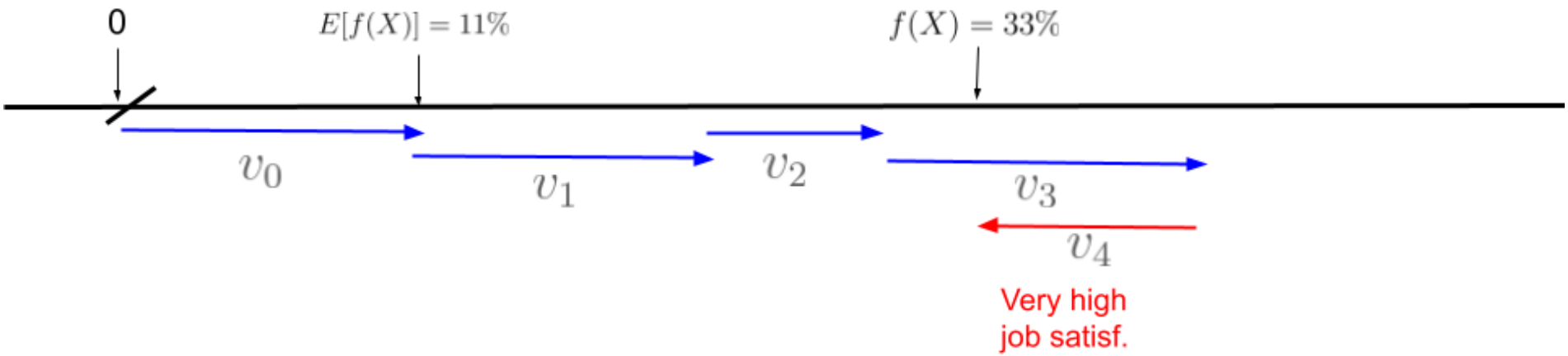
Are we done?



Local methods: SHAP values

Non linear model.

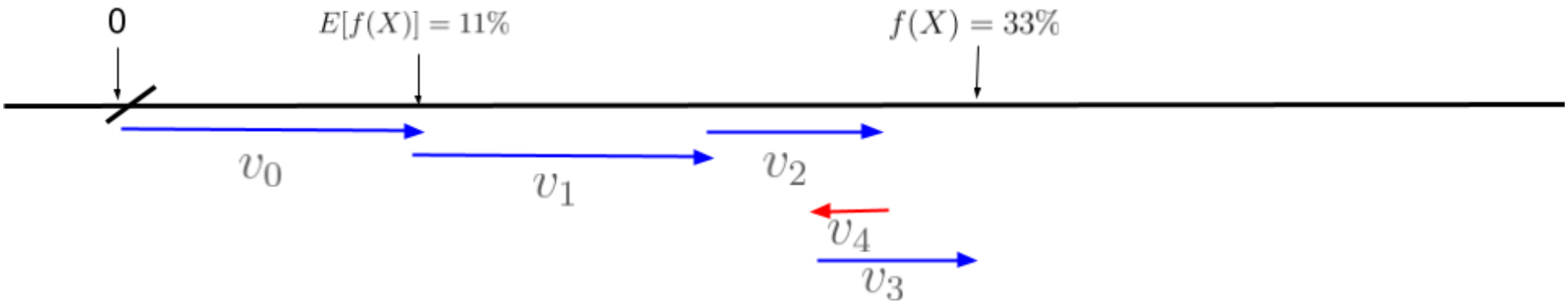
The order matters!



Local methods: SHAP values

Non linear model.

The order matters!

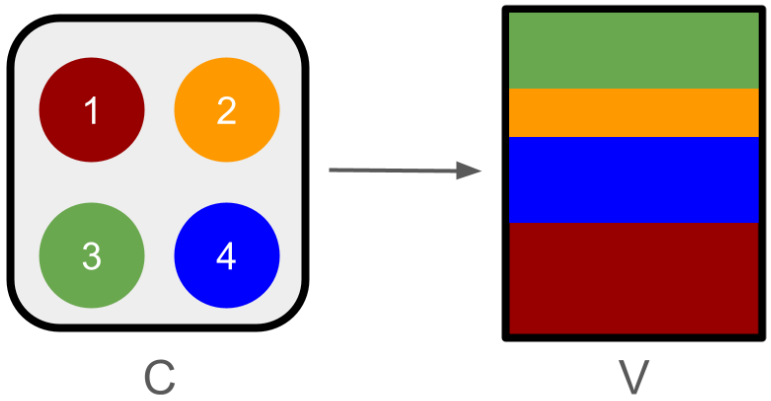


Shapley values

- To be able to estimate the contribution of each predictor, Lundberg and Lee (2017) rely on **Shapley values**.
- Shapley values were introduced in cooperative game theory as a way of providing a fair solution to the following question:
  - If we have a coalition C that collaborates to produce a value V:
  - How much did each **individual member** contribute to the final value?

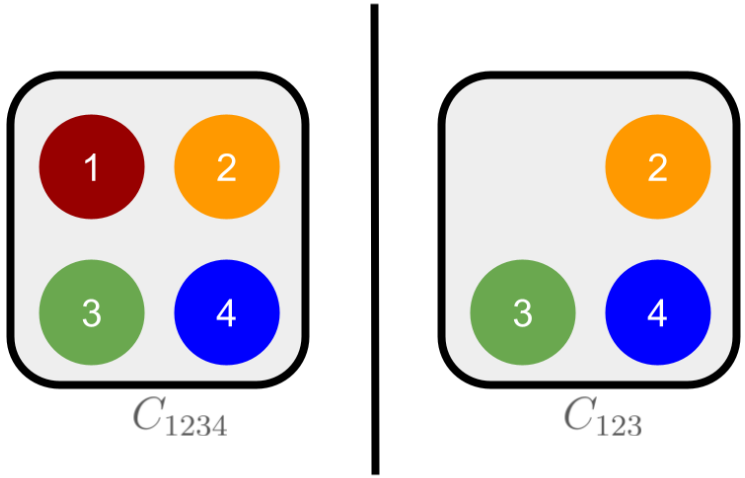
Shapley values

- Suppose 4 of us work on a project.
- We want to know exactly how much each of us contributed to the final coalition value?
- What share of the profit does each of us deserve?



Shapley values

- Let's try to compute Shapley value for the first member.
- We start by sampling a coalition that contains the first member.
- Then look at the coalition formed by removing that member.

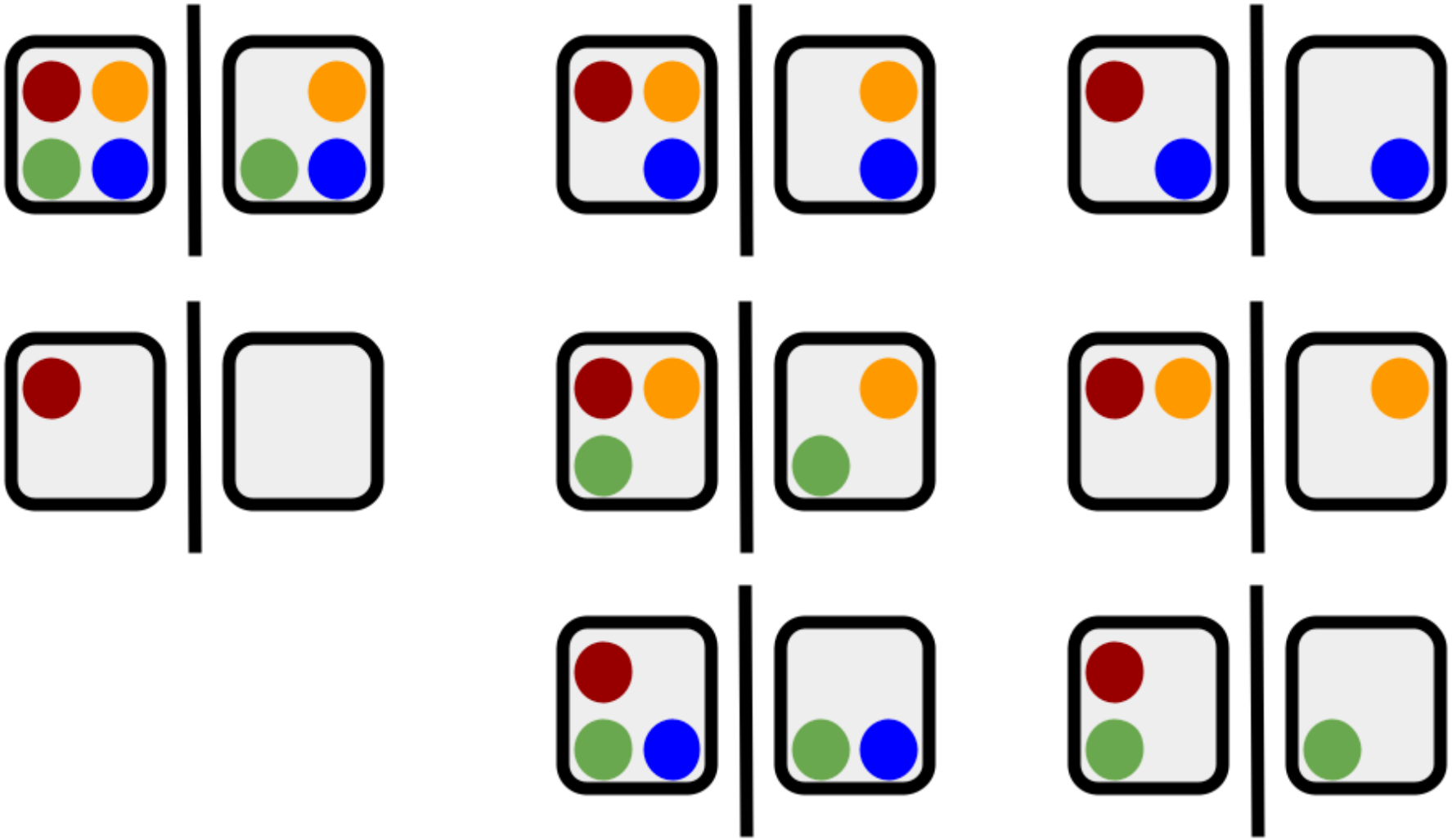


Shapley values

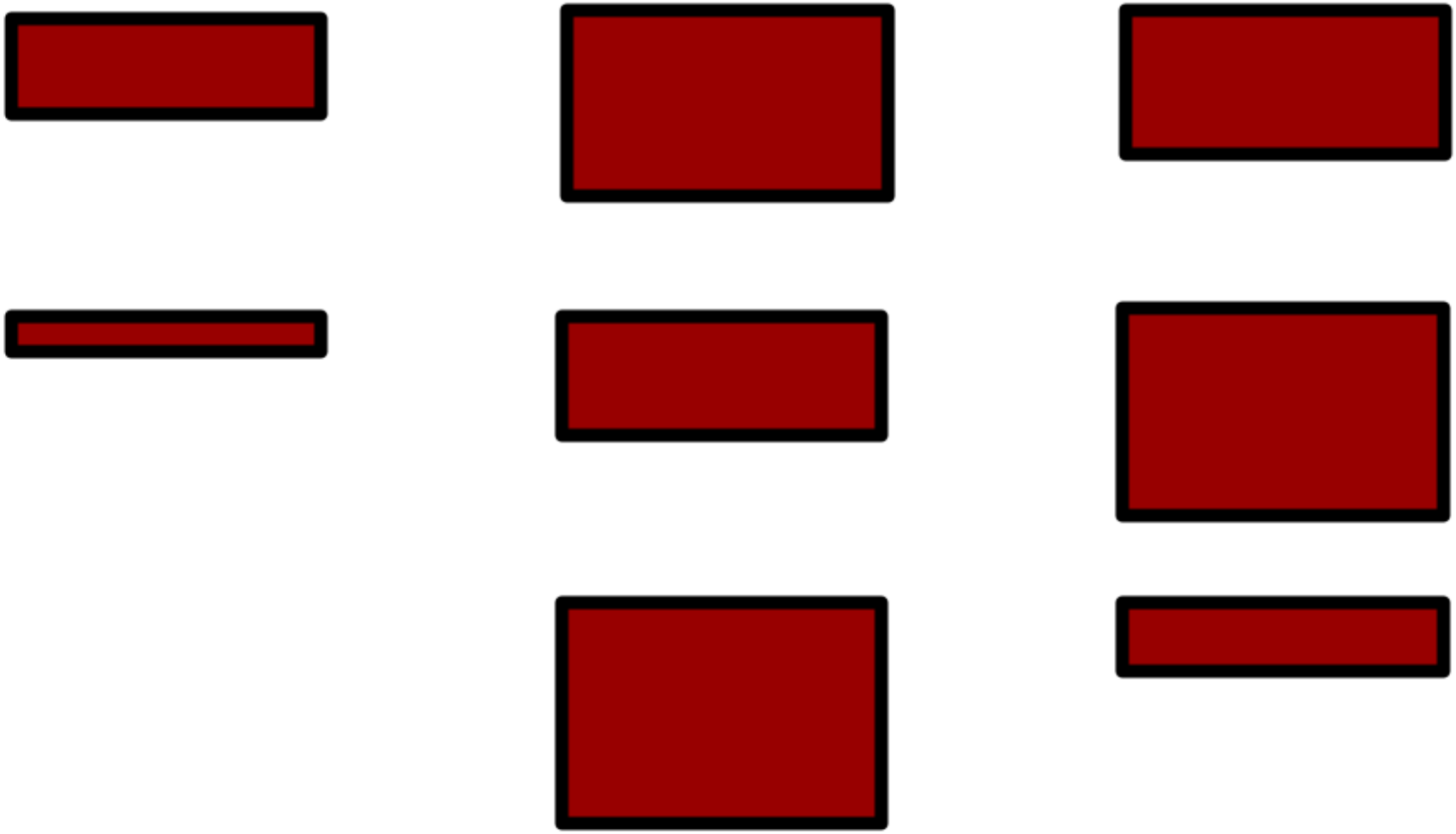
- We then look at the respective values of these two coalitions and compute the difference.
- This difference is a **marginal contribution** of member 1 to the coalition consisting of member $C_{2,3,4} = \{2, 3, 4\}$
- It shows how much member 1 contributed to that specific group.

$$V_{1234} - V_{234} =$$ 

Shapley values



Shapley values

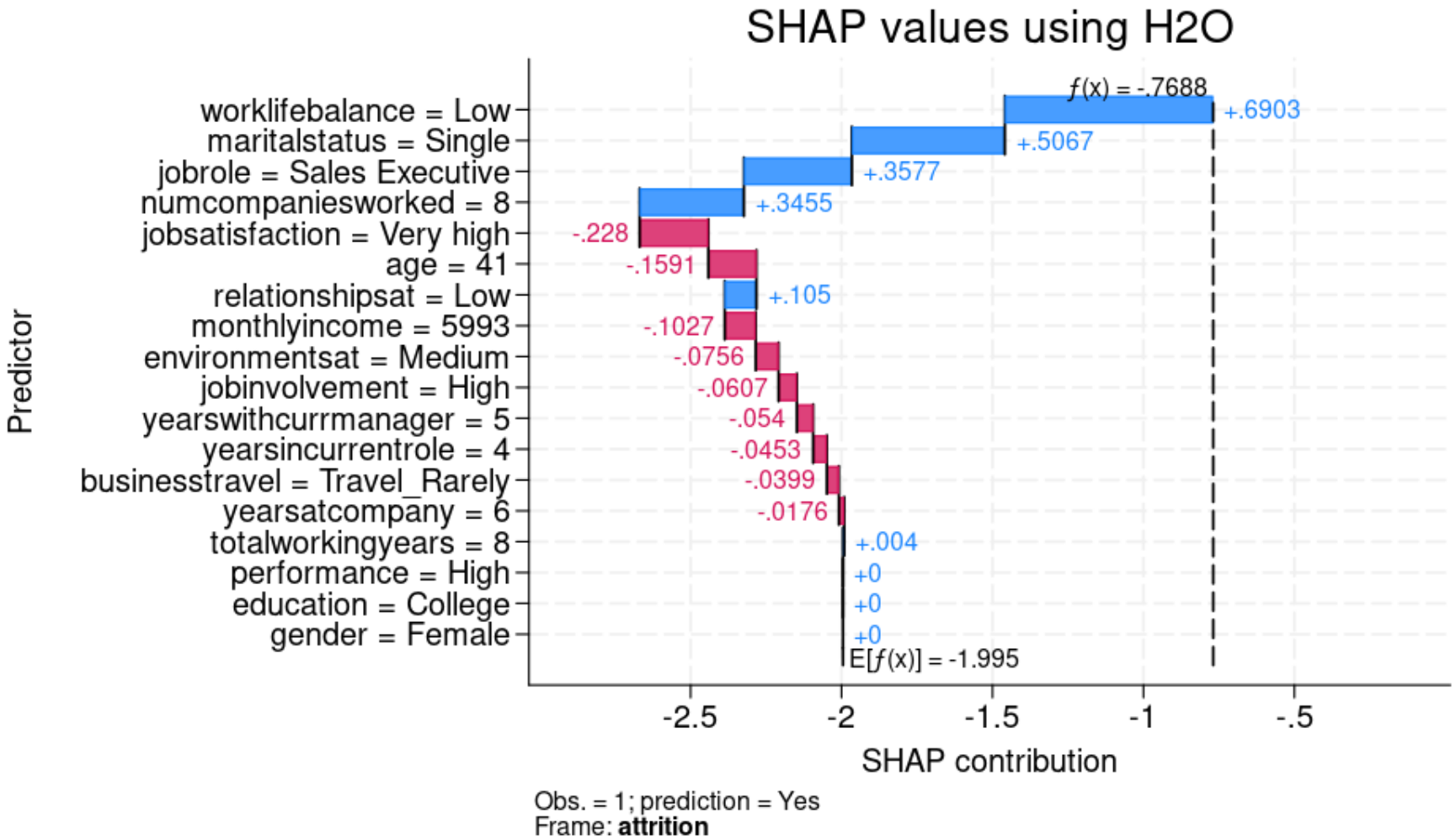


Shapley values

- The **mean marginal contribution** is the Shapley value of that member.
- This involves summing over $2^{|C|-1}$ combinations (NP-hard).
- Lundberg et al. (2020) proposed TREESHAP algorithm to mitigate the NP-hard problem for tree-based methods.
- Nice interactive blog post on TREESHAP: `https://hughchen.github.io/its_blog/index.html`

Hands-on: SHAP

```
. h2omlest restore gbm
(results gbm are active now)

. h2omlgraph shapvalues, obs(1) frame(attrition)
```



Hands-on: SHAP

- After GBM, when `loss()` is not Gaussian, the returned values are "raw" values.
- For binary classification, those values correspond to log odds.
- For details, see `h2oml gbm`.

- Check for $f(x) = -0.7688$

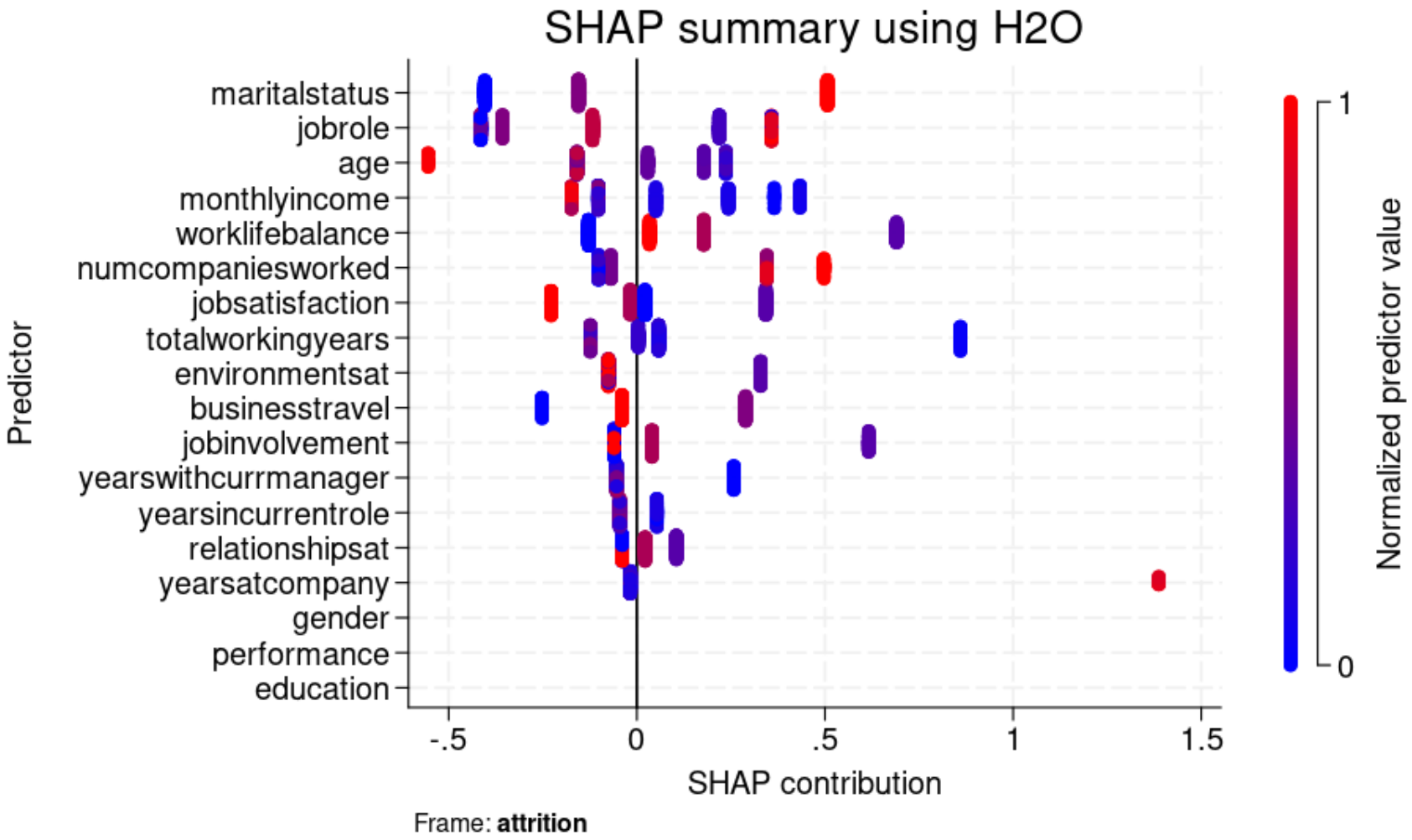
```
. display invlogit(-0.7688)
.31673875
```

- Check for $E[f(x)] = -1.995$

```
. h2omlpredict attr_train_pr*, frame(train) pr
Progress (%): 0 100
. _h2oframe generate logodds = log(attr_train_pr2)
. _h2oframe summarize logodds
. mat list r(mean)
symmetric r(mean)[1,1]
             c1
r1  -1.9951461
```

Hands-on: beeswarm plot

```
. h2omlgraph shapsummary, rseed(19) frame(attrition)
Progress (%): 0 100
```



Pros and Cons: SHAP

- Pros:
  - Solid theoretical foundation in game theory.
  - Contrastive explanations that compare the prediction with the average prediction.
  - Fast implementation for tree-based models.
- Cons:
  - When predictors are dependent, SHAP values might be unreliable.
  - SHAP values can be misinterpreted. A positive SHAP value doesn't mean that increasing the predictor would increase the prediction by that amount.
  - It is possible to create intentionally misleading interpretations with SHAP, which can hide biases (Rudin, 2019; Slack et al., 2020).

Global method: Partial Dependence Plot (PDP)

- PDP visualizes the average effect of a predictor on the prediction.
- For regression, the PDP graphs the average prediction versus the values of a predictor of interest.
- For classification, the PDP graphs the average of the predicted probabilities versus the values of a predictor of interest.

How does it work?

- Pick a predictor $\boldsymbol{x}_S$ (e.g., age) or set of predictors
- Vary this predictor across its range $\boldsymbol{x}_S$ (e.g., age = $20, 23, \cdots, 80$)
- For each value $\boldsymbol{x}_S^{(j)}$:
  - **For each observation $i = 1, \ldots, n$ in the dataset:**
    - Replace that observation's value of $\boldsymbol{x}_S$ with $\boldsymbol{x}_S^{(j)}$
    - Keep that observation's values for $\boldsymbol{X}_C$ unchanged (i.e., use $\boldsymbol{x}_{Ci}$)
    - Predict: $\widehat{f}(\boldsymbol{x}_S^{(j)}, \boldsymbol{x}_{Ci})$
  - Average predictions across all n observations:

$$\widehat{f}_S(\boldsymbol{x}_S^{(j)}) = \frac{1}{n}\sum_{i=1}^{n}\widehat{f}(\boldsymbol{x}_S^{(j)}, \boldsymbol{x}_{Ci})$$

- Plot the pairs $\{(\boldsymbol{x}_S^{(j)}, \widehat{f}_S(\boldsymbol{x}_S^{(j)}))\}$ for each grid point j

How does it work?

Income	Age	...	Attrition
5098	20	...	No
7096	34	...	Yes
4236	58	...	No
5136	63	...	Yes
...	...	...	...
...	...	...	...
5036	26	...	Yes
8236	80	...	Yes
...	...	...	...

How does it work?

Income	Age	...	Attrition
5098	20	...	No
7096	34	...	Yes
4236	58	...	No
5136	63	...	Yes
...	...	...	...
...	...	...	...
5036	26	...	Yes
8236	80	...	Yes
...	...	...	...

How does it work?

Income	Age	...	Attrition
5098	20	...	No
7096	34	...	Yes
4236	58	...	No
5136	63	...	Yes
...	...	...	...
...	...	...	...
5036	26	...	Yes
8236	80	...	Yes
...	...	...	...

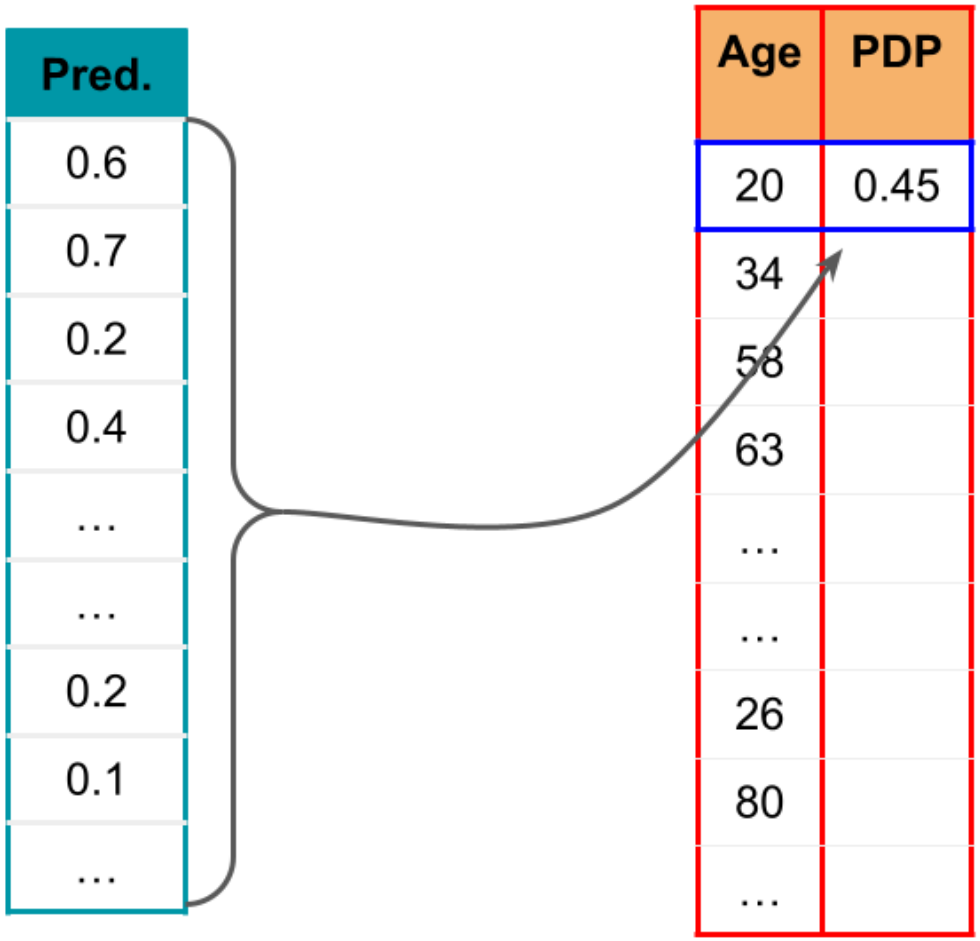
How does it work?

Income	Age	...	Attrition
5098	20	...	No
7096	20	...	Yes
4236	20	...	No
5136	20	...	Yes
...	...	...	...
...	...	...	...
5036	20	...	Yes
8236	20	...	Yes
...	...	...	...

How does it work?

Income	Age	…	Attrition	Pred.
5098	20	…	No	0.6
7096	20	…	Yes	0.7
4236	20	…	No	0.2
5136	20	…	Yes	0.4
…	…	…	…	…
…	…	…	…	…
5036	20	…	Yes	0.2
8236	20	…	Yes	0.1
…	…	…	…	…

How does it work?



How does it work?

Income	Age	...	Attrition
5098	20	...	No
7096	34	...	Yes
4236	58	...	No
5136	63	...	Yes
...	...	...	...
...	...	...	...
5036	26	...	Yes
8236	80	...	Yes
...	...	...	...

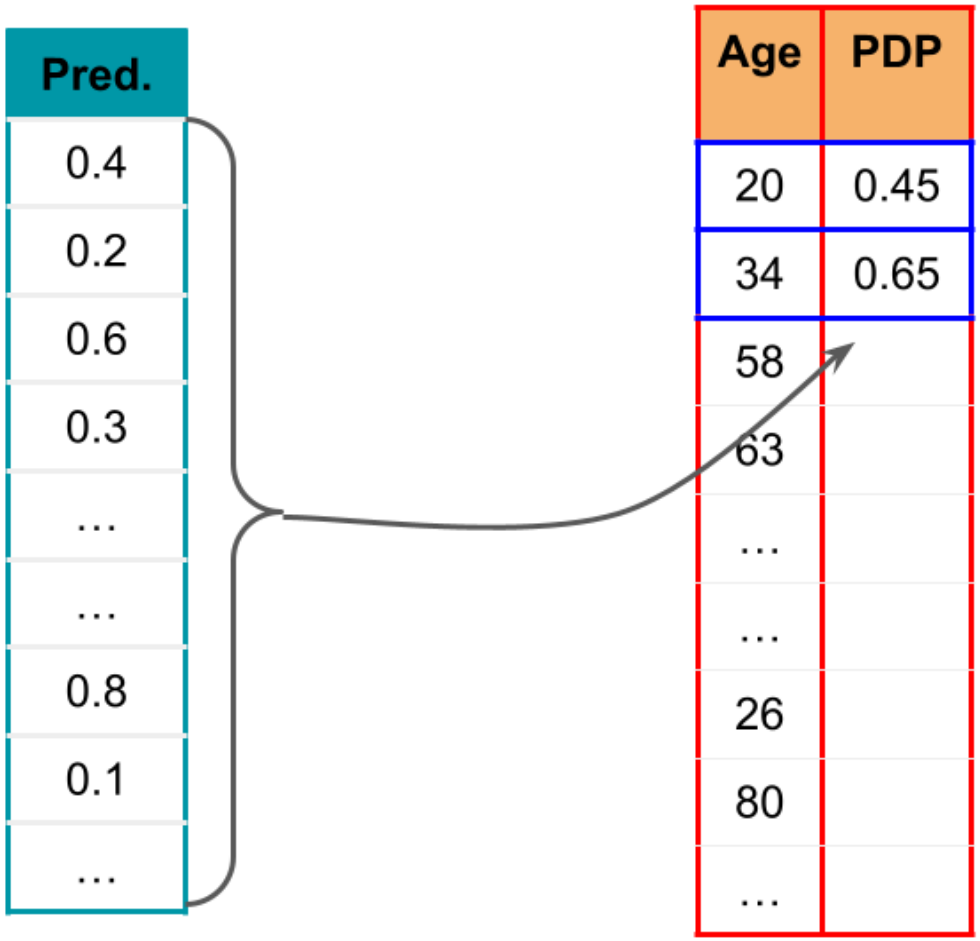
How does it work?

Income	Age	...	Attrition
5098	34	...	No
7096	34	...	Yes
4236	34	...	No
5136	34	...	Yes
...	...	...	...
...	...	...	...
5036	34	...	Yes
8236	34	...	Yes
...	...	...	...

How does it work?

Income	Age	...	Attrition	Pred.
5098	34	...	No	0.4
7096	34	...	Yes	0.2
4236	34	...	No	0.6
5136	34	...	Yes	0.3
...	...	...	...	...
...	...	...	...	...
5036	34	...	Yes	0.8
8236	34	...	Yes	0.1
...	...	...	...	...

How does it work?



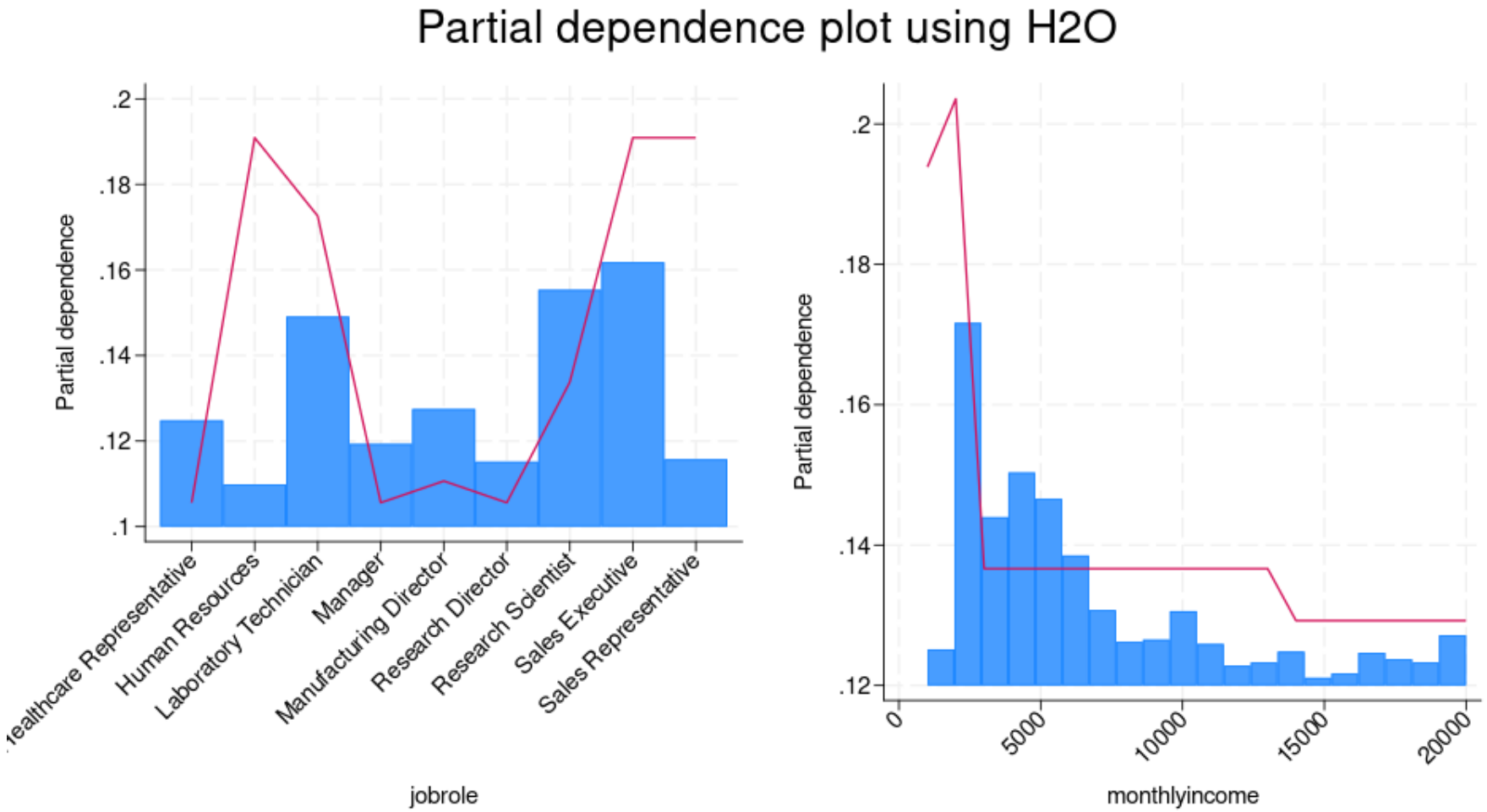
How does it work?

Income	Age	…	Attrition
5098	20	…	No
7096	34	…	Yes
4236	58	…	No
5136	63	…	Yes
…	…	…	…
…	…	…	…
5036	26	…	Yes
8236	80	…	Yes
…	…	…	…

Age	PDP
20	0.45
34	0.65
58	0.24
63	
…	
…	
26	
80	
…	

Hands-on PDP

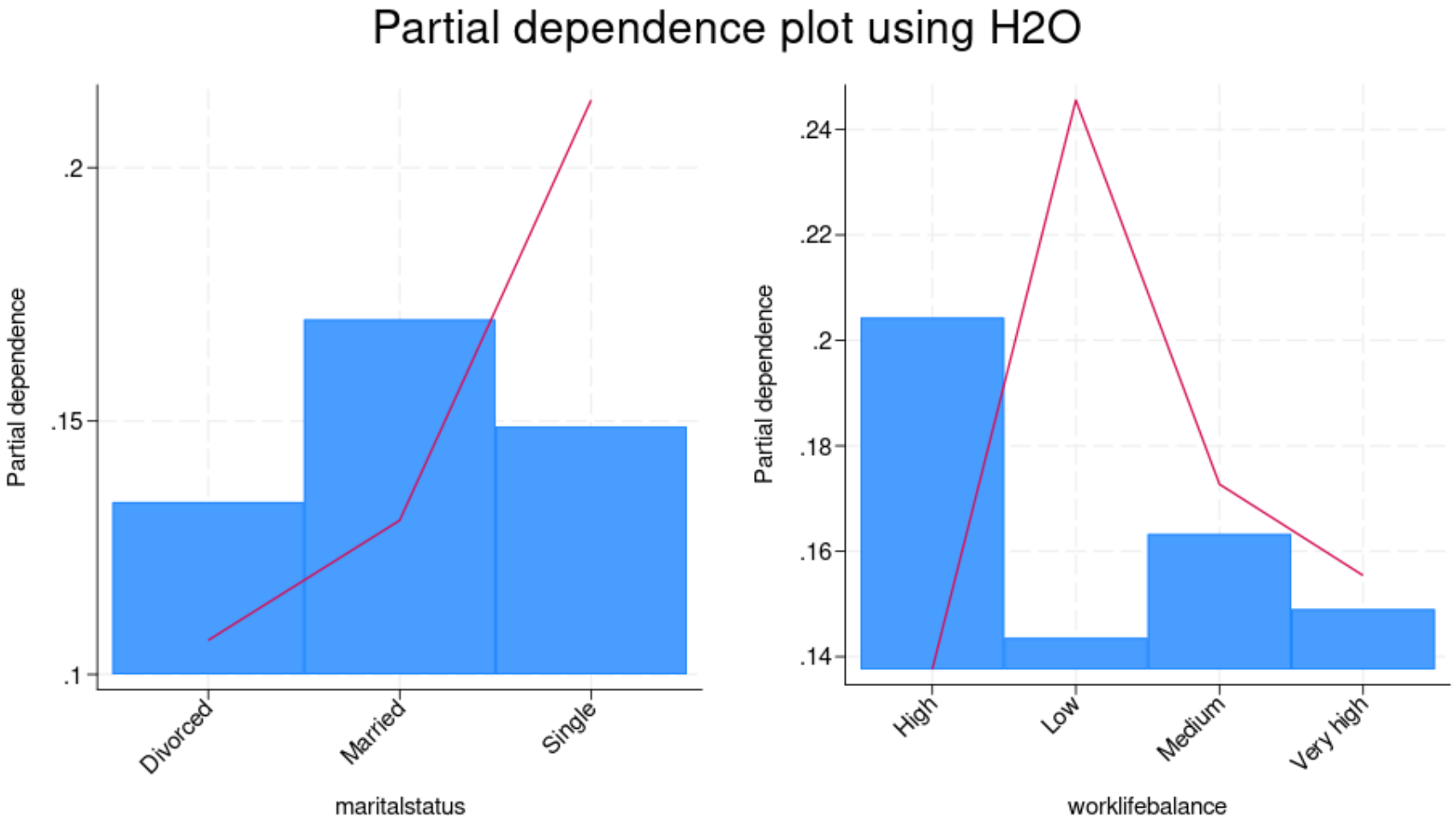
```
. h2omlgraph pdp jobrole monthlyincome, combineopts(cols(2)) ///
> xlabel(, angle(45)) frame(attrition)

Progress (%): 0 100
```



Frame: **attrition**

Hands-on PDP

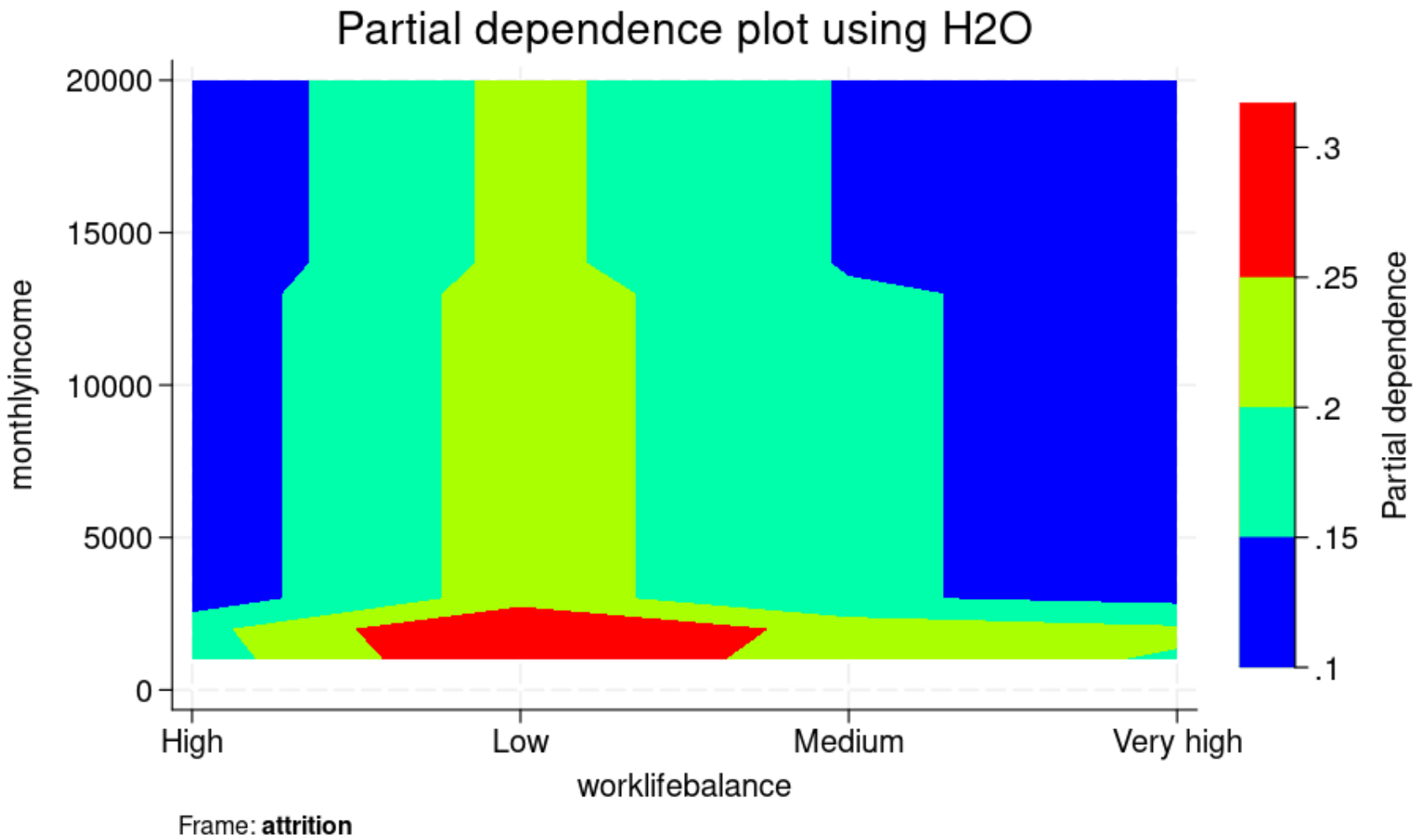
```
. h2omlgraph pdp maritalstatus worklifebalance, combine ///
> combineopts(cols(2)) xlabel(, angle(45)) frame(attrition)

Progress (%): 0 100
```



Frame: **attrition**

Hands-on PDP

```
. h2omlgraph pdp monthlyincome worklifebalance, pair frame(attrition)
Progress (%): 0 100
```



Pros and Cons: PDP

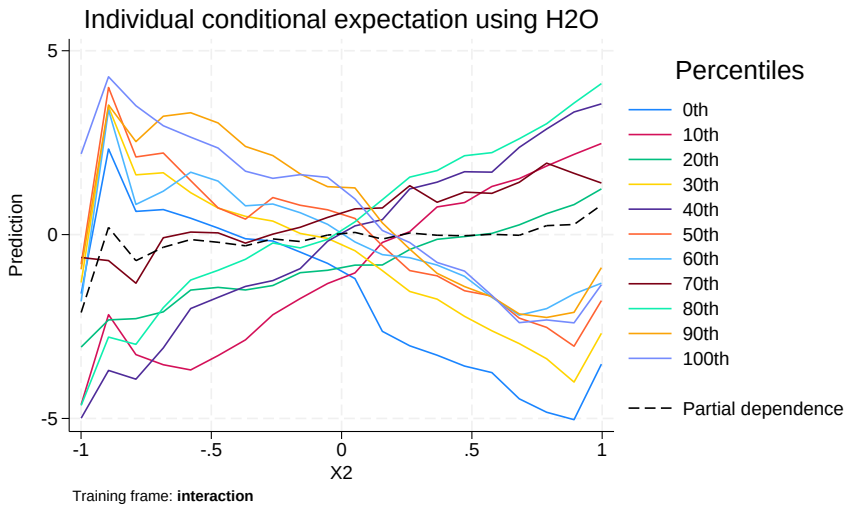
- Pros:
  - Intuitive computation and interpretations.
  - If a predictor is uncorrelated with other predictors, PDP represents how the predictor influences the prediction on average.
  - Under certain conditions (e.g. no backdoor path) has a causal interpretation (Zhao and Hastie, 2021).
- Cons:
  - The maximum number of predictors in a PD function that can be visualized is two or three.
  - Assumes that the predictors for which the PD is computed are **not correlated** with other predictors.
  - Heterogeneous effects might not be reflected in the PDP. For details, see ICE.

Local method: individual conditional expectation (ICE)

- PDP plots **the average** predictions across the values of a predictor of interest.
- When there is an interaction effect among predictors, the PDP cannot fully capture the effect.
- There may be no average effect, flat curve in the PDP, while there are substantial effects at various levels of the predictor.

$$Y = 0.2X_1 + 5X_2 + \varepsilon \text{ if } X_3 \geq 0$$

$$Y = 0.2X_1 - 5X_2 + \varepsilon \text{ otherwise}$$



ICE

- ICE is like PDP, but instead of showing the average effect of a predictor, it shows the effect for individual cases (i.e., deciles).
- It helps detect if the model treats different observations (e.g., employee) differently, even for the same predictor.

How does it work?

- Pick a predictor $\boldsymbol{x}_S$ (e.g., monthly income)
- Define a grid of values for $\boldsymbol{x}_S$ (e.g., income ranging from \$30k to \$150k, using deciles or evenly spaced points)
- For **each employee** i:
  - Vary their value of $\boldsymbol{x}_S$ across the grid
  - Keep all their other predictors $\boldsymbol{x}_{Ci}$ fixed at their observed values
  - Predict the outcome for each grid value: $\widehat{f}(\boldsymbol{x}_S^{(j)}, \boldsymbol{x}_{Ci})$ for $j = 1, \ldots, m$ grid points
  - This creates an individual curve for employee i
- Plot all individual curves on the same graph

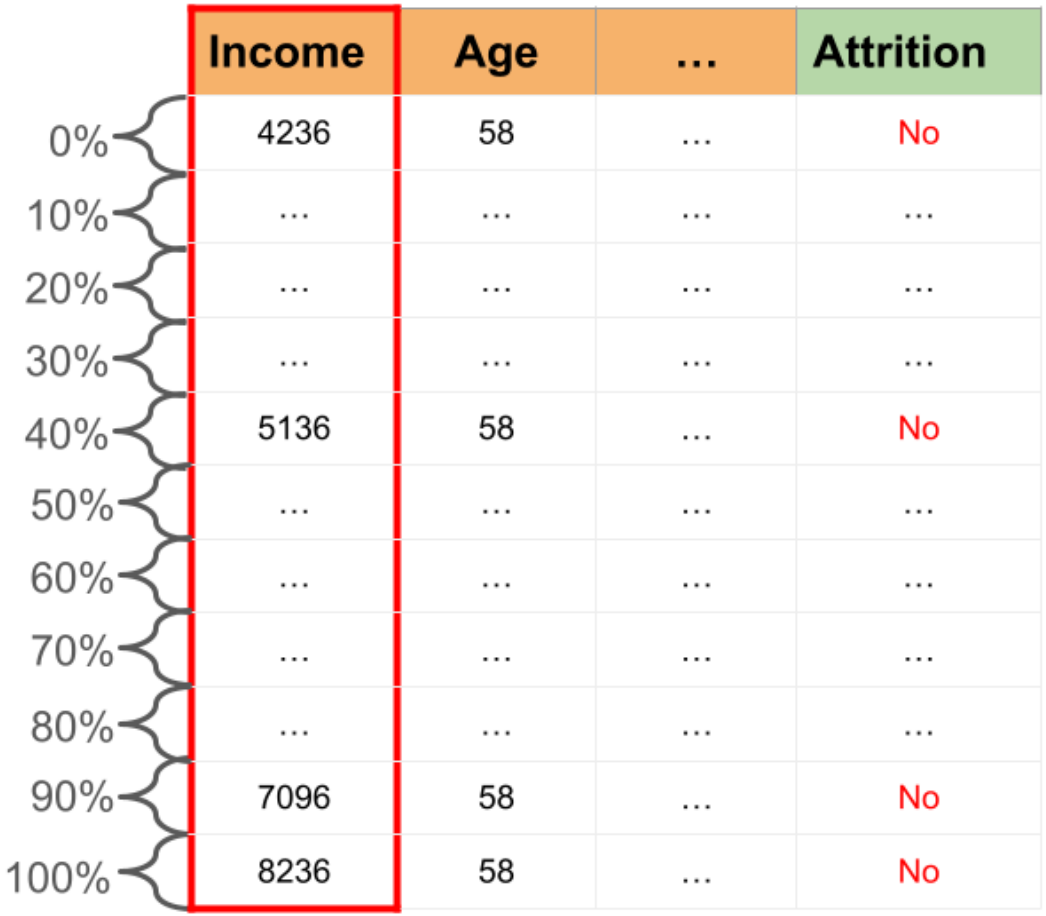
How does it work?

Income	Age	...	Attrition
4236	58	...	No
5098	20	...	No
...	...	...	...
5136	63	...	Yes
...	...	...	...
...	...	...	...
7096	34	...	Yes
...	...	...	...
8236	80	...	Yes

How does it work?

	Income	Age	…	Attrition
0%	4236	58	…	No
	5098	20	…	No
	…	…	…	…
40%	5136	63	…	Yes
	…	…	…	…
	…	…	…	…
90%	7096	34	…	Yes
	…	…	…	…
100%	8236	80	…	Yes

How does it work?

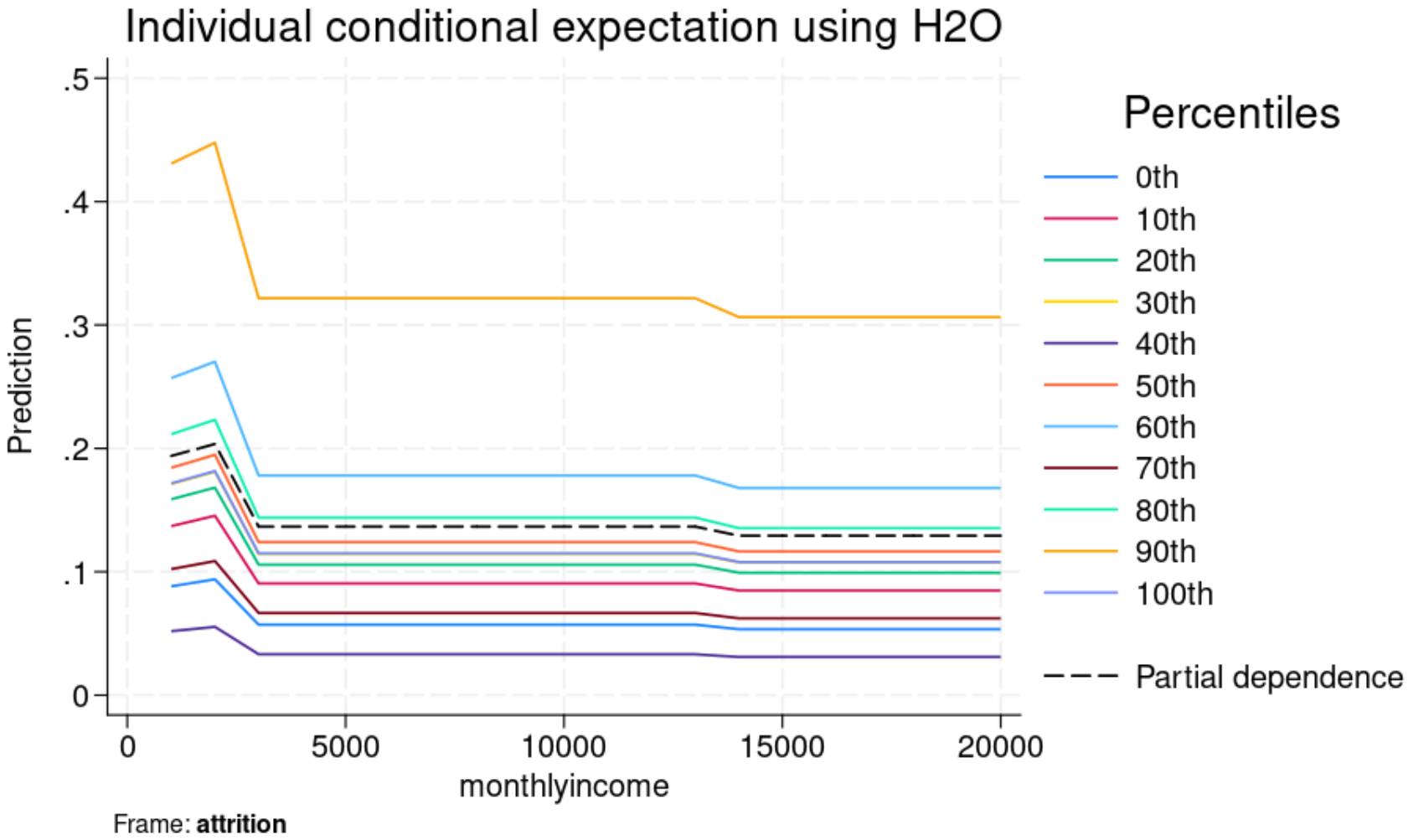


How does it work?

	Income	Age	…	Attrition	Pred.
0%	4236	58	…	No	0.3
10%	5098	58	…	No	0.6
20%	…	…	…	…	…
30%	…	…	…	…	…
40%	5136	58	…	No	0.2
50%	…	…	…	…	…
60%	…	…	…	…	…
70%	…	…	…	…	…
80%	…	…	…	…	…
90%	7096	58	…	No	0.7
100%	8236	58	…	No	0.65

Hands-on: ICE

```
. h2omlgraph ice monthlyincome, frame(attrition)
```

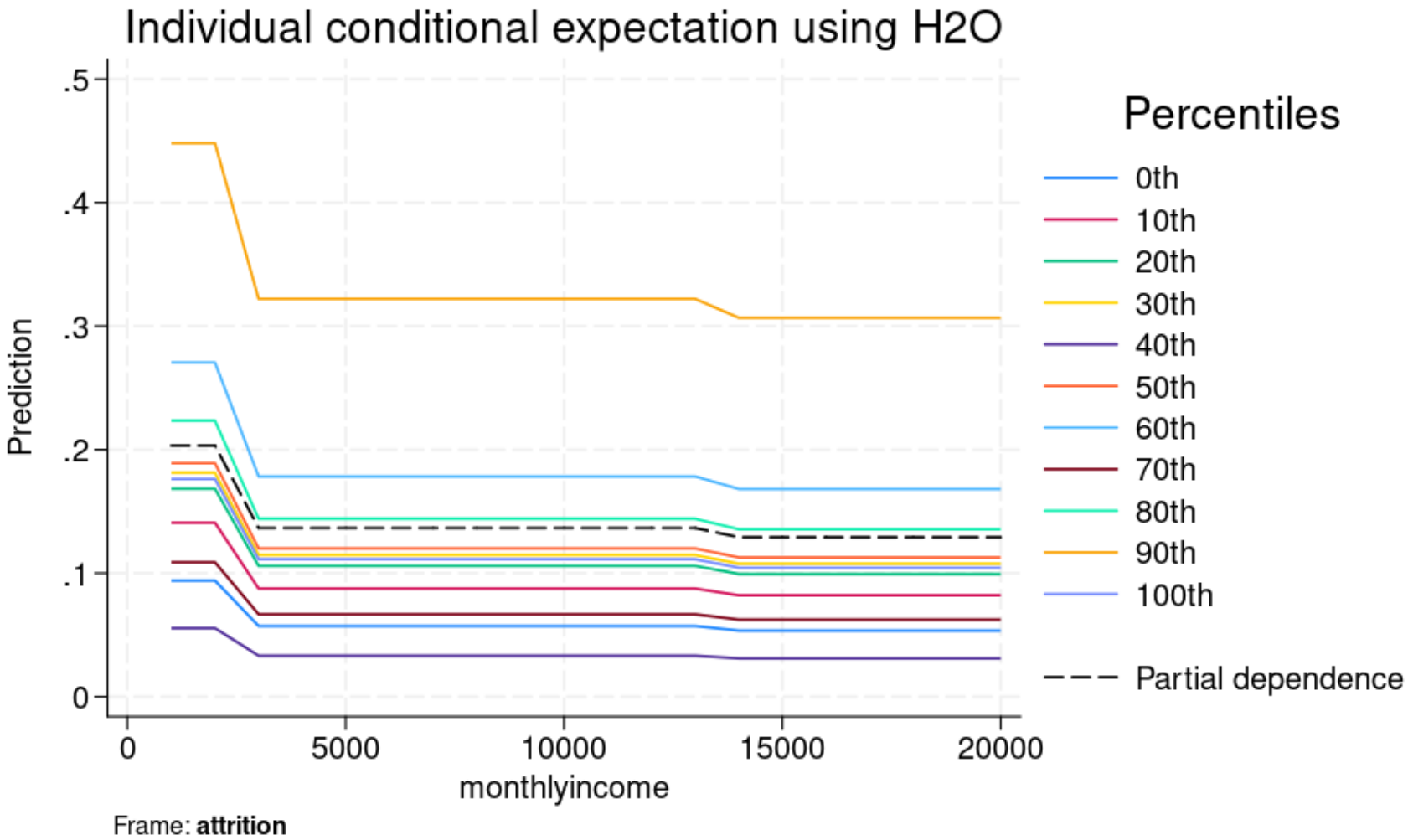


Hands-on: monotone

```
. h2oml gbbinclass attrition $predictors, h2orseed(19) ///
      ntrees(150) maxdepth(1) predsamprate(0.4) ///
      monotone(monthlyincome,decreasing)

(output omitted)
```

Hands-on: monotone

```
. h2omlgraph ice monthlyincome, frame(attrition)
```



Pros and Cons: ICE

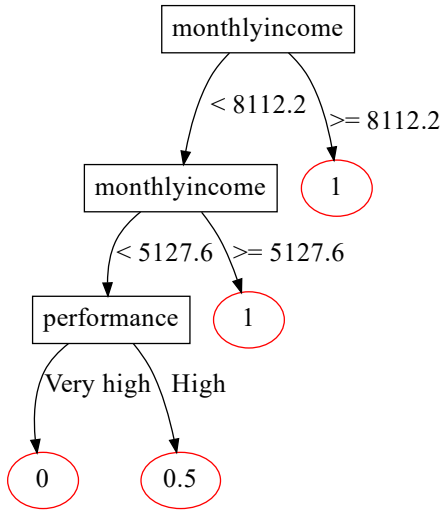
- Pros:
  - ICE is intuitive to understand. One line represents the predictions for one decile if we vary the predictor of interest.
  - Uncover heterogeneous relationships.
- Cons:
  - ICE curves can only display one predictor meaningfully. No pair plots.
  - If the predictor of interest is correlated with the other predictors, then some points in the lines might be invalid.

Global method: Variable importance

- Variable importance identifies the most influential predictors.
- Two approaches:
  - Impurity-based importance: `h2omlgraph varimp`
  - Permutation-based importance: `h2omlgraph permimp`

Impurity-based importance

- Quantifies how much each predictor contributes to **splitting the training data** and **reducing** uncertainty using entropy or MSE.
- Reflects how frequently a predictor is used to split the data.
- But not necessarily **how essential** that predictor is for improving prediction accuracy.
- Can be biased toward predictors with more categories.



Permutation-based importance

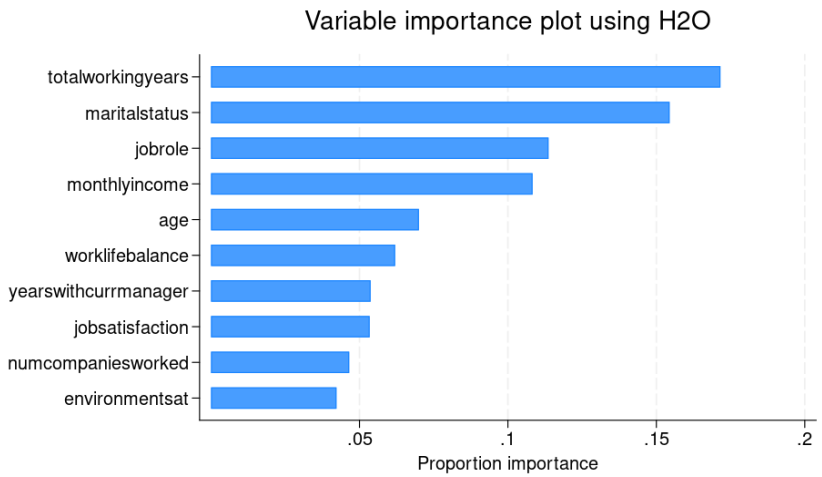
- Designed to answer a more fundamental question: **What happens to a prediction when we break the relationship between a predictor and the response?**
- Predictor is considered important if shuffling its values affects the model's ability to make accurate predictions.
- Permutation variable importance permutes one predictor at a time and measures the resulting change in model performance.

How it works?

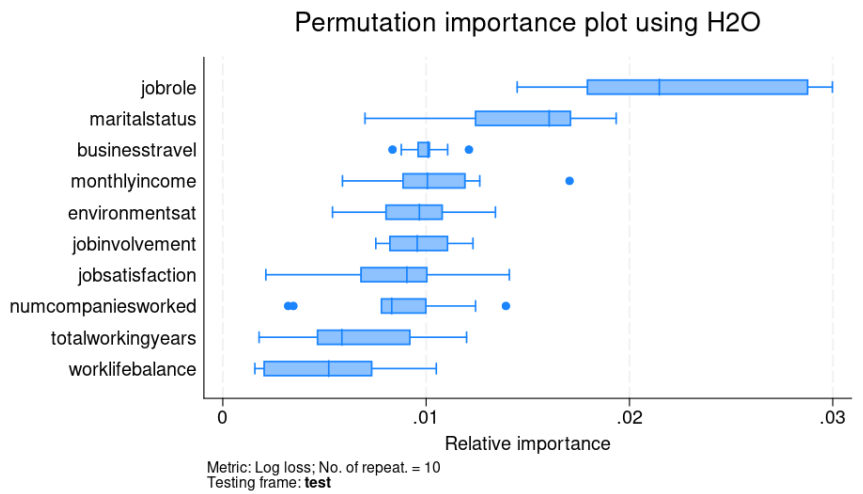
1. For $j = 1, 2, \ldots, B$,
   i. Randomly shuffle predictor X_i for $1 \leq i \leq p$
   ii. Use the machine learning model (e.g. GBM) to generate predictions on the permuted data.
   iii. Compute the new performance metric M_j^{perm}.
   iv. Compute $|M^{\text{orig}} - M_j^{\text{perm}}|$.

Hands-on: Impurity vs permutation-based importance

```
. h2omlgraph varimp
```




```
. h2omlgraph permimp, h2orseed(19)
```



Pros and Cons: Impurity-based importance

- Pros:
  - Calculated during model training. There is no need for additional computation.
  - Helps in early model diagnostics and predictor selection.
- Cons:
  - Biased towards predictors with more categories.
  - Can be misleading if the model is overfitting.

Pros and Cons: Permutation-based importance

- Pros:
  - Can be used with any predictive model.
  - Tends to have less bias toward predictors with many categories.
  - Directly linked to model performance. Shows how much each predictor changes the performance.
  - Takes into account both the main effect and the interaction effects. By permuting the predictor, we destroy the interaction effects with other predictors.
- Cons:
  - Computationally expensive: Requires multiple evaluations of the model.
  - Sensitive to correlated predictors. If two predictors are correlated, the importance of both can be underestimated.
  - Depends on the testing data.
  - The importance of the interaction between two predictors is included in the importance measurements of both predictors.

Surrogate models

- Surrogate models approximate the predictions of a black-box model.
- It uses an interpretable model to explain a black-box model.
- It contains the following steps:
  1. Obtain predictions from a well-tuned black-box model fit to the testing data.
  2. Select and train an interpretable model (for example, a decision tree) for predictions on the testing data.
  3. Measure the goodness of fit of the surrogate model for the predictions, and interpret the model.

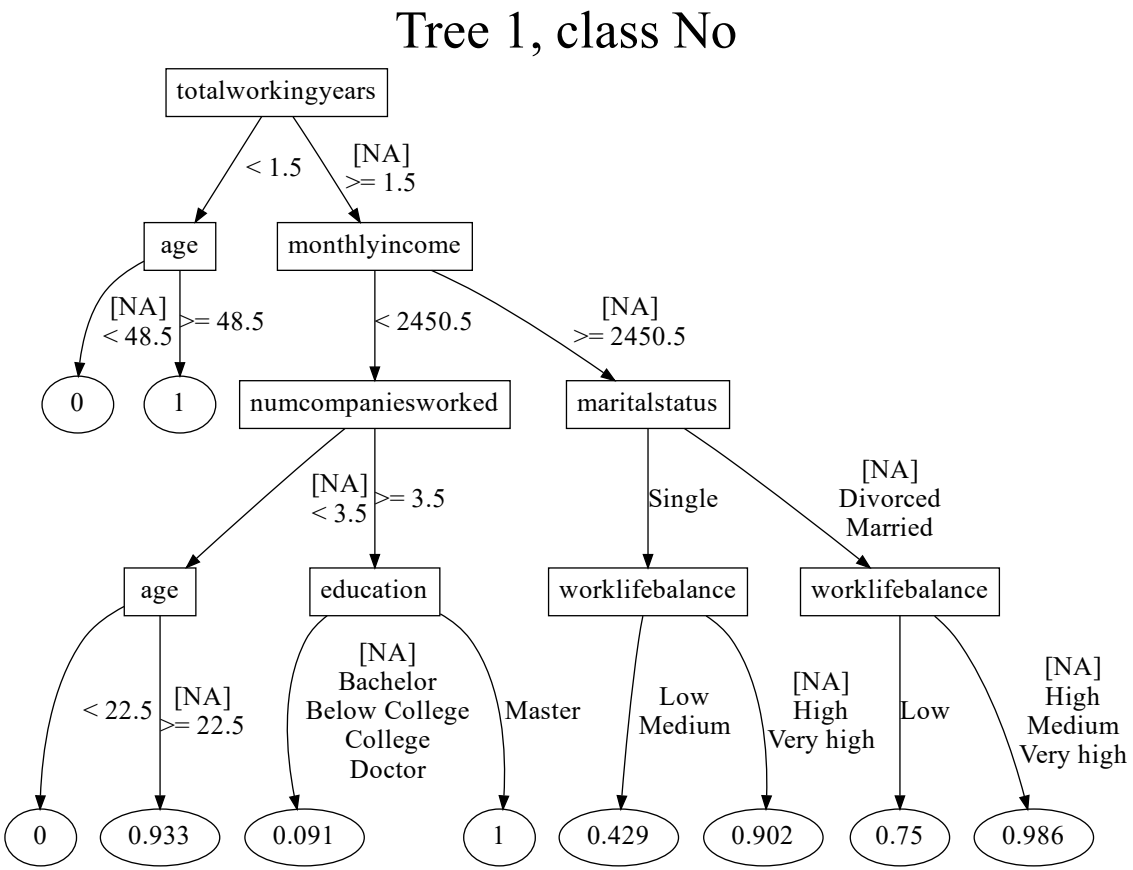
Hands-on surrogate models

```
. _h2oframe change test

. h2oml rfbinclass attr_hat $predictors, ntrees(1) h2orseed(19) ///
> predsampvalue(-2) maxdepth(4)

Progress (%): 0 100

Random forest binary classification using H2O

Response: attr_hat
Frame:                                  Number of observations:
  Training: test                                  Training =    174

Model parameters

Number of trees      =   1
              actual =   1
Tree depth:                             Pred. sampling value =     -2
           Input max =   4              Sampling rate        =   .632
                 min =   4              No. of bins cat.     =  1,024
                 avg = 4.0              No. of bins root     =  1,024
                 max =   4              No. of bins cont.    =     20
Min. obs. leaf split =   1              Min. split thresh.   = .00001

Metric summary
```

Metric	Training
Log loss	.817041
Mean class error	.3404366
AUC	.7370062
AUCPR	.4471749
Gini coefficient	.4740125
MSE	.1236771
RMSE	.3516776

Hands-on surrogate models

```
. h2omltree, id(1) dotsaving(surrogate, replace)
. shell dot -Tpdf surrogate.dot -o surrogate.pdf
```



Requires `Graphviz`. For details, see DOT extension.

Pros and Cons: Surrogate models

- Pros:
  - An interpretable model can be used.
  - Intuitive and straightforward approach,
- Cons:
  - The surrogate model **never sees** the real response.
  - It draws conclusions about the model and not about the data.
  - The chosen interpretable model as a surrogate comes with all its advantages and disadvantages.
  - The interpretation for the simple model would not be equally good for all data points.

Helpful links I

- More on setting up H2O in Stata:
  https://www.stata.com/manuals/h2omlh2osetup.pdf
- More on `h2o` and `_h2oframe` commands, visit
  https://www.stata.com/h2o/h2o19/
- More on `h2oml` suite, visit
  https://www.stata.com/manuals/h2oml.pdf
- Also read our blog posts:
  - Approximate statistical tests for comparing binary classifier error rates using H2OML
  - Prediction intervals with gradient boosting machine
  - Heterogeneous treatment-effect estimation with S-, T-, and X-learners using H2OML

Others: Explainable ML is hard
Me: Using h2oml in Stata

References I

Biecek, P. and Samek, W. (2024). Position: explain to question not to justify. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Breiman, L. (2001). Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical Science*, 16(3):199 – 231.

Lipton, Z. C. (2018). The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57.

Lundberg, S., Erion, G., Chen, H., DeGrave, A., Prutkin, J., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2:56–67.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS17, pages 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

References II

Molnar, C. (2025). *Interpretable Machine Learning:A Guide for Making Black Box Models Explainable*. 3 edition.

Ozkurt, C. (2024). Enhancing financial decision-making: Predictive modeling for personal loan eligibility with gradient boosting, xgboost, and adaboost. *ADBA Information Technology and Publishing Limited Company*, 1.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1135–1144, New York, NY, USA. Association for Computing Machinery.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1:206–215.

References III

Slack, D., Hilgard, S., Jia, E., Singh, S., and Lakkaraju, H. (2020). Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, AIES 20, pages 180–186, New York, NY, USA. Association for Computing Machinery.

Zhao, Q. and Hastie, T. (2021). Causal interpretations of black-box models. *Journal of Business & Economic Statistics*, 39(1):272–281.

Thank you!