Reinforcement Learning in Digital Finance

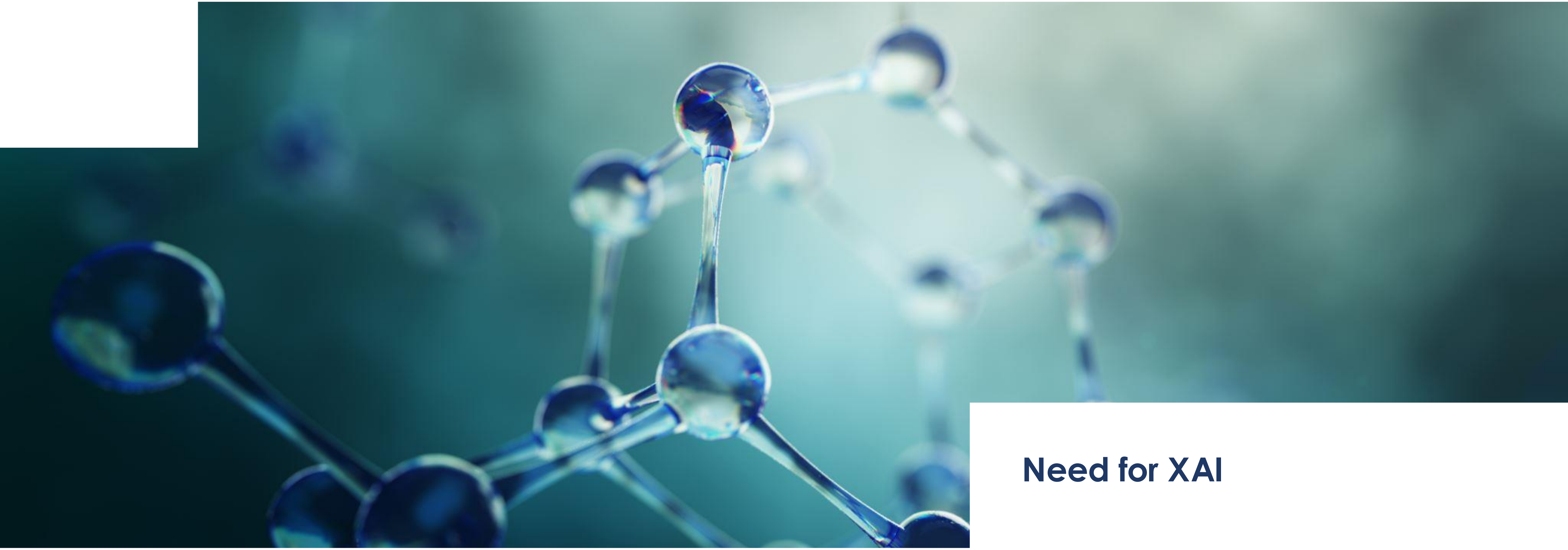
Introduction to Explainable AI

hey.

Prof. Dr. **Branka** Hadji Misheva
Bern University of Applied Science **(BFH)**

**Funded by
the European Union**

F
H
Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences



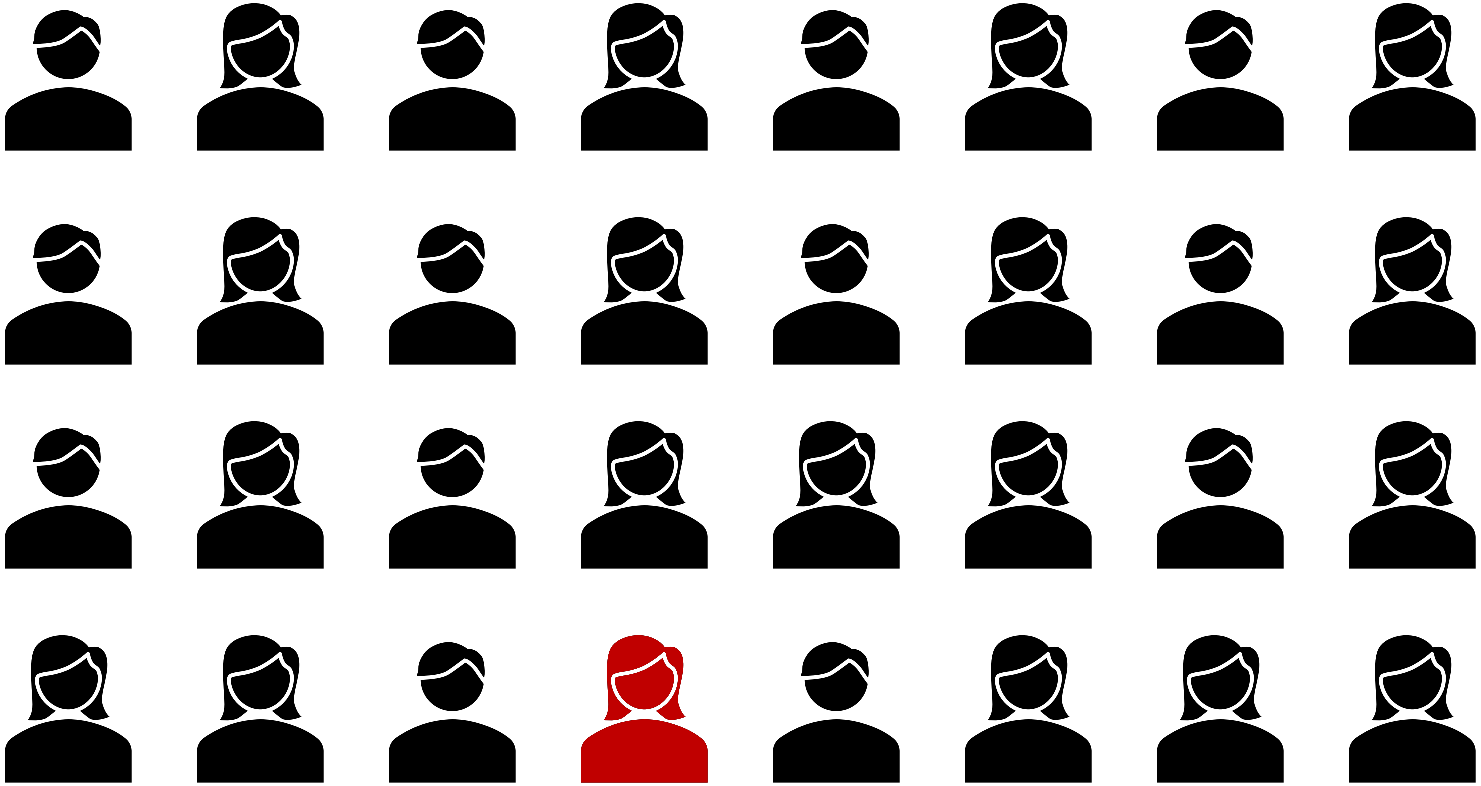
Need for XAI

2018 NeurIPS Explainable ML Challenge

- **NeurIPS** is one of the world's most well-known and prestigious machine learning conferences.

> *'Suppose you have a tumour and need surgery.* ***Would you rather trust an AI surgeon who cannot tell anything about its inner workings but has a 2% chance of making a fatal mistake or a human surgeon who can explain every step in detail but has a 15% chance of making a fatal mistake?'***

NEURAL INFORMATION PROCESSING SYSTEMS



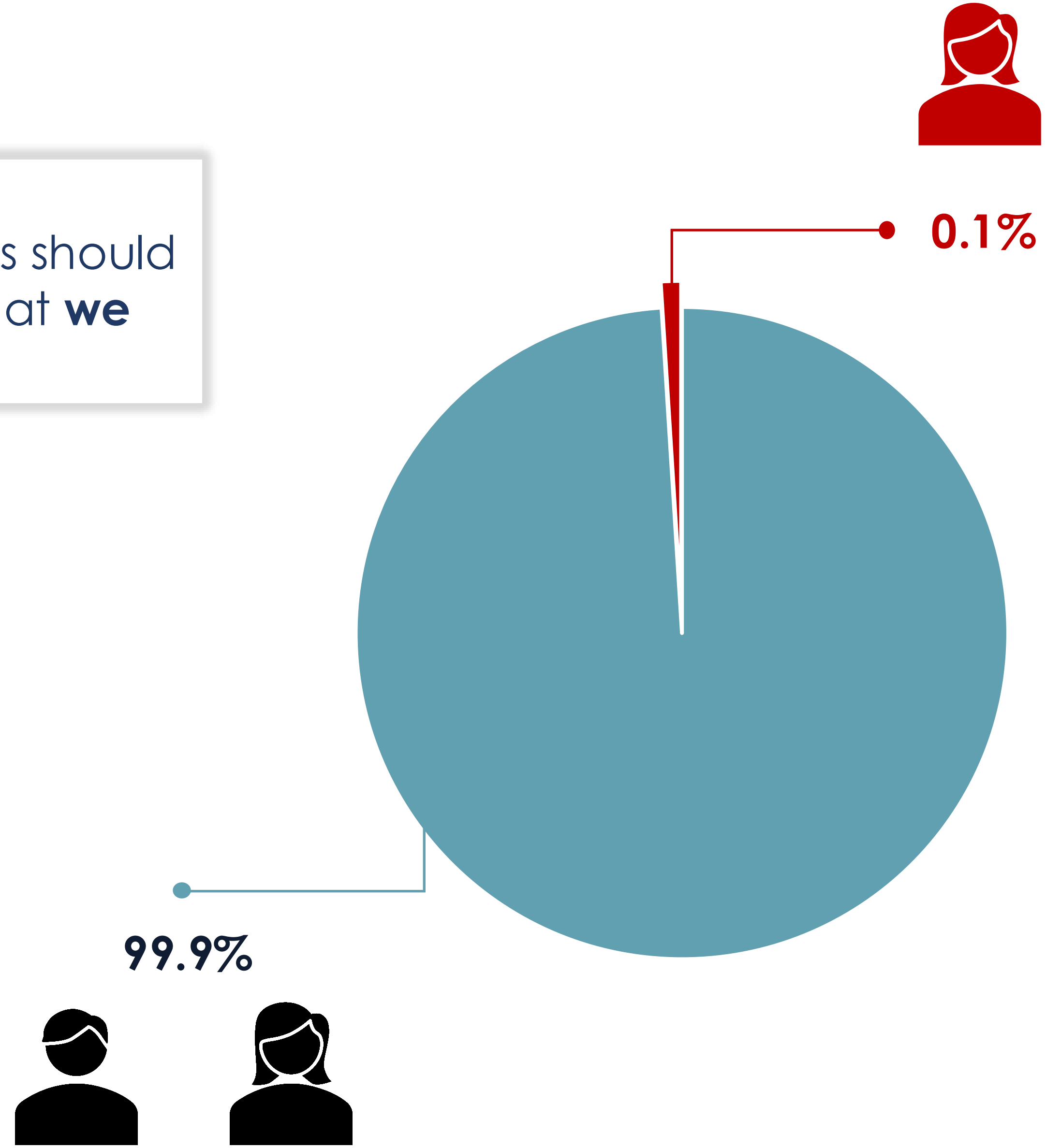
Robotics

ML

Finance

What % of the audience you think chose the human surgeon?

Accurate (desirable) performance of models should be a good indicator that **we can trust the model**



Hypothesis

Accurate (desirable) performance of models should be a good indicator that **we can trust the model**

Predicted: Husky
True: Husky

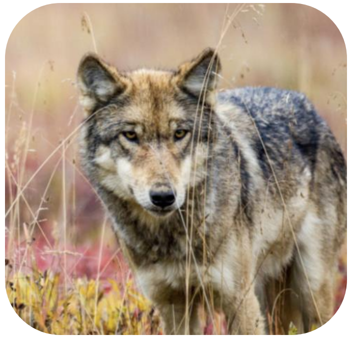
Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Husky
True: Husky



Predicted: Husky
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Husky
True: Husky

Hypothesis

Accurate (desirable) performance of models should be a good indicator that **we can trust the model**

Predicted: Husky
True: Husky

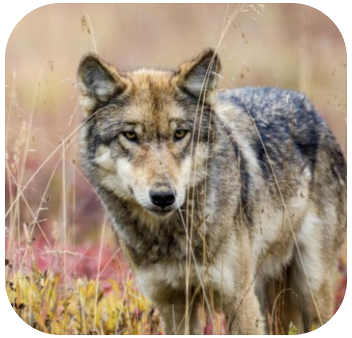
Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Husky
True: Husky



Predicted: Husky
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Husky
True: Husky

Hypothesis

Accurate (desirable) performance of models should be a good indicator that **we can trust the model**

Predicted: Husky
True: Husky

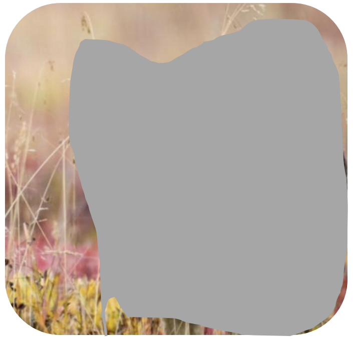
Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Husky
True: Husky



Predicted: Husky
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Husky
True: Husky

Accurate (desirable) performance of models should be a good indicator that **we can trust the model** ✗

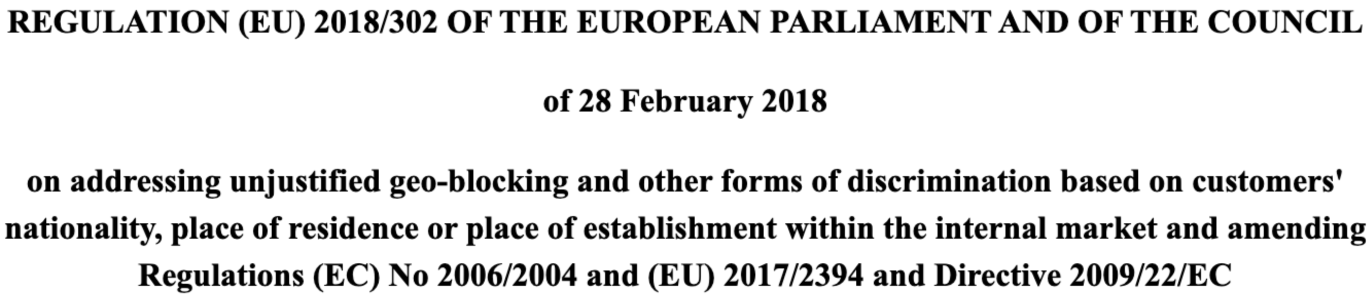
Accurate (desirable) performance of models should be a good indicator that **we can trust the model** ✗

Accurate (desirable) performance of models should be the result of the **model capturing true dependencies** ✓

Imagine that you **work for an insurance company that wants to adopt an AI system to determine car insurance premiums for its clients**.

To start, **what data** do you think should be used in such an AI system?

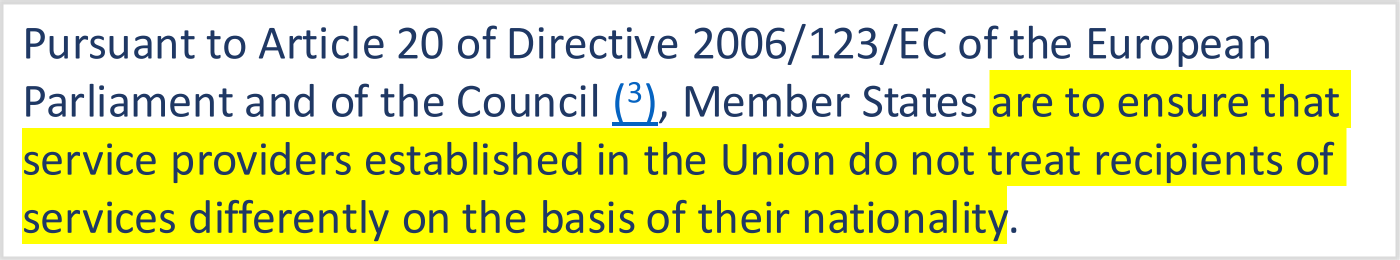
Should **nationality** be considered when estimating premiums?



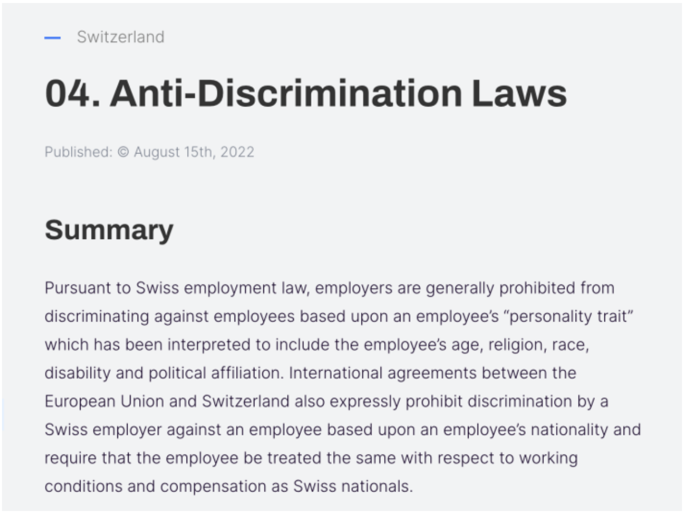
REGULATION (EU) 2018/302 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 28 February 2018

on addressing unjustified geo-blocking and other forms of discrimination based on customers' nationality, place of residence or place of establishment within the internal market and amending Regulations (EC) No 2006/2004 and (EU) 2017/2394 and Directive 2009/22/EC



Pursuant to Article 20 of Directive 2006/123/EC of the European Parliament and of the Council [3], Member States are to ensure that service providers established in the Union do not treat recipients of services differently on the basis of their nationality.



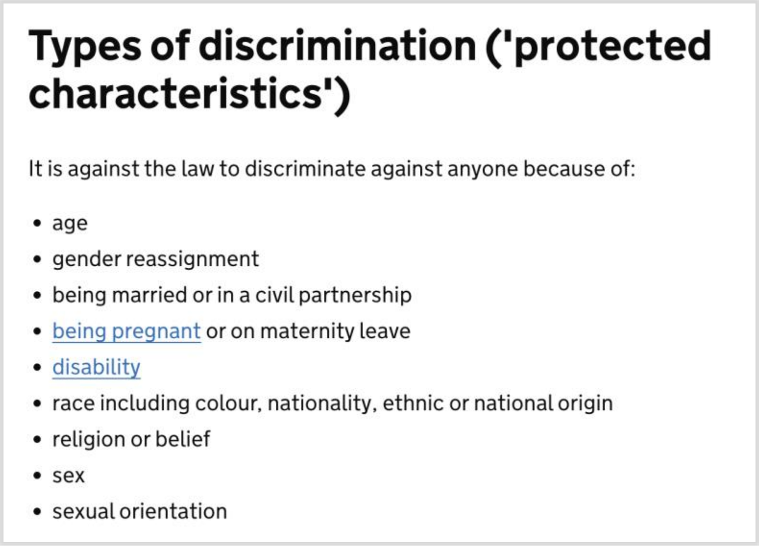
— Switzerland

04. Anti-Discrimination Laws

Published: © August 15th, 2022

Summary

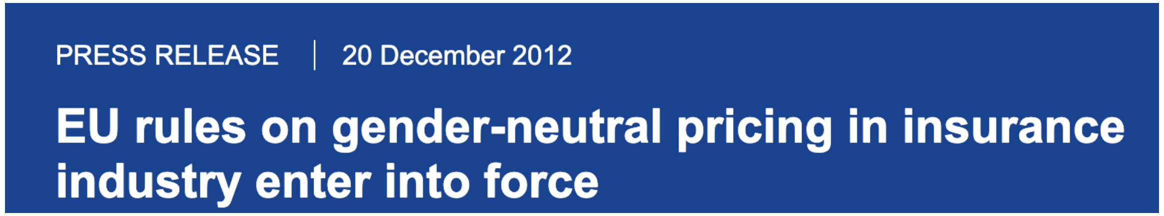
Pursuant to Swiss employment law, employers are generally prohibited from discriminating against employees based upon an employee's "personality trait" which has been interpreted to include the employee's age, religion, race, disability and political affiliation. International agreements between the European Union and Switzerland also expressly prohibit discrimination by a Swiss employer against an employee based upon an employee's nationality and require that the employee be treated the same with respect to working conditions and compensation as Swiss nationals.



Types of discrimination ('protected characteristics')

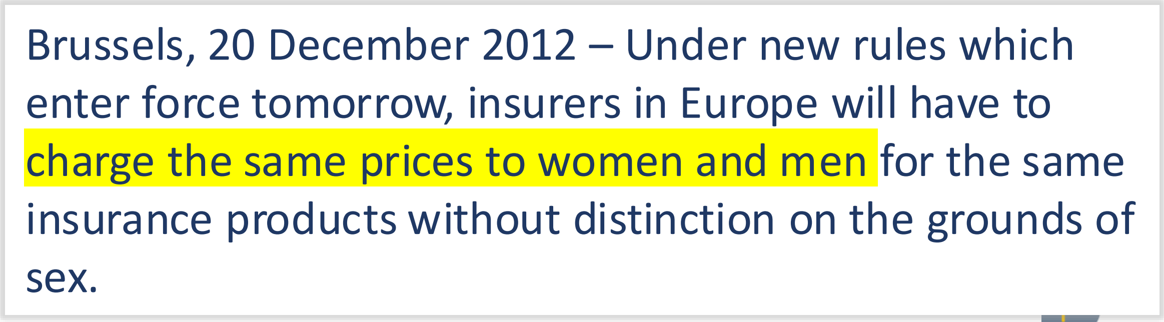
It is against the law to discriminate against anyone because of:

- age
- gender reassignment
- being married or in a civil partnership
- being pregnant or on maternity leave
- disability
- race including colour, nationality, ethnic or national origin
- religion or belief
- sex
- sexual orientation



PRESS RELEASE | 20 December 2012

EU rules on gender-neutral pricing in insurance industry enter into force



Brussels, 20 December 2012 – Under new rules which enter force tomorrow, insurers in Europe will have to charge the same prices to women and men for the same insurance products without distinction on the grounds of sex.

Demographic information

Eg. Age, marital status, location

Driving history

Eg. Years of driving, past claims, accident records, traffic violations)

Vehicle information

Eg. Make and model of car, age, mileage

Location data

Eg. Residential area, garage location

Is there any inputs which might be a **proxy** for a sensitive feature?

Demographic information

Eg. Age, marital status, **location**

Driving history

Eg. Years of driving, past claims, accident records, traffic violations)

Vehicle information

Eg. Make and model of car, age, mileage

Location data

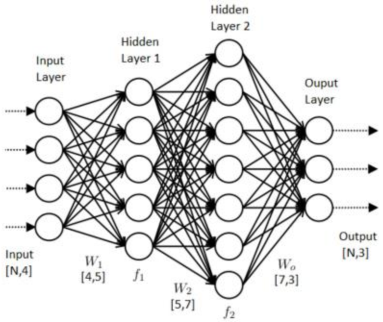
Eg. **Residential area, garage location**

Even if nationality is not directly used, **other features might act as proxies**, **indirectly introducing bias**.

Explainability can help developers understand the innerworkings of the models better.

What we want to **ACHIEVE**?

Training data



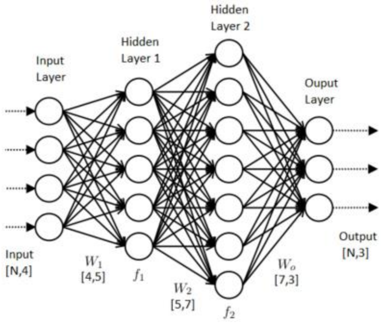
Trained model

Predicted: Husky
p: 94%

Output

What we want to **ACHIEVE**?

Training data



Trained model

Predicted: Husky
Because: broad skull, short snout, almond-shaped and blue eyes

Output

Deploying eXplainability

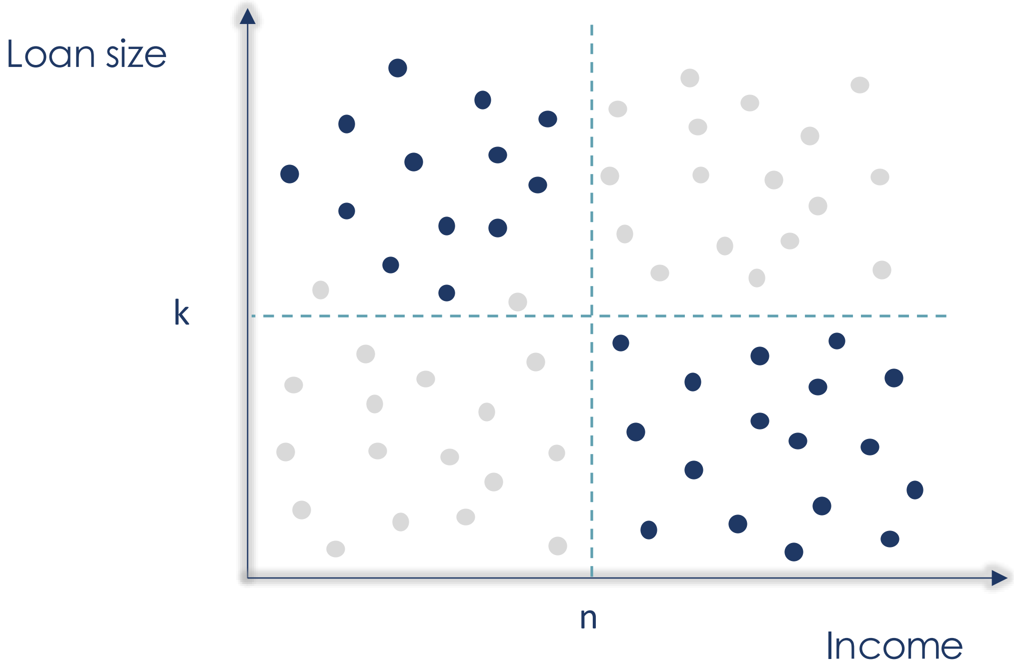
WHEN is explainability an issue?

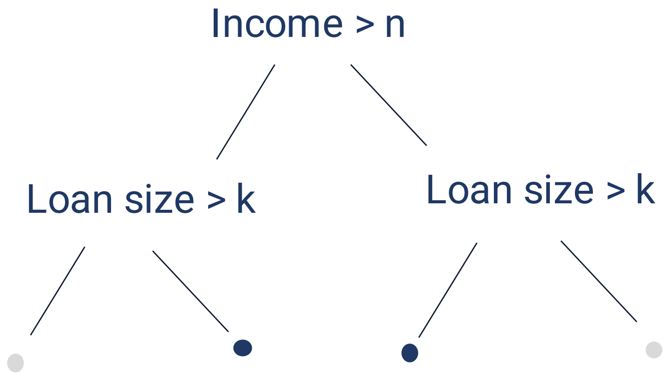
Credit risk managem

WHEN is explainability an issue?

What about **non-linear relationships**?

Still interpretable!





N-dimensions and **HIGH COMPLEXITY**

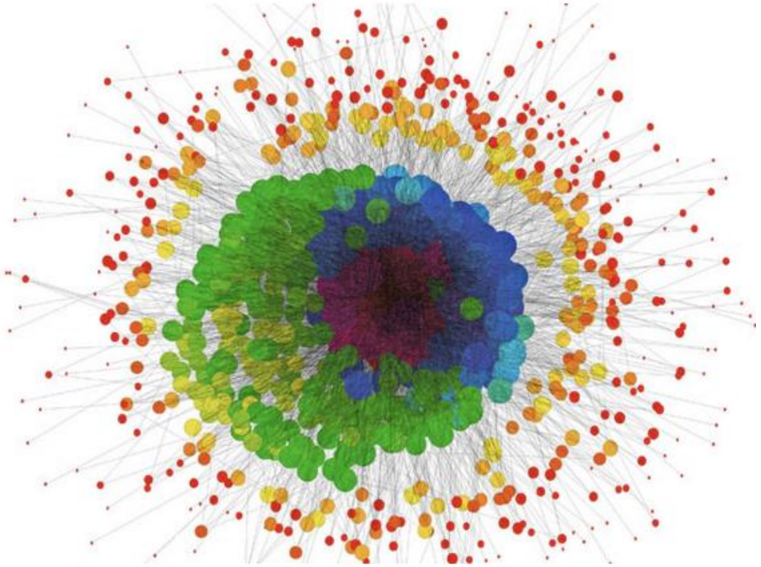


Image source: https://www.datanami.com/

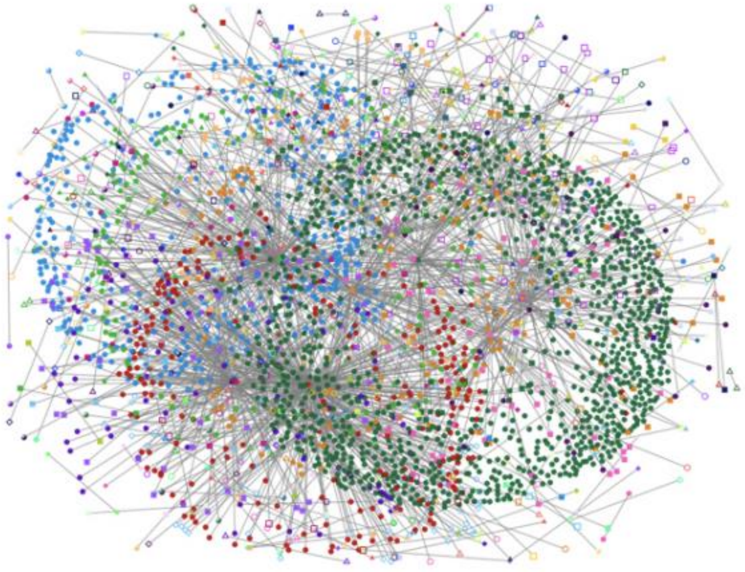


Image source: towardsdatascience.com

TIMELINE

TIMELINE

1990's	**Features of simple models** LR/DT

TIMELINE

1990’s	**Features of simple models** LR/DT
2000’s	**Feature importance**, can be used on any model

TIMELINE

1990's	**Features of simple models** LR/DT
2000's	**Feature importance**, can be used on any model
2017's	**LIME & SHAP**, model-agnostic feature attribution

TIMELINE

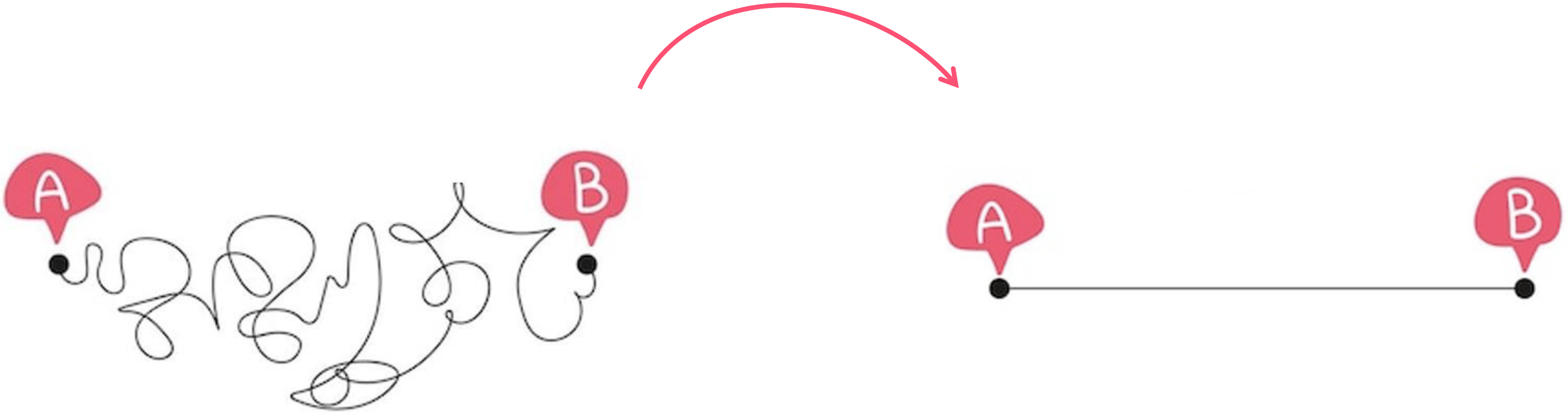
1990's	**Features of simple models** LR/DT
2000's	**Feature importance**, can be used on any model
2017's	**LIME & SHAP**, model-agnostic feature attribution
2017's	Deep learning explanations, mostly **gradient-based**

TIMELINE

1990's	**Features of simple models** LR/DT
2000's	**Feature importance**, can be used on any model
2017's	**LIME & SHAP**, model-agnostic feature attribution
2017's	Deep learning explanations, mostly **gradient-based**
2020's	**Counterfactual** explanations

TIMELINE

1990's	**Features of simple models** LR/DT
2000's	**Feature importance**, can be used on any model
2017's	**LIME & SHAP**, model-agnostic feature attribution
2017's	Deep learning explanations, mostly **gradient-based**
2020's	**Counterfactual** explanations
…	…



The Flash **TOUR** …

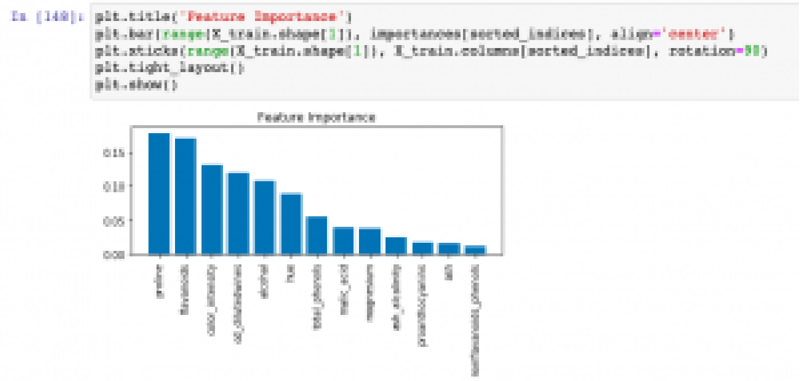
TIMELINE

1990's	**Features of simple models** LR/DT
2000's	**Feature importance**, can be used on any model
2017's	**LIME & SHAP**, model-agnostic feature attribution
2017's	Deep learning explanations, mostly **gradient-based**
2020's	**Counterfactual** explanations
…	…

FEATURE IMPORTANCE

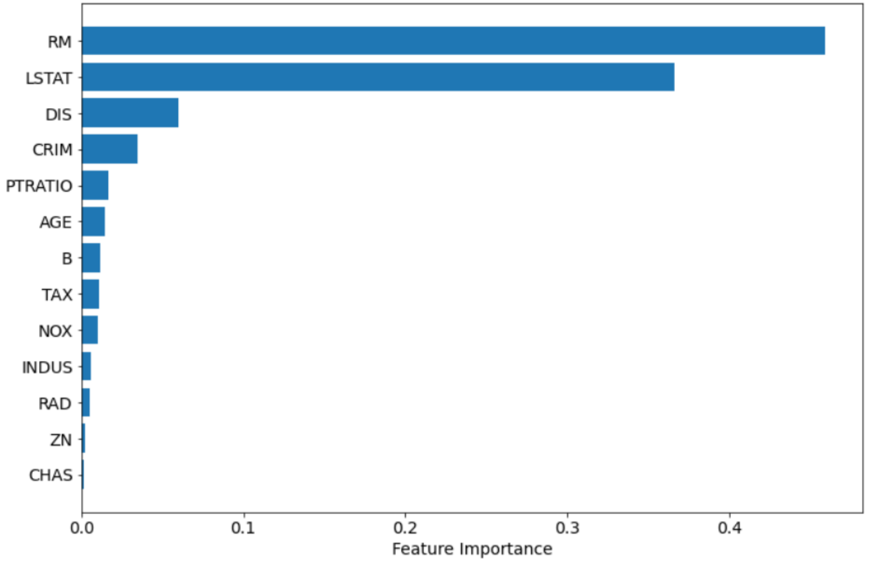
- **Feature importance** – one of the most commonly used method for understanding the inner workings of complex, ML/DL models

- There are different ways to calculate feature importance:
  - **Gini importance**
    - Calculates each feature importance as the sum over the number of splits that include the feature, proportionally to the number of samples it splits.
  - **Permutation feature importance**
    - Feature importance is measured as the increase in the model's prediction error after permuting the feature. A feature is "important" if permuting its values results in higher model error. A feature is "unimportant" if shuffling its values leaves the model error unchanged

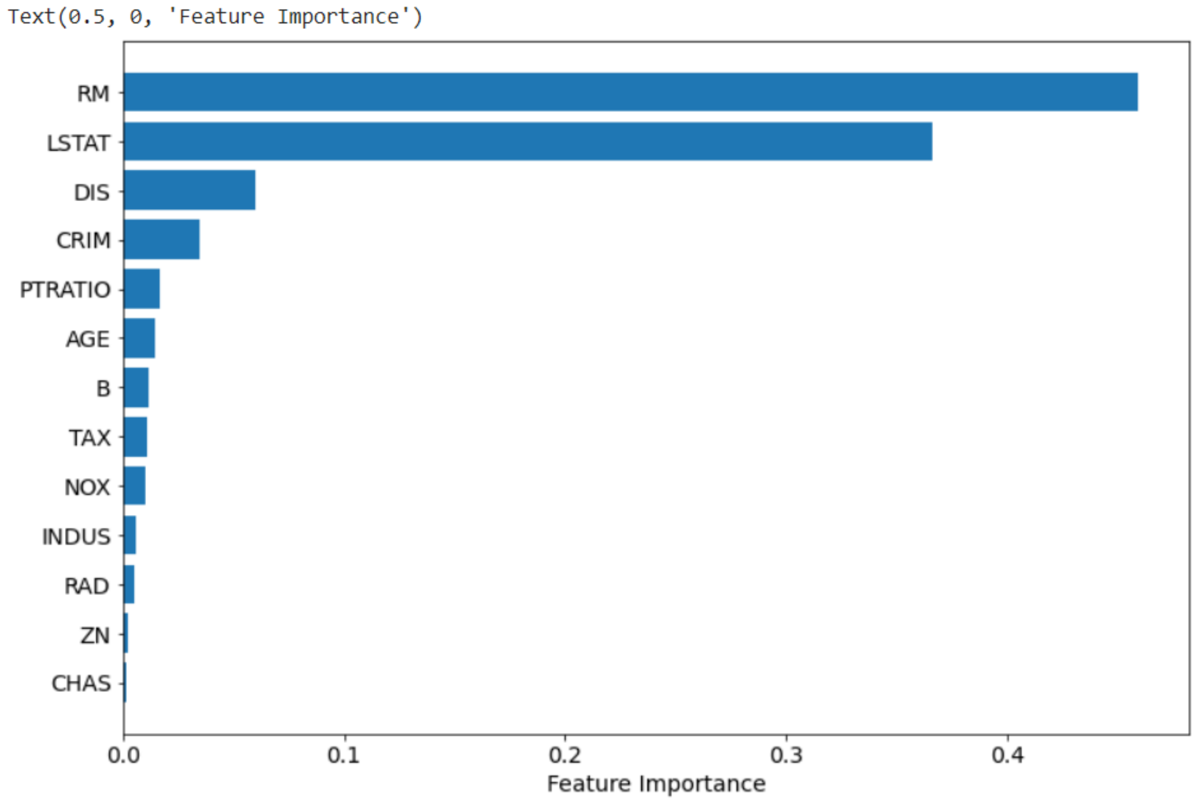


```
In [140]: plt.title('Feature Importance')
          plt.bar(range(X_train.shape[1]), importances[sorted_indices], align='center')
          plt.xticks(range(X_train.shape[1]), X_train.columns[sorted_indices], rotation=90)
          plt.tight_layout()
          plt.show()
```

```
Text(0.5, 0, 'Feature Importance')
```



FEATURE IMPORTANCE: Issue



```
Text(0.5, 0, 'Feature Importance')
```

Give some insights

BUT, no info on the relationship!

What is the relationship between each feature and the response?

Partial Dependency Plots **(PDP)**

- Proposed in *Friedman* (2001)

- They show the **marginal effect one feature has on the predicted outcome of a ML model** while accounting for the average effect of the other predictors in the model

- A partial dependence plot can show **whether the relationship between the target and a feature is linear, monotonic or more complex**

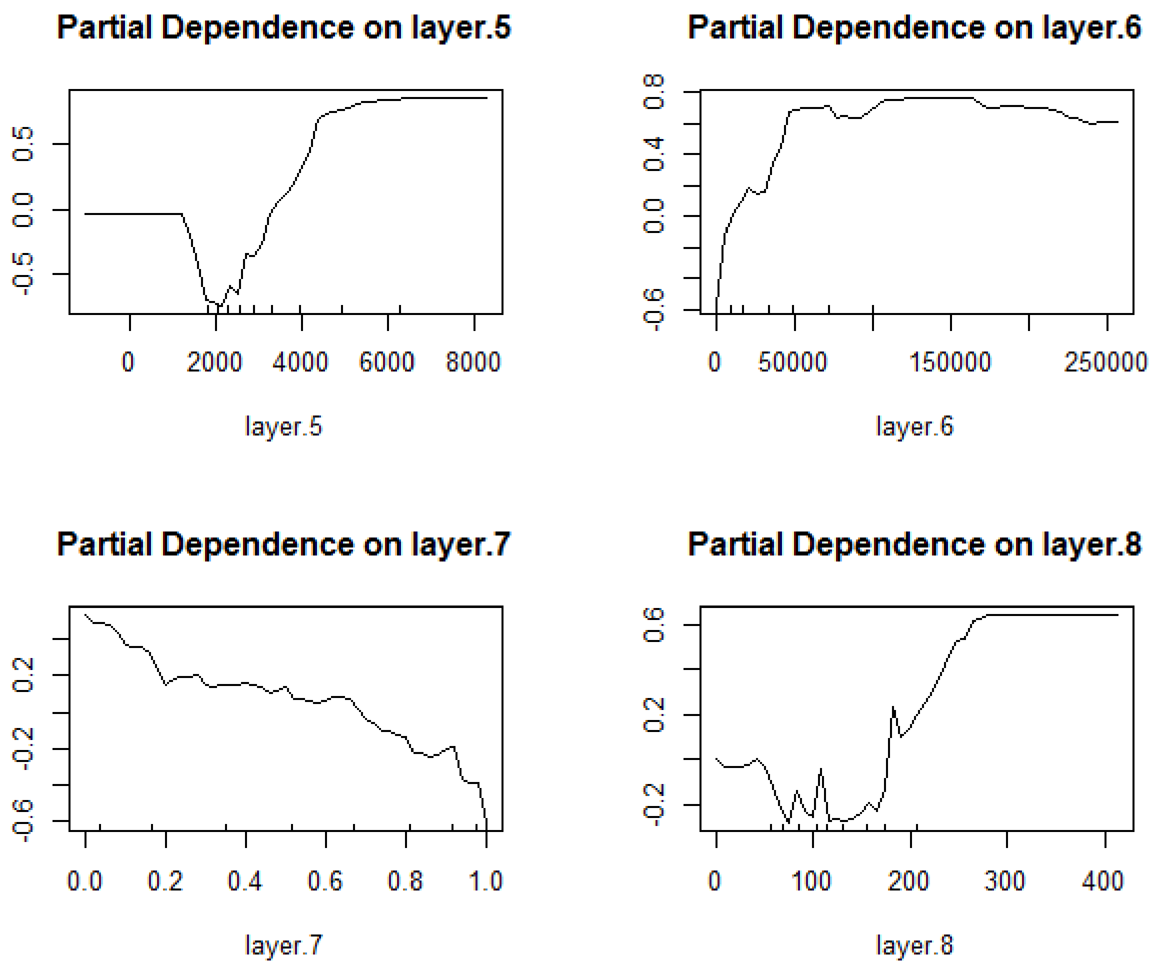


Image source: stats.stackexchange.com

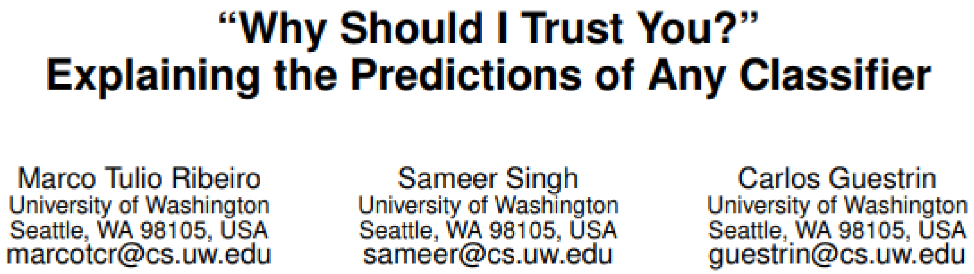
Molnar (2023)

TIMELINE

1990’s	**Features of simple models** LR/DT
2000’s	**Feature importance**, can be used on any model
2017’s	**LIME & SHAP**, model-agnostic feature attribution
2017’s	Deep learning explanations, mostly **gradient-based**
2020’s	**Counterfactual** explanations
…	…

Local Interpretable Model-Agnostic Explanations

- LIME → explains the prediction of **any machine learning model** by learning an interpretable model **locally** around a specific instance of interest
- Works with classification & regression
- Works with tabular data, text and pictures



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

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

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

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

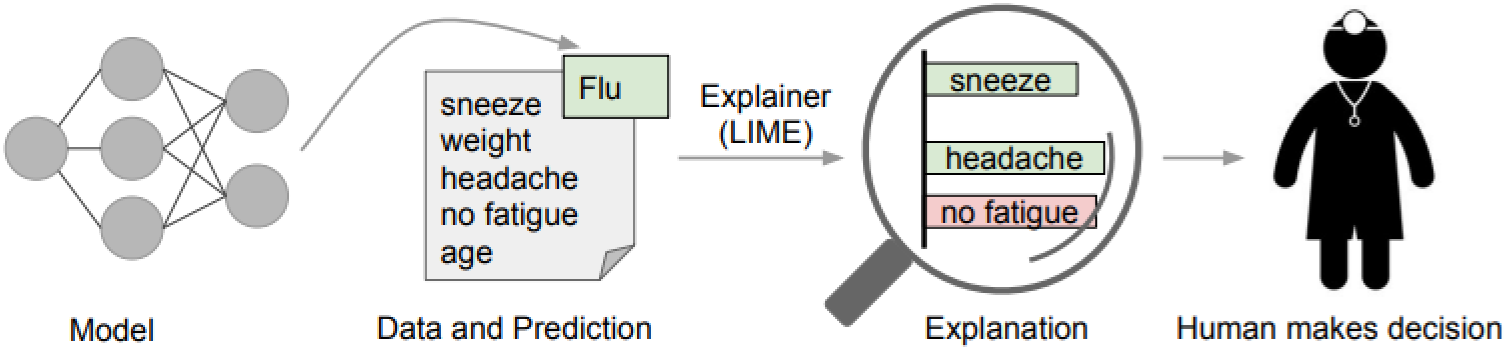
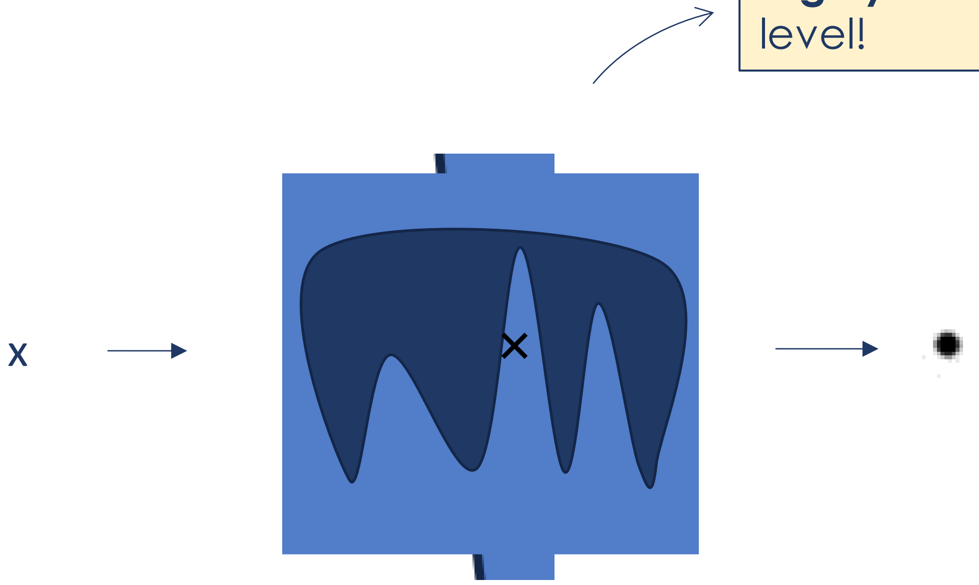


Image source: Ribeiro et al. (2016)

LIME – **How does it work?**



Highly accurate at local level!

So local that one can **apply a simple, interpretable model** to understand the relationship!

Steps

For which you require explanations

- **Pick an observation**, create and permute data;
- Calculate similarity between the original observations and the permutations;
- Make predictions on new data using your black box;
- **Fit a simple model** to the permuted data with n features and similarity scores as weights;
- **Coefficients from the simple model serve as an explanation of the model behavior** at the local level.

LIME: **Formally**

$$\xi(x) = argmin_{g \in G} L(f, g, \pi_x) + \Omega(g)$$

- $\xi(x)$ is the **explanation function.**
- f is the **black-box model** we want to explain.
- $g \in$ G represents the set of **interpretable models** (e.g., linear regression, decision trees).
- $L(f, g, \pi_x)$ is the **loss function.**
- π_x is the **proximity function.**
- $\Omega(g)$ is a **complexity penalty**.

Molnar (2023)

LIME: **Formally**

$$\xi(x) = argmin_{g\in G} L(f, g, \pi_x) + \Omega(g) \qquad (1)$$

$$\boldsymbol{L(f, g, \pi_x)} = \sum_i \pi_x(x_i)\big(f(x_i) - g(x_i)\big)^2 \quad (2)$$

- We want to ensure that the interpretable model g approximates the black-box model f **locally**. The typical choice is the **weighted squared error**.

- x_i are the perturbed samples around x.

- $\pi_x(x_i)$ are their proximity weights.

Molnar (2023)

LIME: Formally

$$\xi(x) = argmin_{g \in G} L(f, g, \pi_x) + \Omega(g) \qquad (1)$$

$$L(f, g, \pi_x) = \sum_i \pi_x(x_i)\big(f(x_i) - g(x_i)\big)^2 \qquad (2)$$

$$\pi_x(x_i) = \exp\left(\frac{-D(x, x_i)^2}{\sigma^2}\right) \qquad (3)$$

- Controls which points are considered more relevant for the explanation.

- $D(x, x_i)^2$ is the **Euclidean distance** between the perturbed point x and the original instance.

- σ controls the **scale of locality** (how fast weights decrease as distance increases).

Molnar (2023)

LIME: **Formally**

$$\xi(x) = argmin_{g \in G} L(f, g, \pi_x) + \boldsymbol{\Omega}(\boldsymbol{g}) \qquad (1)$$

- Complexity parameter.
- Prevents the local model g from being too complex.
- Encourages simpler explanations (e.g., fewer features in a linear model).
  - Example: If g is a linear model, $\boldsymbol{\Omega}(\boldsymbol{g})$ could be the number of non-zero coefficients.

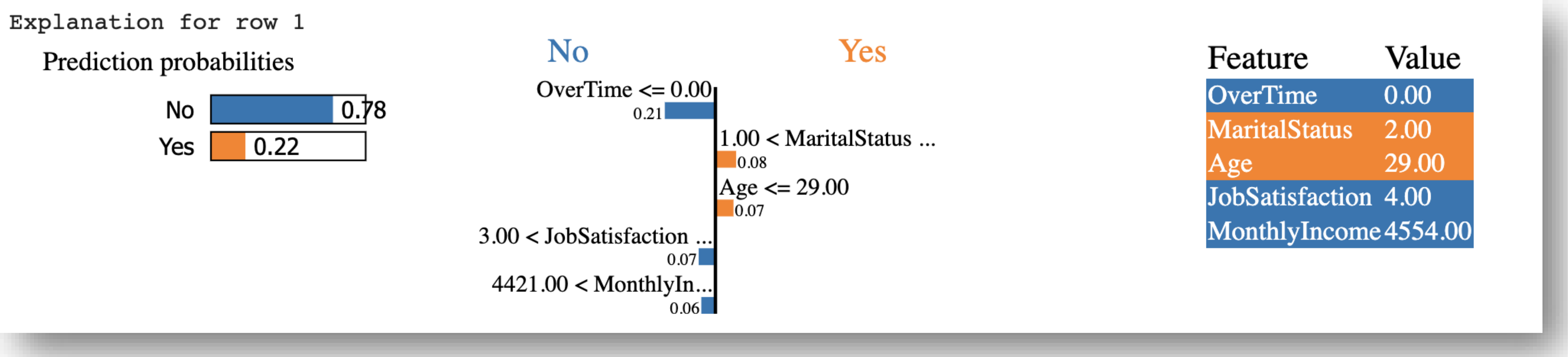
Molnar (2023)

LIME: **Example**

- We train a BB model to predict whether a person leaves their job based on different demographic (gender, age, etc.) and job-related data (wage, whether they have overtime, whether they travel for their job, etc.)

- Once trained, we apply LIME to obtain how the BB model has made the prediction for a specific person in the dataset

- LIME allows us to observe **which features push the prediction toward "STAY" and which push the prediction towards "LEAVE" by looking at a specific row/person**



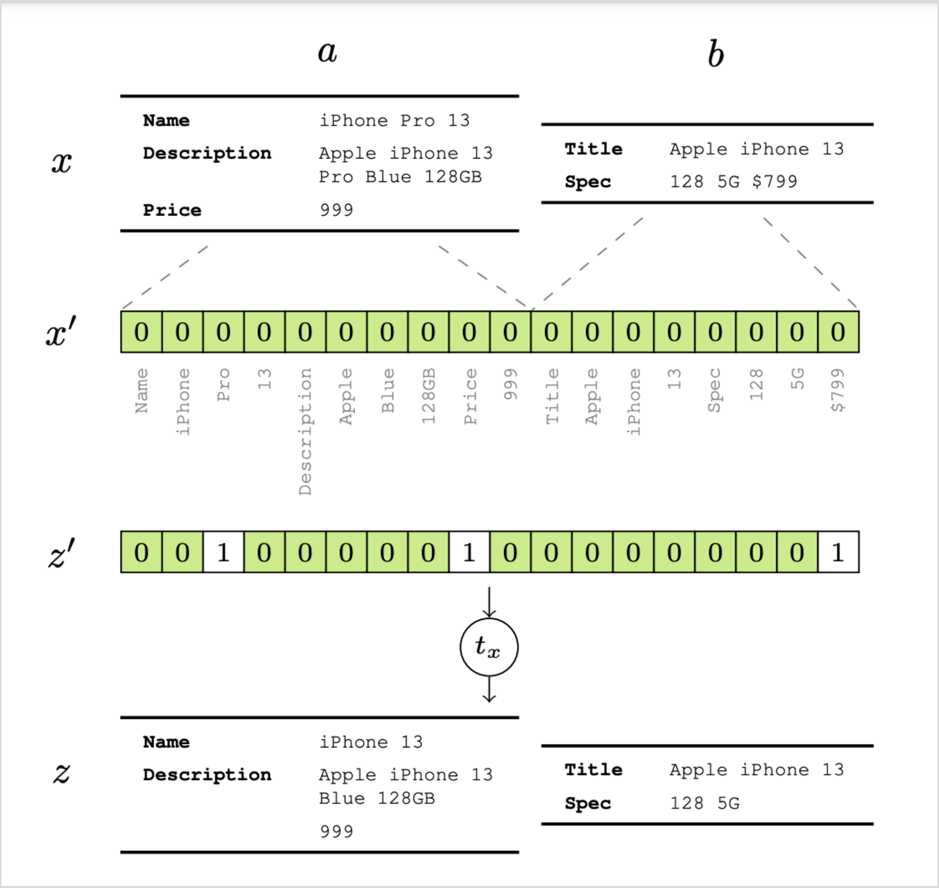
```
Explanation for row 1
```

Feature	Value
OverTime	0.00
MaritalStatus	2.00
Age	29.00
JobSatisfaction	4.00
MonthlyIncome	4554.00

Note: LIME for **different input**

Example of entity matching

Can be applied to different **problem sets** …



x, a:

Name	iPhone Pro 13
Description	Apple iPhone 13 Pro Blue 128GB
Price	999

x, b:

Title	Apple iPhone 13
Spec	128 5G $799

Name	iPhone	Pro	13	Description	Apple	Blue	128GB	Price	999	Title	Apple	iPhone	13	Spec	128	5G	$799
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1

z, a:

Name	iPhone 13
Description	Apple iPhone 13 Blue 128GB
	999

z, b:

Title	Apple iPhone 13
Spec	128 5G

https://arxiv.org/abs/2110.00516

LIME: Advantages **vs** Disadvantages

Advantages

- **Model agnostic** – can be applied to any BB model
- Can provide **human-friendly explanations** (also useful to a non-technical audience)
- Works with **different data types**
- The explanations can be created with another subset of features than the original model was trained on

Disadvantages

- The **correct definition of the neighbourhood** is a very big, unsolved problem
- **Instability** of the explanations (two similar points can get very different explanations)
- Some implementations **ignore correlation of features** (sampling can be improved)

Molnar (2023)

SHAP Values

- Stands for **Shapley Additive exPlainations**
- Based on Shapley values
  - Shapley values are a concept from **cooperative game theory**, developed by Lloyd Shapley.
  - They provide a fair way to distribute the total gain (or cost) among players based on their individual contributions.

Game

The prediction task

Players

Features

Gain

Actual prediction for an observation minus the average prediction for all instances

Shapley Values: **DETAILS**

- Given:
  - A set N of n players: $N = \{1,2,\dots,n\}$
  - A characteristic function v that assigns a value to every coalition (subset of players)

The **Shapley value for a player i** is a measure of the **average contribution of i to all possible coalitions**.

Kumar (2020)

The **Math**

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

The **Math**

$$\boxed{\phi_i(v)} = \sum_{S \subseteq N\{i\}} \frac{|S|!\,(n-|S|-1)!}{n!}\left(v(S \cup \{i\}) - v(S)\right)$$

Shapley value for a given feature i

We calculate the contribution of each feature to a prediction by considering all possible subsets of features and computing the marginal contribution of each feature across these subsets

The **Math**

$$\phi_i(v) = \boxed{\sum_{S \subseteq N\{i\}}} \frac{|S|!\,(n - |S| - 1)!}{n!} \left(v(S \cup \{i\}) - v(S)\right)$$

Sum over all possible coalitions that do not contain i

The Shapley value aims to measure the average contribution of feature i to the prediction, **considering all possible scenarios where i could join a coalition**

The **Math**

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

Coalition without feature i

The **Math**

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

Coalition with feature i

The **Math**

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

Marginal contribution of i to the coalition

Marginal change in the model’s score **after adding feature i**

The **Math**

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

Weighting the contributions of i by its share in the number of total coalitions

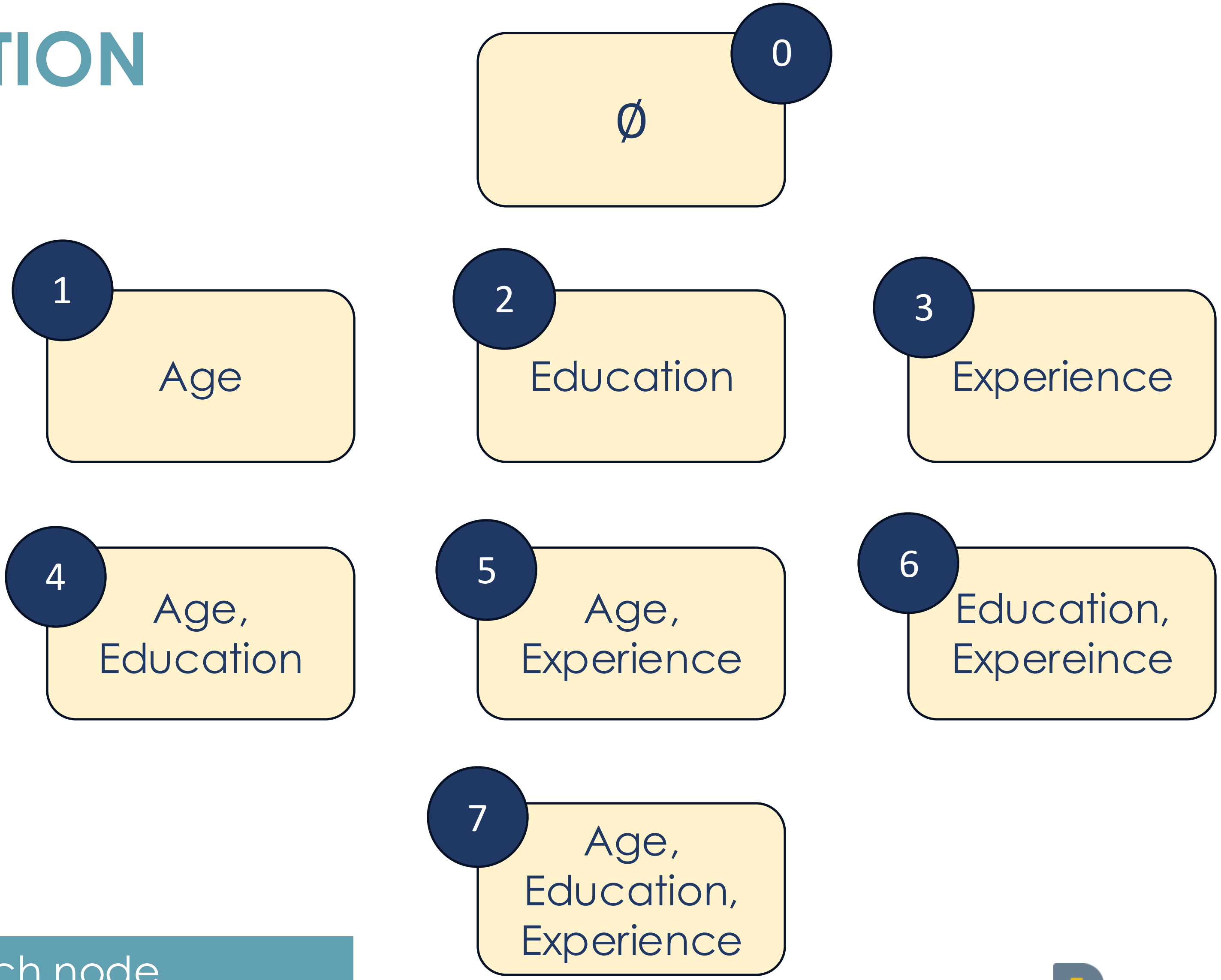
- |S| is the **size of the coalition** S (excluding feature i)
- n is the total **number of feature**

Shapley Values: **INTUITION**

- Let's train a BB model to predict **a person's wage based on their age, education and experience**

- We also want to obtain the Shapley values for each feature

- **<u>The cardinality of a power set is</u>** $\mathbf{2^n}$, where n is the number of elements of the original set



0 ∅

1 Age

2 Education

3 Experience

4 Age, Education

5 Age, Experience

6 Education, Expereince

7 Age, Education, Experience

Each node represents a model

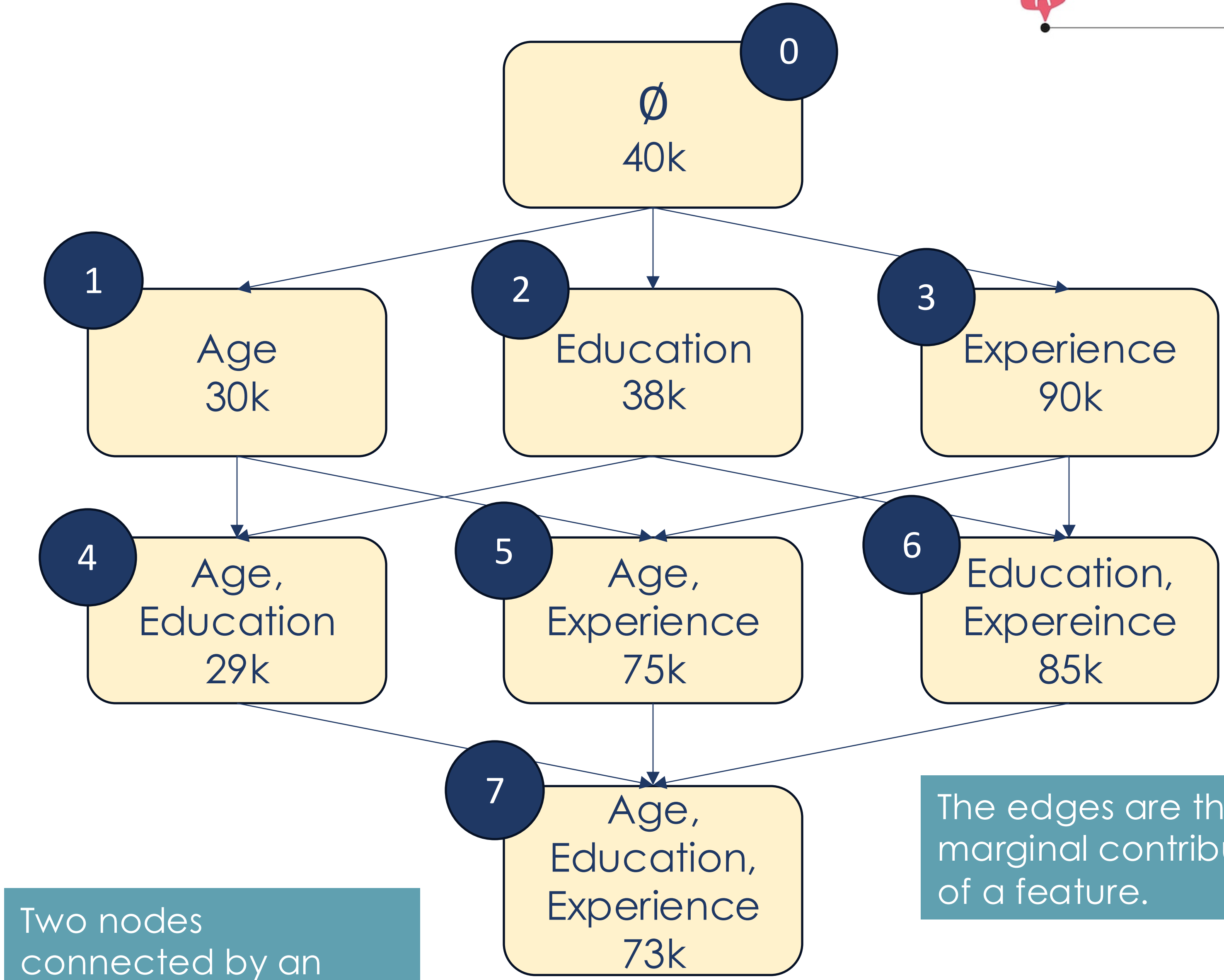
Let’s see the predictions.

Q: Model 0 has no features. How do we obtain the 40k?

Model 1 contains only the Age and gives a prediction of 30k.

Model 2 contains only Education and gives a predictuon of 38k.

So on …

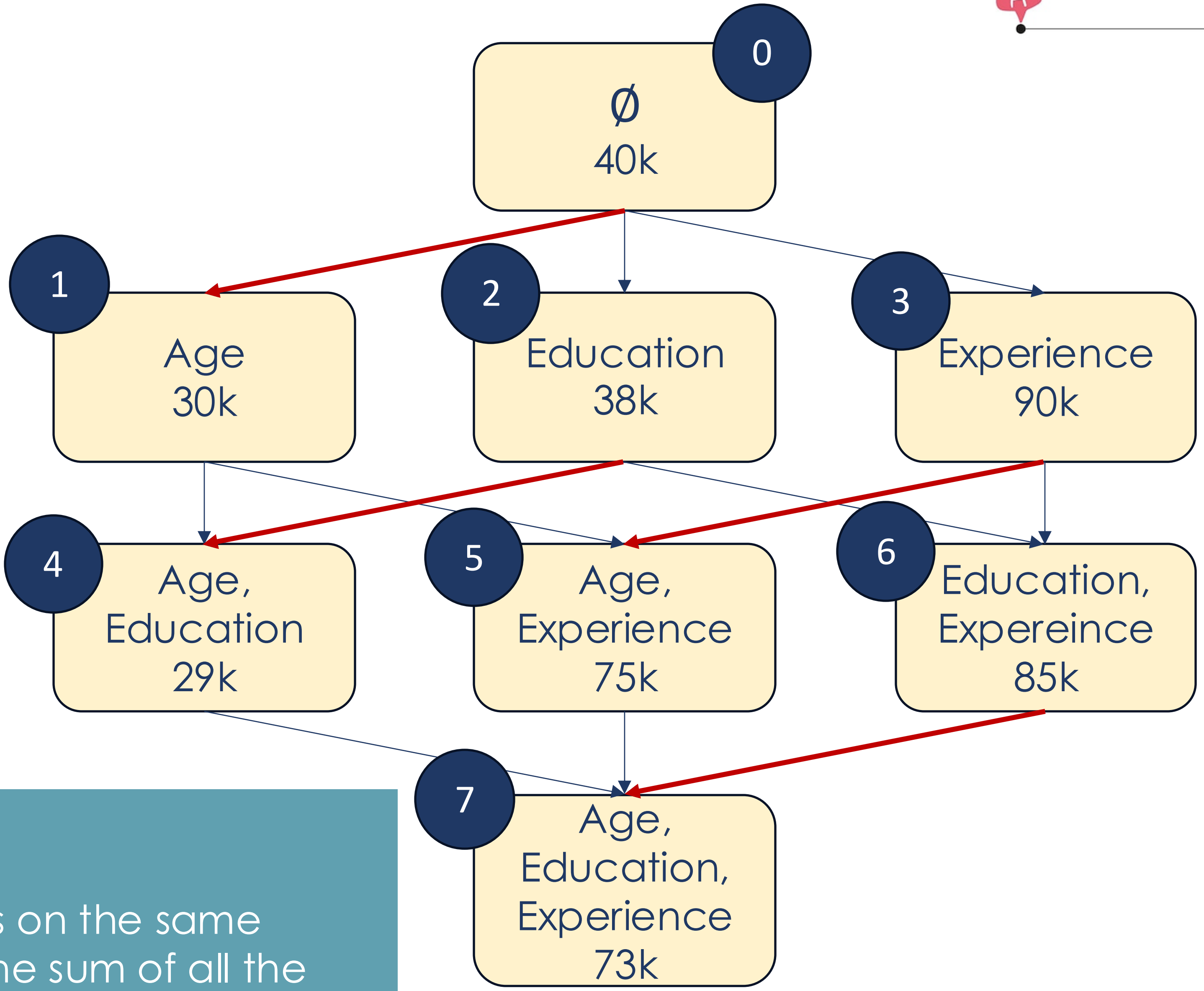


Two nodes connected by an edge differ by just one feature!

The edges are the marginal contributions of a feature.

In order to obtain the overall contribution of a single feature to the final prediction, we need to consider the marginal contribution of that feature in **all models where the feature is present.**

Let’s take Age: **which edges do we take?**



0
Ø
40k

1
Age
30k

2
Education
38k

3
Experience
90k

4
Age,
Education
29k

5
Age,
Experience
75k

6
Education,
Expereince
85k

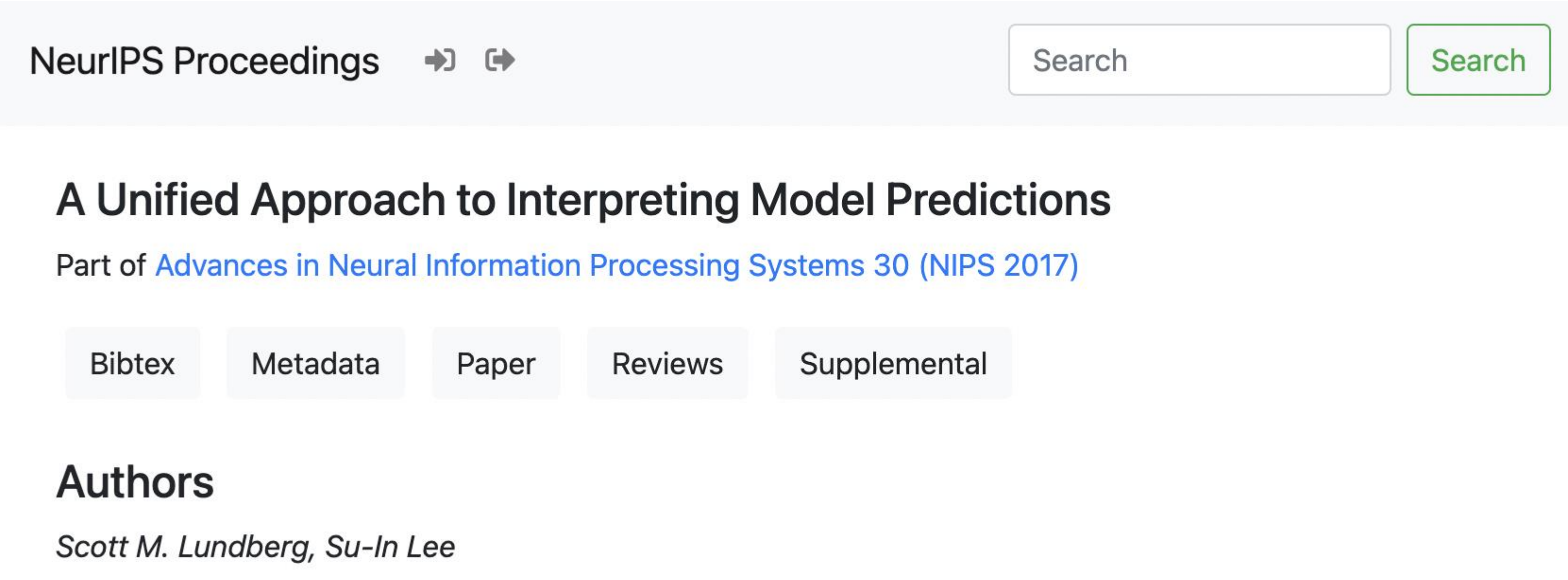
7
Age,
Education,
Experience
73k

Weighting:

- sum of all the weights on the same “row” should equal the sum of all the weights on any other “row”.
- all the edges on the same “row” should equal to each other

Back to **SHAP**

- Proposed by Lundberg and Lee (2017)
- The original Shapley formula requires to train $\mathbf{2^n}$ models. For a model with 50 features, **this would mean to train 1e15 models!**
- The work by Scott Lundberg employ approximations and samplings where **instead of making the computations for all coalitions – you draw a sample and compute contributions for a few samples** of all posible coalitions.

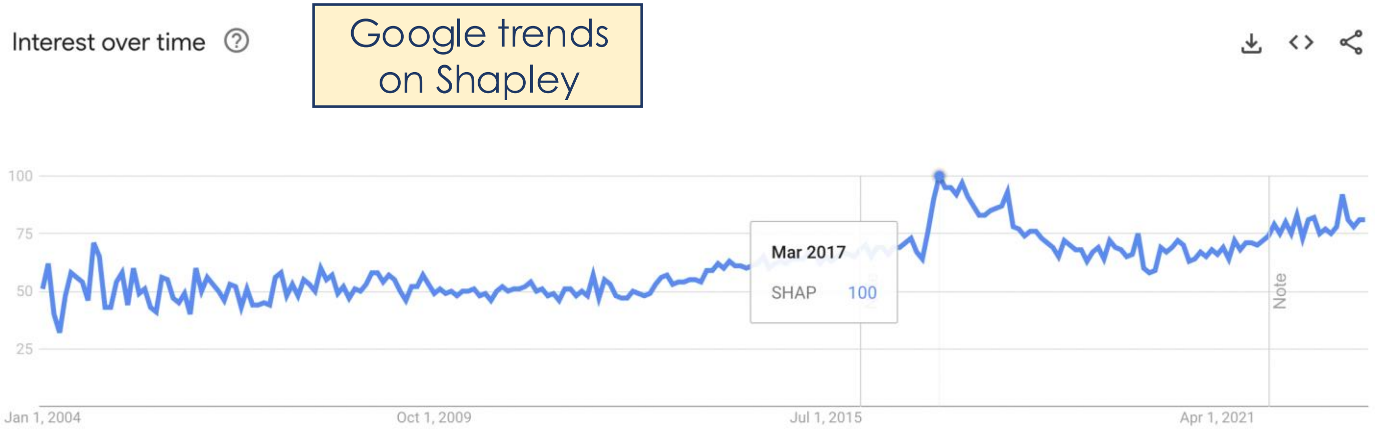


A Unified Approach to Interpreting Model Predictions

Part of Advances in Neural Information Processing Systems 30 (NIPS 2017)

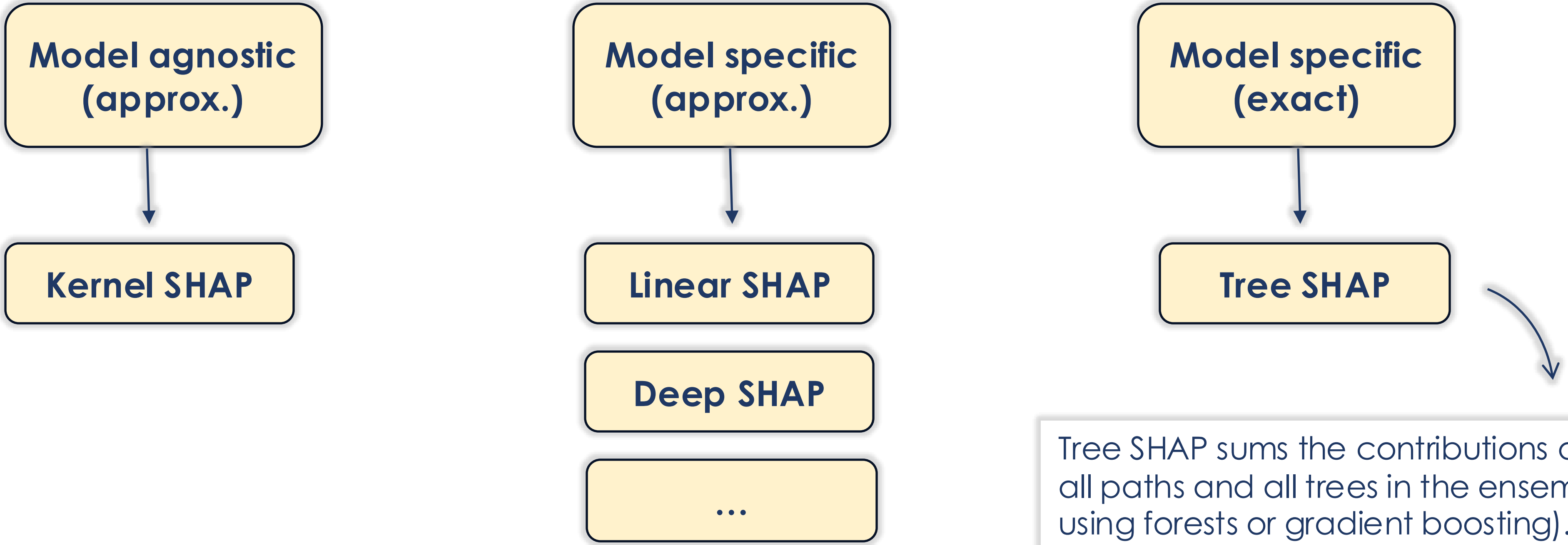
Authors

Scott M. Lundberg, Su-In Lee



Google trends on Shapley

SHAP **Implementations**



Model agnostic (approx.)

Kernel SHAP

Model specific (approx.)

Linear SHAP

Deep SHAP

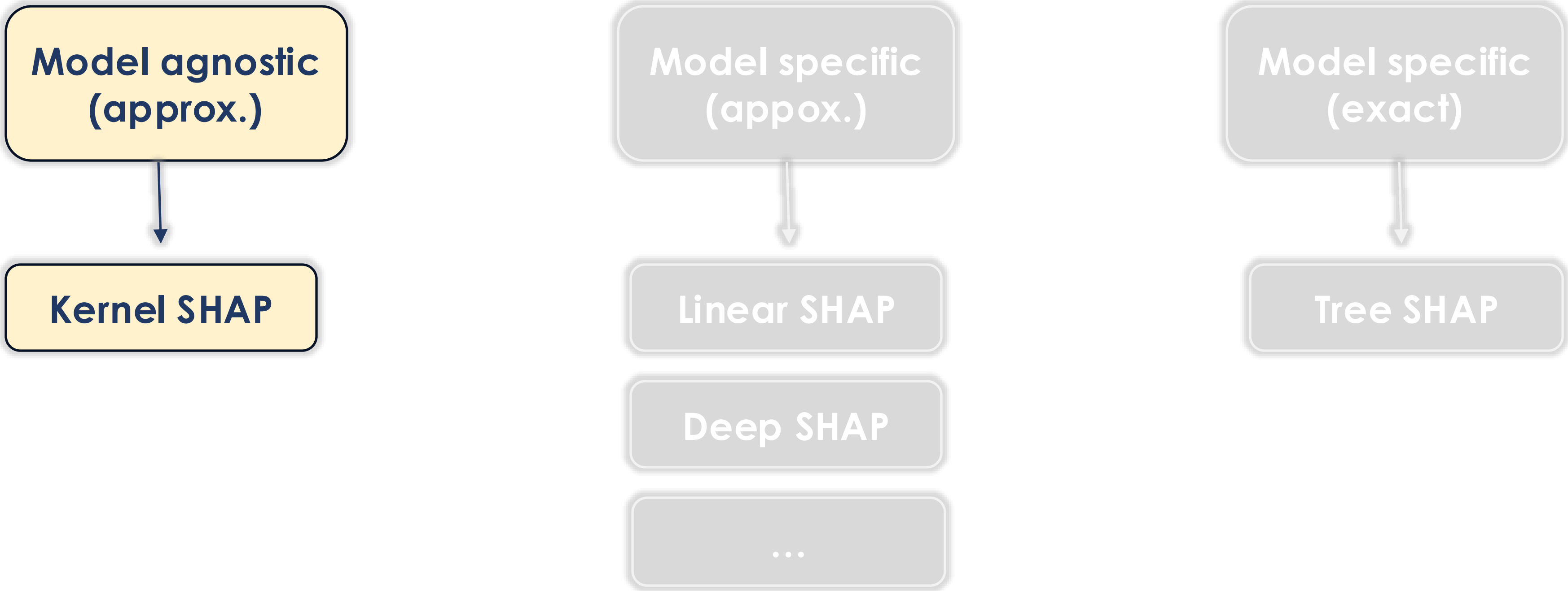
...

Model specific (exact)

Tree SHAP

Tree SHAP sums the contributions across all paths and all trees in the ensemble (if using forests or gradient boosting).The result is the exact Shapley value for each feature.

SHAP **Implementations**



KernelSHAP

Steps:

01. **Sample coalitions** (random – chain of 0s and 1s)

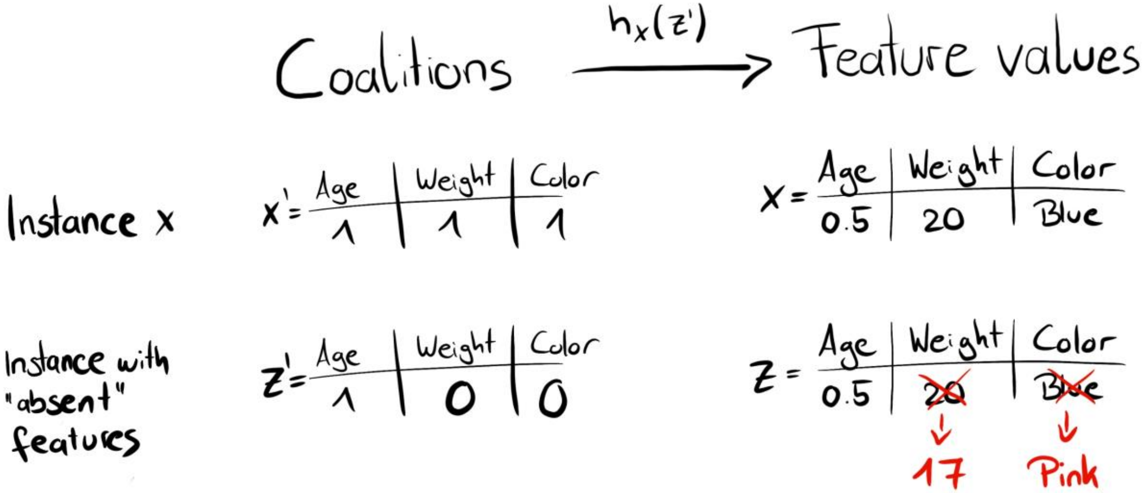
02. **Get predictions** from the BB model for each coalition

03. **Compute the weights** for each coalition

04. **Fit a weighted linear model**

05. **Return SHAP values** (coefficients)

For example, the vector of (1,0,1,0,0,1) means that we have a coalition of the first, third and sixth feature.



$x' =$

Age	Weight	Color
1	1	1

$x =$

Age	Weight	Color
0.5	20	Blue

$z' =$

Age	Weight	Color
1	0	0

$z =$

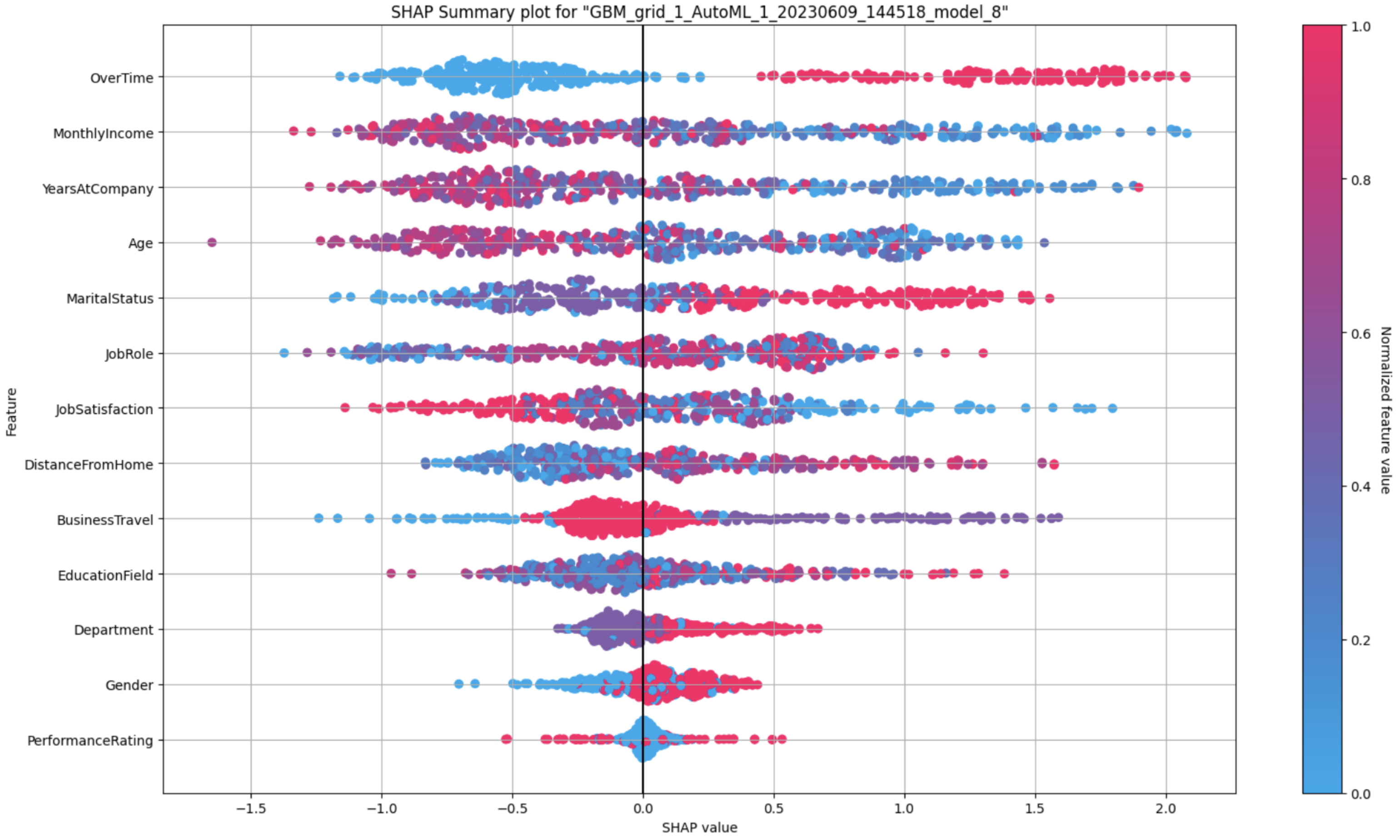
Age	Weight	Color
0.5	~~20~~ ↓ 17	~~Blue~~ ↓ Pink

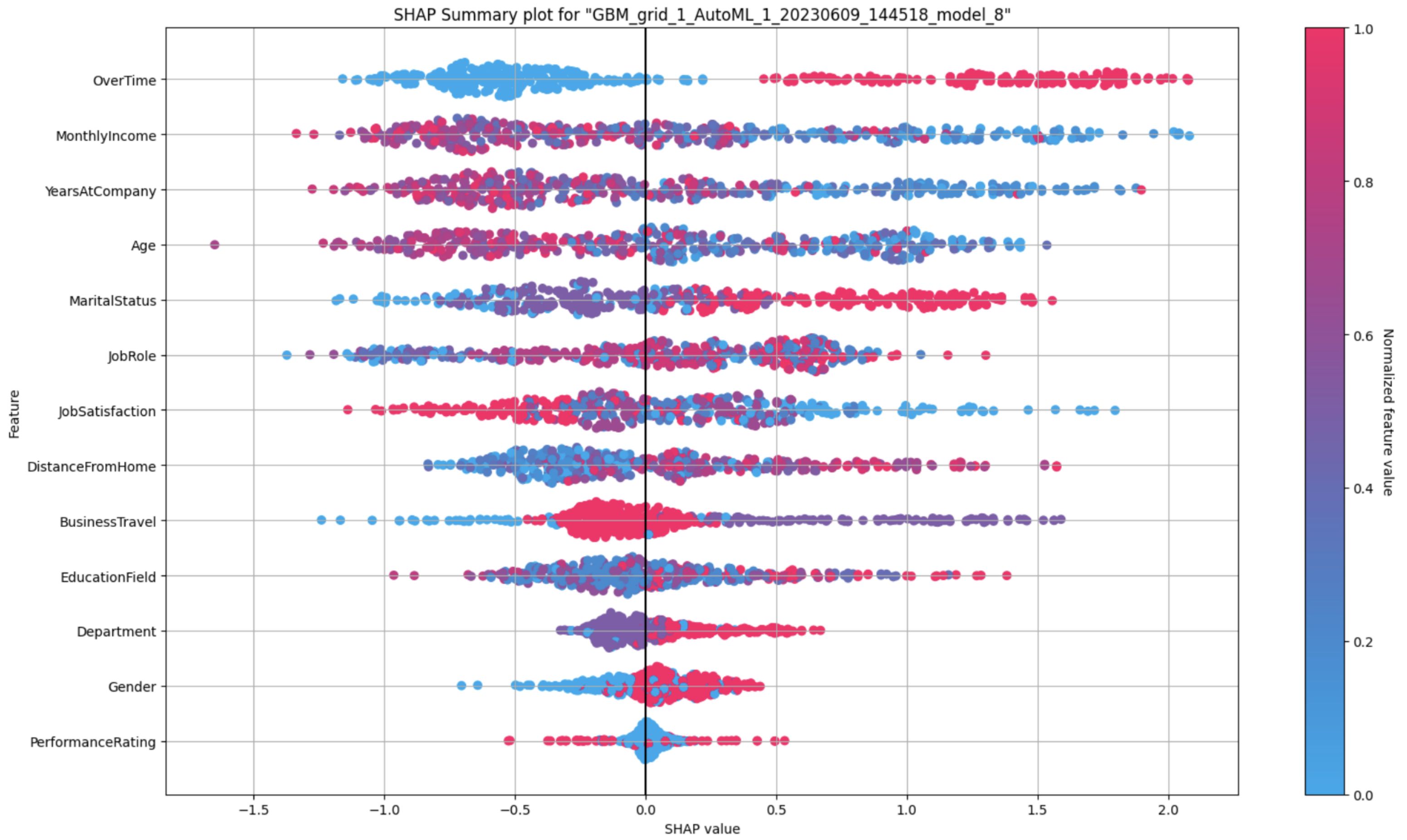
SHAP: **Examples**

- Let's go back to our example of training a BB model to predict whether a person leaves their job.

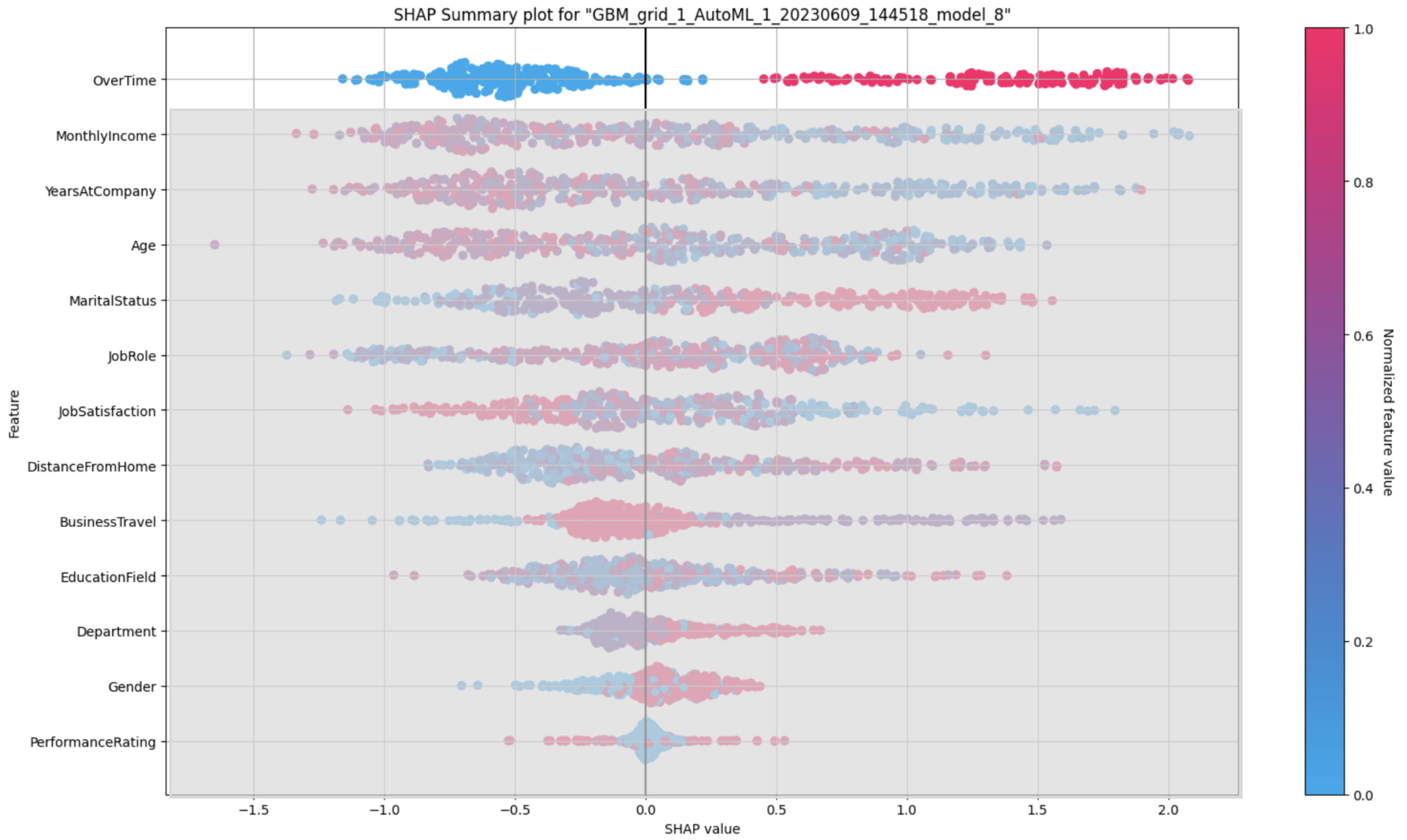
- Once trained, we wish to obtain the SHAP values for each feature included in the analysis.

- **This is the visualization we will obtain!**

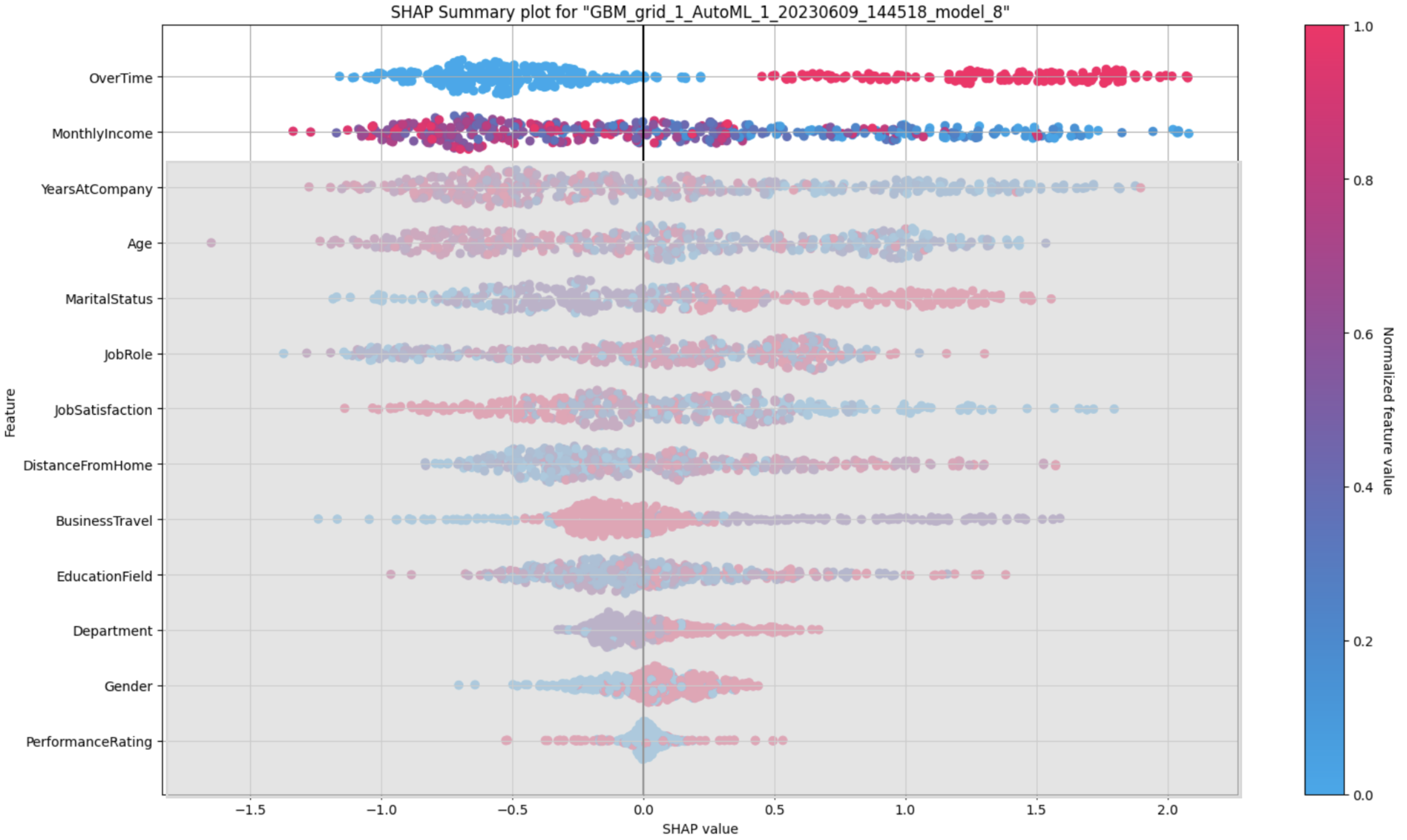




- The summary plot **combines feature importance with feature effects.**
- Each point on the summary plot is a SHAP value for a feature and an instance.
- The position on the y-axis is determined by the feature and on the x-axis by the Shapley value.
- The colour represents the value of the feature from low to high.



- The most important feature: **OverTime.**
- The x-axis give the impact of the model.
- Most of the **blue points (i.e. Overtime = "No", or = 0) are concentrated on the left and are associated with negative SHAP values.**
- **No Overtime reduces the probability of people leaving their jobs.**
- Overtime = "YES (i.e. = 1) (red dots) are associated with positive SHAP values hence increase the probability of people leaving their jobs.

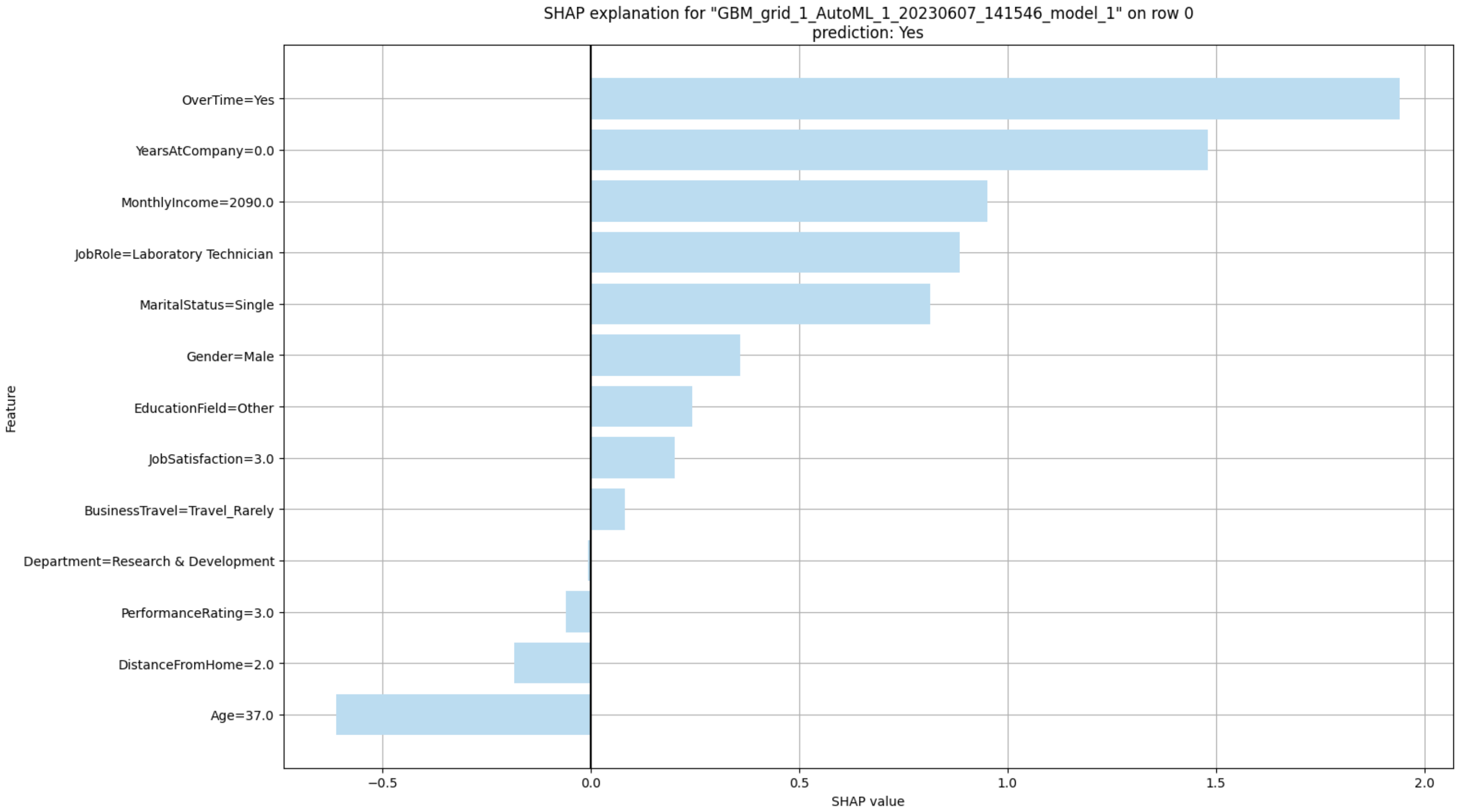


- The second most important feature: **MontlyIncome.**
- Most of the **blue points are concentrated on the right and are associated with positive SHAP values.**
- **Lower monthly income increases the probability of people leaving their jobs.**
- Higher monthly income (red dots) are associated with negative SHAP values hence decrease the probability of people leaving their jobs.

SHAP: **Examples**

- We can also obtain local explanations, i.e. **how the model arrived at a certain decision (prediction) for a specific unit/row in our data.**

Example for a **specific person** (first row in a dataset, i.e. row = 0)



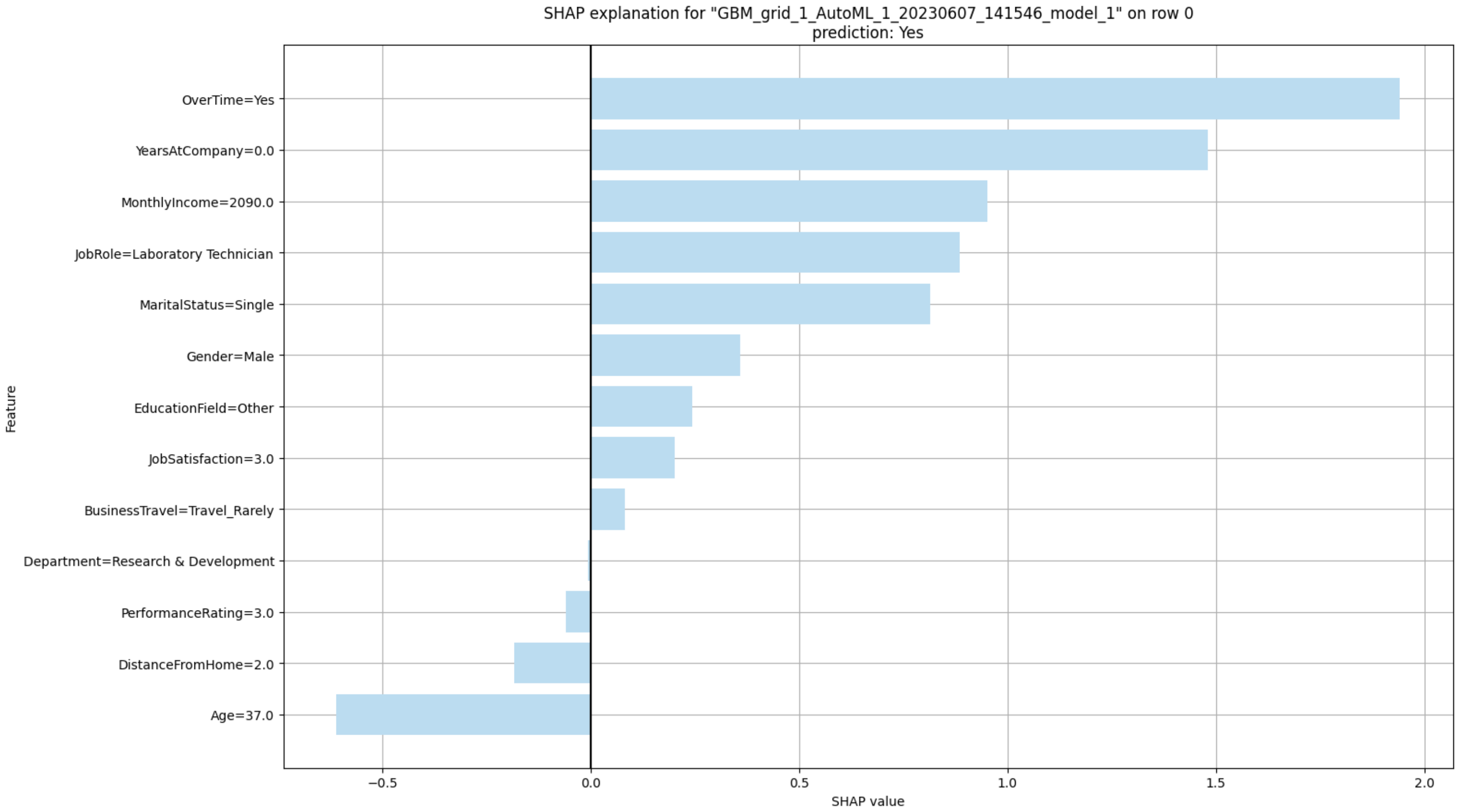
SHAP explanation for "GBM_grid_1_AutoML_1_20230607_141546_model_1" on row 0
prediction: Yes

SHAP: **Examples**

- For this specific instance, the model predicts that the person will leave their job.

- SHAP further indicates that the most features that push the prediction towards leaving their job are: **Overtime = Yes; YearAtCompany=0.0** …

- Only **Age = 37, DistanceFromHome = 2.0 and PeformanceRating = 3.0**, pushes the prediction towards "Attrition" = No



SHAP: Advantages **vs** Disadvantages

Advantages

- SHAP has a **solid theoretical foundation** in game theory.
- The prediction is **fairly distributed among the feature values.**
- We get **contrastive explanations** that compare the prediction with the average prediction.

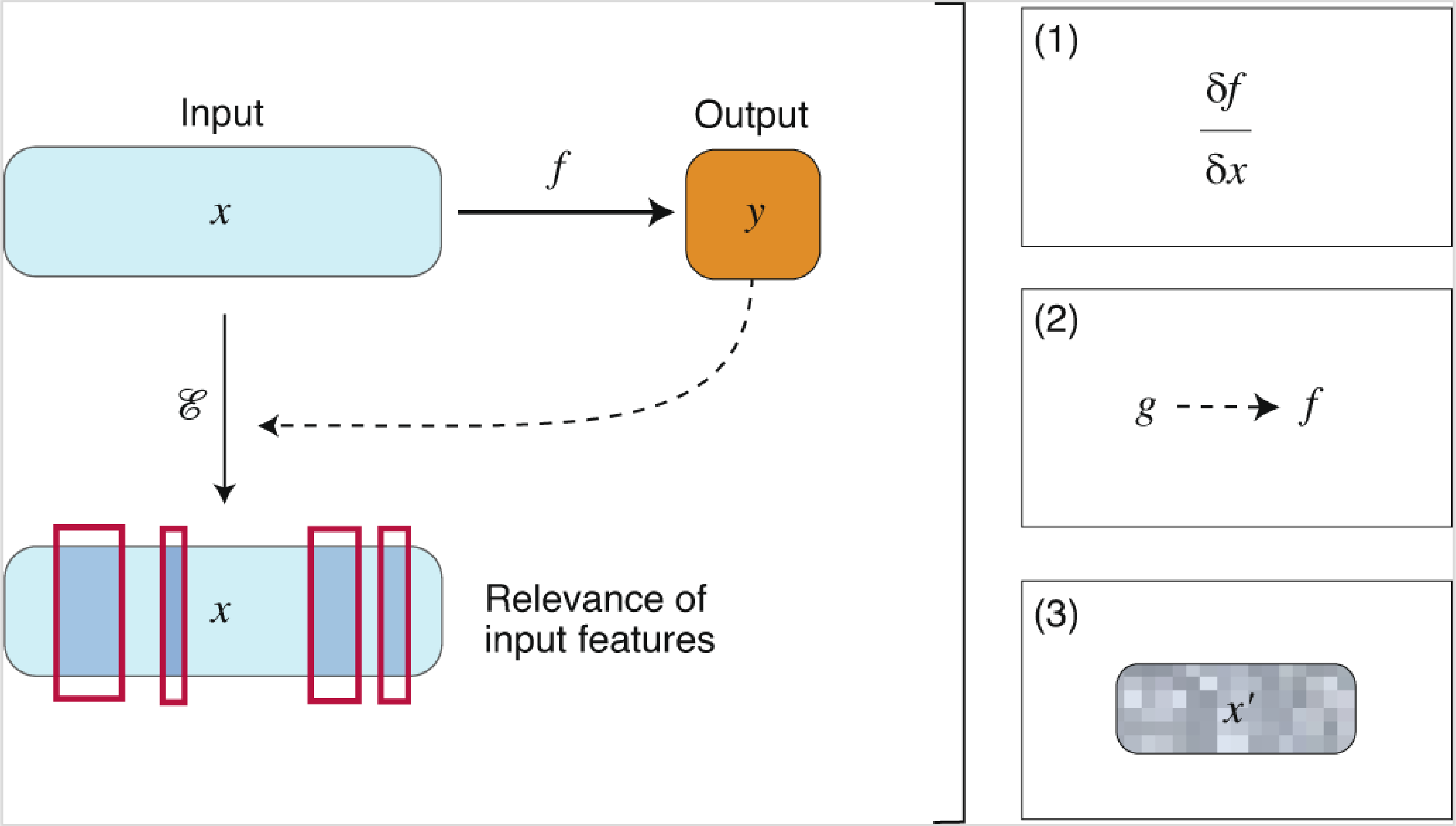
Disadvantages

- Can be **computationally very intensive.**
- **Ignores feature dependence**, same as all permutation-based methods.
- **Explanations may change** based on which features are considered.

Molnar (2023)

TIMELINE

1990's	**Features of simple models** LR/DT
2000's	**Feature importance**, can be used on any model
2017's	**LIME & SHAP**, model-agnostic feature attribution
2017's	Deep learning explanations, mostly **gradient-based**
2020's	**Counterfactual** explanations
…	…

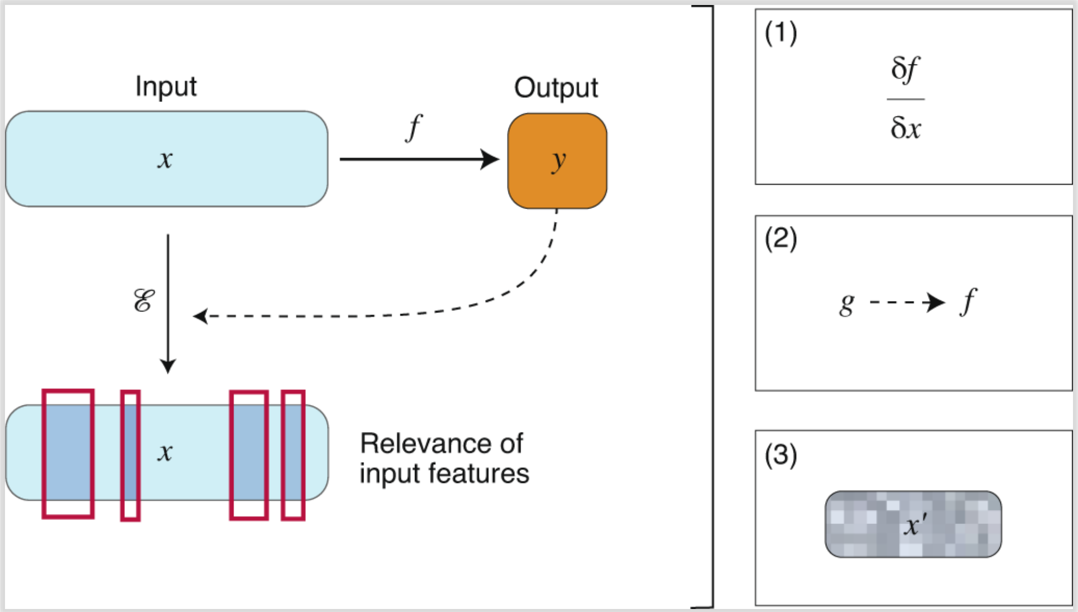

Input
Output
x
f
y
$\mathscr{E}$
x
Relevance of input features
(1)
$\frac{\delta f}{\delta x}$
(2)
g
f
(3)
x'

GRADIENT-BASED Explainability

- Gradient-based explainability methods analyse **how the model's output changes when small changes occur in the input**.

- The **gradient of the output with respect to the input** tells us which parts of the input have the **strongest influence** on the model's decision.

- The larger the gradient – **the more relevant the feature** (!)



GRADIENT-BASED Explainability

Vanila Gradient

The vanilla gradient calculates how **sensitive the output is to each input** by computing the derivative of the output with respect to each input feature.

Simple and fast

Can be **noisy**

Smooth Gradient

The smooth gradient reduces the noise of vanilla gradients by adding **small random noise** to the input multiple times and **averaging the gradients**.

More **stable and reliable**

Requires more computation

Integrated Gradient

Instead of looking at the gradient at a single point, integrated gradients **accumulate the gradients** along a path from a **baseline** to the actual input.

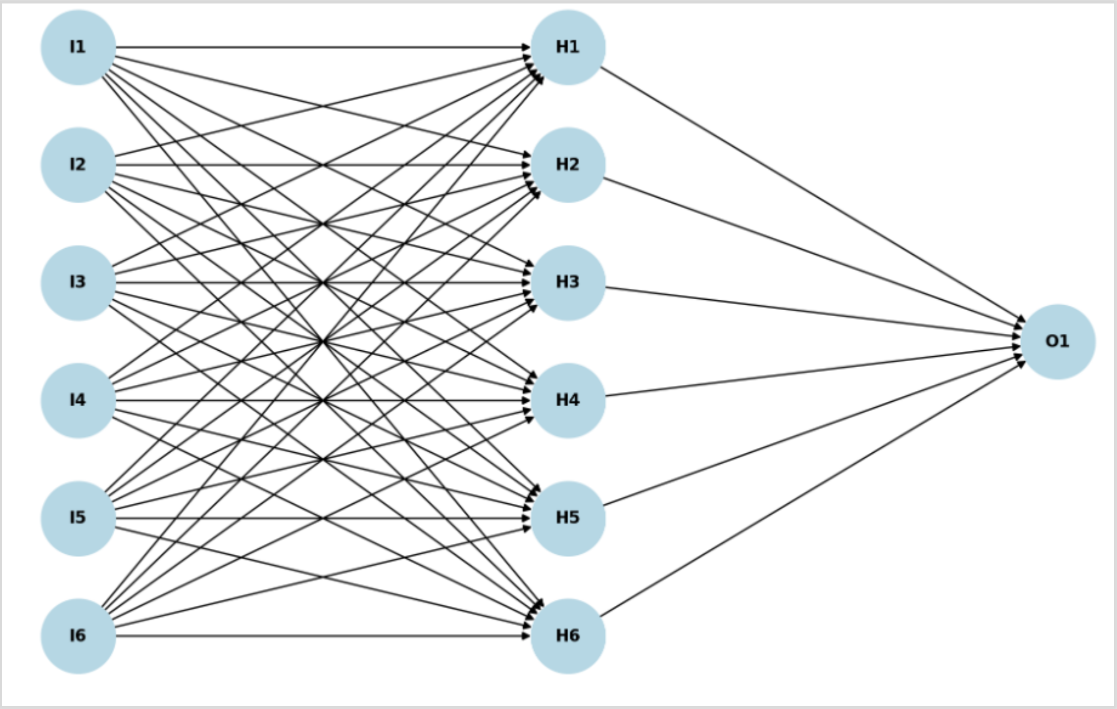
Theoretically **robust**

Very computationally intensive

TOY Example

- Let's consider an example where we want to model **the returns of a stock Y** by considering **the returns of other related stocks (X1-X6)**

- **Model**: Simple feedforward neural network
- **Inputs**: Returns of other stocks in the sector (X1,X2,...,X6)
- **Output**: Predicted return of the target stock
- **Activation function:** Sigmoid



Input returns: $X = [0.02, -0.01, 0.03, 0.015, -0.02, 0.01]$

Predicted return for Stock Y: $\widehat{Y} = 0.018$

TOY Example

Vanila Gradient

For Each X, we calculate the first-order derivative:

$$\frac{\partial Y}{\partial X_i}$$

This derivative shows **how Y** when X_i changes slightly.

Smooth Gradient

Add **tiny random noise** to the inputs.

For each noisy input, compute the vanilla gradient.

Do this **100 times** with different noises.

Average the gradients.

Integrated Gradient

Choose a baseline.

Move from the baseline to the actual input in steps.

Compute gradients at each step.

Average the gradients and multiply by the input difference.

TIMELINE

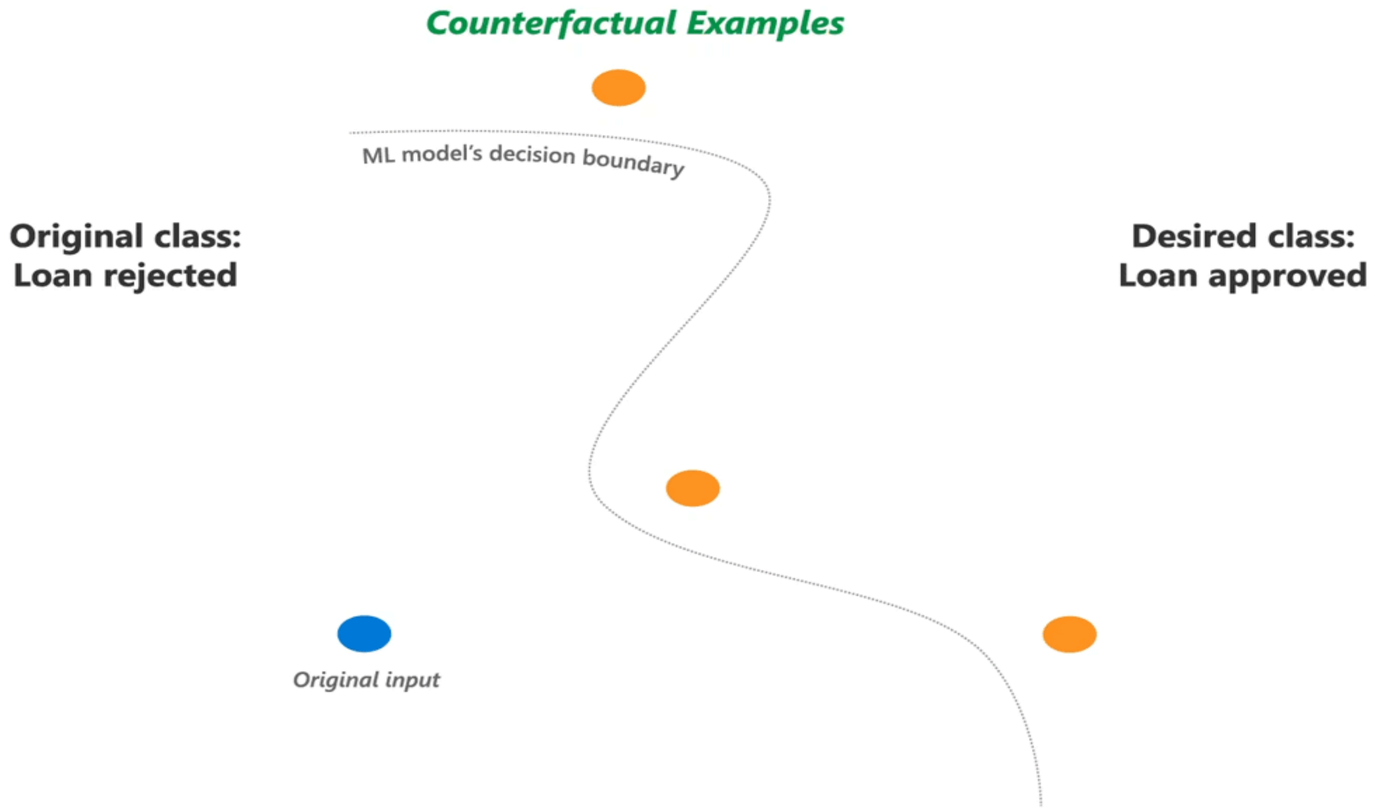
1990's	**Features of simple models** LR/DT
2000's	**Feature importance**, can be used on any model
2017's	**LIME & SHAP**, model-agnostic feature attribution
2017's	Deep learning explanations, mostly **gradient-based**
2020's	**Counterfactual** explanations
…	…

COUNTERFACTUAL Explanations

- Imagine you applied for a loan, and it was **rejected**. A counterfactual explanation answers the question:

> *"What could I have changed to get my loan* ***approved*** *instead?"*

- It allows us to identify **the smallest possible change in inputs, that will lead to a different outcome.**



Counterfactual Examples

ML model's decision boundary

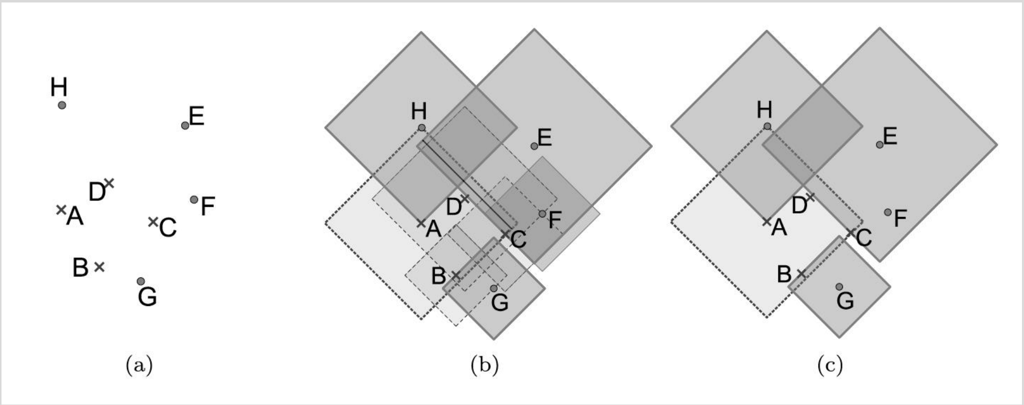
Original class:
Loan rejected

Desired class:
Loan approved

Original input

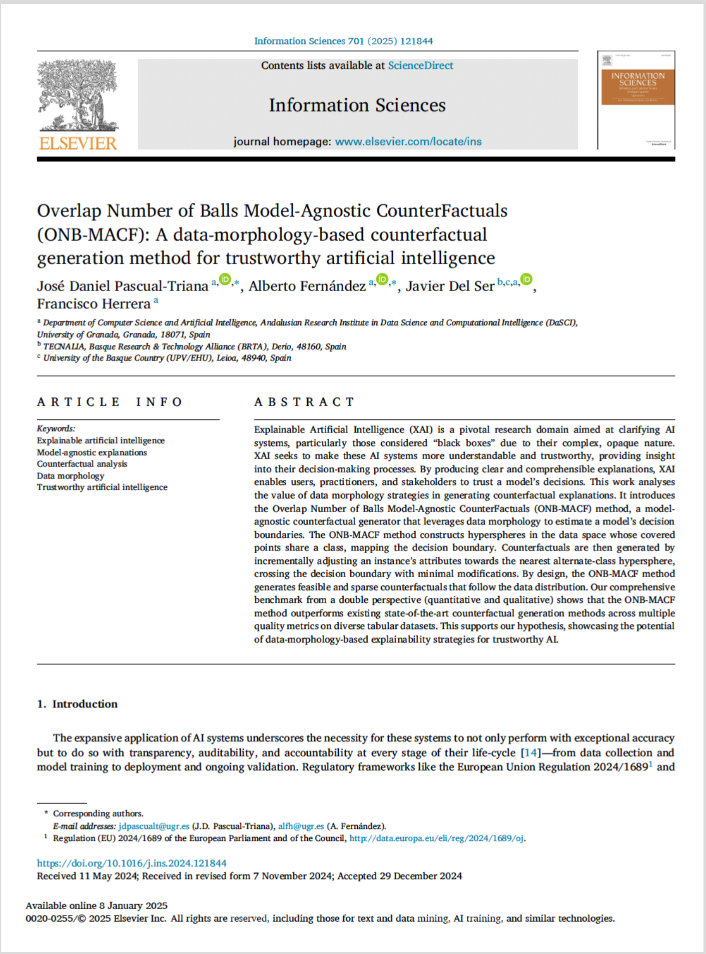
COUNTERFACTUAL Explanations

- Optimization-based methods
- Generative methods
- Gradient-based methods
- …



(a) (b) (c)

Pascual-Triana, J. D., Fernández, A., Del Ser, J., & Herrera, F. (2023). *Overlap Number of Balls Model-Agnostic Counterfactuals (ONB-MACF): A data-morphology-based counterfactual generation method for trustworthy artificial intelligence*. *Information Fusion*, 98, 101783. https://doi.org/10.1016/j.inffus.2023.101783



Information Sciences 701 (2025) 121844

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Overlap Number of Balls Model-Agnostic CounterFactuals (ONB-MACF): A data-morphology-based counterfactual generation method for trustworthy artificial intelligence

José Daniel Pascual-Triana [a,*], Alberto Fernández [a,*], Javier Del Ser [b,c,a], Francisco Herrera [a]

[a] *Department of Computer Science and Artificial Intelligence, Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada, 18071, Spain*
[b] *TECNALIA, Basque Research & Technology Alliance (BRTA), Derio, 48160, Spain*
[c] *University of the Basque Country (UPV/EHU), Leioa, 48940, Spain*

ARTICLE INFO

Keywords:
Explainable artificial intelligence
Model-agnostic explanations
Counterfactual analysis
Data morphology
Trustworthy artificial intelligence

ABSTRACT

Explainable Artificial Intelligence (XAI) is a pivotal research domain aimed at clarifying AI systems, particularly those considered “black boxes” due to their complex, opaque nature. XAI seeks to make these AI systems more understandable and trustworthy, providing insight into their decision-making processes. By producing clear and comprehensible explanations, XAI enables users, practitioners, and stakeholders to trust a model’s decisions. This work analyses the value of data morphology strategies in generating counterfactual explanations. It introduces the Overlap Number of Balls Model-Agnostic CounterFactuals (ONB-MACF) method, a model-agnostic counterfactual generator that leverages data morphology to estimate a model’s decision boundaries. The ONB-MACF method constructs hyperspheres in the data space whose covered points share a class, mapping the decision boundary. Counterfactuals are then generated by incrementally adjusting an instance’s attributes towards the nearest alternate-class hypersphere, crossing the decision boundary with minimal modifications. By design, the ONB-MACF method generates feasible and sparse counterfactuals that follow the data distribution. Our comprehensive benchmark from a double perspective (quantitative and qualitative) shows that the ONB-MACF method outperforms existing state-of-the-art counterfactual generation methods across multiple quality metrics on diverse tabular datasets. This supports our hypothesis, showcasing the potential of data-morphology-based explainability strategies for trustworthy AI.

1. Introduction

The expansive application of AI systems underscores the necessity for these systems to not only perform with exceptional accuracy but to do so with transparency, auditability, and accountability at every stage of their life-cycle [14]—from data collection and model training to deployment and ongoing validation. Regulatory frameworks like the European Union Regulation 2024/1689[1] and

[*] Corresponding authors.
E-mail addresses: jdpascualt@ugr.es (J.D. Pascual-Triana), alfh@ugr.es (A. Fernández).

[1] Regulation (EU) 2024/1689 of the European Parliament and of the Council, http://data.europa.eu/eli/reg/2024/1689/oj.

https://doi.org/10.1016/j.ins.2024.121844
Received 11 May 2024; Received in revised form 7 November 2024; Accepted 29 December 2024

Available online 8 January 2025

NEXT … how does that apply to RL?

Resources & Further Useful Links

- Molar, V. (2023). Interpretable Machine Learning. A Guide to Making Black Box Models Explainable. https://christophm.github.io/interpretable-ml-book/

- Lundberg, S. and Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. https://arxiv.org/abs/1705.07874

- Ribeiro et al., (2016). Why Should I trust You? https://arxiv.org/abs/1602.04938
- Kumar, et al., (2020). Problems with Shapley-value-based explanations as feature importance measures. https://arxiv.org/abs/2002.11097

- Wildi, M. and Hadji Misheva, B. (2022). A Time Series Approach to Explainability for Neural Nets with Applications to Risk-Management and Fraud Detection. https://arxiv.org/abs/2212.02906

- H20 documentation - https://docs.h2o.ai/h2o/latest-stable/h2o-docs/explain.html