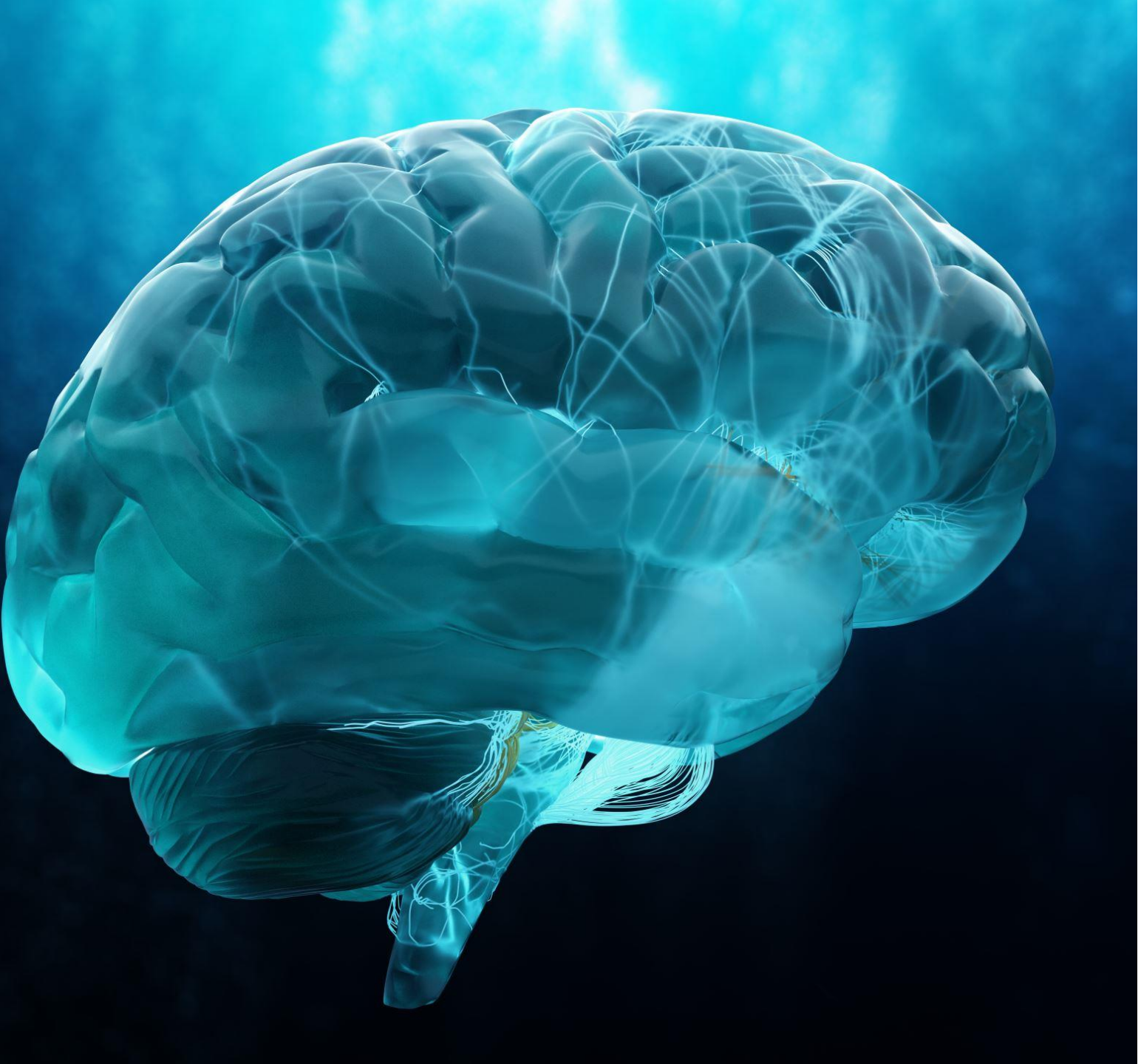
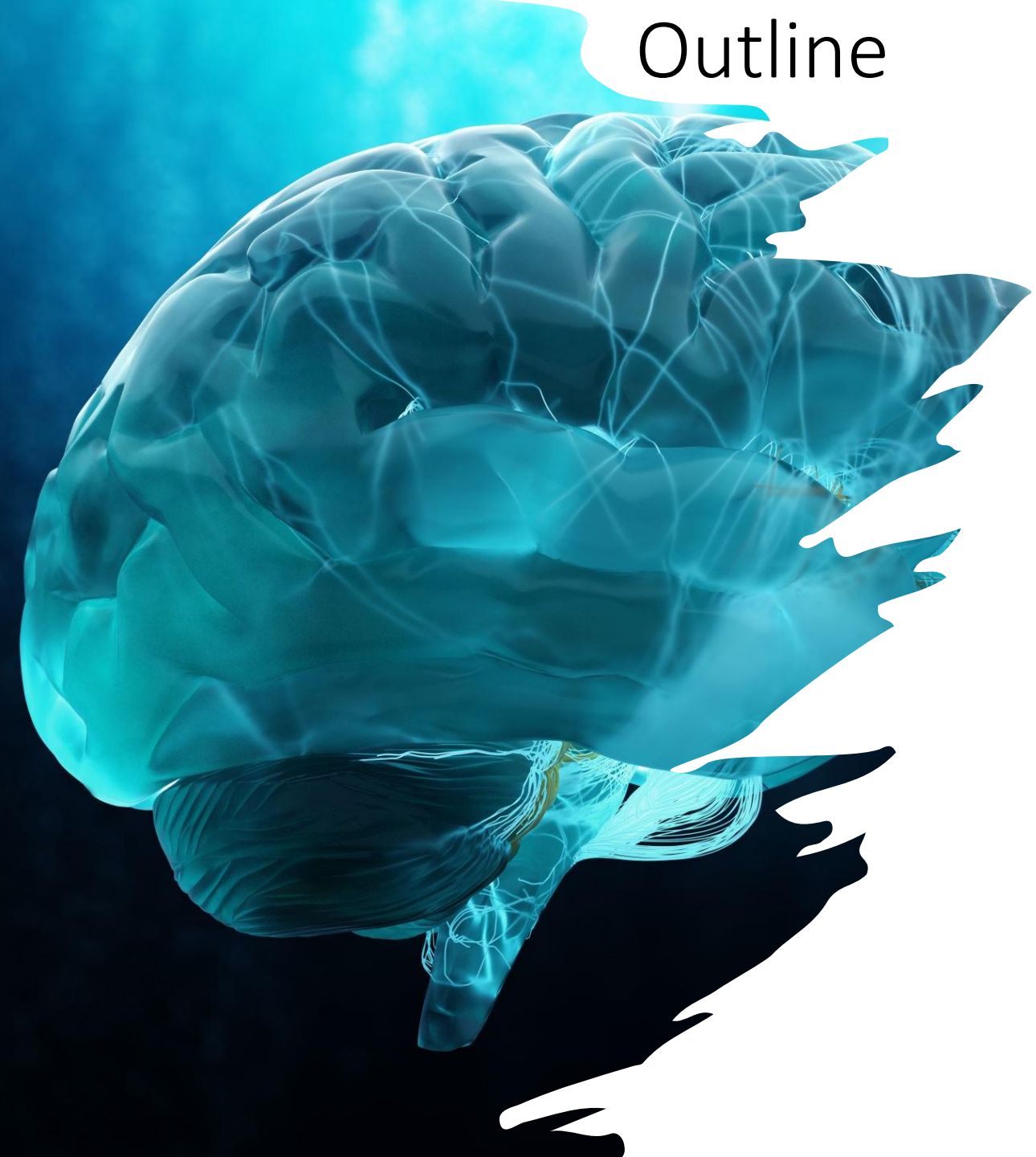


# Managing Artificial Intelligence

Nahla Ben Amor
Professor in Business Computing
University of Tunis

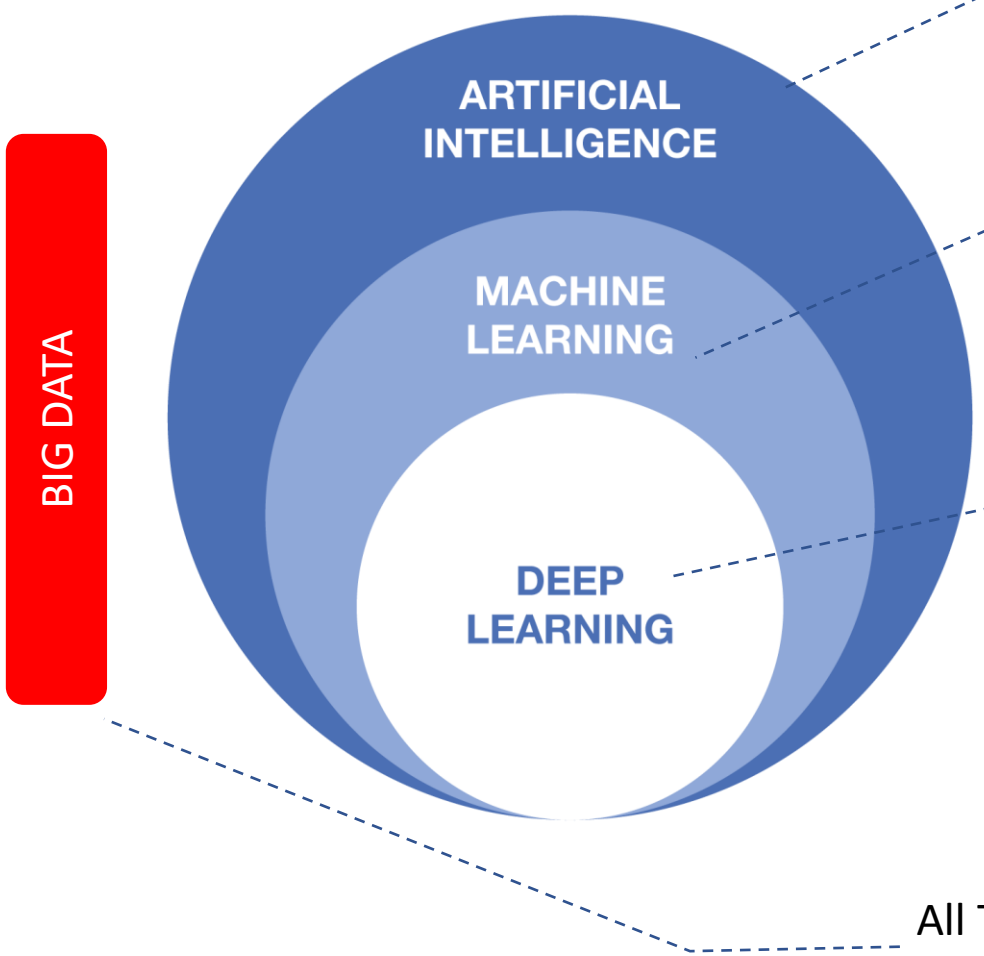
nahla.benamor@gmx.fr
nah.benamor@gmail.com

July, 2023



# Outline

- ▷ 1. Basics of Artificial Intelligence
  - ▷ 1.1 Definitions
    - Machine Learning
    - Deep Learning
  - ▷ 1.2 Main Domains of AI
- ▷ 2. Ethics and Responsibility in the Use of AI
  - ▷ 2.1 Ethical Issues Related to AI
  - ▷ 2.2 Sources of Bias in ML Lifecycle
  - ▷ 2.3 Regulatory and Normative Framework for AI
  - ▷ 2.4 Best Practices for Bias Avoidance/Mitigation
  - **Task at Hand: Ethical Implications and Trustworthy Use of AI**
- ▷ 3. Risk Management and Compliance with AI-Related Regulations
  - ▷ 3.1 Risk Management in AI Systems
  - ▷ 3.2 Adversarial Machine Learning
  - ▷ 3.3 AI Regulation
- ▷ 4. Impacts of AI on the World of Work
  - ▷ 4.1 Task Automation and Job Transformation
  - ▷ 4.2 How Artificial Intelligence Will Redefine Management?
- ▷ 5. Managing Artificial Intelligence
- ▷ Conclusion

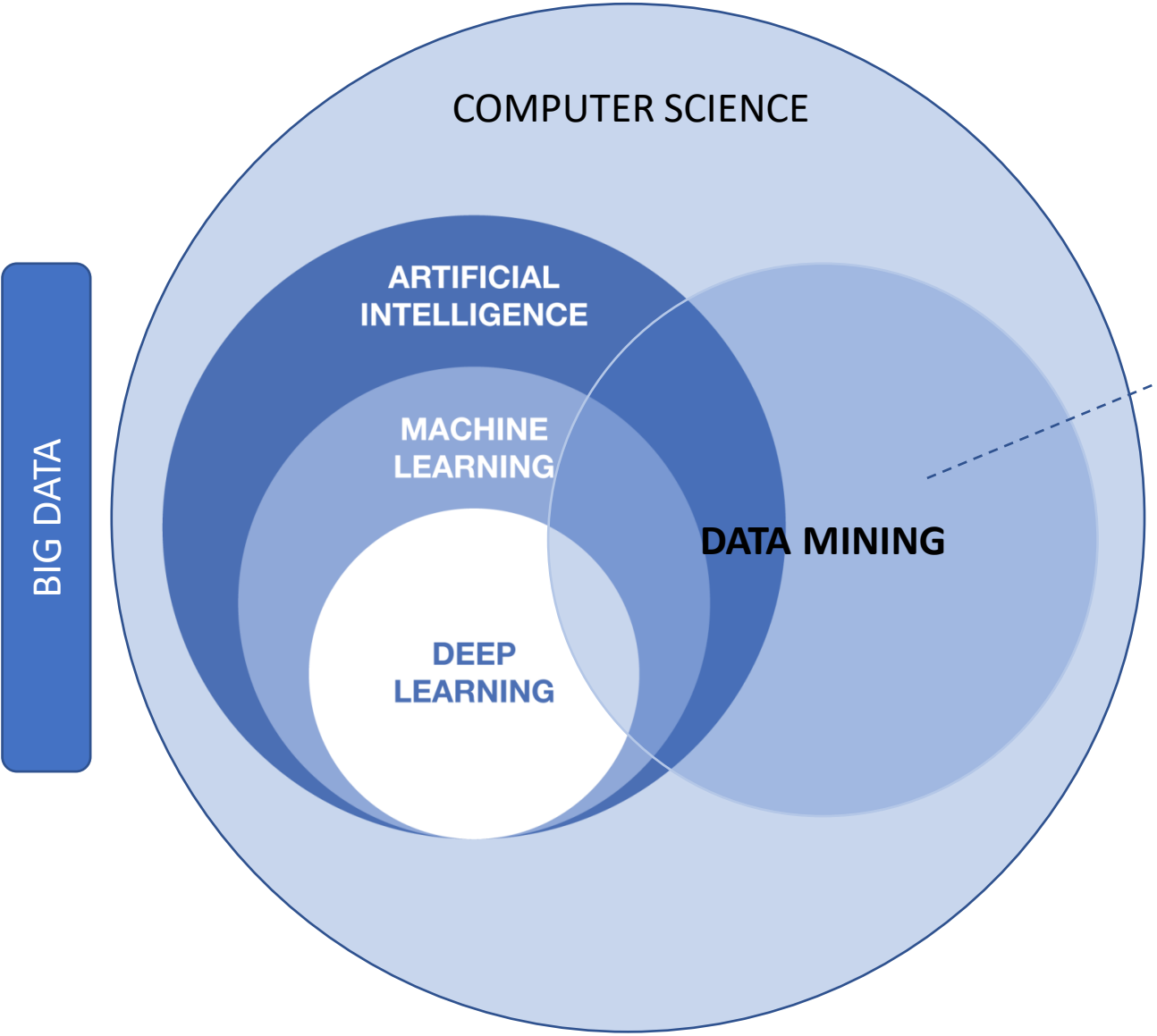



**AI: ARTIFICIAL INTELLIGENCE**
Enable machines to mimic human behavior

**ML: MACHINE LEARNING**
Subset of AI technique which enable machines learning from data without explicit programming.

**DL: DEEP LEARNING**
Subset of ML using artificial neural networks with multiple layers

All These techniques heavily rely on the availability of large amount of data for accurate results

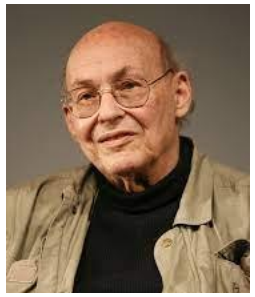



**DATA MINING**

**KNOWLEDGE DISCOVERY IN DATABASES (KDD)**

Refers to the process of discovering patterns, relationships, and insights from data.

# 70 years of History



**TURING TEST**

Alan Turing imagines the intelligent machine

**PERCEPTRON**

Frank Rosenblatt creates the Perceptron, the first neural network capable of learning.

**ELIZA**

The 1st chatbot

**DENDRAL**

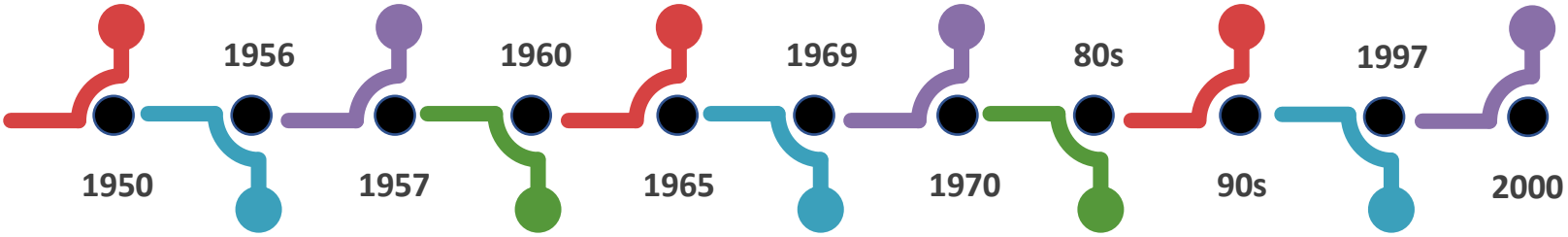
The 1st Expert System

**SEMANTIC NETWORKS**

Marvin Minsky and his colleagues at MIT developed the 1st form of semantic networks.

**MAJOR ADVANCES**

Multi-agent planning, data analysis, NLP, machine translation, computer vision, VR, games, etc.




**DARTMOUTH CONVENTION**

The Dartmouth Convention marks the official start of AI as a research field

**LISP**

Programming language designed for AI applications.

**1ST IJCAI CONFERENCE**

International join conf on AI

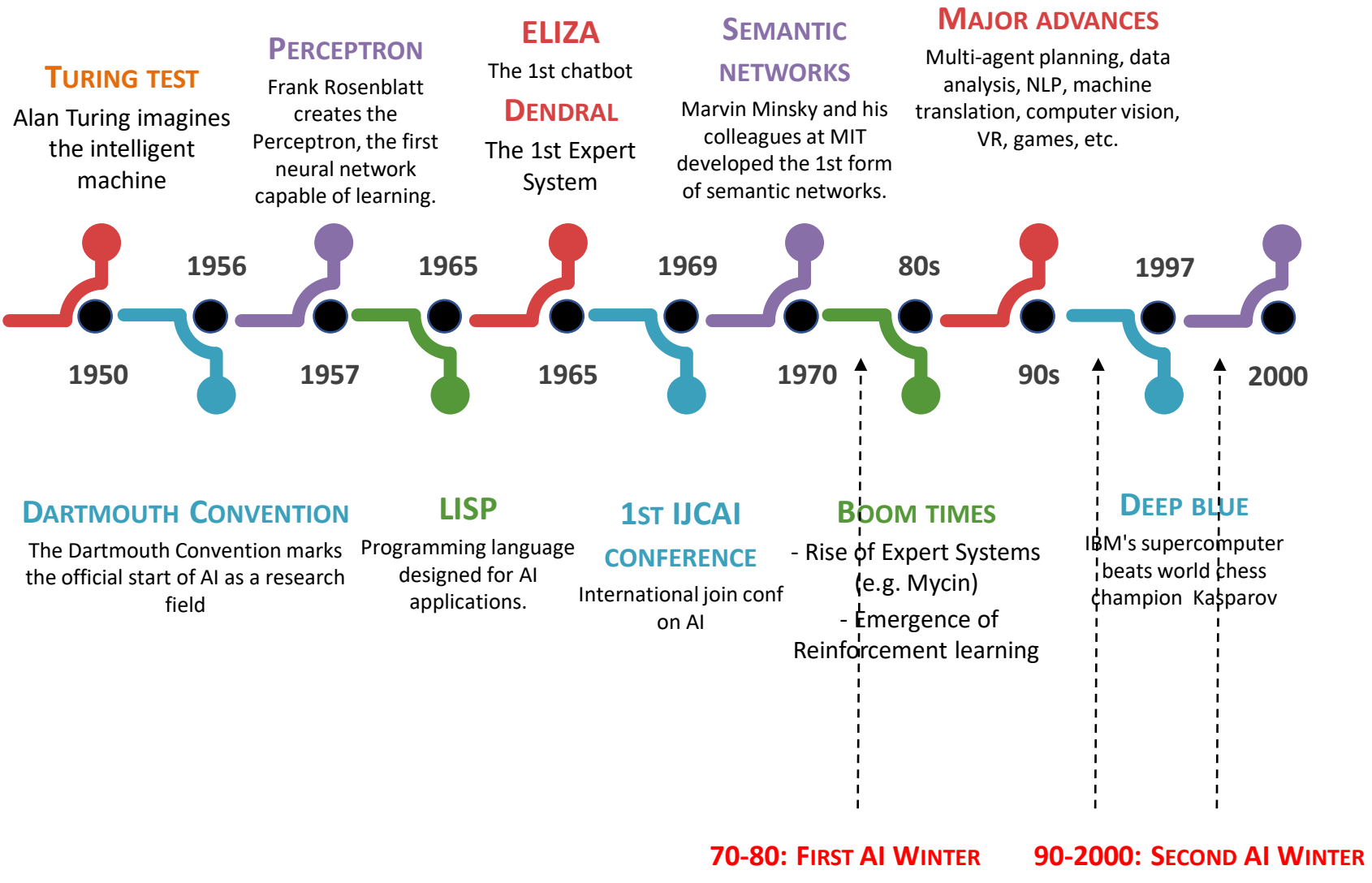
**BOOM TIMES**

- Rise of Expert Systems (e.g. Mycin)
- Emergence of Reinforcement learning

**DEEP BLUE**

IBM's supercomputer beats world chess champion Kasparov


# 70 years of History




- The 'Two AI Winters' refer to two periods of dissilusion and stagnation in the development of AI.
- During each winter, interest and progress in AI were revitalized by new technological advancements and a better understanding of the challenges.

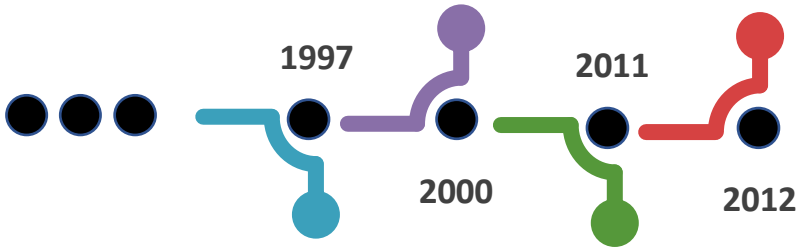
# 70 years of History

**EMERGENCE OF DEEP LEARNING**

Geoffrey Hinton laid the foundations for the concepts and techniques that are now at the heart of DL

**POPULARITY OF DEEP LEARNING**

Deep learning and artificial neural networks gain in popularity (SuperVision)




**DEEP BLUE**

Deep Blue beats Kasparov at chess.

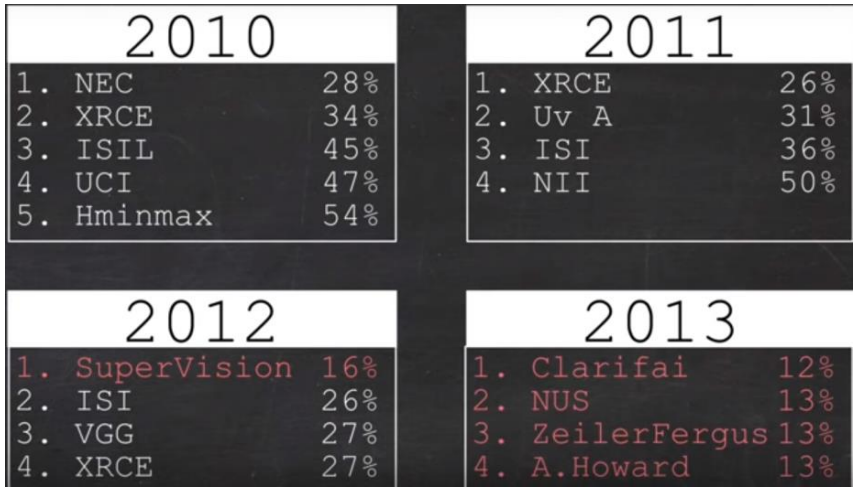
**WATSON**

IBM Watson wins Jeopardy game show

**SIRI**

Apple integrates Siri, an intelligent virtual assistant

- The popularity of deep learning rise primarily due to the release of the famous **IMAGENET** database in 2007 by Stanford's Vision Lab, where several million photos were collected and labeled.
- In 2010, IMAGENET had gathered 15,000,000 categorized images based on their specific features (vehicles, animals, etc.).
- In 2012, Hinton and his team (Supervision) won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with a convolutional neural network called AlexNet.



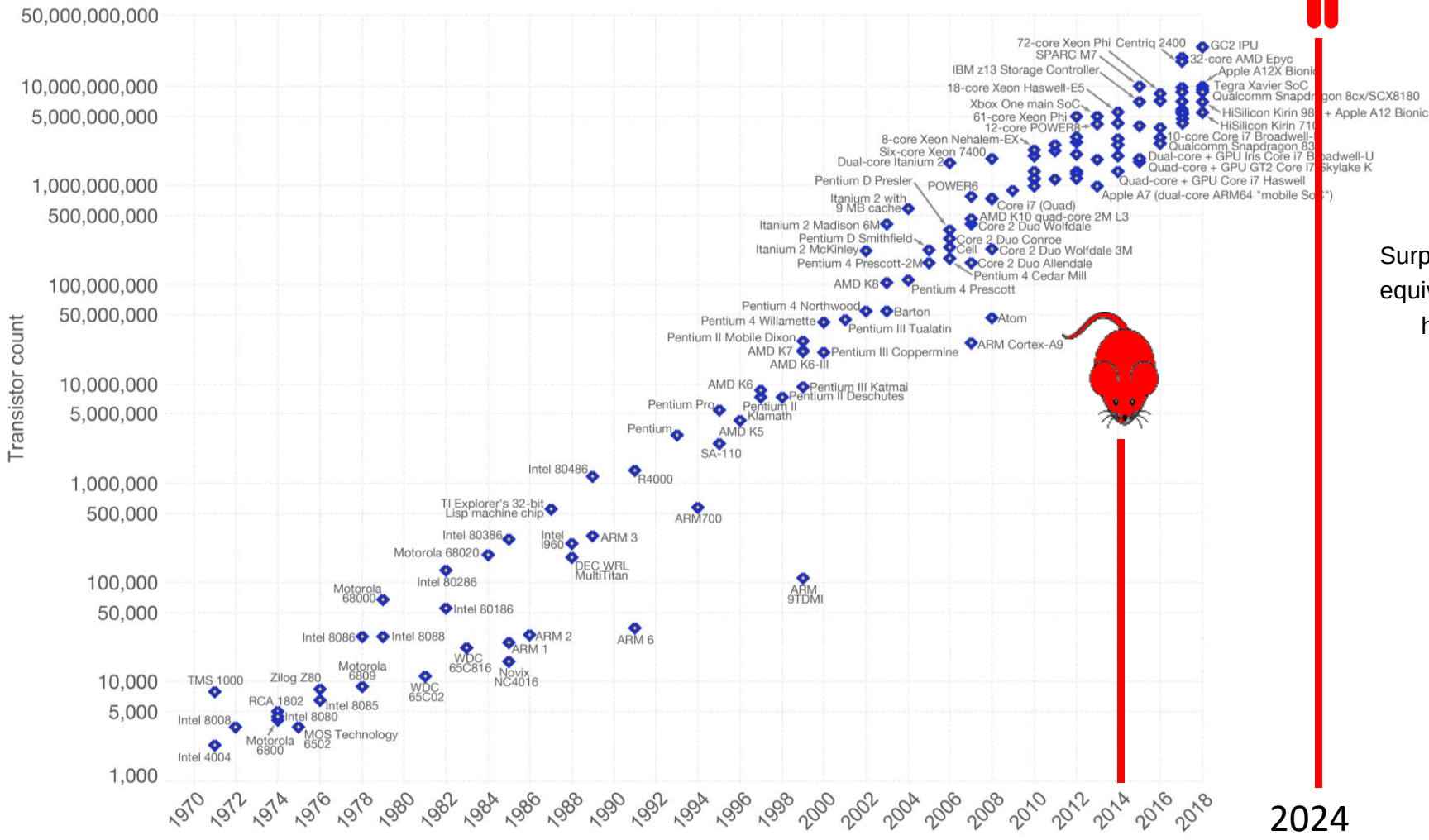

IMAGENET

# WHY DOES IT WORK?

# Why does it work?

1) Technological progress → Exponential growth in computing power.

- Moore's Law: describes the empirical regularity that the number of transistors on a computer chip doubles every two years, while the cost of computers is halved (making it more accessible).
- We estimate that in 2014, a computer had the processing power equivalent to that of a mouse's brain, and we are now at the level of a human brain




2024

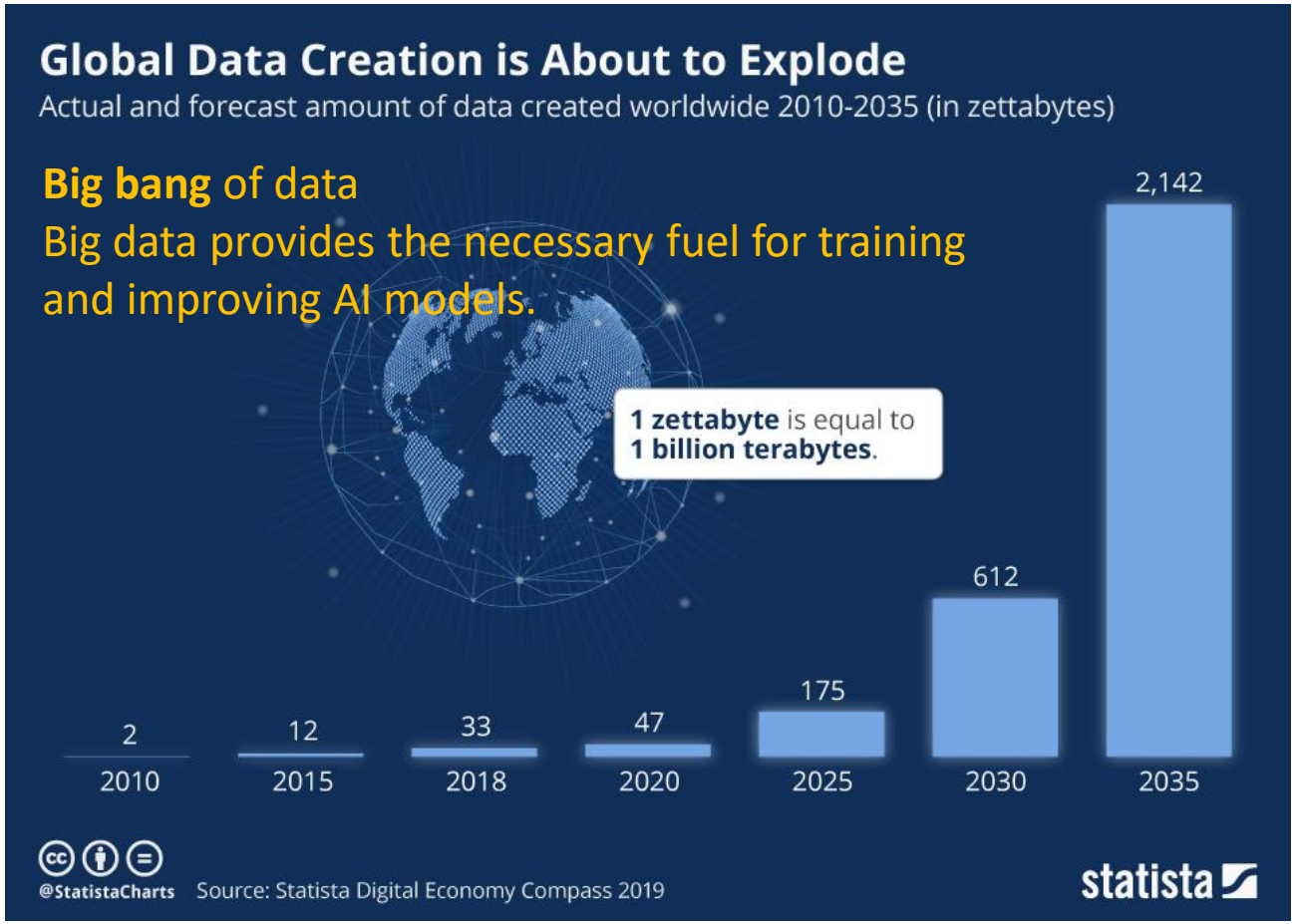
2045

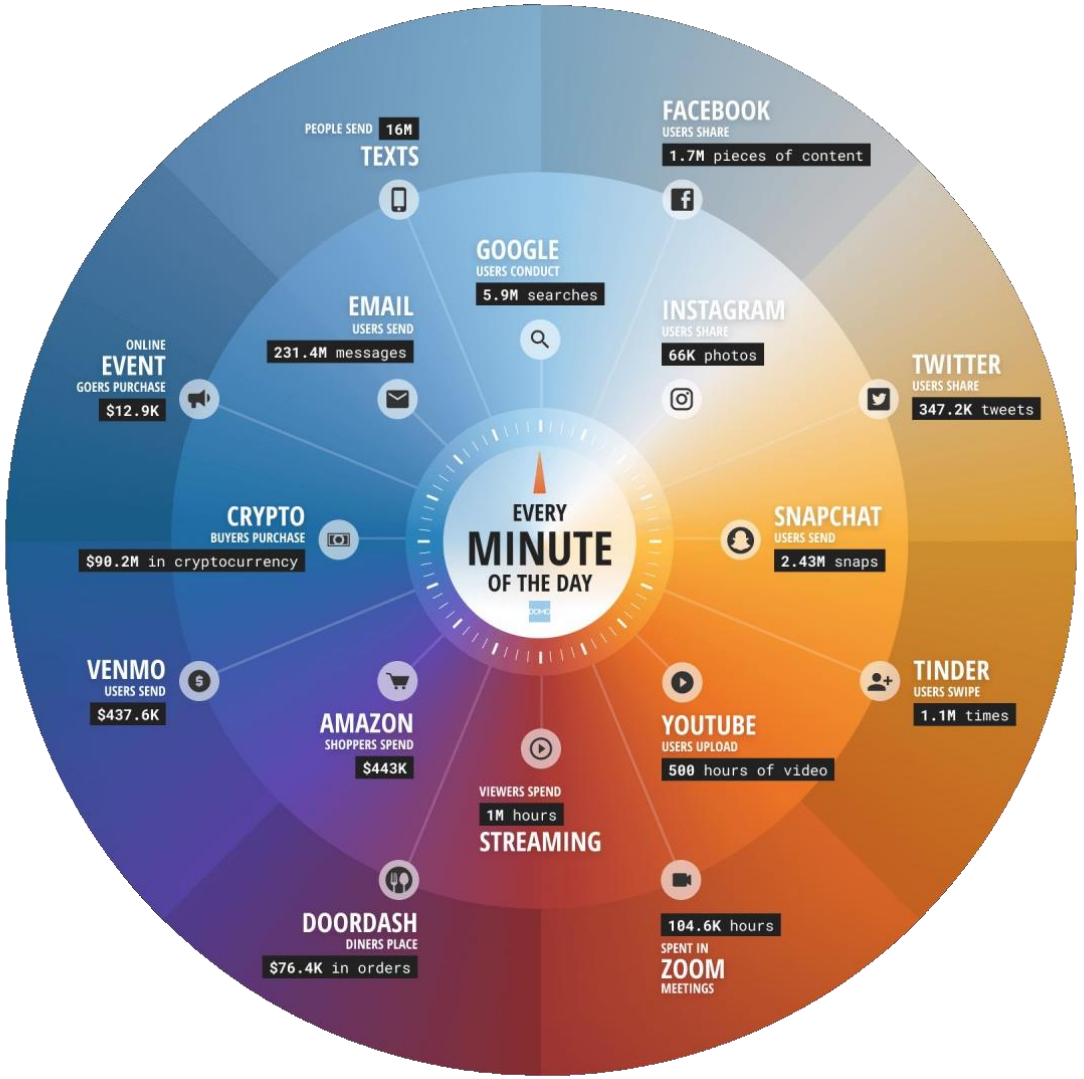
Surpasses brainpower equivalent to that of all humans brains combined

Data source: Wikipedia (https://en.wikipedia.org/wiki/Transistor_count)
The data visualization is available at OurWorldinData.org. There you find more visualizations and research on this topic.

# Why does it work?

2) Big data

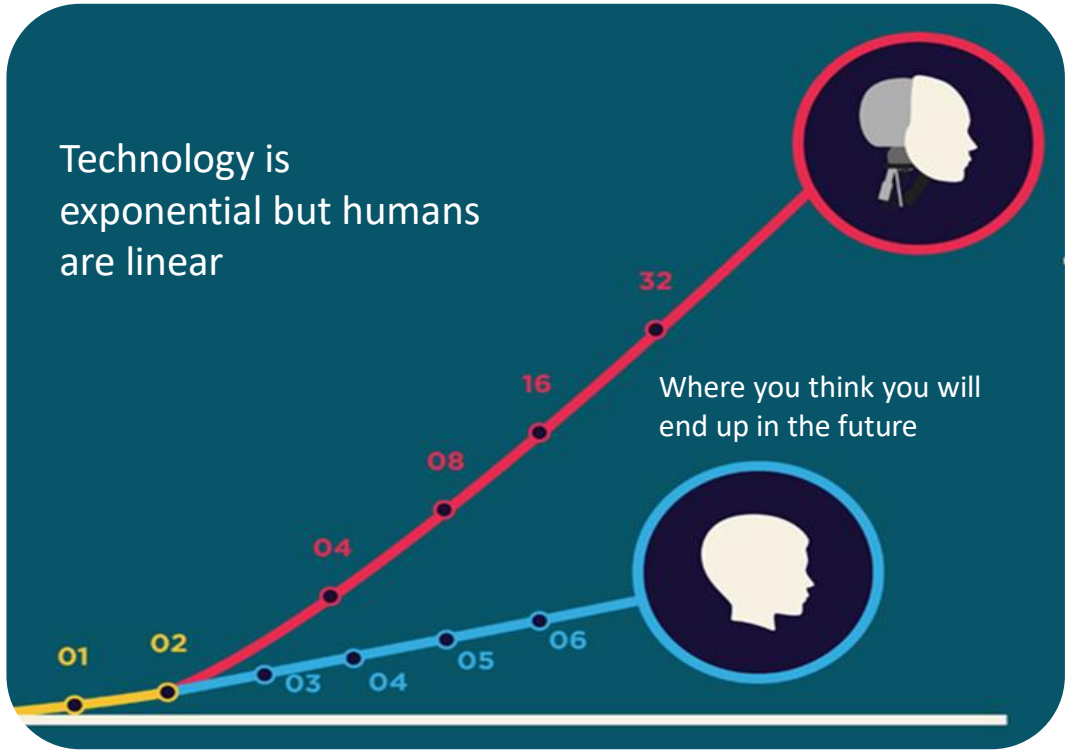



1 minute Internet 2022

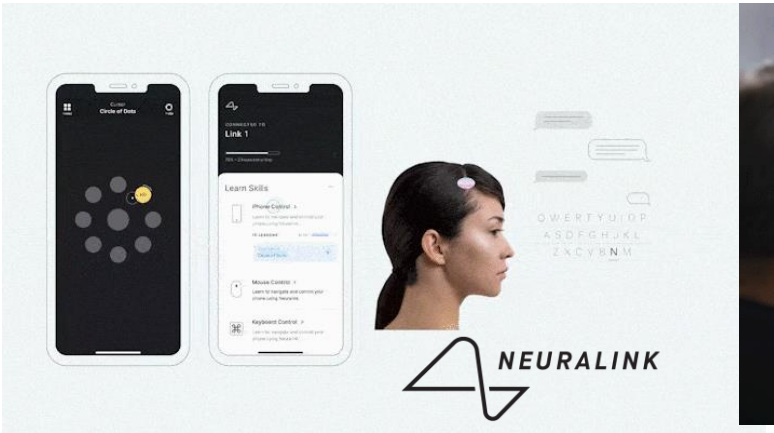
# Why does it work?

## 3) Acceptation / Adaptation



# Technology is exponential but humans are linear
## Evolution of telephony


- Looking back at the evolution of telephony from 1876 until now
- In the 1980s, it was impossible to imagine the advanced phones 2000, and in 2000, it was unimaginable to predict the voice, image, and multimedia capabilities of the phones we have today.
- Next 20 years ? enhanced cameras
- But it is unlikely that we can envision technologies like neuralink, which involves communicating through a brain chip, or seeing holographic representations of our conversation partners, as demonstrated by projects like Google's Project Starline.

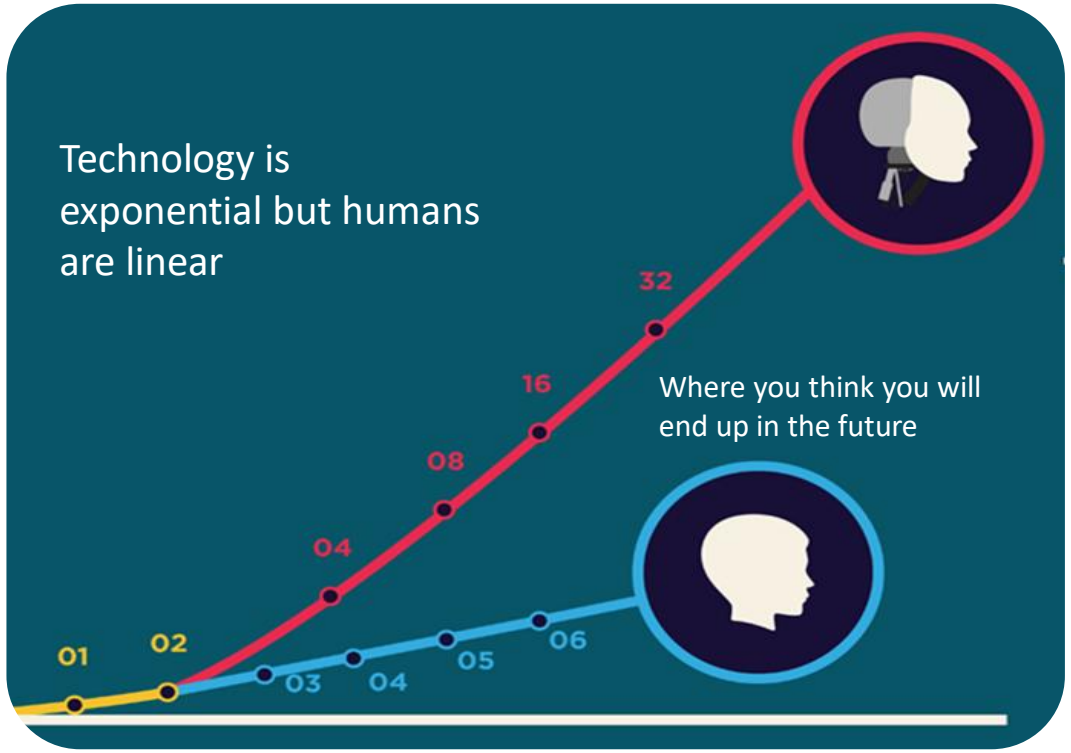


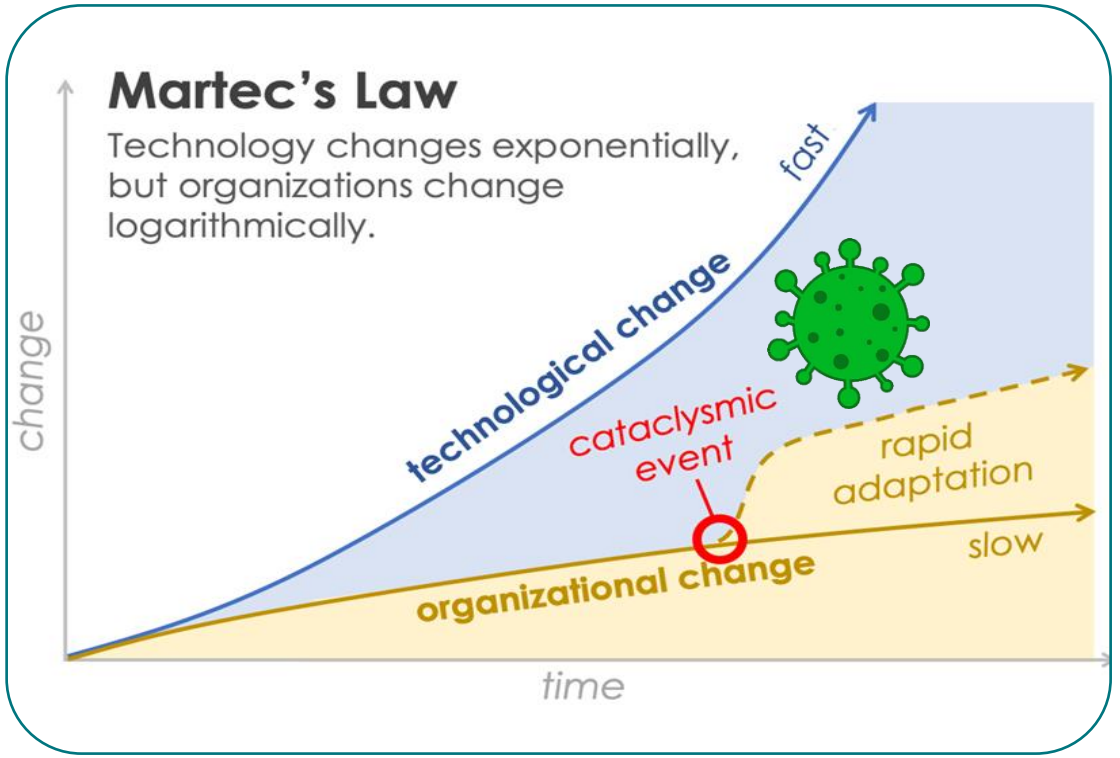

**Neuralink** : Brain implant

**Project Starline**: Google's new video chat technology offering a "life-size" 3D experience (prototype)

# Why does it work?

## 3) Acceptation / Adaptation



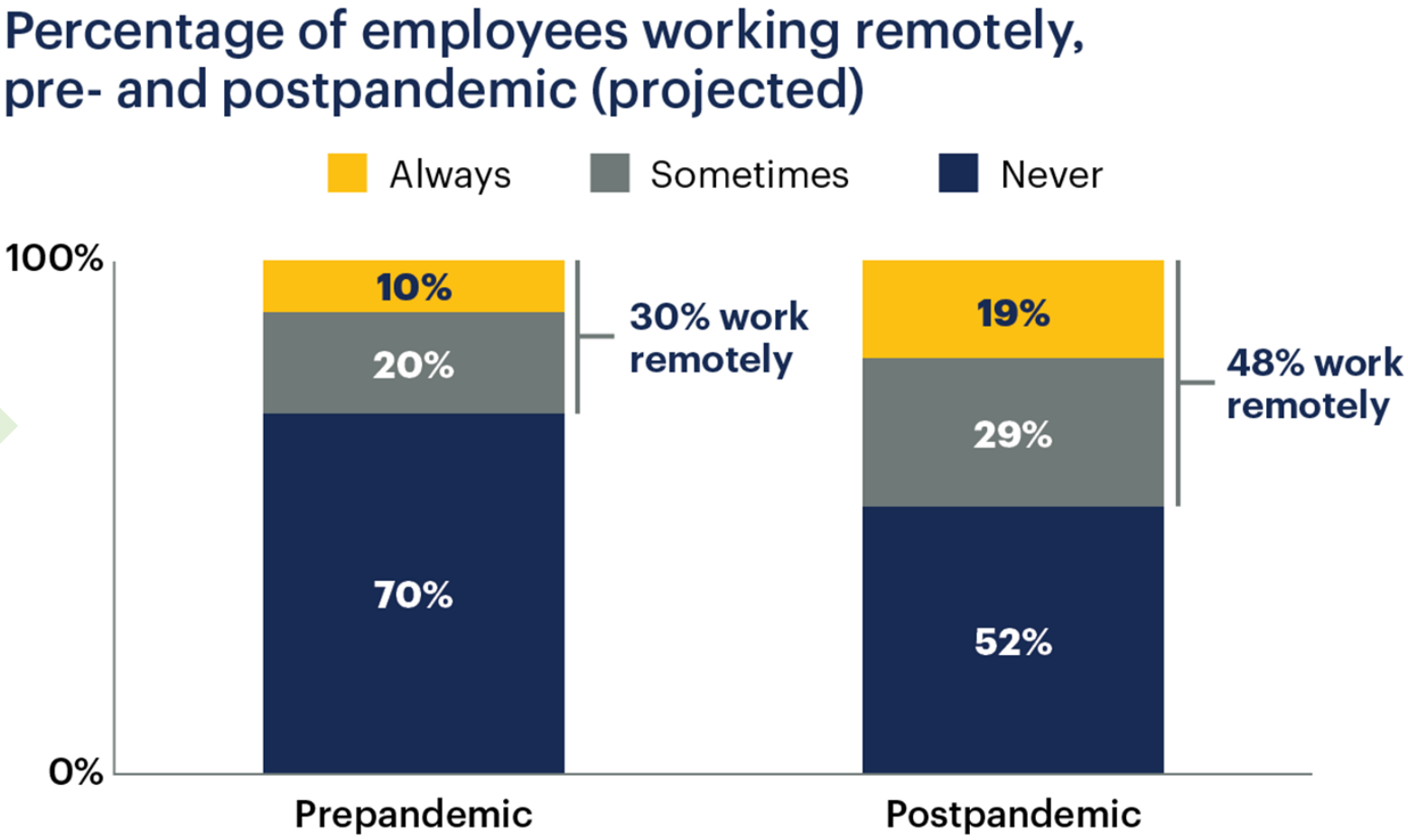




The ability to underestimate the unintuitive rate of change is not isolated to individuals — it also applies to organizations.

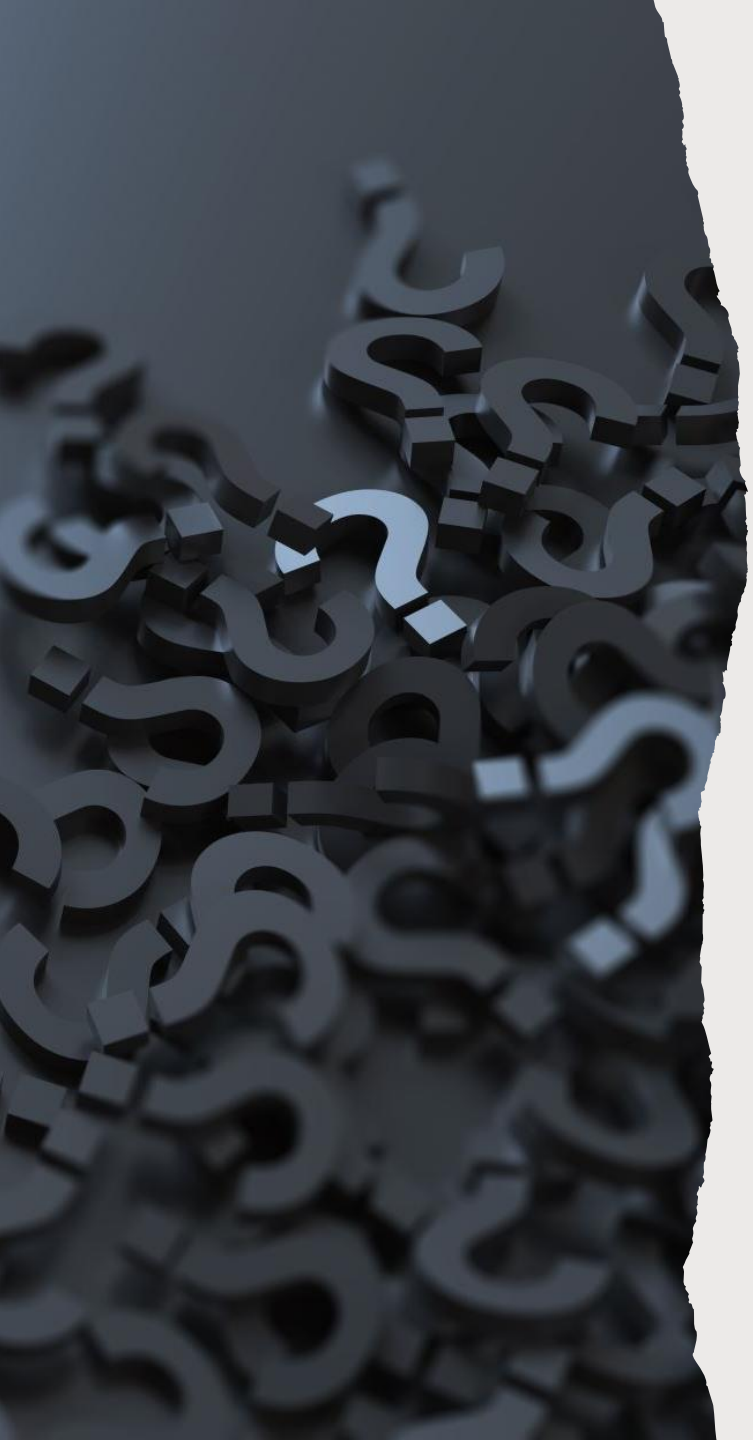
# Covid-19 Pandemic accelerates remote work

A trend likely to remain

- Nearly half of employees worked remotely full-time during the pandemic
- 62% of employees now expect their employers will allow them to work remotely moving forward




Source: Gartner.com (2020)



# Formal Definitions

# Traditional programming vs Machine learning

## TRADITIONAL PROGRAMMING

Input

1. Cut vegetables
2. Season chicken
3. Preheat oven
4. Cook chicken for 30-mn
5. Add vegetables

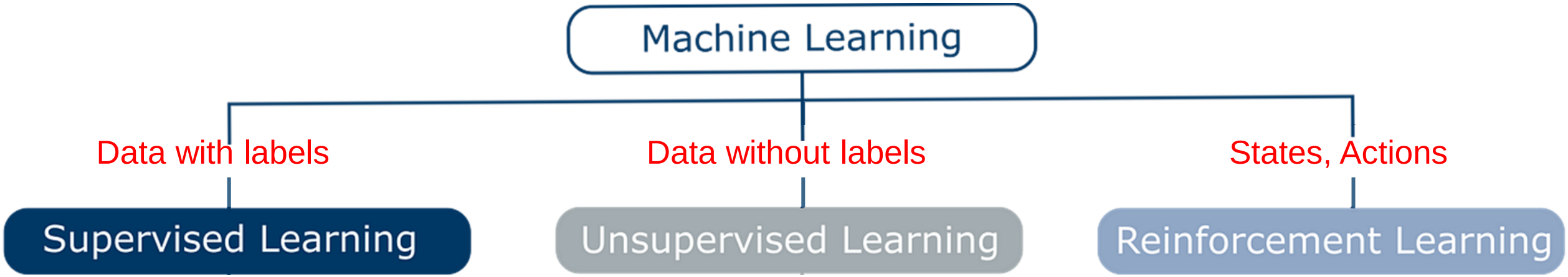
Output

## MACHINE LEARNING

Input

Output

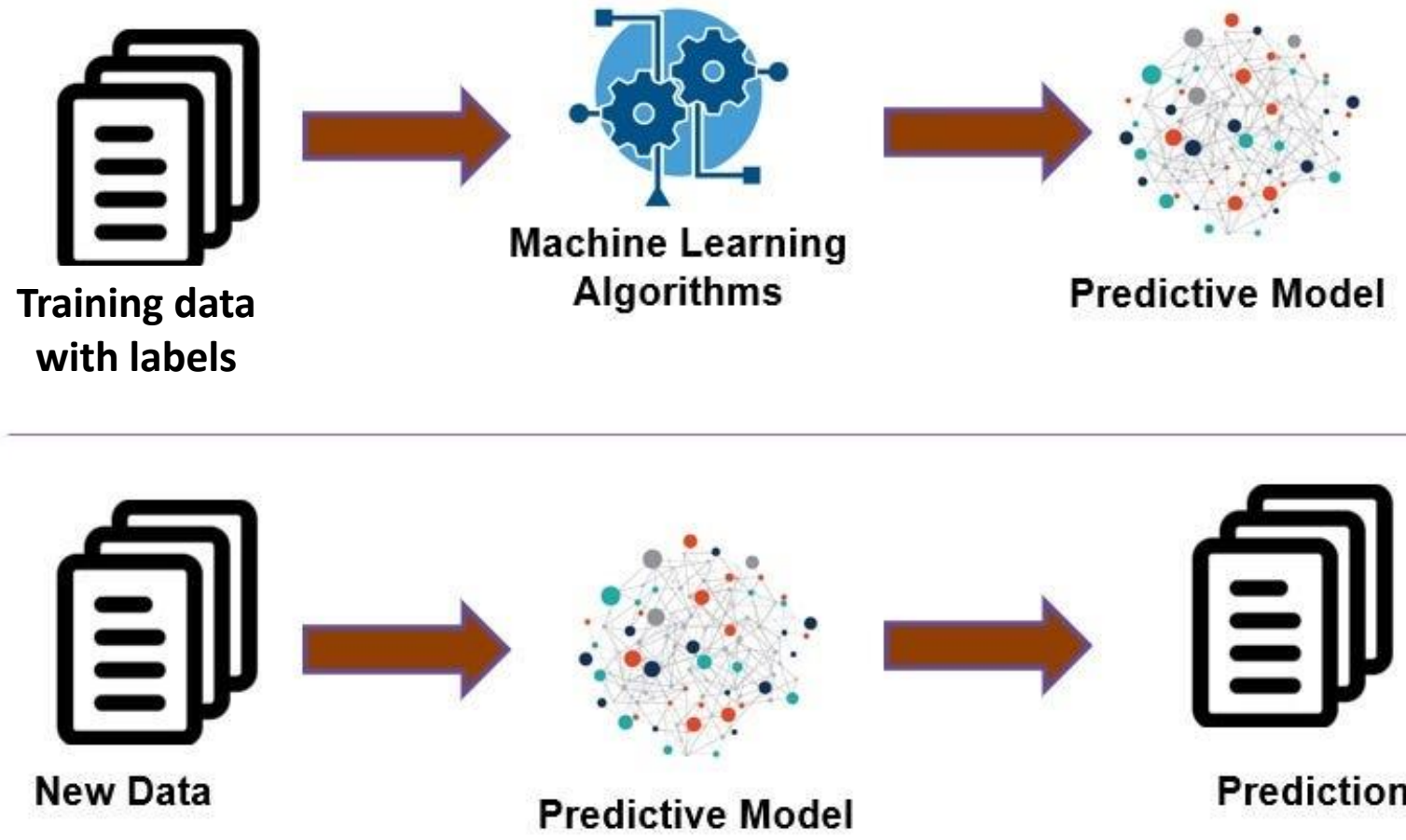
1. Cut vegetables
2. Season chicken
3. Preheat oven
4. Cook chicken for 30-mn
5. Add vegetables

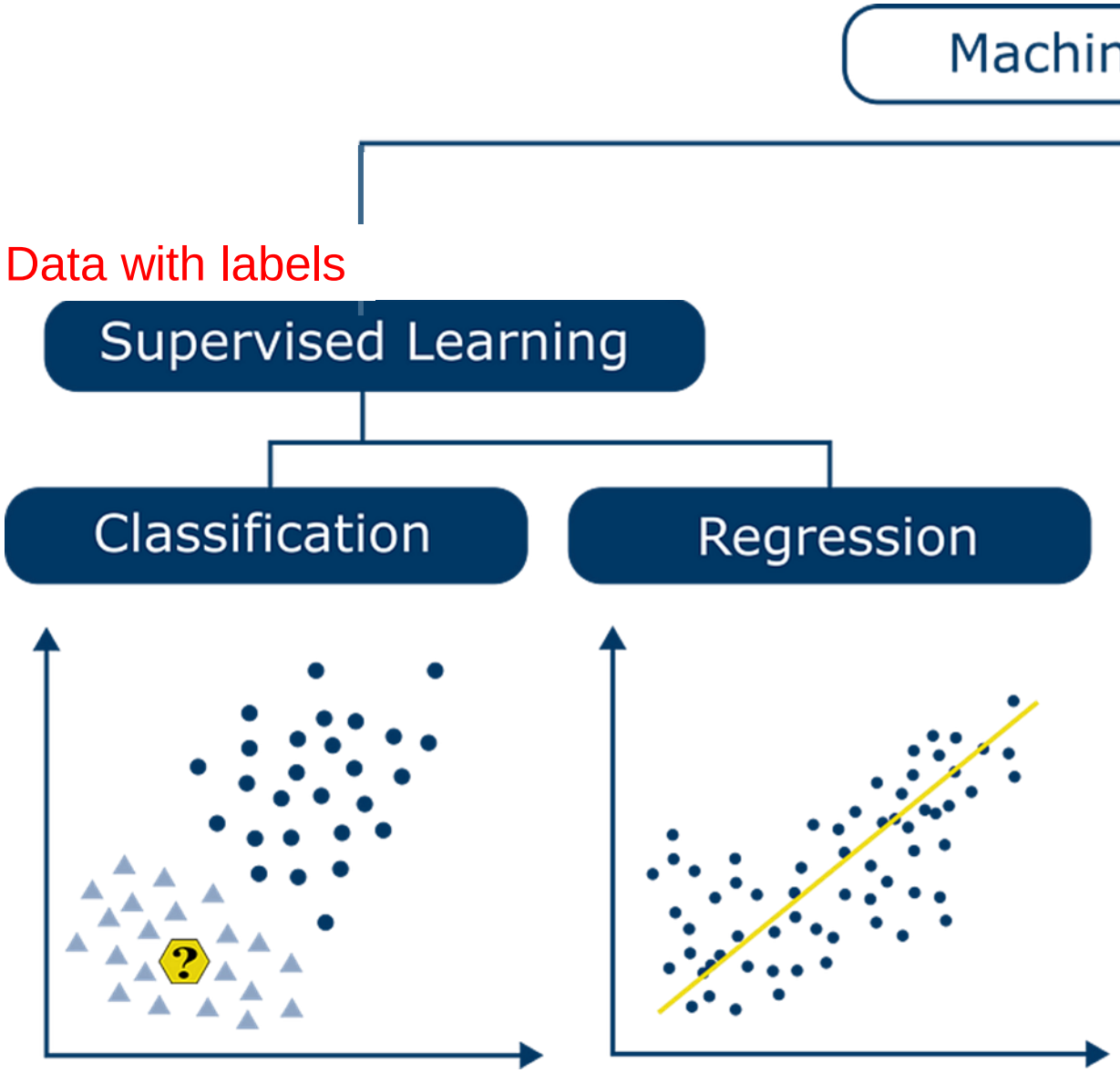

Machine Learning
Data with labels
Data without labels
States, Actions
Supervised Learning
Unsupervised Learning
Reinforcement Learning

Machine Learning

Data with labels

Supervised Learning



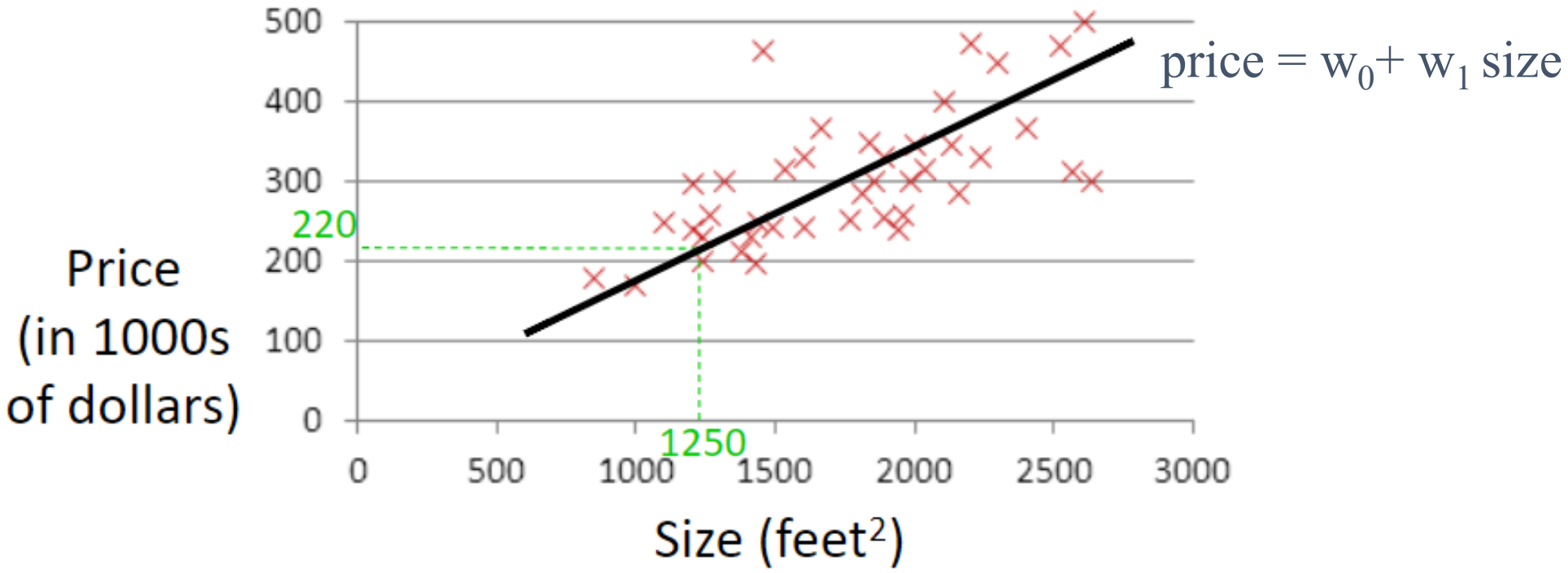



- **Classification** is the task of assigning input data to predefined categories or classes, predicting **discrete values** or labels based on input features.
- Applications: classifying emails as spam or not spam or categorizing images into different object classes.

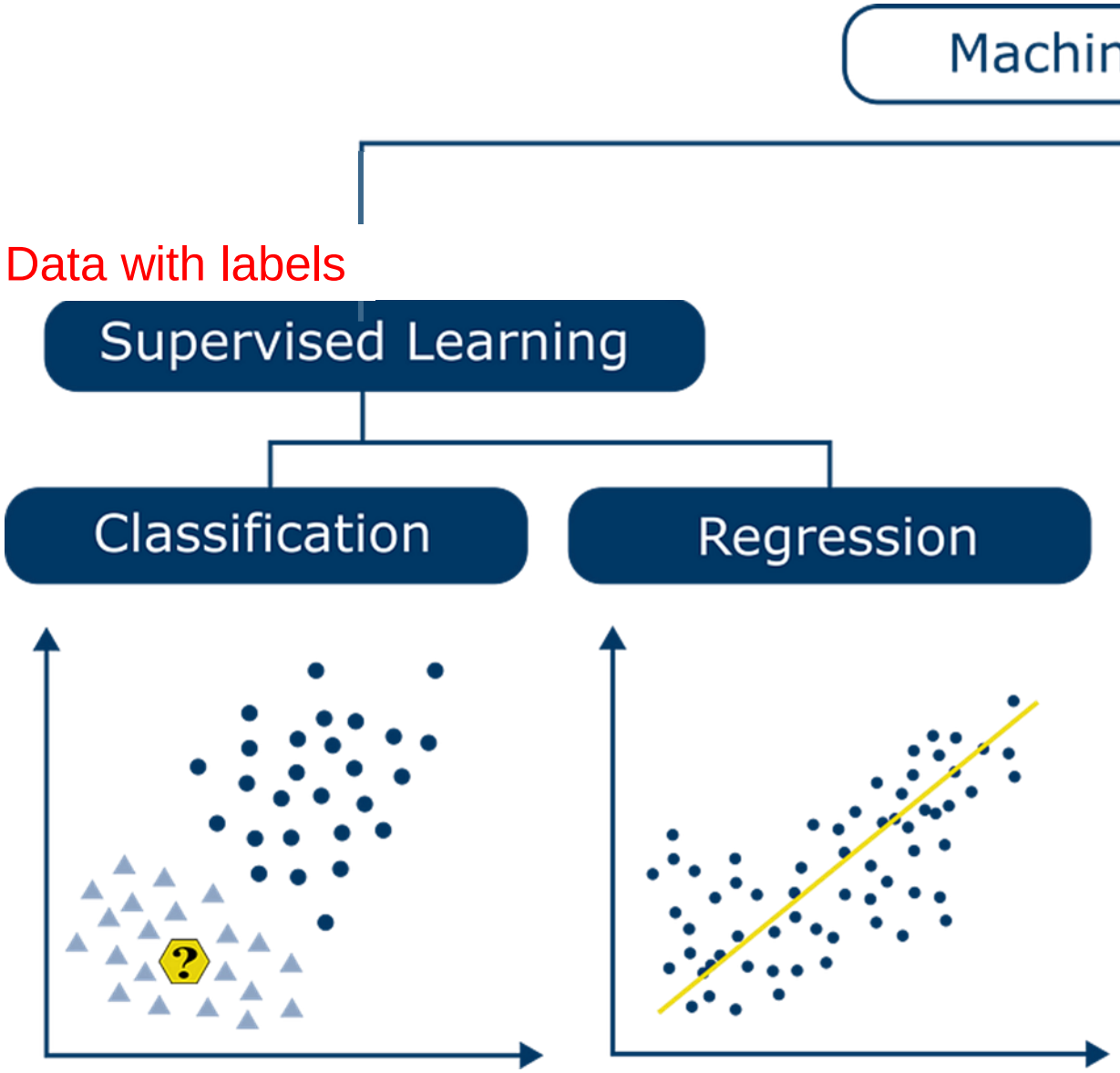
- **Regression**, involves predicting **continuous or real-valued output variables** based on input features
- Applications: Regression problems include predicting house prices based on factors like square footage, number of bedrooms, and location, or forecasting product sales volume based on advertising expenditure and other factors.

# Simple Example: Linear Regression

- The easiest model is regression, which is widely used in many fields.
- [Linear regression: mother of all learning algorithms]

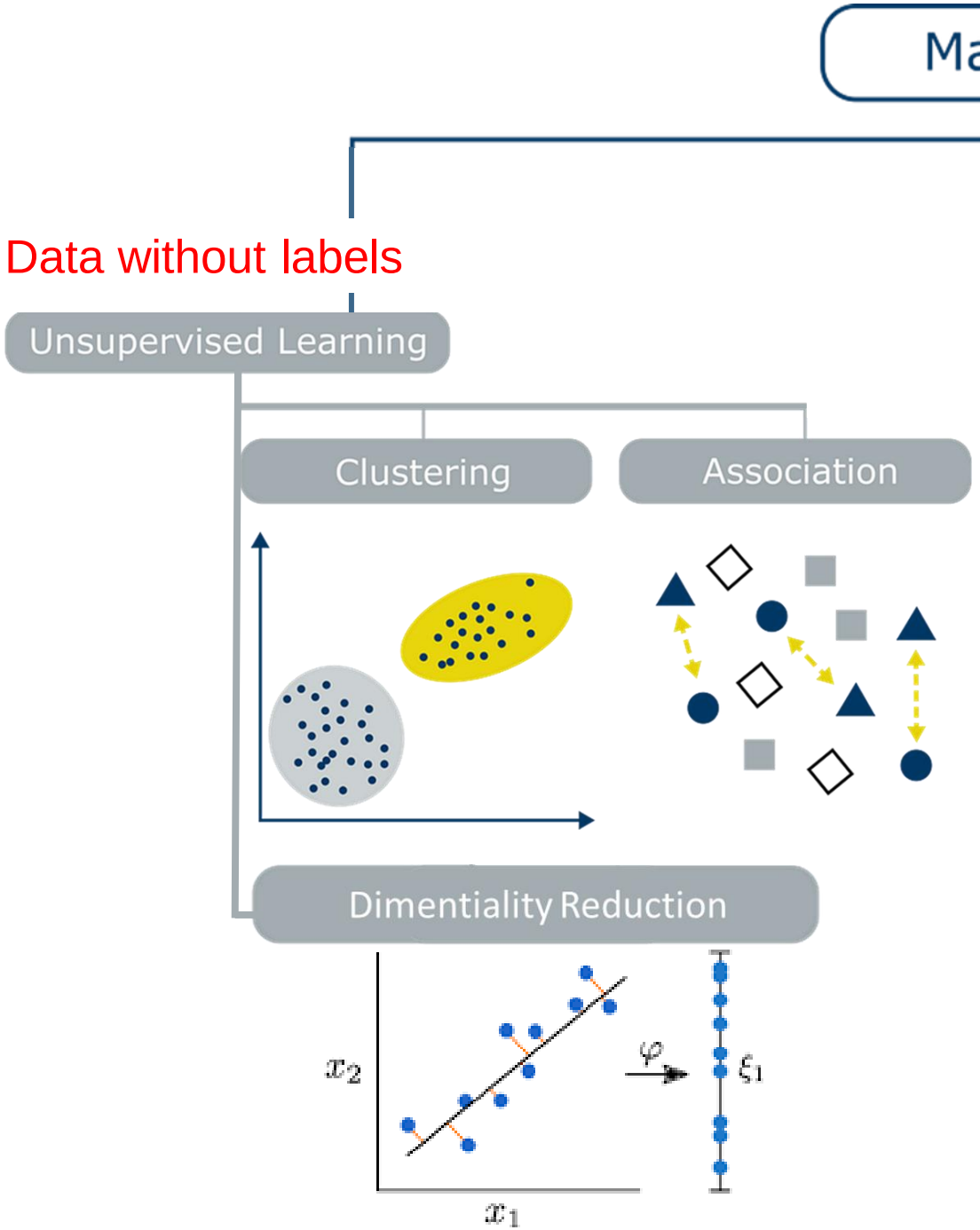



Model the price of a house according to its m² surface area




- **Classification Algorithms**
  - Support Vector Machines (SVM)
  - K-Nearest Neighbors (KNN)
  - Naive Bayes
  - Decision Trees
  - Random Forests
  - Logistic Regression
  - Neural Networks (e.g., Multilayer Perceptron)

- **Regression Algorithms**
  - Linear Regression
  - Lasso Regression
  - Decision Trees
  - Random Forests
  - Gradient Boosting (e.g., XGBoost, LightGBM)
  - Neural Networks (e.g., Feedforward Neural Networks)




- **Clustering** : Group similar data based on their common characteristics. It involves finding inherent structures or patterns within the data without having predefined categories to follow.
- Applications: Market segmentation, Recommander systems

- **Association analysis** focuses on finding relationships or associations among items in a dataset. It identifies co-occurrence patterns and dependencies between items
- Applications: market basket analysis, recommendation systems, understanding customer behavior.

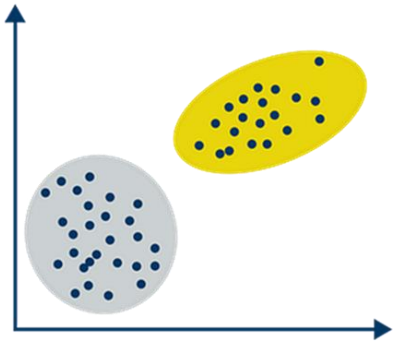
- **Dimensionality reduction** aim to reduce the number of input variables or features while preserving the important information in the data. This helps in simplifying the dataset, removing redundant or irrelevant features, and improving computational efficiency.
- Applications: data visualization, feature extraction, Structure discovery

# Machine Learning

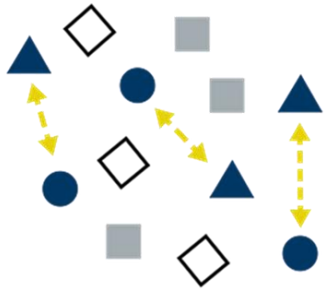
Data without labels

## Unsupervised Learning

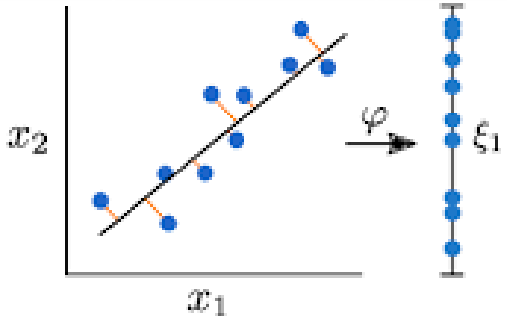
### Clustering

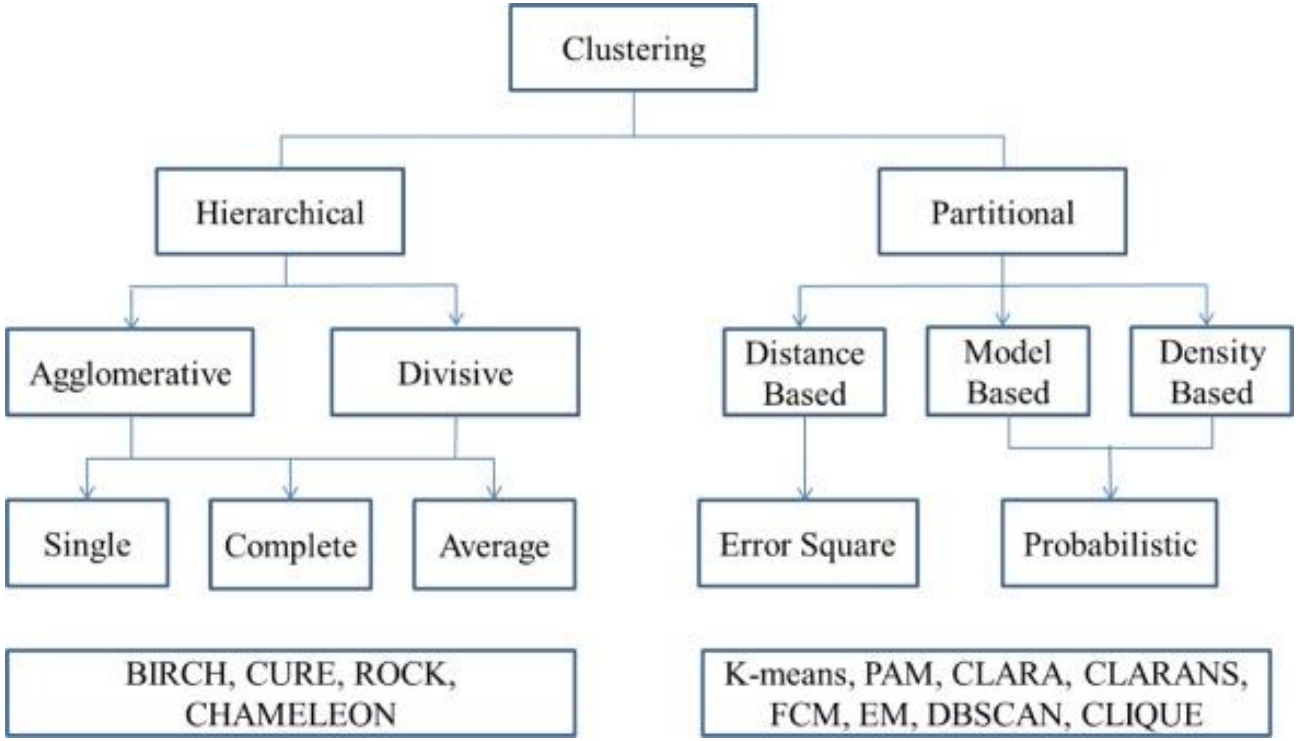


### Association



### Dimentiality Reduction



# Real Application of Clustering:

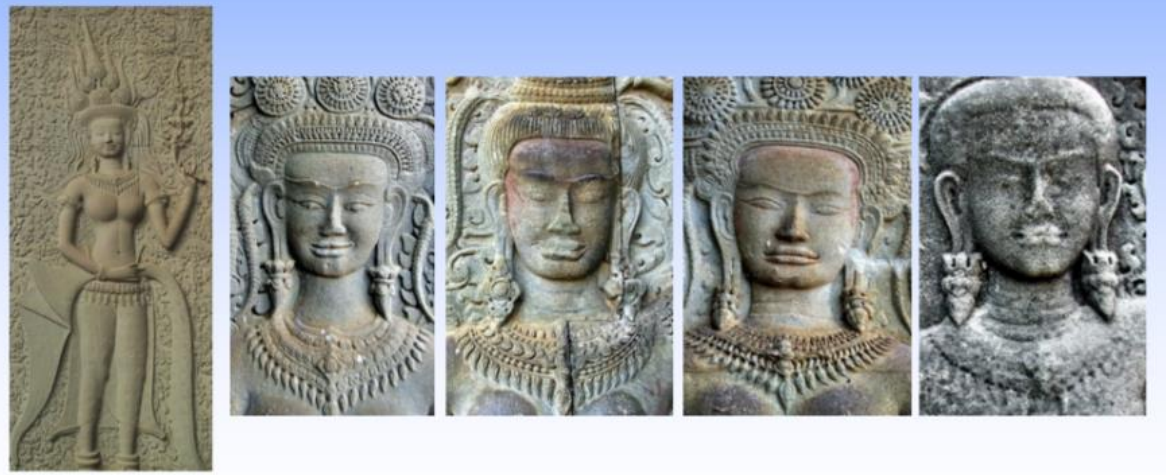
Clustering of Apsara Faces [divine dancers] (Cambodge)
https://pdfs.semanticscholar.org/1823/98ca4d85c25c64ba238dad10cafc92203660.pdf

## Angkor Wat


Hindu temple built by a Khmer king ~1,150AD; Khmer kingdom declined in the 15th century; French explorers discovered the hidden ruins in late 1800's

## Apsaras of Angkor Wat

- Angkor Wat contains the most unique gallery of ~2,000 women depicted by detailed full body portraits
- What facial types are represented in these portraits?



Kent Davis, "Biometrics of the Godedess", DatAsia, Aug 2008
S. Marchal, "Costumes et Parures Khmers: D'apres les devata D'Angkor-Vat", 1927

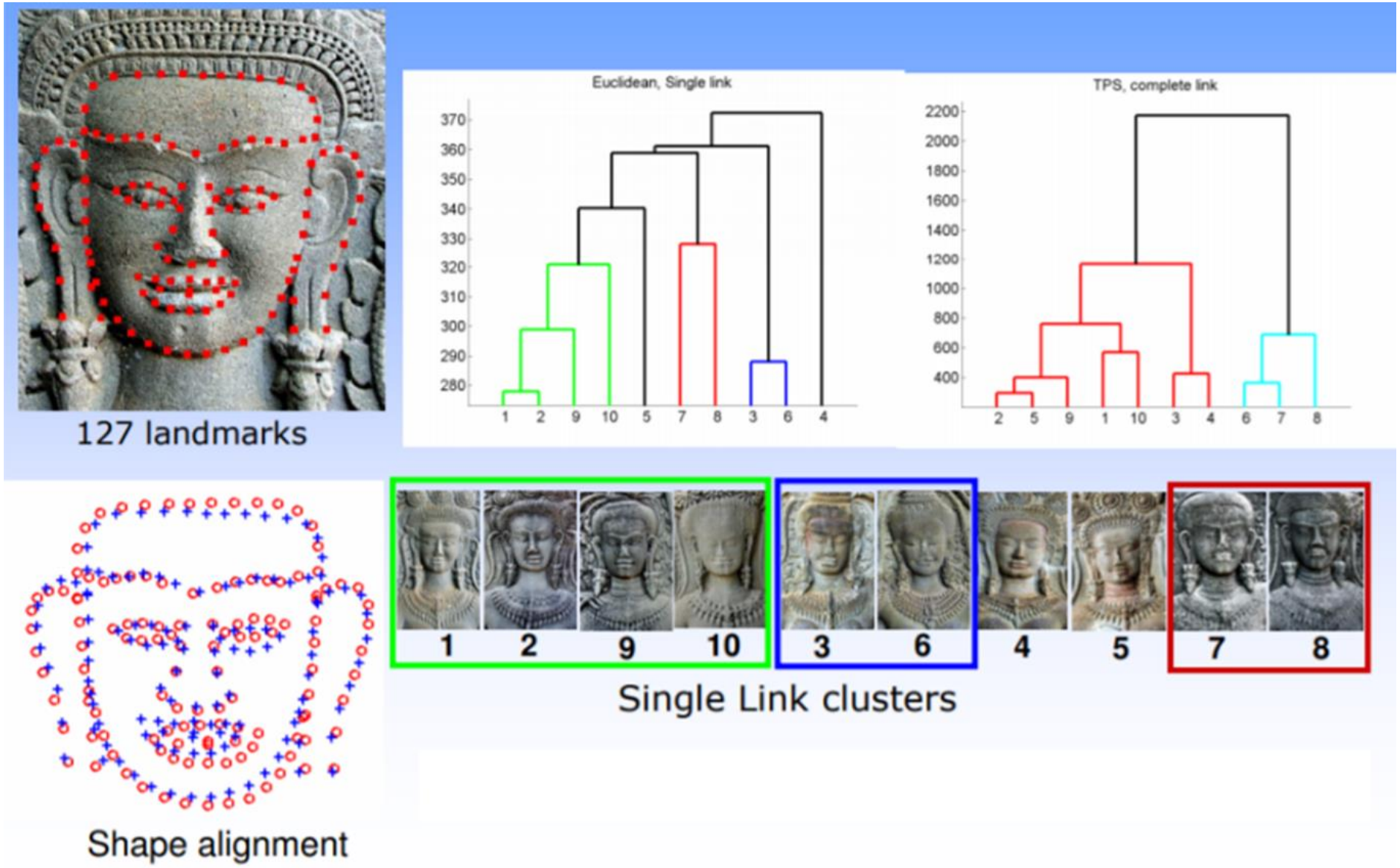


25

# Real Application of Clustering:

Clustering of Apsara Faces (Cambodge)

https://pdfs.semanticscholar.org/1823/98ca4d85c25c64ba238dad10cafc92203660.pdf

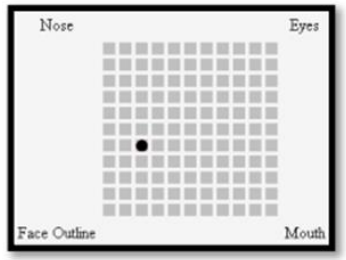





26

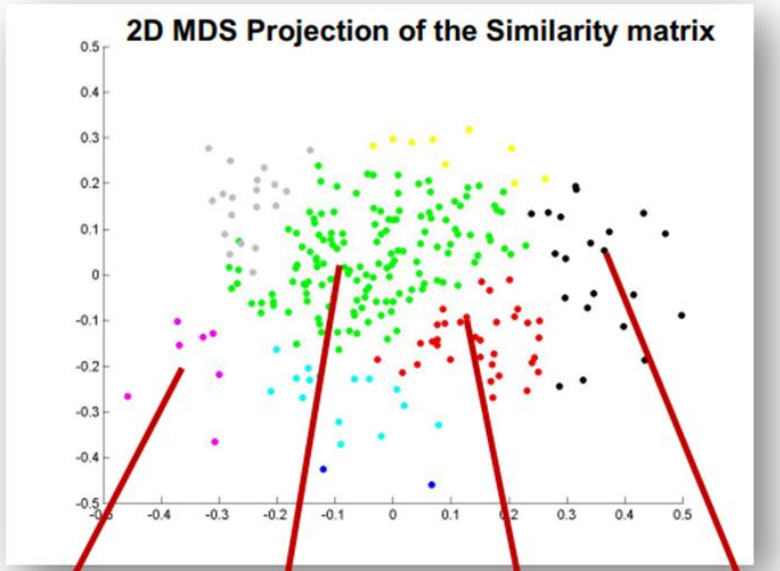
# Real Application of Clustering:

Clustering of Apsara Faces (Cambodge)
https://pdfs.semanticscholar.org/1823/98ca4d85c25c64ba238dad10cafc92203660.pdf

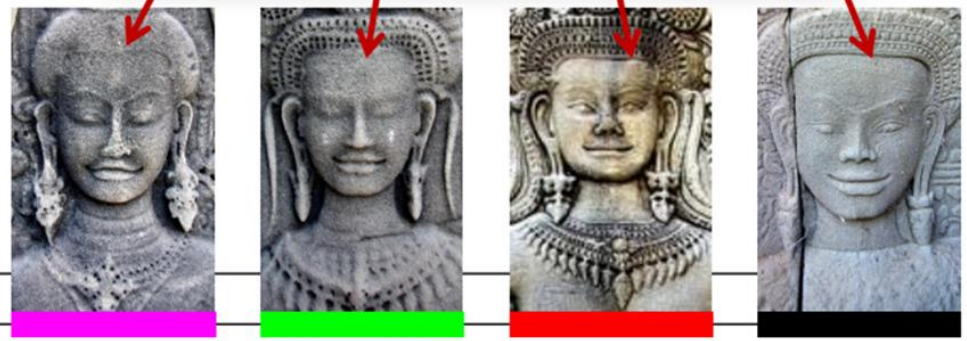



| Weights: | |
|---|---|
| Nose | 0.36 |
| Eyes | 0.00 |
| Chin | 0.54 |
| Mouth | 0.10 |

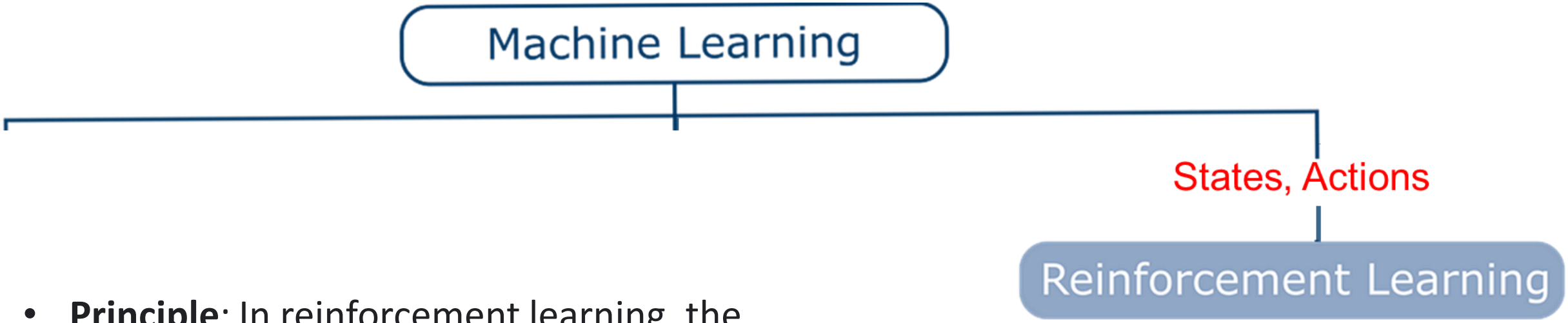
**Clustering with large weights assigned to chin and nose**



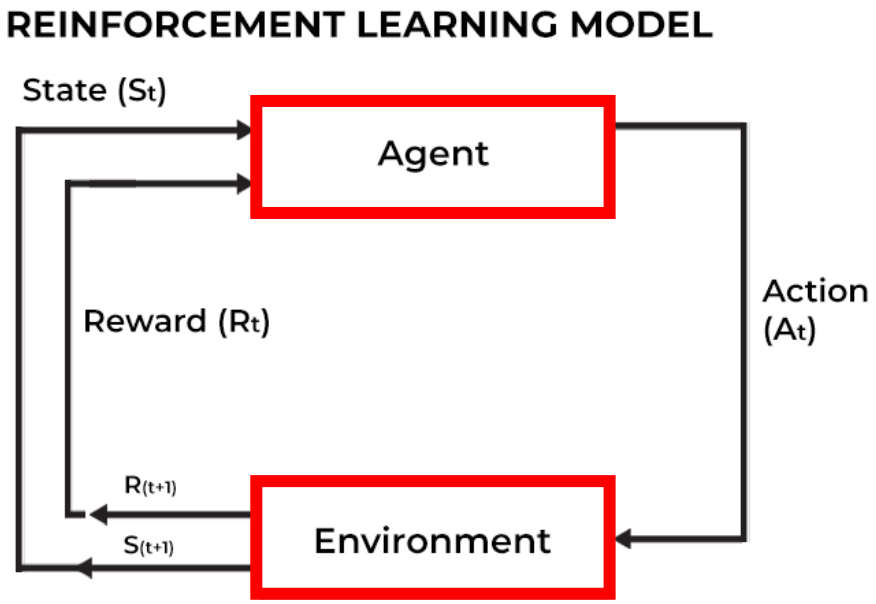

**the clusters differ largely in chin and nose, thereby reflecting the weights chosen for similarity**




An ethnologist needs to validate the groups




- **Principle**: In reinforcement learning, the agent learns by taking actions in the environment and receiving rewards or punishments based on its actions.
- The objective is to find an optimal policy that maximizes the cumulative reward over time, considering the agent's current state and future potential states.
- Reinforcement learning is commonly used in applications such as game playing, robotics, and autonomous systems.



# Machine Learning vs Deep Learning

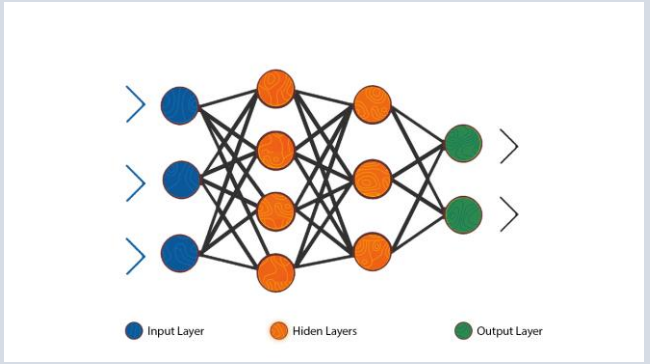
INPUT

**MACHINE LEARNING**

| FEATURE EXTRACTION | CLASSIFICATION |
| --- | --- |
|  |  SVM, logistic, KNN<br>Decision trees,<br>Naïve Bayes |

**DEEP LEARNING**

FEATURE EXTRACTION & CLASSIFICATION




CNN
LSTM
RNN

OUTPUT

**CAT**

**NOT CAT**

# Machine Learning vs Deep Learning

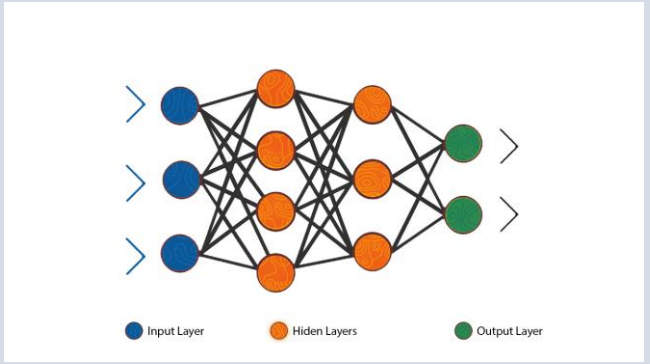
INPUT

**MACHINE LEARNING**

| FEATURE EXTRACTION | CLASSIFICATION |
| --- | --- |
| | SVM, logistic, KNN<br>Decision trees,<br>Naïve Bayes |

**DEEP LEARNING**

FEATURE EXTRACTION & CLASSIFICATION




CNN
LSTM
RNN

OUTPUT

**CAT**

**DOG**

# Structured / Unstructured data

### Unstructured Data

The university has 5600 students. Shaun (ID Number: 160801), 18 years old Communication study. Linh with ID number 160802, majoring in Accounting and is 20 years old. Ahmed from Psychology study program, 19 years old, ID number 160803.

### Semi-Structured Data

```
<University>
  <ID Number="160801">
    <Name="Shaun">
    <Age="18">
    <Program="Communication">
  <ID Number="160802">
    <Name="Linh">
    <Age="20">
    <Program="Accounting">
..... </University>
```

### Structured Data

| ID | Name | Age | Program |
|---|---|---|---|
| 160801 | Shaun | 18 | Communication |
| 160802 | Linh | 20 | Accounting |
| 160803 | Ahmed | 19 | Psychology |

# Example Unstructured Data - Sentiment Analysis

In [8]:
```
# import SentimentIntensityAnalyzer class from vaderSentiment.vaderSentiment
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')
# polarity_scores method of SentimentIntensityAnalyzer
SentimentIntensityAnalyzer().polarity_scores('Today is a good day.')
# output:
# {'neg': 0.0, 'neu': 0.58, 'pos': 0.42, 'compound': 0.4404}
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\nah\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

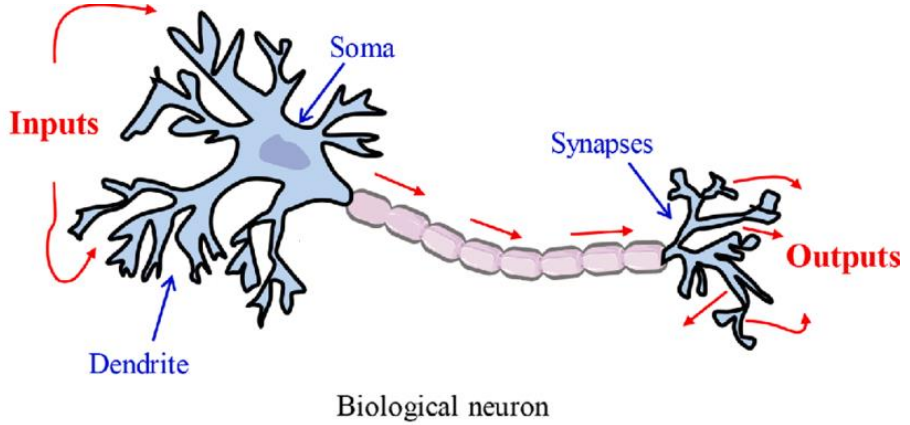
Out[8]:
```
{'neg': 0.0, 'neu': 0.508, 'pos': 0.492, 'compound': 0.4404}
```

# DL Algorithmic explanation

To explain deep learning, consider a foundational model in DL : Multilayer Perceptron (MLP)

Understand the concept of Black box

# Artificial Neuron



Biological neuron

weights represent a synaptic connection

Biais $x_0$

Inputs $x_1$ $x_2$ $x_n$

$w_1$ $w_2$ $w_n$

$\sum$

output

$$y = \begin{cases} 1 & si \sum_{i=0..n} w_i x_i \geq 0 \\ 0 & \text{sinon} \end{cases}$$

If activated

If not activated

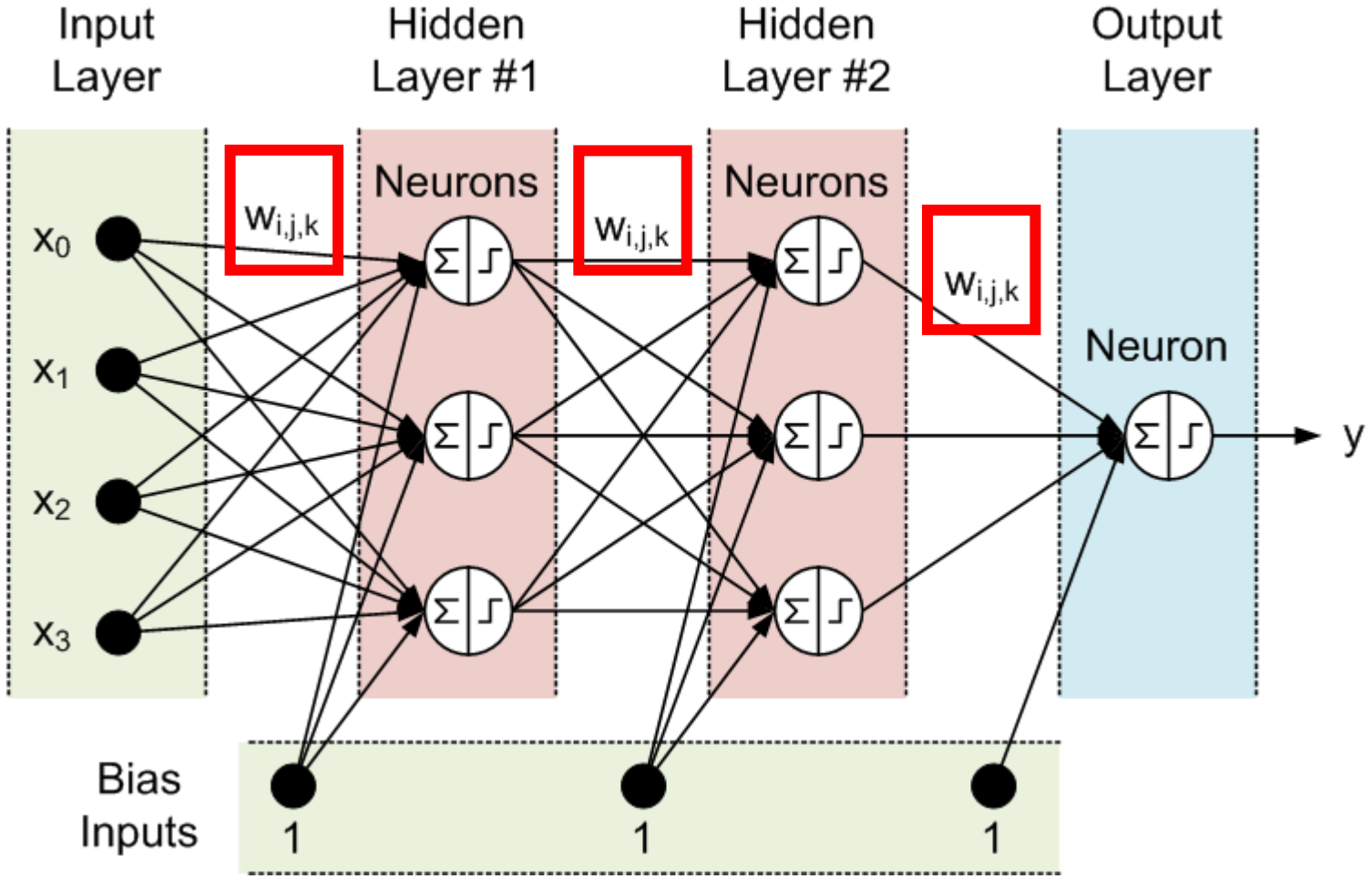
Apply dot product between each input and its weight

$$\sum_{i=0..n} w_i x_i$$

Apply activation function on summation
For example g: Heaviside
*Common activation functions include sigmoid, tanh, ReLU (Rectified Linear Unit), and softmax.*

# MLP architecture

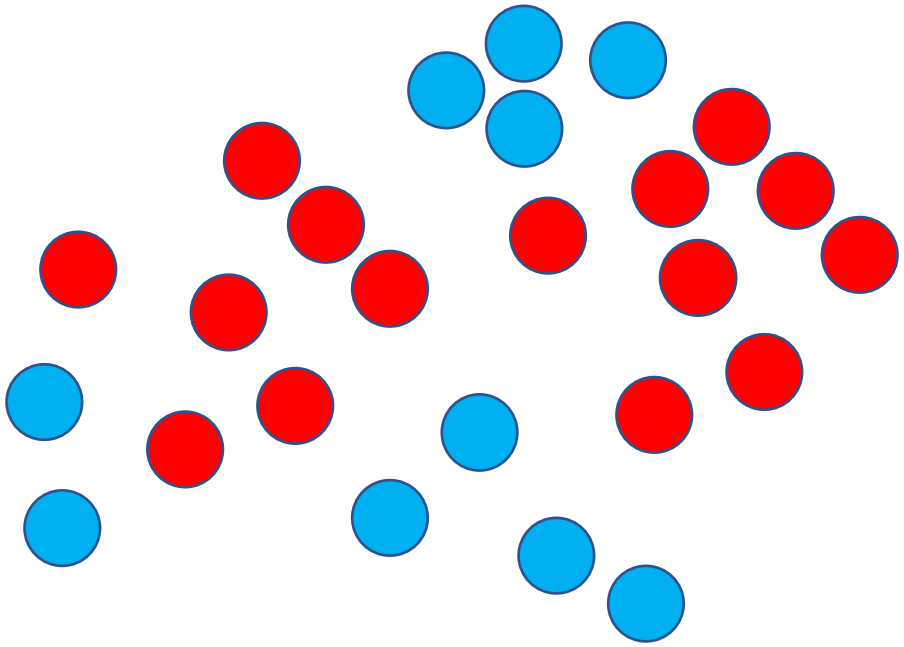
- A key challenge is determining the optimal weights.
- This task is complex due to the network's architecture and the large number of parameters involved



# Principle

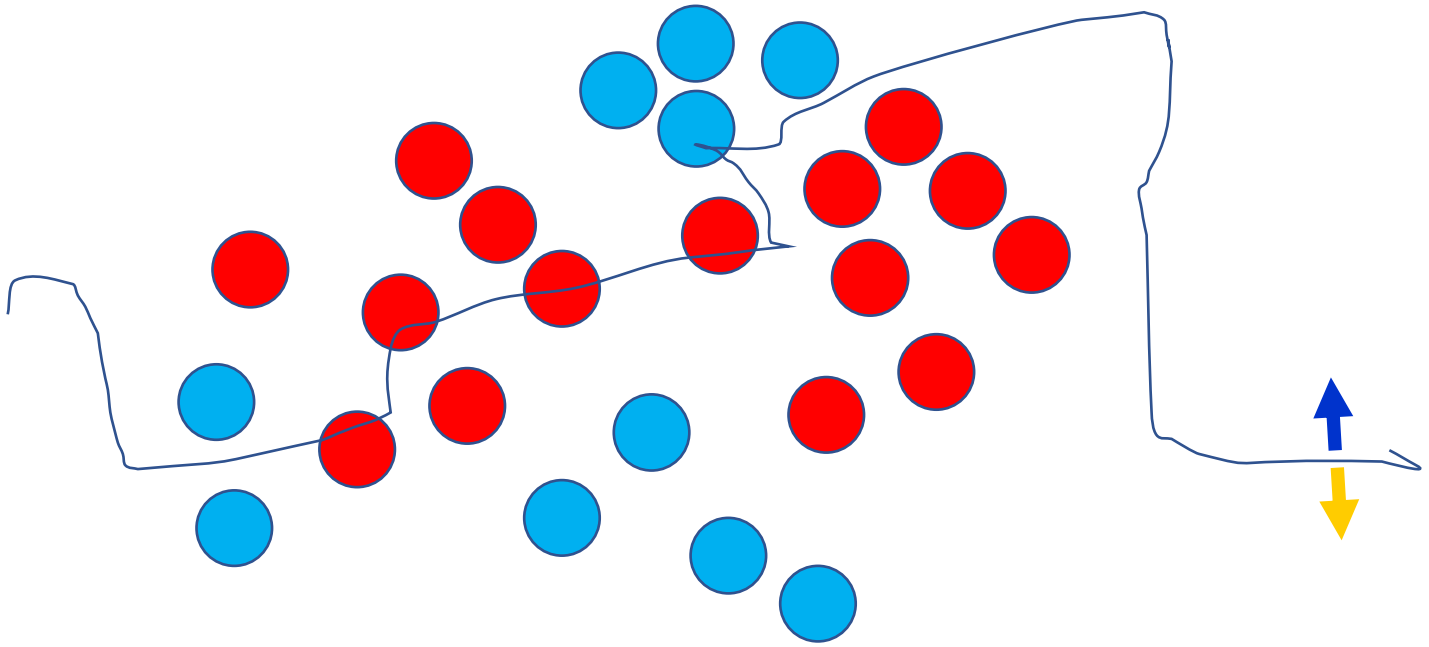
**Objective : separate blue and red points in population**

Population



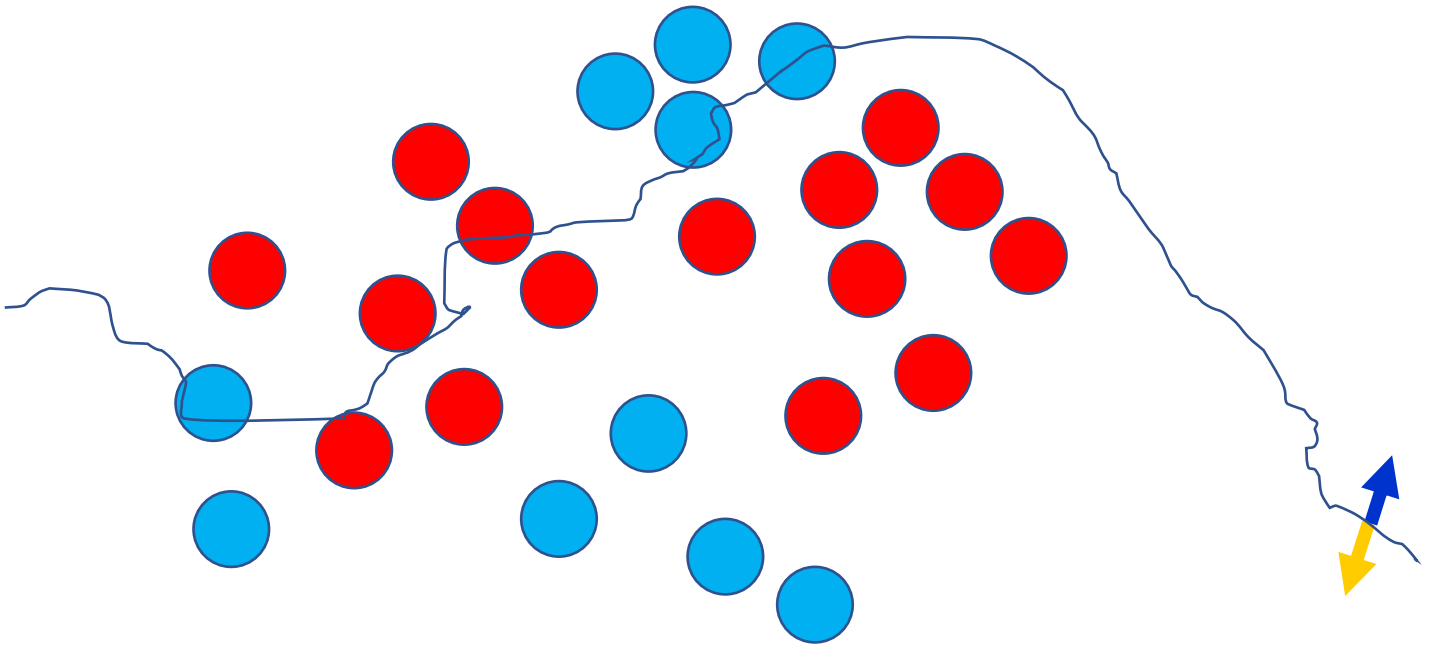
# Principle

**Initial random weights**



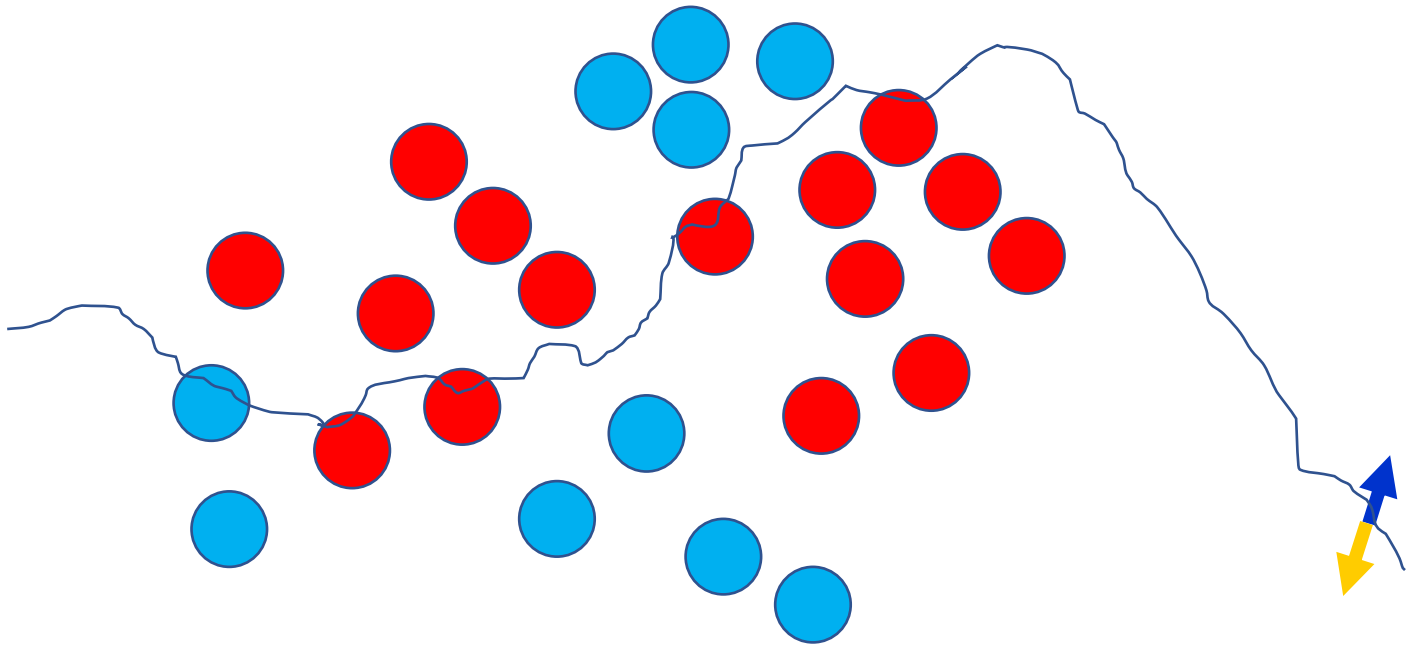
# Principle

**Present a training instance / adjust the weights**



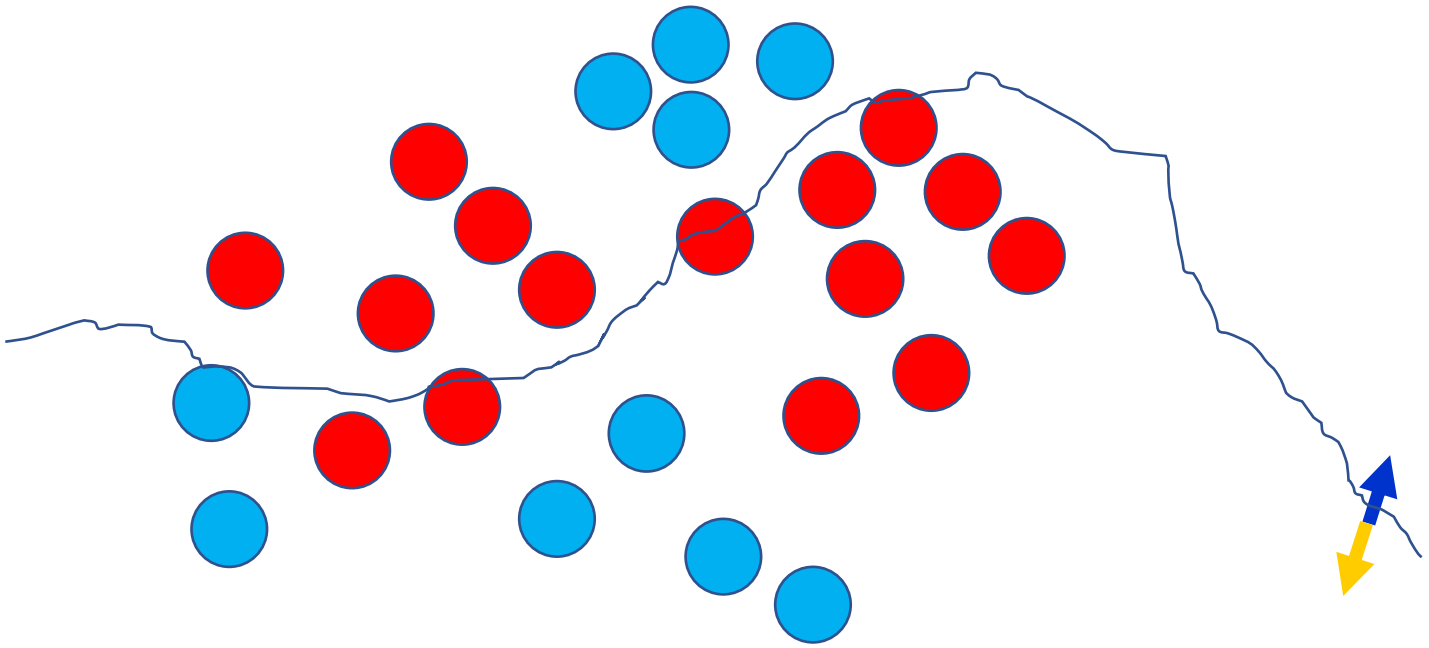
# Principle

**Present a training instance / adjust the weights**



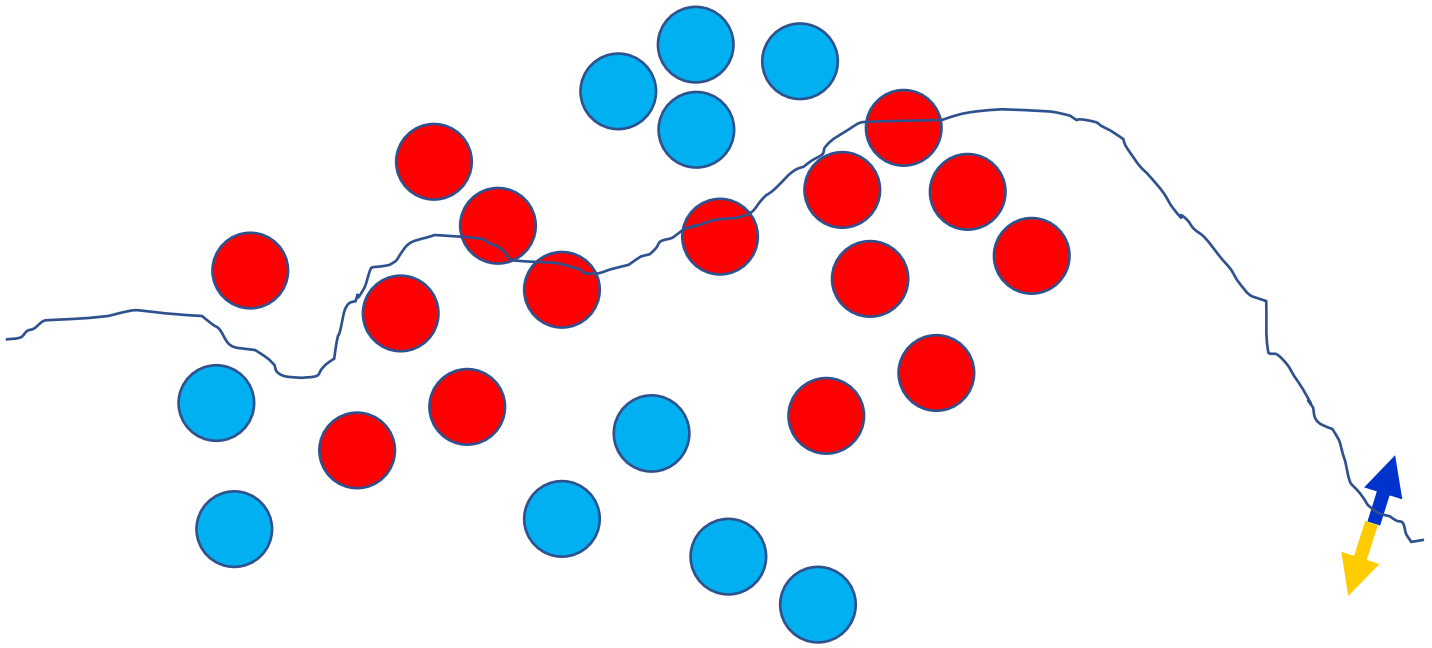
# Principle

**Present a training instance / adjust the weights**



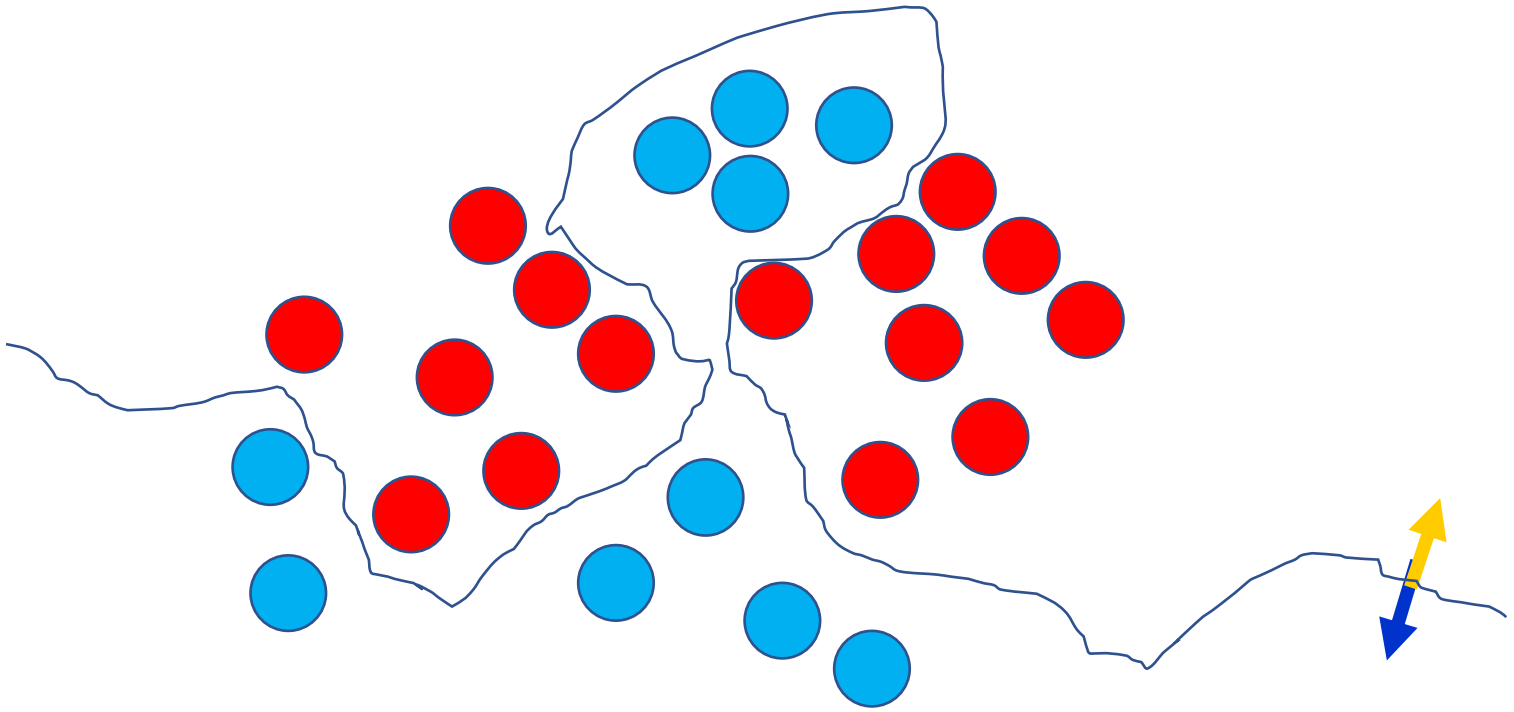
# Principle

**Present a training instance / adjust the weights**



# Principle

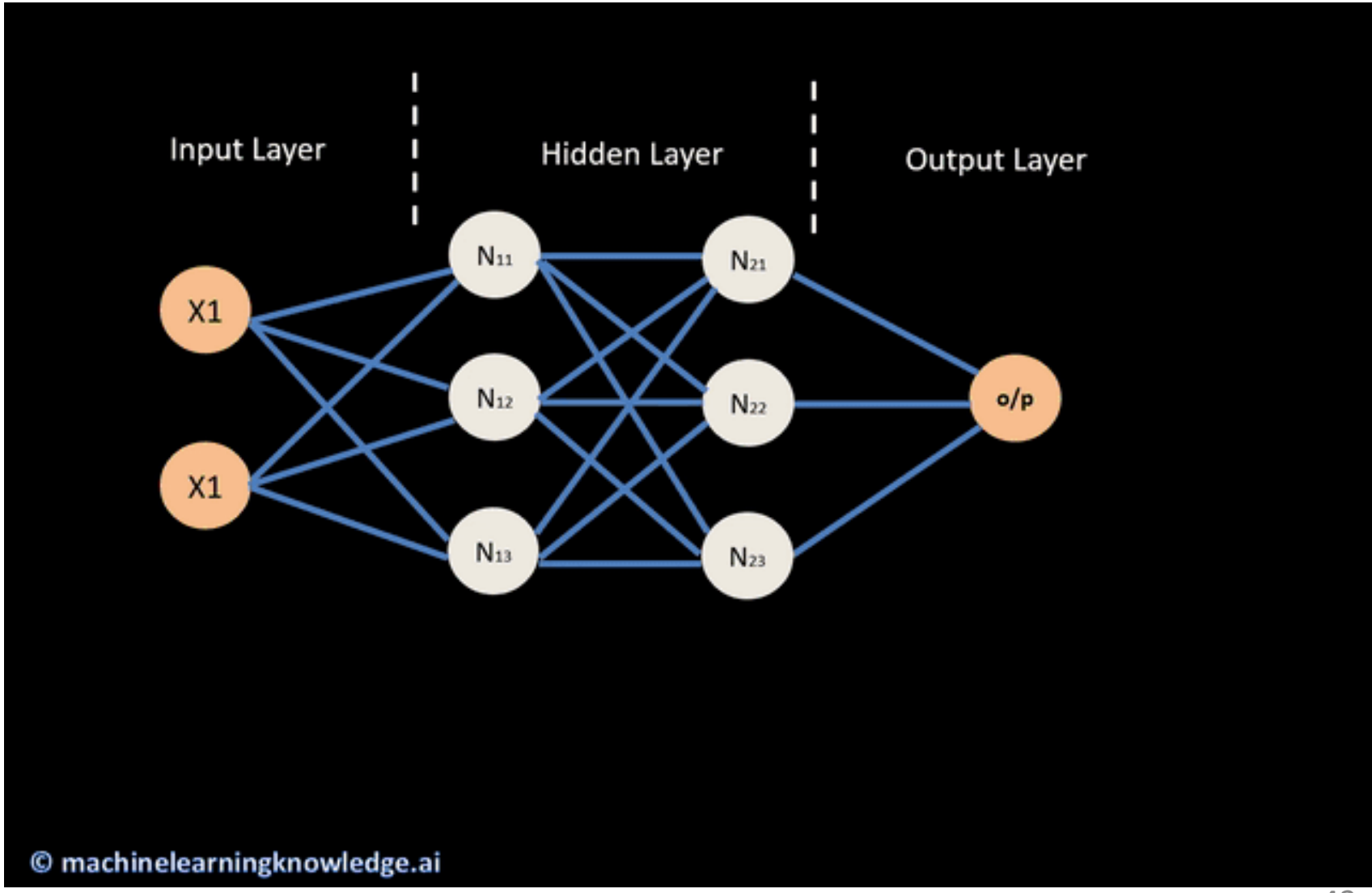
**Finally,**



# MLP (Multi-Layer Perceptron) learning priniple: Backpropagation

- Key idea: Learn by adjusting weights to reduce loss (difference between predicted output and real output)
- 2 passes:
  - Forward pass
  - Backpropagation

→ update weights repeatedly for each example.



# Backpropagation Learning Algorithm

- S = {($x_i$, $y_i$): i = 1, 2, ..., m} is the training set.
- $x$: the vector of inputs
- $y$ : the vector of 'real' outputs
- $h_w(x)$ : the predicted function for $x$ (output of the MLP) using the vector of weights $w$.

→ **Objective**: Minimize the loss $\boldsymbol{l(y, h_w(x))}$ where $l$ is the loss function

- The error depends on the weights $w$, so we will use the gradient (partial derivative) to determine the update direction for the weights. By definition:

$$\frac{\partial}{\partial w} l(y, h_w(x))$$

- To update the parameter $w$, we move in the opposite direction of this gradient ($\alpha$: Learning rate)

$$w \leftarrow w - \alpha \frac{\partial}{\partial w} l(y, h_w(x))$$

# Backpropagation Learning Algorithm

**function** BACK-PROP-LEARNING (*examples, network,* $\alpha$) returns a neural network
**Inputs**: *examples*: a set of examples, each with input vector **x** and output vector **y**
*network*: a multilayer network with L layers, weights $W_{j,i}$, activation function $g$
$\alpha$: learning rate
**Local variables**: $\Delta$: a vector of errors, indexed by network node

**for each** weight $W_{i,j}$ in network **do**
$W_{i,j} \leftarrow$ a small random number

Initialize the weights with small random values.

**Repeat**

epoch

**for each** example **(x,y) in** examples **do**

Forward pass

Backward pass

*/*update every weight in network using deltas*/*
**for each** weight $W_{i,j}$ in network **do**
$W_{i,j} \leftarrow W_{i,j} + \alpha \times a_i \times \Delta_j$

**Until some stopping criterion is satisfied**
**Return Network**

To update the parameters, we move in the opposite direction of the gradient

$w_{ij} \leftarrow w_{ij} - \alpha \frac{\partial E}{\partial w_{ij}}$ The value of the gradient: $-a_i \times \Delta_i$

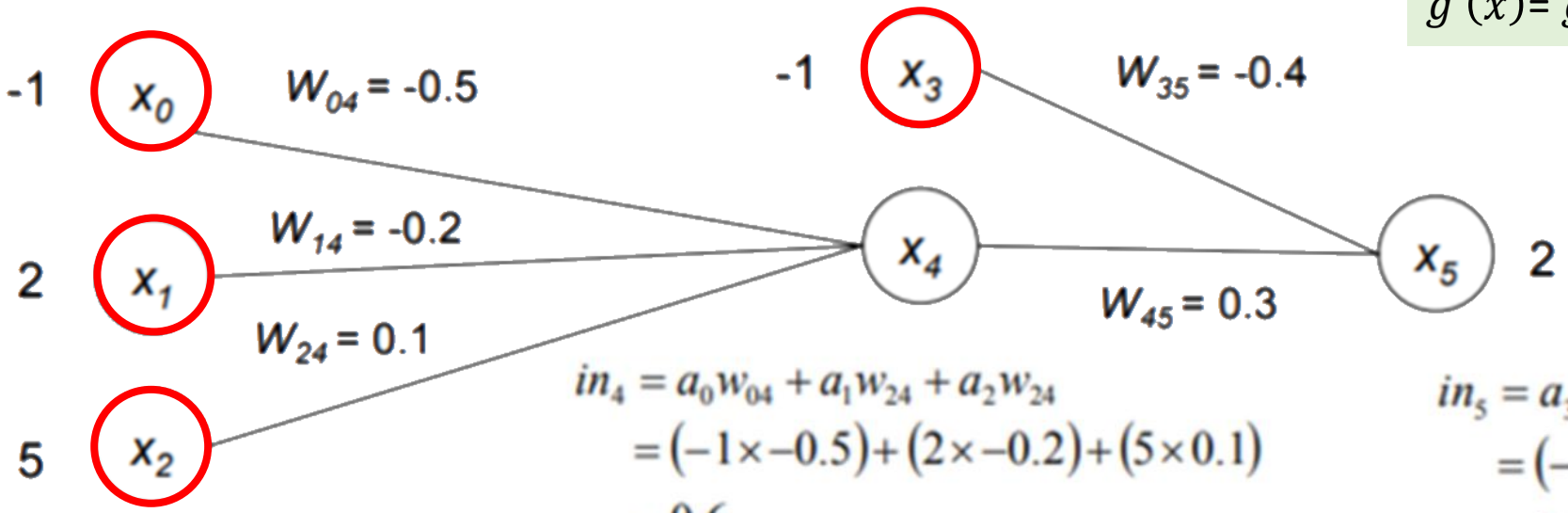
There are several possible termination conditions:
1. After a fixed number of iterations.
2. When the weights stabilize.
3. When the error in the outputs of the training examples drops below a certain threshold.
4. Etc.

# Example (1)

- 4 input nodes: $x_0$ .. $x_3$
- 1 hidden node : $x_4$
- 1 binary output $x_5$ (1 or 0)
- Activation function = Sigmoïde at $x_4$and $x_5$

$$g(x) = \frac{1}{1+e^{x}}$$

$$g'(x) = g(x)\big(1-g(x)\big)$$




$$in_4 = a_0 w_{04} + a_1 w_{24} + a_2 w_{24}$$
$$= (-1 \times -0.5) + (2 \times -0.2) + (5 \times 0.1)$$
$$= 0.6$$

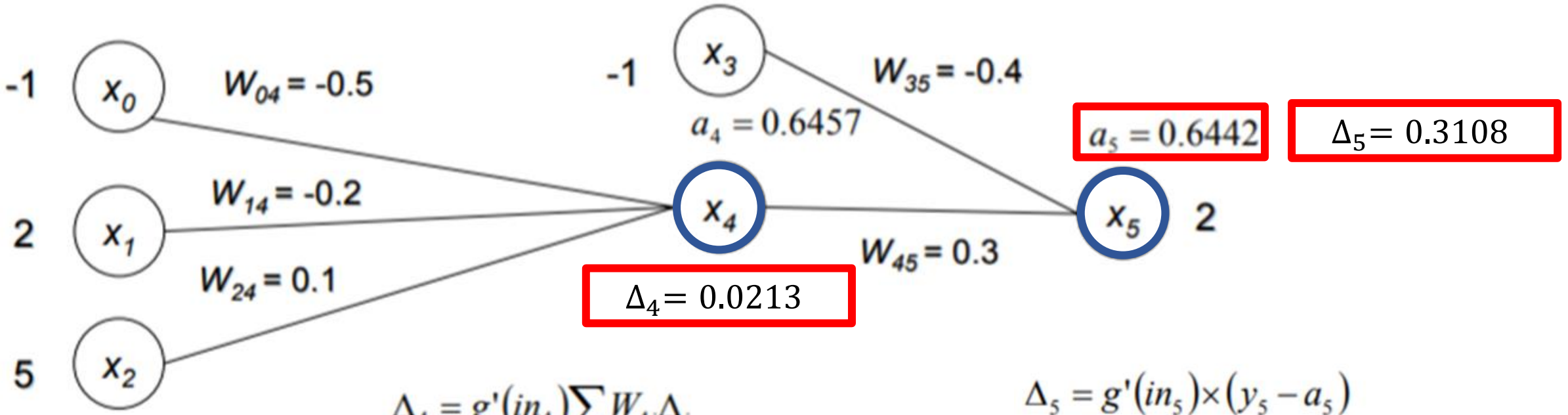
$$a_4 = g(in_4) = \frac{1}{1+e^{-0.6}} = 0.6457$$

$$in_5 = a_3 w_{35} + a_4 w_{45}$$
$$= (-1 \times -0.4) + (0.6457 \times 0.3)$$
$$= 0.5937$$

$$a_5 = g(in_5) = \frac{1}{1+e^{-0.5937}} = 0.6442$$

predicted

# Example (2)

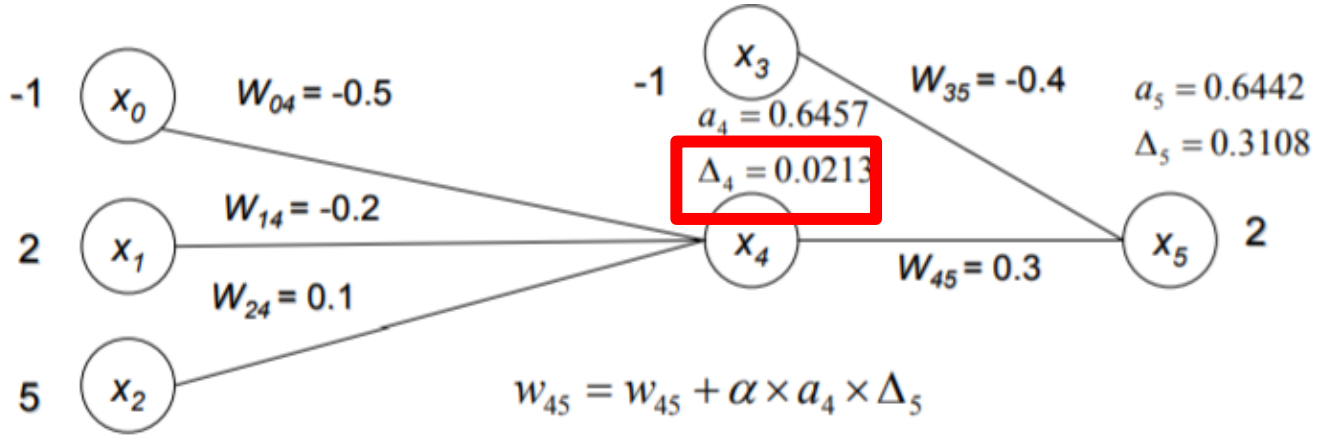



$$\Delta_4 = g'(in_4)\sum_i W_{4i}\Delta_i$$
$$= a_4 \times (1-a_4) \times W_{45}\Delta_5$$
$$= 0.6457 \times (1-0.6457) \times (0.3 \times 0.3108)$$
$$= 0.0213$$

$$\Delta_5 = g'(in_5) \times (y_5 - a_5)$$
$$= a_5 \times (1-a_5) \times (y_5 - a_5)$$
$$= 0.6442 \times (1-0.6442) \times (2-0.6442)$$
$$= 0.3108$$

Learning rate $\alpha$=0.05

# Example (3)




$$w_{45} = w_{45} + \alpha \times a_4 \times \Delta_5$$
$$= 0.3 + 0.05 \times 0.6457 \times 0.3108 = 0.3100$$

$$w_{35} = w_{35} + \alpha \times a_3 \times \Delta_5$$
$$= -0.4 + 0.05 \times -1 \times 0.3108 = -0.4155$$

$$w_{04} = w_{04} + \alpha \times a_0 \times \Delta_4$$
$$= -0.5 + 0.05 \times -1 \times 0.0213 = -0.5011$$

$$w_{14} = w_{14} + \alpha \times a_1 \times \Delta_4$$
$$= -0.2 + 0.05 \times 2 \times 0.0213 = -0.1979$$

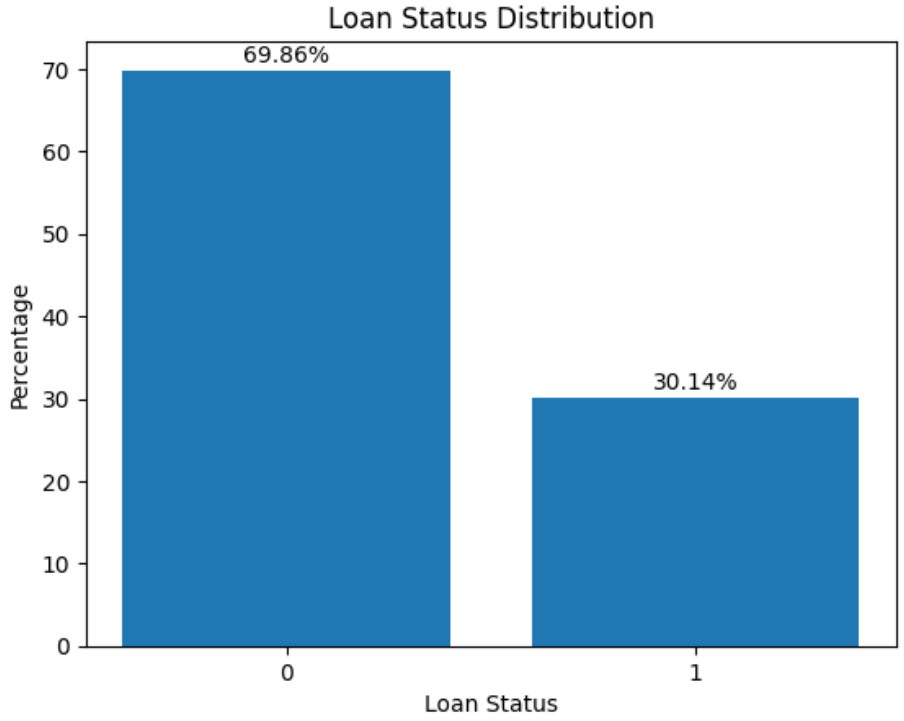
$$w_{24} = w_{24} + \alpha \times a_2 \times \Delta_4$$
$$= 0.1 + 0.05 \times 5 \times 0.0213 = 0.1053$$

# Neural networks : Black Boxes ?

1. **Number and dimensionality of weights**: Neural networks can have millions or even billions of weights, especially in deep architectures.
2. **Complex interactions**: Weights determine the strength of connections between neurons, and each weight influences the activations and outputs of neurons in the network. Modifying a specific weight can have impact throughout the entire network, making it difficult to understand the **individual contributions of weights to a given prediction**.
3. **Nonlinearity**: Neural networks often use nonlinear activation functions. This means that the influence of a particular weight can be nonlinear and dependent on the values of other weights. Therefore, assigning a simple and direct interpretation to each individual weight becomes challenging.

# Example : Loan application

loan new.ipynb




- An application using multiple customer-related data to enable the system to recommend whether to grant a loan or not.
- https://raw.githubusercontent.com/dphi-official/Datasets/master/Loan_Data/loan_train.csv
- Dataset: 491 cases / 11 attributes
- Class: Loan-status : 0 (not attributed) 1 (attributed)

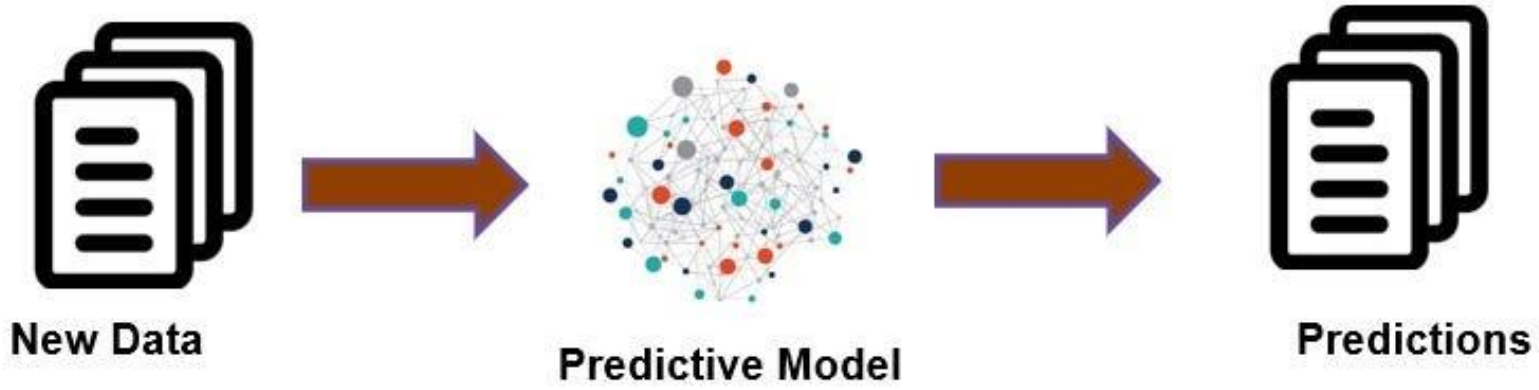
- **One-hot encoding** : convert categorical variables into distinct binary variables, indicating the presence or absence of each category – thus we pass from 11 to 21 attributes

| Gender | Married | Dependents | Education | Self_Employed | ApplicantIncome | CoapplicantIncome | LoanAmount | Loan_Amount_Term | Credit_History | Property_Area | Loan_Status |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Female | No | 0 | Graduate | No | 4547 | 0.0 | 115.0 | 360.0 | 1.0 | Semiurban | 1 |
| Male | Yes | 3+ | Not Graduate | Yes | 5703 | 0.0 | 130.0 | 360.0 | 1.0 | Rural | 1 |
| Female | Yes | 0 | Graduate | No | 4333 | 2451.0 | 110.0 | 360.0 | 1.0 | Urban | 0 |
| Male | Yes | 0 | Not Graduate | Yes | 4695 | 0.0 | 96.0 | NaN | 1.0 | Urban | 1 |
| Male | Yes | 2 | Graduate | No | 6700 | 1750.0 | 230.0 | 300.0 | 1.0 | Semiurban | 1 |

# Classification principle


**Training data:** loan_train.csv

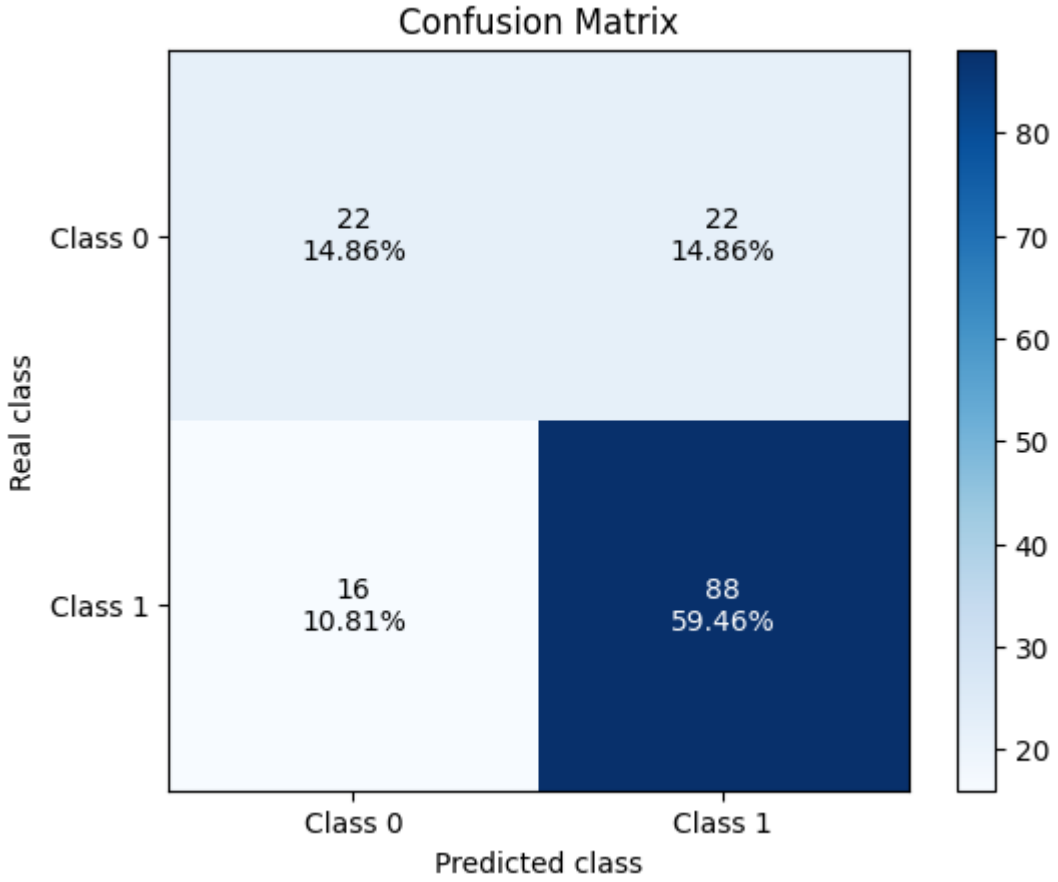
**Decision tree**
**Deep neural network**



# Example : Loan application Decision tree (1)

```
clf = DecisionTreeClassifier(random_state=1)
```

```
Test accuracy :
0.7432432432432432
```

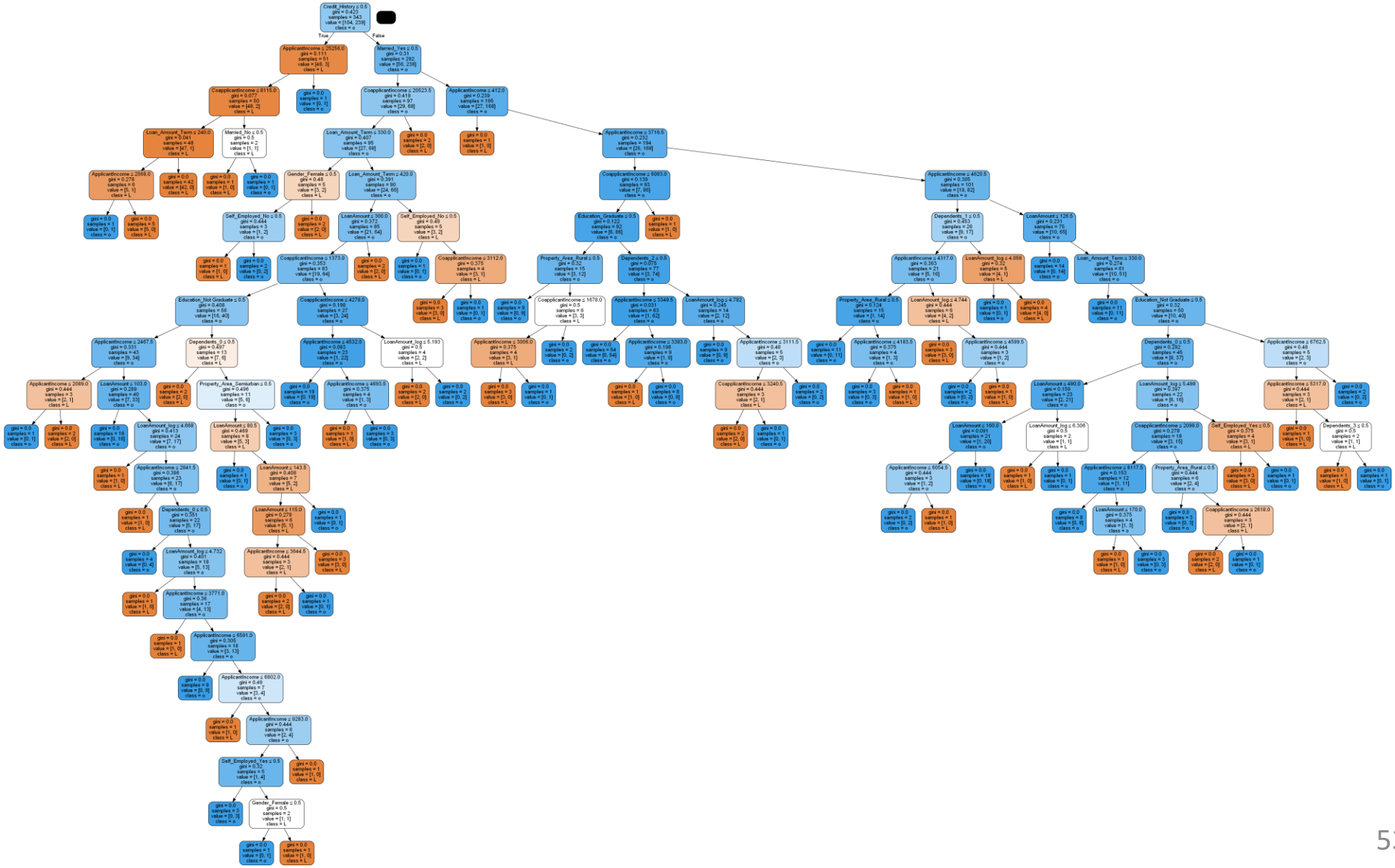



**Visualize the Decision tree**

```
from io import StringIO
import pydotplus
from IPython.display import Image
plot_decision_tree(clf, x_train.columns,loan_data.columns[11])
```

*Accuracy = (TP + TN) / (TP + TN + FP + FN)
=22+88/22+22+16+88

# Example : Loan application
# Decision tree (2)



# Example : Loan application Decision Tree (3)

```
from sklearn.tree import export_text

# Extract the rules from the decision tree
rules = export_text(clf, feature_names=list(X.columns))
print(rules)

num_rules = len(rule_list)
print("The number of rules in the decision tree is:", num_rules)
```

```
|--- Credit_History <= 0.50
|   |--- ApplicantIncome <= 25256.00
|   |   |--- CoapplicantIncome <= 8115.00
|   |   |   |--- Loan_Amount_Term <= 240.00
|   |   |   |   |--- ApplicantIncome <= 2568.00
|   |   |   |   |   |--- class: 1
|   |   |   |   |--- ApplicantIncome >  2568.00
|   |   |   |   |   |--- class: 0
|   |   |   |--- Loan_Amount_Term >  240.00
|   |   |   |   |--- class: 0
|   |   |--- CoapplicantIncome >  8115.00
|   |   |   |--- Married_No <= 0.50
|   |   |   |   |--- class: 0
|   |   |   |--- Married_No >  0.50
|   |   |   |   |--- class: 1
|   |--- ApplicantIncome >  25256.00
|   |   |--- class: 1
```

58 rules

If the credit history is less than or equal to 0.50:
|--- If the applicant's income is less than or equal to 25256.00:
| |--- If the co-applicant's income is less than or equal to 8115.00:
| | |--- If the loan amount term is less than or equal to 240.00:
| | | |--- If the applicant's income is less than or equal to 2568.00: The predicted class is 1 (loan approved).
| | | |--- If the applicant's income is greater than 2568.00: The predicted class is 0 (loan not approved).
| | |--- If the loan amount term is greater than 240.00: The predicted class is 0 (loan not approved).
| |--- If the co-applicant's income is greater than 8115.00:
| | |--- If the applicant is not married: The predicted class is 0 (loan not approved).
| | |--- If the applicant is married: The predicted class is 1 (loan approved).
| |--- If the applicant's income is greater than 25256.00: The predicted class is 1 (loan approved).

# Example : Loan application – Deep learning (1)

```
# Define the model class
class Model(nn.Module):
    def __init__(self, input_size):
        super(Model, self).__init__()
        self.fc1 = nn.Linear(input_size, 32)
        self.fc2 = nn.Linear(32, 16)
        self.fc3 = nn.Linear(16, 1)
        self.sigmoid = nn.Sigmoid()

    def forward(self, x):
        x = self.fc1(x)
        x = self.fc2(x)
        x = self.fc3(x)
        x = self.sigmoid(x)
        return x
```

Deep neural network model:

- **Number of layers**: 3 linear layers
- **Number of inputs of the first layer**: determined by the shape of the input data
- **Activation function**: sigmoid used after the last linear layer for binary classification.

# Example : Loan application – Deep learning (2)

```
Test Accuracy: 0.75
```

Initial weights :

fc1.weight tensor([[-0.1587, -0.1832, -0.0520, -0.1832, -0.1063, 0.0826, 0.1902, -0.0642,
0.1841, -0.2046, 0.0528, 0.2051, -0.0112, 0.0243, 0.0377, 0.1300, -0.0824, 0.0804, -0.0789, -0.1017, 0.1405],
[-0.0527, 0.1408, 0.1966, -0.1951, 0.1630, -0.0704, 0.0336, -0.1555, 0.0692, 0.1308, -0.1989, -0.0622, -0.0036, 0.0868, -0.0575, -0.0133, -0.0103, 0.1980, 0.1471, 0.0165, 0.1218],
[-0.0321, -0.0786, 0.1684, -0.1928, -0.0683, 0.0792, 0.0538, -0.1863, -0.1580, -0.1293, 0.1322, -0.0504, -0.0940, -0.2013, 0.0966, -0.1581, -0.0482, -0.1099, -0.1275, 0.1952, 0.0878],
[-0.0195, 0.1745, 0.1080, 0.1334, -0.2173, 0.1719, 0.1317, -0.1210, 0.0569, -0.0286, 0.1919, -0.0343, 0.1744, -0.0856, -0.0887, 0.1576, -0.0972, -0.0668, -0.0949, -0.0555, 0.0567],
[-0.0472, 0.1827, 0.0174, 0.1832, -0.1797, -0.1702, -0.0230, -0.0072, 0.1069, -0.0774, -0.0116, 0.1401, 0.0753, 0.1449, 0.0869, 0.0717, 0.1501, 0.1671, 0.0022, 0.0848, -0.0858],
....

fc1.bias tensor([-0.0586, -0.0251, -0.0282, 0.0275, 0.1395, -0.1052, -0.1971, 0.1654, 0.0404, 0.0314, -0.0927, -0.0440, -0.1732, 0.2061, 0.1703, -0.1492,
0.0007, -0.1791, -0.1684, 0.0644, 0.1032, 0.1480, 0.0020, 0.1137, 0.0332, -0.1227, -0.1031, 0.1970, -0.0183, 0.0654, 0.0724, 0.1789])

fc2.weight tensor([[ 8.3723e-02, 1.3322e-01, -5.9647e-02, 8.9700e-02, -8.7064e-02, 1.5882e-01, 1.4312e-02, 1.6855e-01, 1.0083e-01, 9.5341e-02,
-7.5547e-02, 1.5756e-01, 6.7507e-02, -1.6330e-01, -7.8047e-02, 1.0389e-01, -2.3302e-03, -4.9479e-02, 1.0473e-01, 1.6773e-01,
8.3254e-02, 1.7482e-01, 1.6021e-01, 1.5268e-01, -6.9719e-02, -1.2814e-01, 1.4385e-01, -9.7902e-02, 1.3354e-02, -1.5641e-01,
....

fc2.bias tensor([-0.0206, -0.1349, 0.0591, 0.1763, -0.1439, 0.0787, -0.1239, -0.0020, -0.0399, -0.0432, 0.1190, -0.0215, -0.0959, -0.0623, 0.1025, -0.0940])

fc3.weight tensor([[-0.1886, 0.0282, 0.1158, -0.1472, 0.0853, 0.1995, -0.2395, 0.0728, 0.1509, 0.0318, 0.0678, -0.1237, 0.0145, -0.1069, -0.1993, -0.0551]])

fc3.bias tensor([0.0969])

441 values ?

# New instance: DT vs DNN (1)

```
new_instance = [[4547, 0.0, 115.0, 360.0, 1.0, 4.744932, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1]]
```

```
#DECISION TREE
# Predict the loan status for the new instance
prediction = clf.predict(new_instance)
print("Predicted Loan Status:", prediction)
```

```
Predicted Loan Status: [1]
```

```
#DNN
with torch.no_grad():
    outputs = model(new_instance)
    predicted = (outputs >= 0.5).float()

print("Predicted Loan Status:", predicted.item())
```

```
Predicted Loan Status: 1.0
```

# New instance: DT vs DNN (2)

Explanation of the result given by the DT
No Explanation For DNN

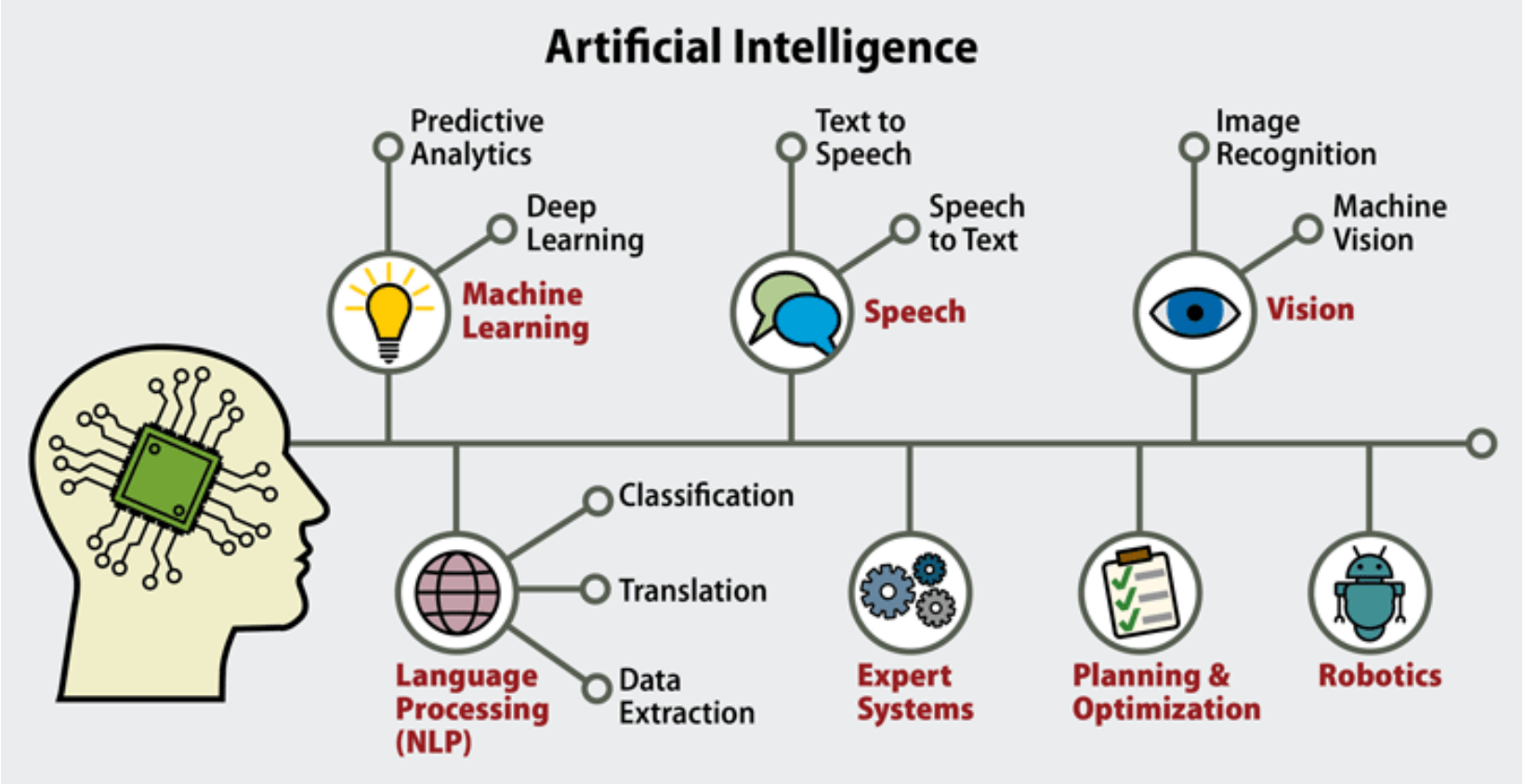
```
# New instance
new_instance = [
    [4547, 0.0, 115.0, 360.0, 1.0, 4.744932, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1]
]
```

The path in the decision tree using the features of **new_instance**:

**1.If "Credit_History" is less than or equal to 0.50**
**2. and "ApplicantIncome" is less than or equal to 25,256.00**
**3. and "CoapplicantIncome" is less than or equal to 8,115.00**
**4. and "Loan_Amount_Term" is less than or equal to 240.00**
**5. and "ApplicantIncome" is less than or equal to 2,568.00,**
**then the predicted class is 1.**

# Main AI domains

AI covers various domains:




- Speech recognition: error rate of only 5.9%
- Image analysis: error rate of only 3.5%
- AI systems now recognize words in a human conversation with human-level accuracy, enabling seamless communication.
- Real-time translation during conversations has become a reality, allowing for effective multilingual communication.

THE FINANCE DEPARTMENT OF A LEADING MULTINATIONAL COMPANY, OCT 2019

POLLUTED BEACH IN INDIA, DEC 2019

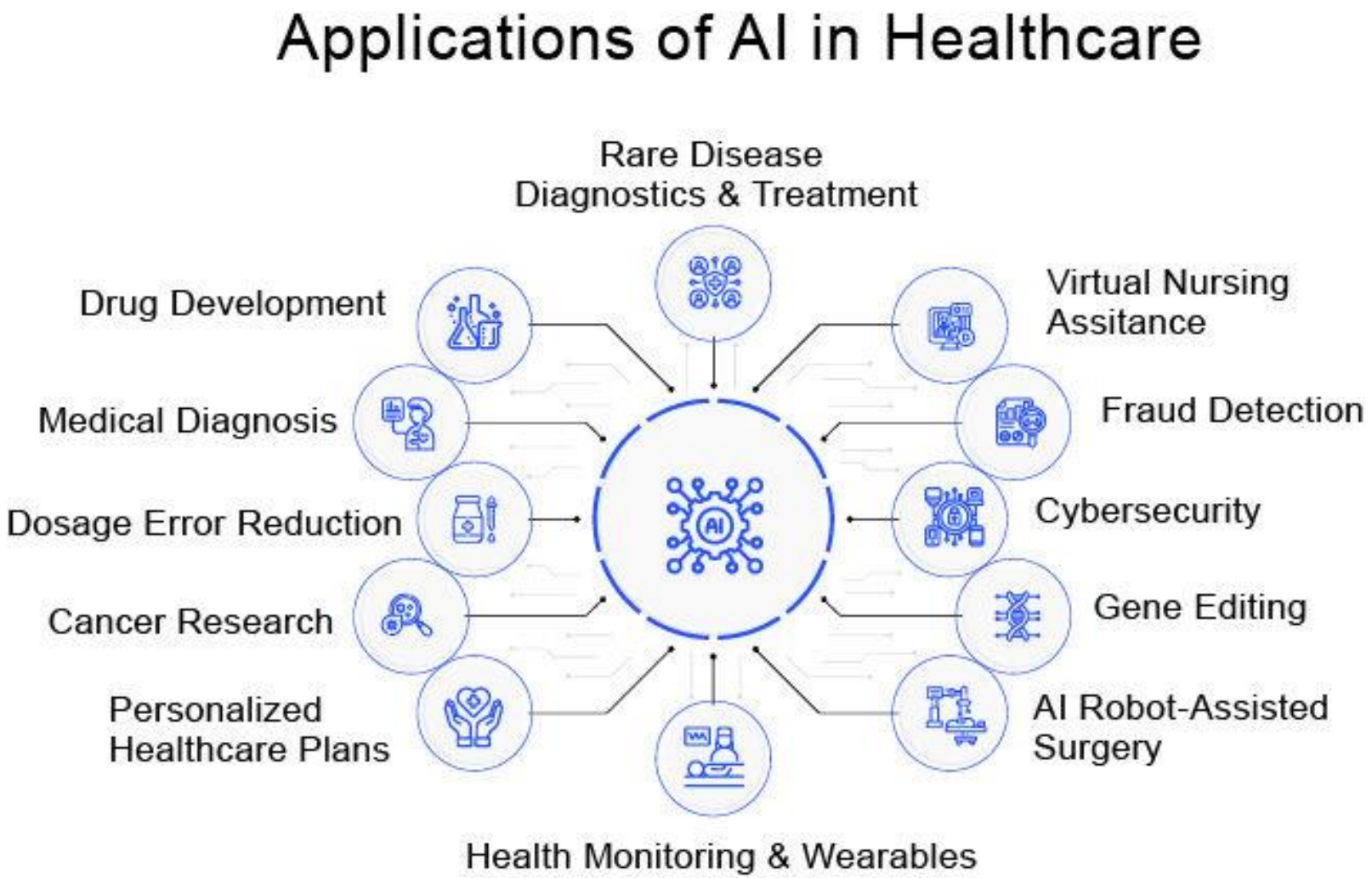
LONG QUEUE OF CLIENTS AT A TRAIN STATION IN CHINA, DEC 2019

SICK PEOPLE LYING ON THE GROUND IN AN OVERCROWDED HOSPITAL IN MADRID, APRIL 2020

*IA aims at changing or at least improving such shocking situations*

# AI: A Promise for a Better World

SICK PEOPLE LYING ON THE GROUND IN AN OVERCROWDED HOSPITAL IN MADRID, APRIL 2020



POLLUTED BEACH IN INDIA, DEC 2019

# AI FOR THE ENVIRONMENT

| *Climate change* | *Biodiversity and conservation* | *Healthy Oceans* | *Water security* | *Clean air* | *Weather and disaster resilience* |
|---|---|---|---|---|---|
| • Clean power | • Habitat protection and restoration | • Fishing sustainably | • Water supply | • Filtering and capture | • Prediction and forecasting |
| • Smart transport options | • Sustainable trade | • Preventing pollution | • Catchment control | • Monitoring and prevention | • Early warning systems; |
| • Sustainable production and consumption | • Pollution control | • Protecting habitats | • Water efficiency | • Early warning | • Resilient infrastructure |
| • Sustainable land-use | • Invasive species and disease control | • Protecting species | • Adequate sanitation | • Clean fuels | • Financial instruments |
| • Smart cities and homes | • Realising natural capital | • Impacts from climate change (including acidification) | • Drought planning | • Real-time, integrated, adaptive urban management | • Resilience planning |

https://ia4marketing.fr/environnement/thierrypires/2021/03/lintelligence-artificielle-nouvelle-main-verte-de-notre-planete/

LONG QUEUE OF CLIENTS AT A TRAIN STATION IN CHINA, DEC 2019

# AI IN TRANSPORTATION

## TRAFFIC MANAGEMENT

- COOPERATIVE NAVIGATION
- SPEED MANAGEMENT
- TRAFFIC INFORMATION
- ROUTING

## ROAD SAFETY

- COLLISION WARNING
- SIGNAL VIOLATION WARNING
- EMERGENCY VEHICULE
- DIAGNISTICS

## AUTONOMOUS DRIVING

- ELECTRONIC STABILITY CONTROL
- AUTOMATIC BRAKING
- ADAPTIVE CRUISE CONTROL

https://espaces-numeriques.org/voiture-autonome/
https://www.actuia.com/actualite/disponibilite-des-wagons-en-usine-loutil-tvms-deverysens-mise-a-jour-avec-une-nouvelle-fonctionnalite/
https://www.lesnumeriques.com/voitures-co/l-intelligence-artificielle-au-service-des-feux-de-circulation-n190975.html


63

# Domains of application for AI

SPACE EXPLORATION

AGRICULTURE

AUTOMOBILE

DISTRIBUTION


EDUCATION

ENERGY

FINANCE

INDUSTRY

JUSTICE

MEDIA


ADVERTISING

HEALTHCARE

TECHNOLOGY

TRANSPORT

GAMING

E-COMMERCE

SOCIAL MEDIA

ENTERTAINMENT

# Stages of AI

**NARROW AI**

**WEAK AI**

Specialized in specific and limited tasks

2. Ethics and Responsibility in the Use of Al

- ▷ 2.1 Ethical Issues Related to Al
- ▷ 2.2 Sources of Bias in ML lifecycle
- ▷ 2.3 Regulatory and Normative Framework for Al
- ▷ 2.4 Best practices for Bias avoidance/mitigation

# Understanding and Addressing Bias: Why Does it Occur and How to Resolve it?

# AI can be sexist and racist

- March 2016: Microsoft's artificial intelligence took its first steps on Twitter, in the guise of a 16-year-old American teenager (Tay).
- But as Tay came into contact with other Internet users, she quickly became racist, misogynistic and conspiracist..




# AI can be sexist and racist


- In 2016, it was found that the software developed by the company **Northpointe**, used in several places in the United States to predict the likelihood of recidivism among individuals convicted of crimes (by rating them on a scale of 1 to 10), exhibited racist biases
- Overall, the tool correctly predicted recidivism **61%** of the time, but the predictions failed differently for black defendants.

| | WHITE | AFRICAN AMERICAN |
|---|---|---|
| Labeled Higher Risk, But Didn't Re-Offend | 23.5% | 44.9% |
| Labeled Lower Risk, Yet Did Re-Offend | 47.7% | 28.0% |

https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing
Source : Analyse de ProPublica des données du comté de Broward, Floride.)

# AI can be sexist and racist

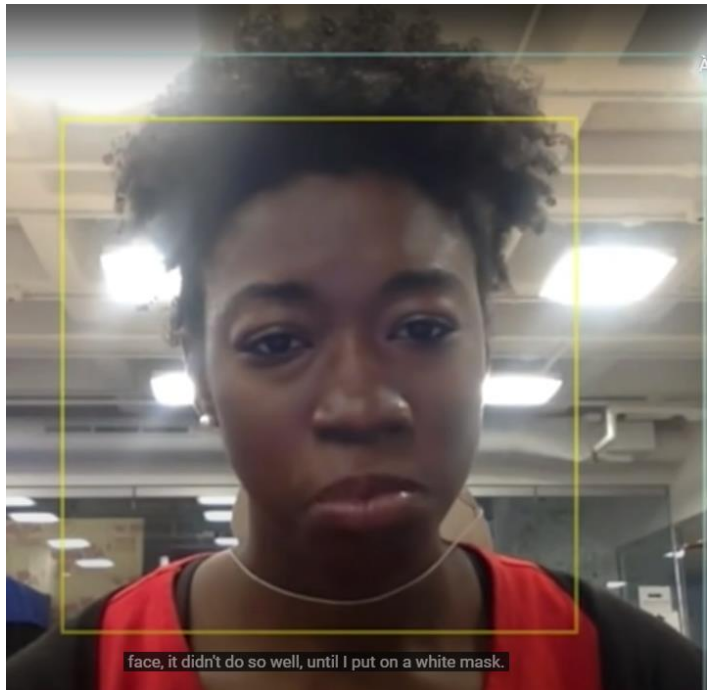
- 2018: In an experiment conducted by the American Civil Liberties Union (ACLU), Amazon's facial recognition system, **Rekognition**, mistakenly identified 28 members of the U.S. Congress as criminals in portraits.
- Nearly 40% of the misidentified members were people of color, despite representing only 20% of the total Congress membership.

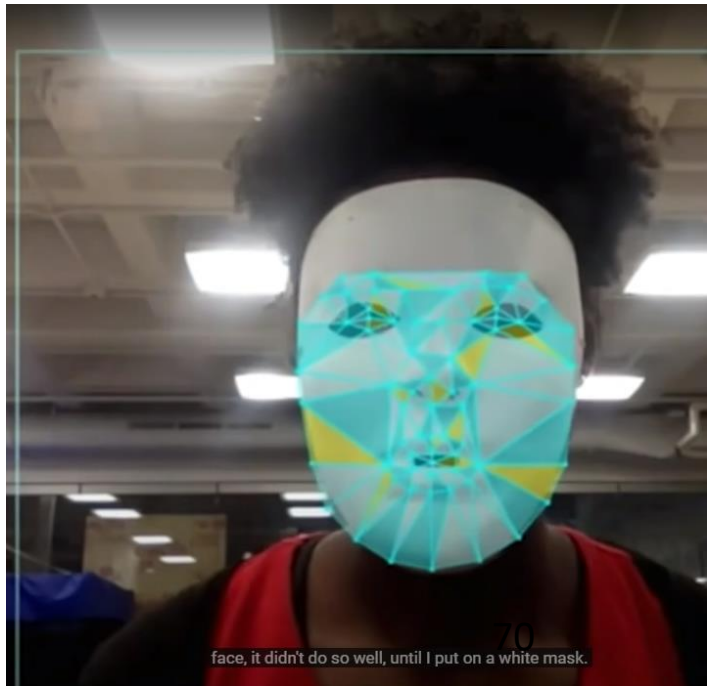
# AI can be sexist and racist

## Face recognition (1)

http://gendershades.org/

- **2018**: As a Phd.  student, Joy Buolamwini discovered that an AI system was better at detecting her when she wore a white mask, which inspired her research project called Gender Shades.
- This project revealed the inherent bias present in commercial AI systems when it comes to gender classification.

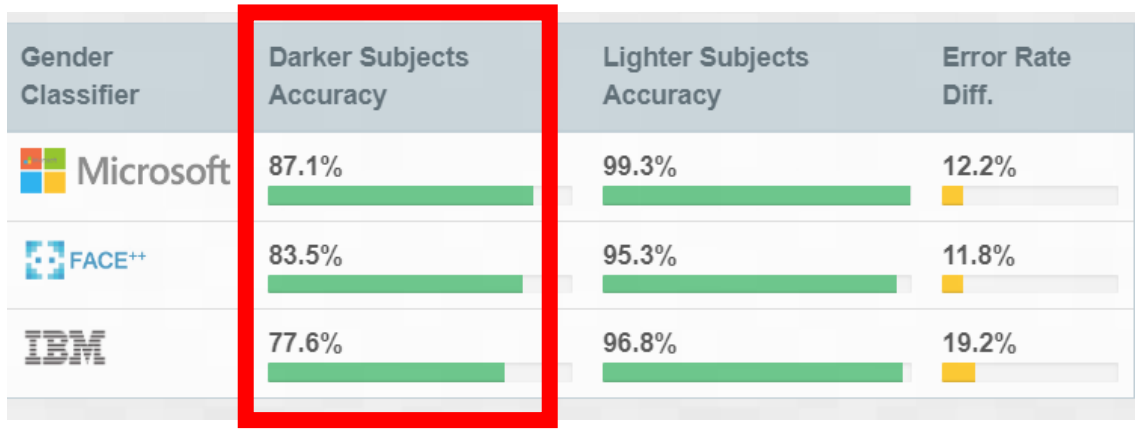


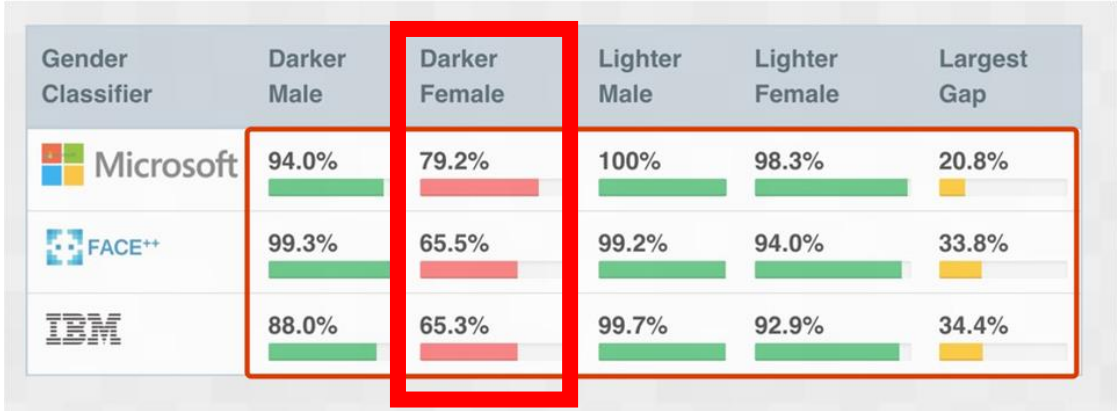
# AI can be sexist and racist

## Face recognition (2)

http://gendershades.org/



| Gender Classifier | Darker Subjects Accuracy | Lighter Subjects Accuracy | Error Rate Diff. |
|---|---|---|---|
| Microsoft | 87.1% | 99.3% | 12.2% |
| FACE++ | 83.5% | 95.3% | 11.8% |
| IBM | 77.6% | 96.8% | 19.2% |



| Gender Classifier | Darker Male | Darker Female | Lighter Male | Lighter Female | Largest Gap |
|---|---|---|---|---|---|
| Microsoft | 94.0% | 79.2% | 100% | 98.3% | 20.8% |
| FACE++ | 99.3% | 65.5% | 99.2% | 94.0% | 33.8% |
| IBM | 88.0% | 65.3% | 99.7% | 92.9% | 34.4% |

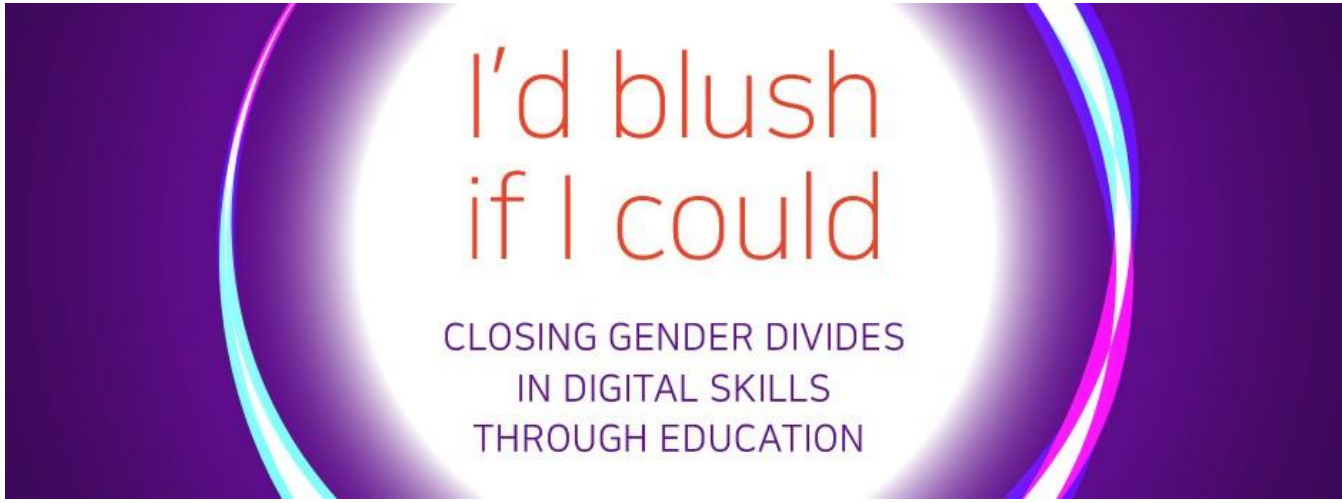
Gender shades project showed that :

- All companies perform better on lighter subjects as a whole than on darker subjects
- All companies perform worst on darker females.

# AI can be sexist and racist


Until 2019, **Siri** responded to
"**Hey Siri, you're a bitch**,"
with
"**I'd blush if I could**,"

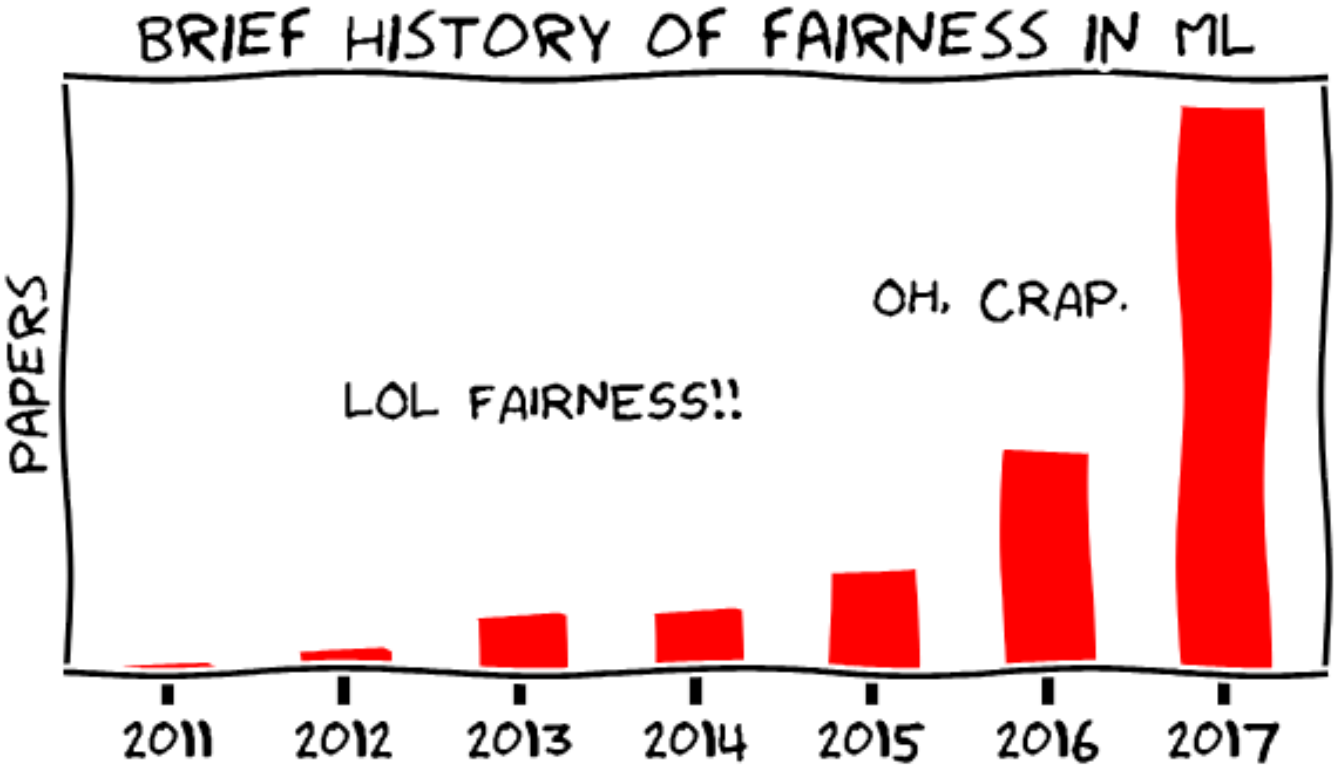
- 2019: A report released by (UNESCO) shows that virtual assistants reflect, reinforce and spread gender bias.
- It also highlighted that AI voice assistants can reinforce harmful general stereotypes




https://en.unesco.org/Id-blush-if-I-could

# Fairness is a hot topic and gaining traction!

- This Figure, illustrating the growth of articles on the topic of fairness in ML, with a peak in 2017 at the specialized **Neural Information Processing Systems (NIPS) conference** shows that the field is actively investigating how to bring fairness to predictive models.
- A signal that fairness is finally being taken seriously.




https://godatadriven.com/blog/towards-fairness-in-ml-with-adversarial-networks/

# EQUALITY

## Equality = SAMENESS

**Equality is about SAMENESS**, it promotes fairness and justice by giving everyone the same thing.

BUT it can **only work IF everone starts from the SAME place**, in this example equality only works if everyone is the same height.

# EQUITY

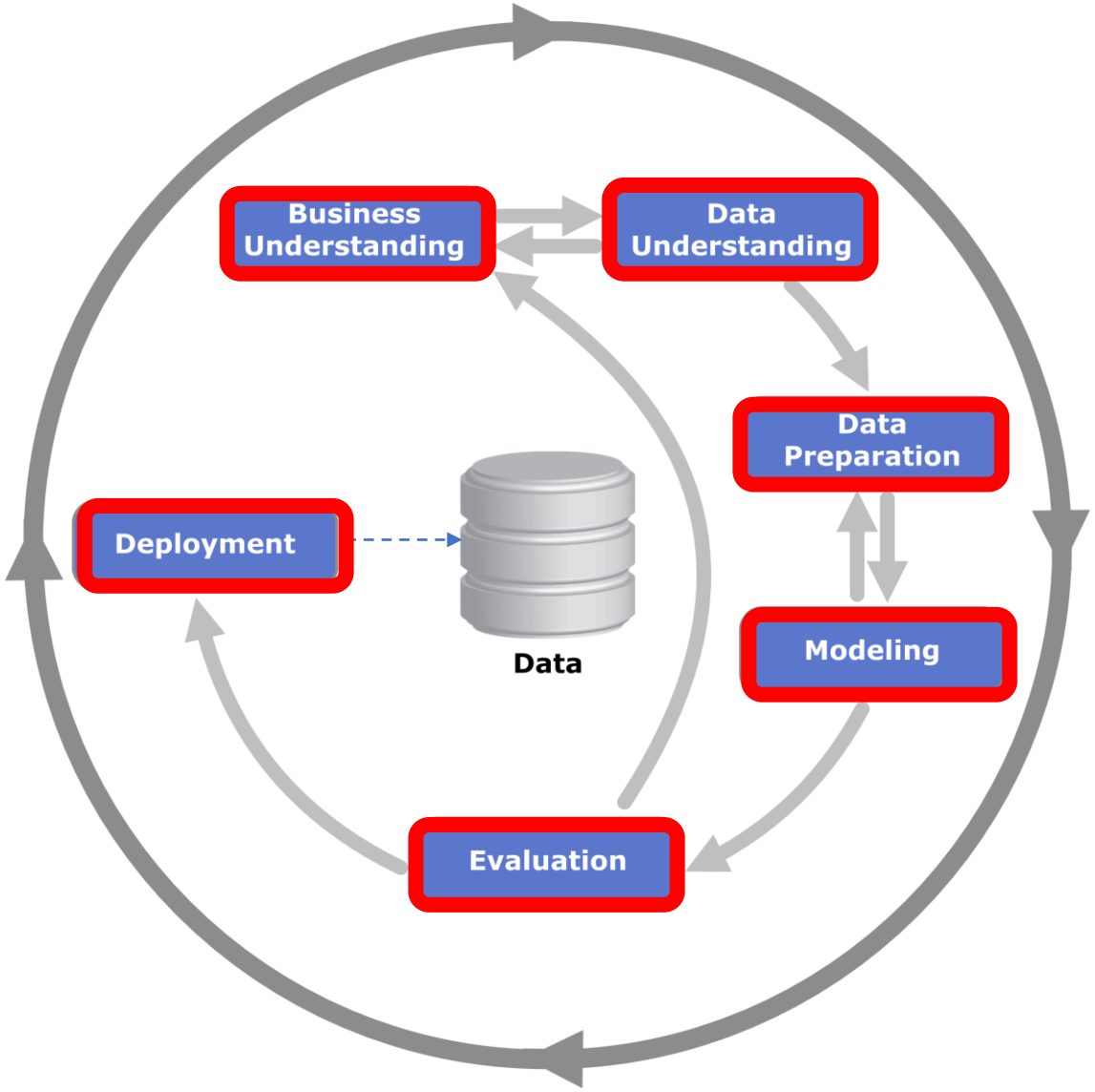
## Equity = FAIRNESS

**Equity is about FAIRNESS**, it's about making sure people get access to the same opportunities.

Sometimes our differences and/or history can make barriers to participation, so we must **FIRST ensure EQUITY** before we can enjoy equality.

# Sources of Bias in ML Lifecycle

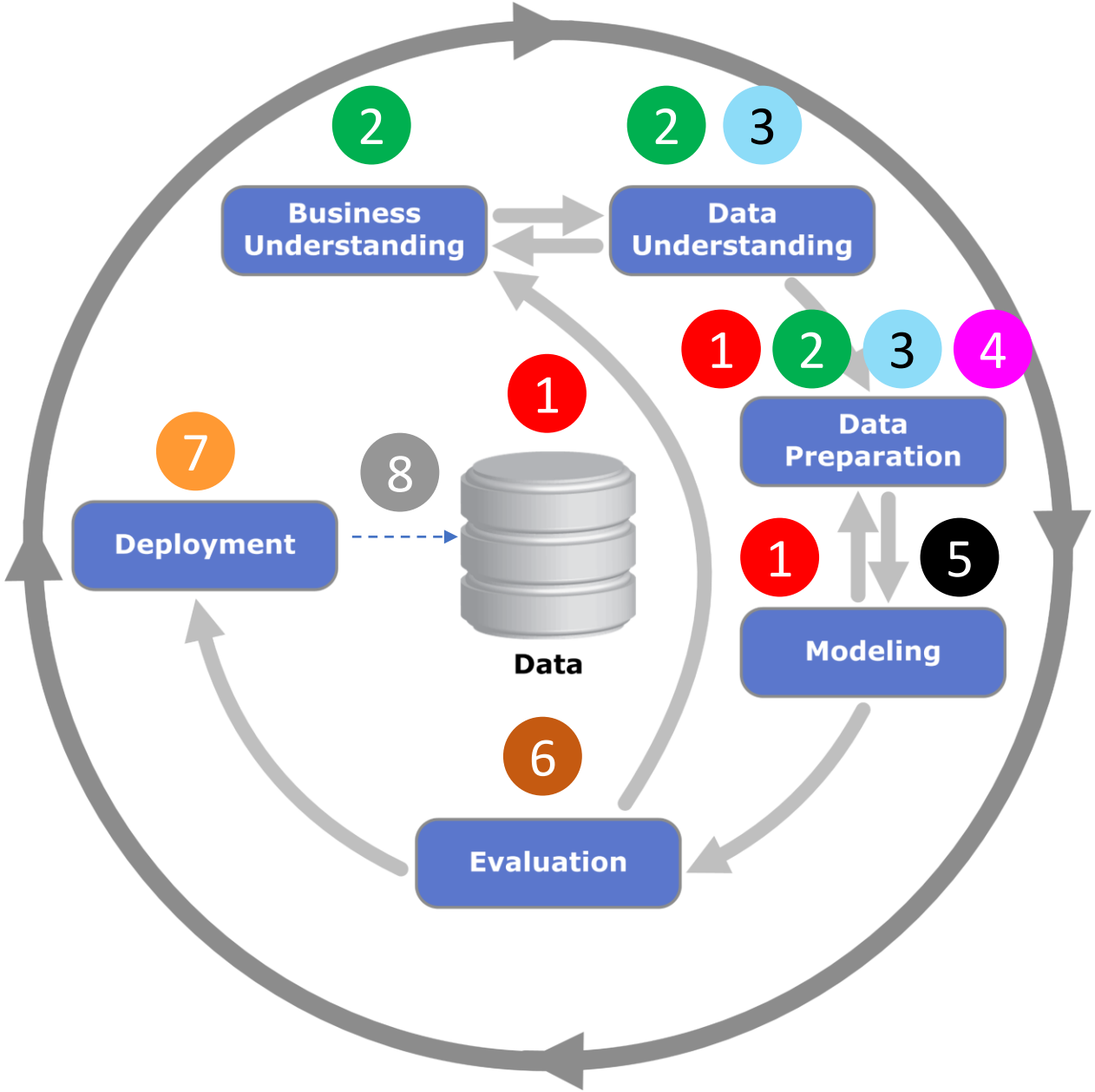
# Process Phases and Potential Biases of Machine Learning




- To show different sources of ML Bias we use The CRoss Industry Standard Process for Data Mining (**CRISP-DM**)
- **CRISP-DM** is the most common methodology for data mining, and data science projects [1990]
- CRISP-DM consists of six sequential phases that form the project lifecycle:

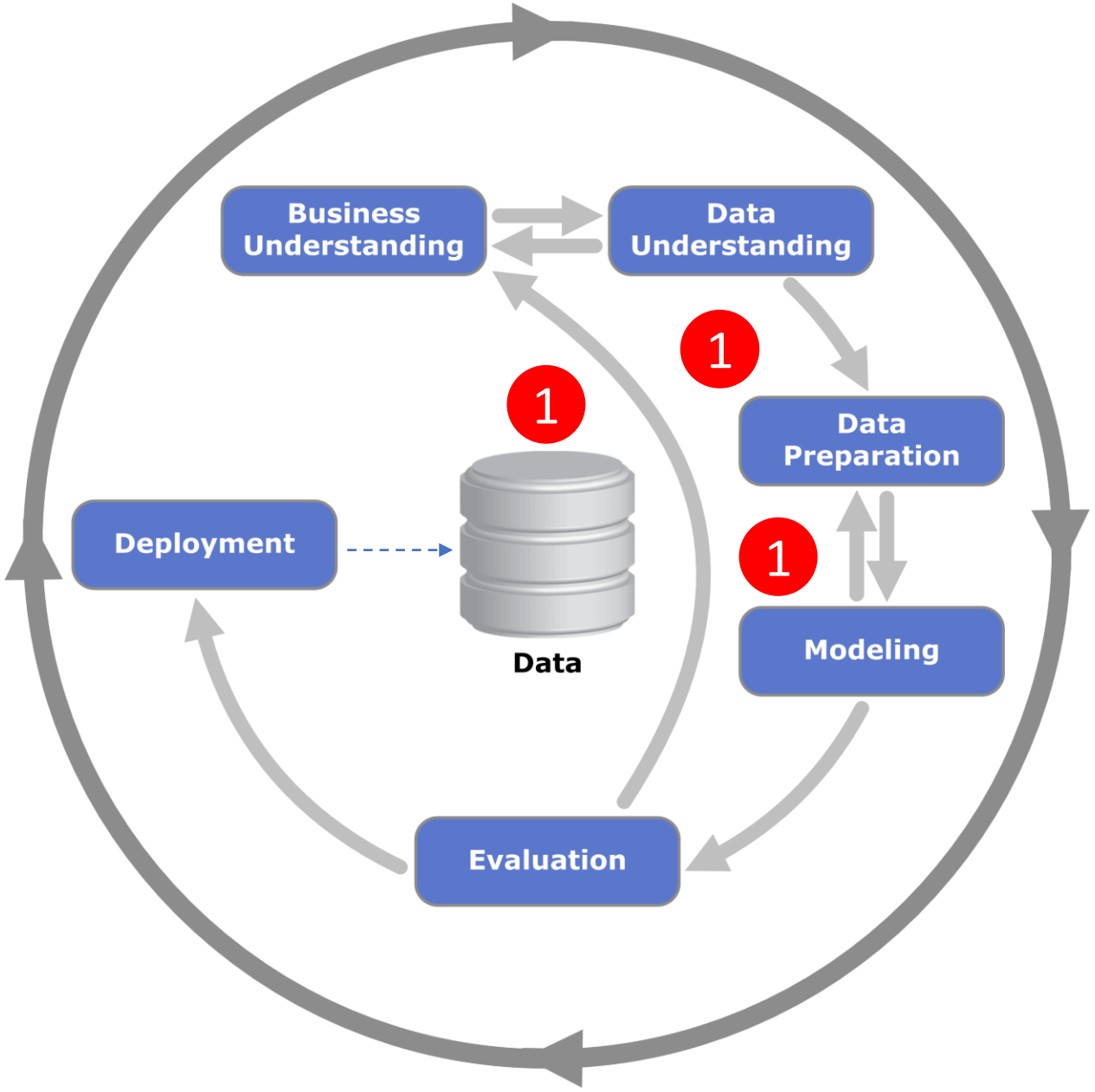
1. Business understanding: What are the business needs?
2. Data understanding: What data do we have/need? Is it clean?
3. Data preparation: How do we organize the data for modeling?
4. Modeling: What modeling techniques should we apply?
5. Evaluation: Which model best meets the business objectives?
6. Deployment: How do stakeholders access the results?

# Process Phases and Potential Biases of Machine Learning




[Van Giffen et al. 2022] defined various sources of bias, particularly according to the lifecycle phases:

1. Social Bias
2. Measurement Bias
3. Representation Bias
4. Label Bias
5. Algorithmic Bias
6. Evaluation Bias
7. Deployment Bias
8. Feed-back Bias




## 1. Social Bias

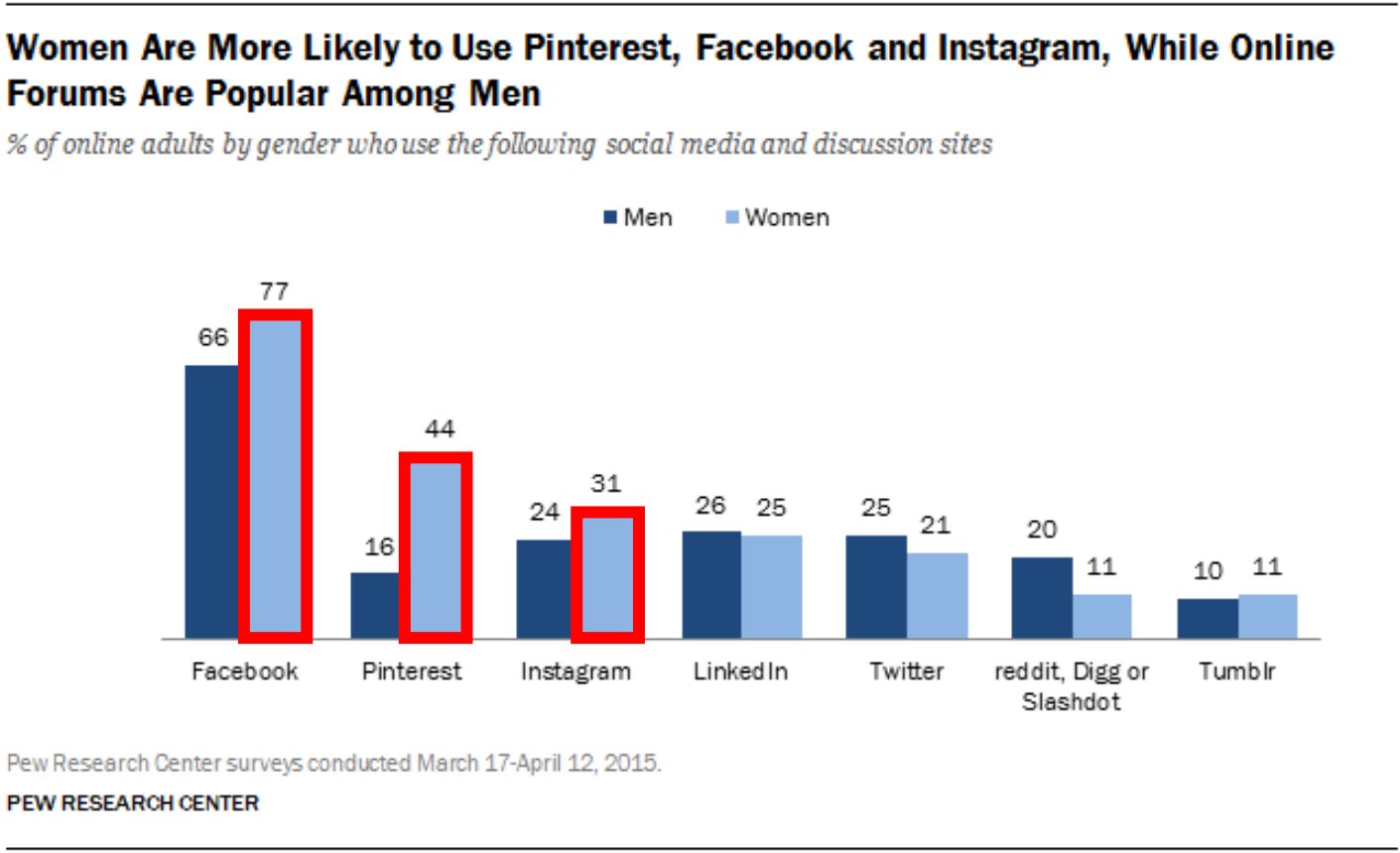
*Historical Bias, Societal Bias, Individual Bias, Pre-existing Bias, Population Bias*

This Bias illustrates the fact that available data reflects existing bias in the relevant population prior to the creation of the ML model.

# Social Bias

## Different user demographics on different social plateforms

- When we collect data from different social platforms, we need to be aware that each platform can attract a specific demographic group since this creates a social bias.
- For example, if the model is trained on data primarily from a social platform used by men, it may fail to make accurate or fair predictions for women.



# Social Bias

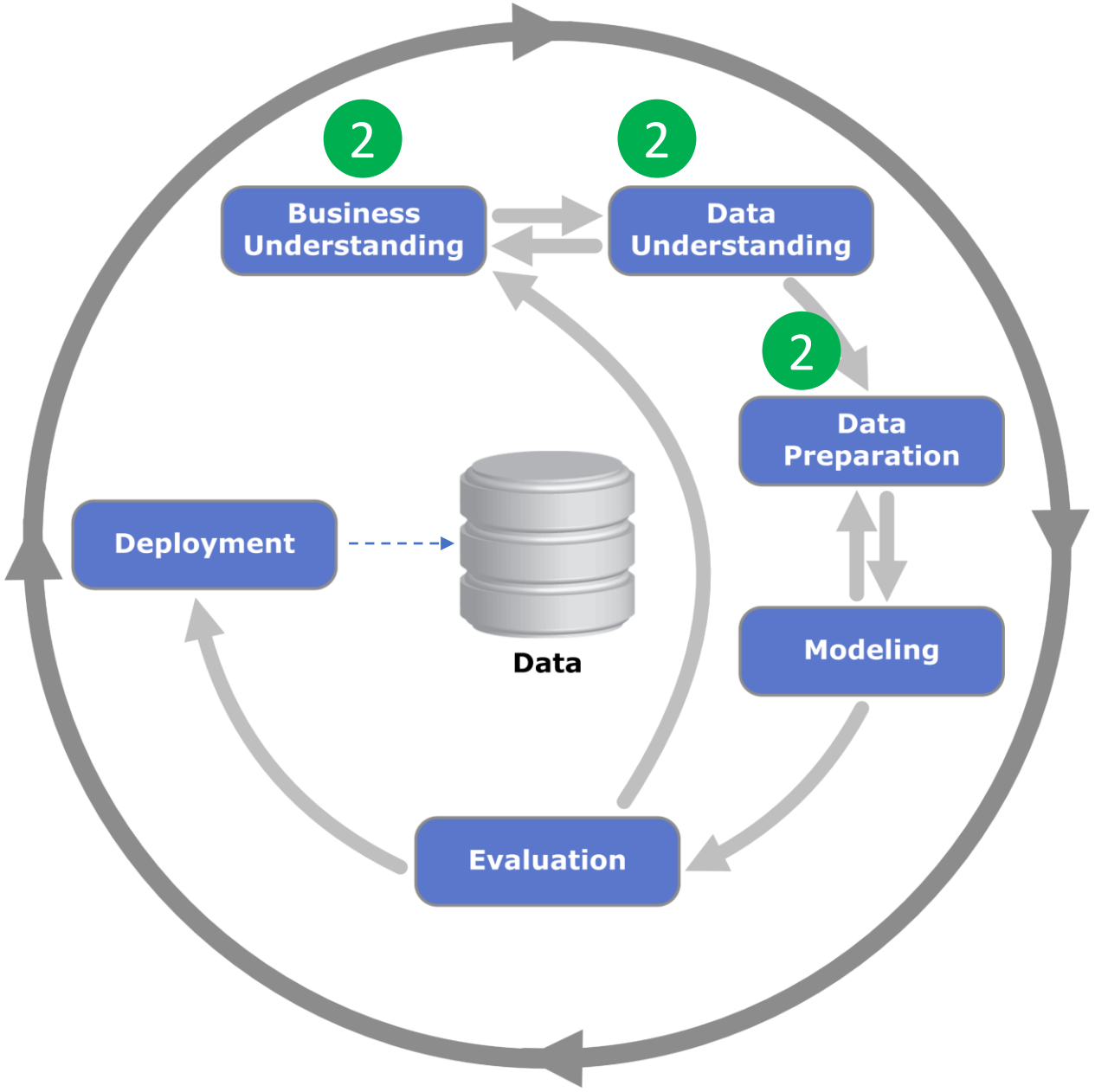
Note that for many tasks, data should match '**target population'**, to improve external validity:

- In fraud detection, including a representative proportion of fraudulent transactions in the training data is crucial to build an accurate and fair model. By using balanced and representative data, a fraud detection model can learn to identify specific patterns associated with fraudulent transactions and make unbiased predictions for all types of transactions.
- When studying breast cancer, it is essential to have a training dataset that specifically includes a representative proportion of positive breast cancer cases among women

## But…

This balanced data is not allowed in loan and recruitment applications where gender discrimination is illegal.

Thus distortions in data should be evaluated in a **task dependent way**




## 2) Measurement Bias

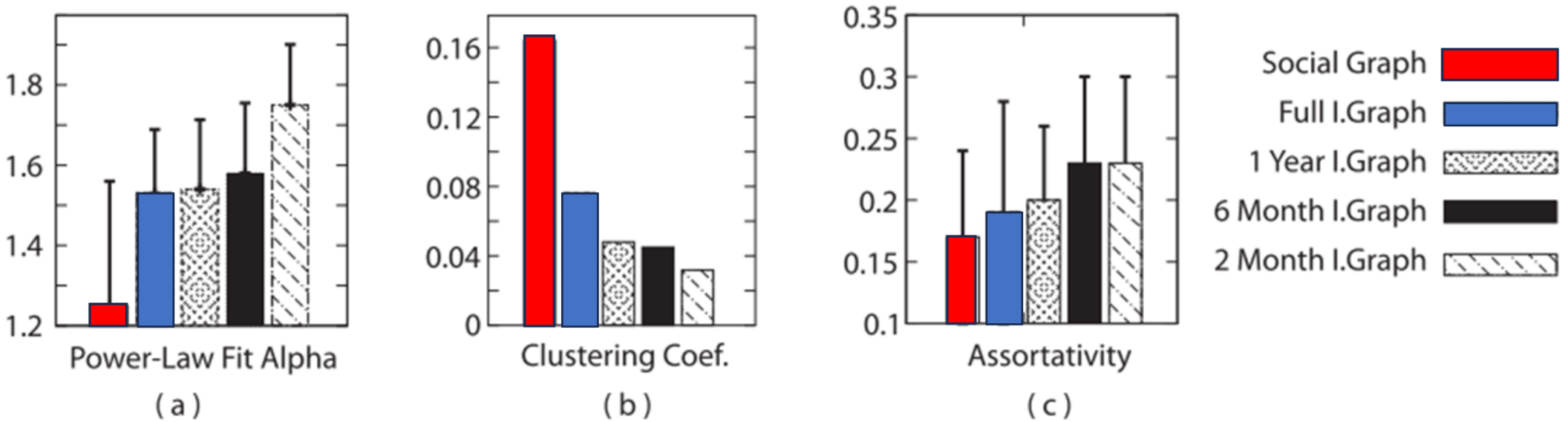
*Linking Bias, Omitted Variable Bias*

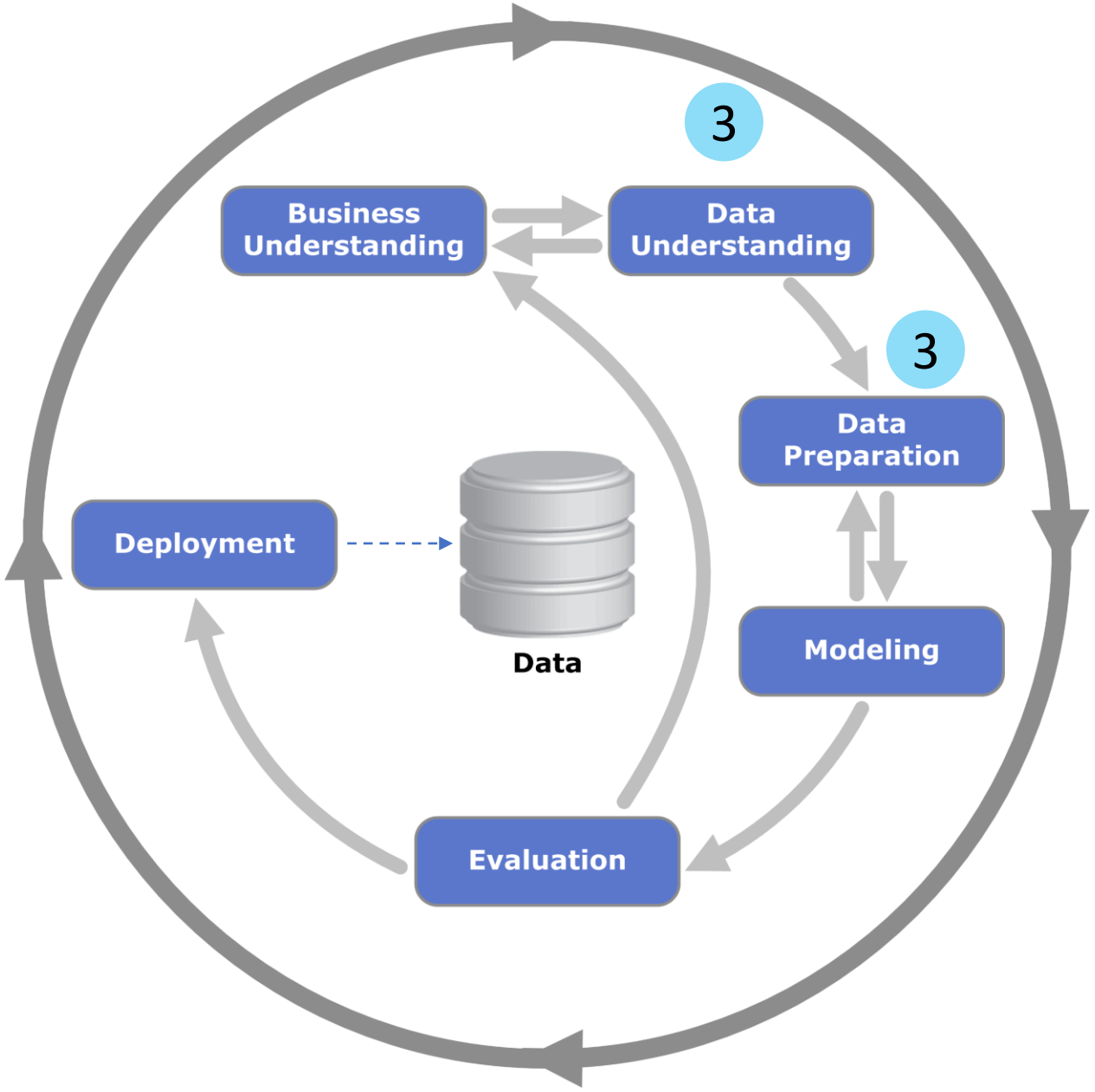
Generally, chosen features and labels are **imperfect proxies** (substitutes) for the real variables of interest.

# Measurement Bias

## Behavior-based and connection-based social links are different

- In their paper "*Beyond Social Graphs: User Interactions in Online Social Networks and their Implications*" [Wilson et al. 2012] analyze interaction graphs derived from Facebook user traces (blue) and show that they exhibit different "**small-world**" properties (e.g. power-law fit, clustering, assortativity, etc.) present in their social graph counterparts (red): larger graph diameters, lower clustering coefficients, and higher assortativity.
- Social links are not valid indicators of real user interactions






## 3) Representation Bias

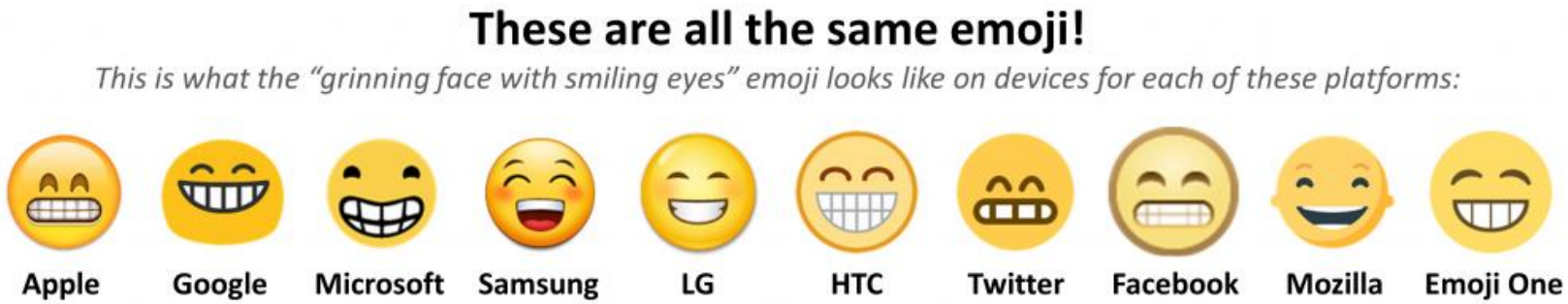
*Temporal Bias, Longitudinal Data Fallacy, Emergent Bias, Population Bias, Group Bias, Aggregation Bias, Behavioral Bias, Sampling Bias, Content Production Bias, (Self) Selection Bias, Availability Bias*

The input data is not representative for the relevant population (e.g. Differences in populations and behaviors over time, Differences in user behavior across plateforms etc.).

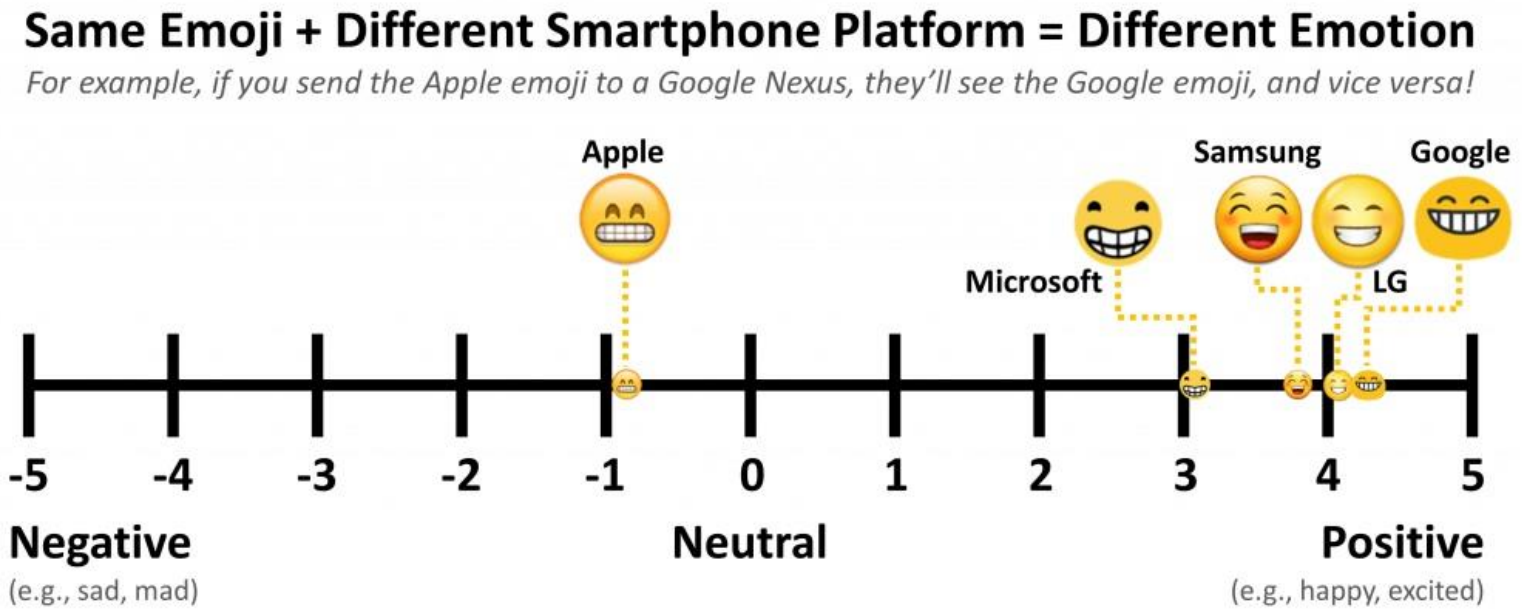
# Representation Bias

Differences in user behavior across plateforms [Miller, 2016]

**Platform functionally and algorithms influence human behaviors and our observations of human behaviors**



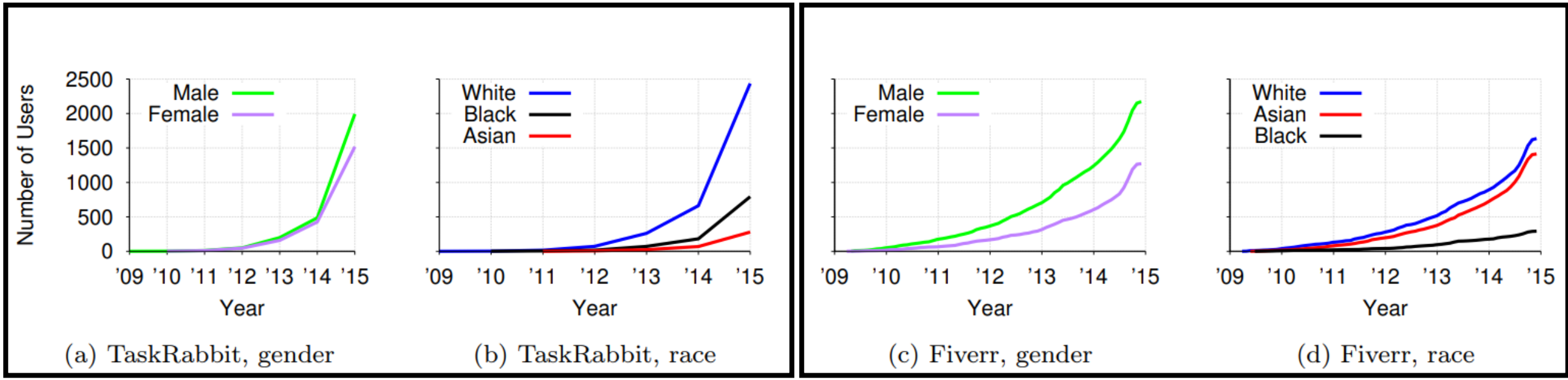

https://grouplens.org/blog/investigating-the-potential-for-miscommunication-using-emoji/



# Representation Bias

Different demographic groups can have varying growth rates on social platforms, both between different platforms and within each platform

In [Hannák et al. 2017], authors study whether two online freelance marketplaces—**TaskRabbit** and **Fiverr**— are impacted by racial and gender bias (they collect 13,500 worker profiles 2009-2015) .




(a) TaskRabbit, gender (b) TaskRabbit, race (c) Fiverr, gender (d) Fiverr, race

Member growth over time on TaskRabbit and Fiverr, broken down by perceived gender and race.

# Representation Bias

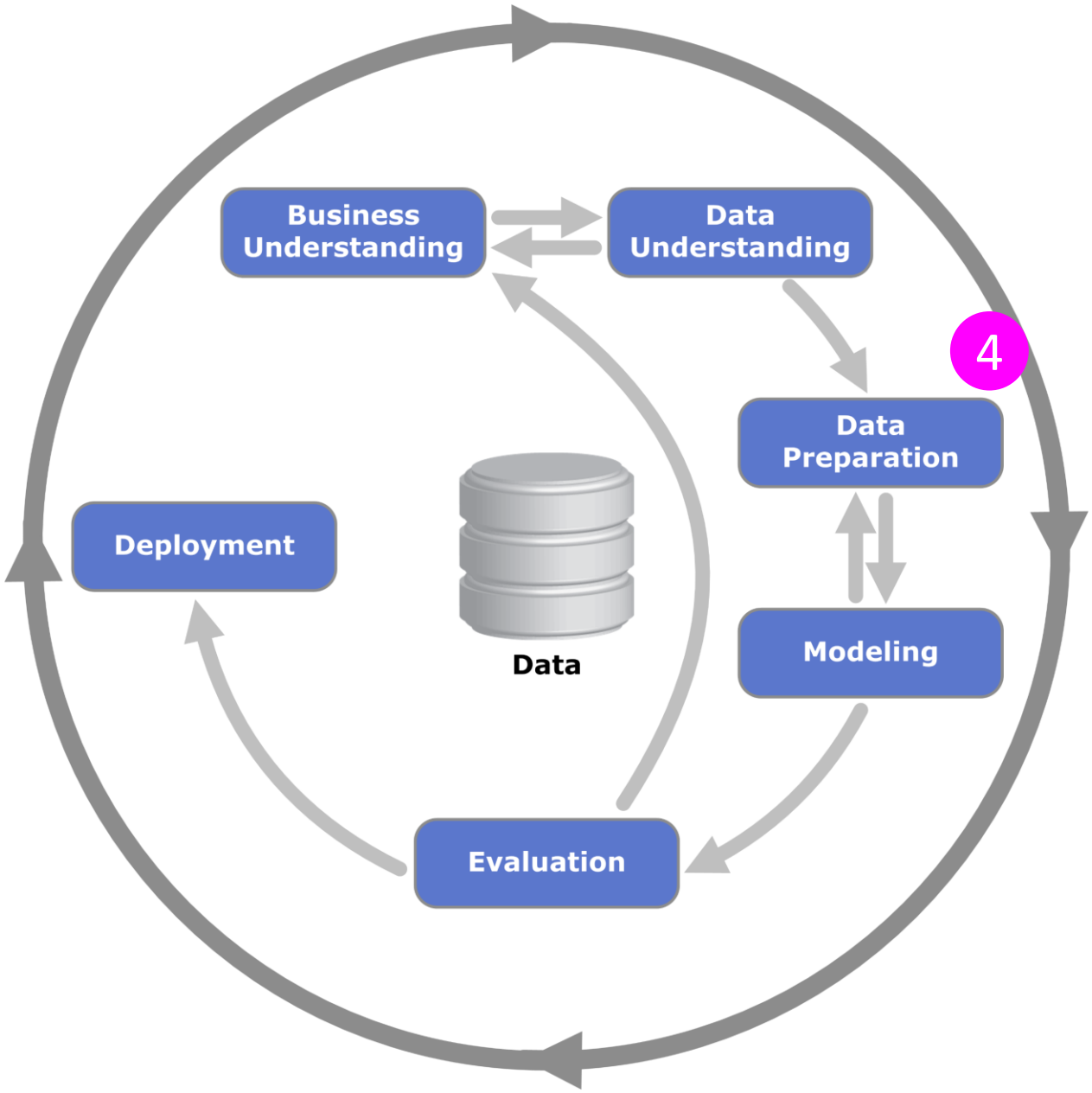
## Community norms and societal biases influence observed behavior and vary across online and offline communities and contexts

What kind of pictures would you share on Facebook, but not on LinkedIn?


The same social media act can have multiple interpretations and may not always reflect a clear consensus or agreement.

*For example, a **retweet** on social media can be interpreted differently by different individuals. It may not always indicate a clear consensus or agreement. One person might retweet a post to show support or agreement, while another might retweet it to criticize or express sarcasm.*

[Tufekci ICWSM'14]
https://arxiv.org/ftp/arxiv/papers/1403/1403.7400.pdf



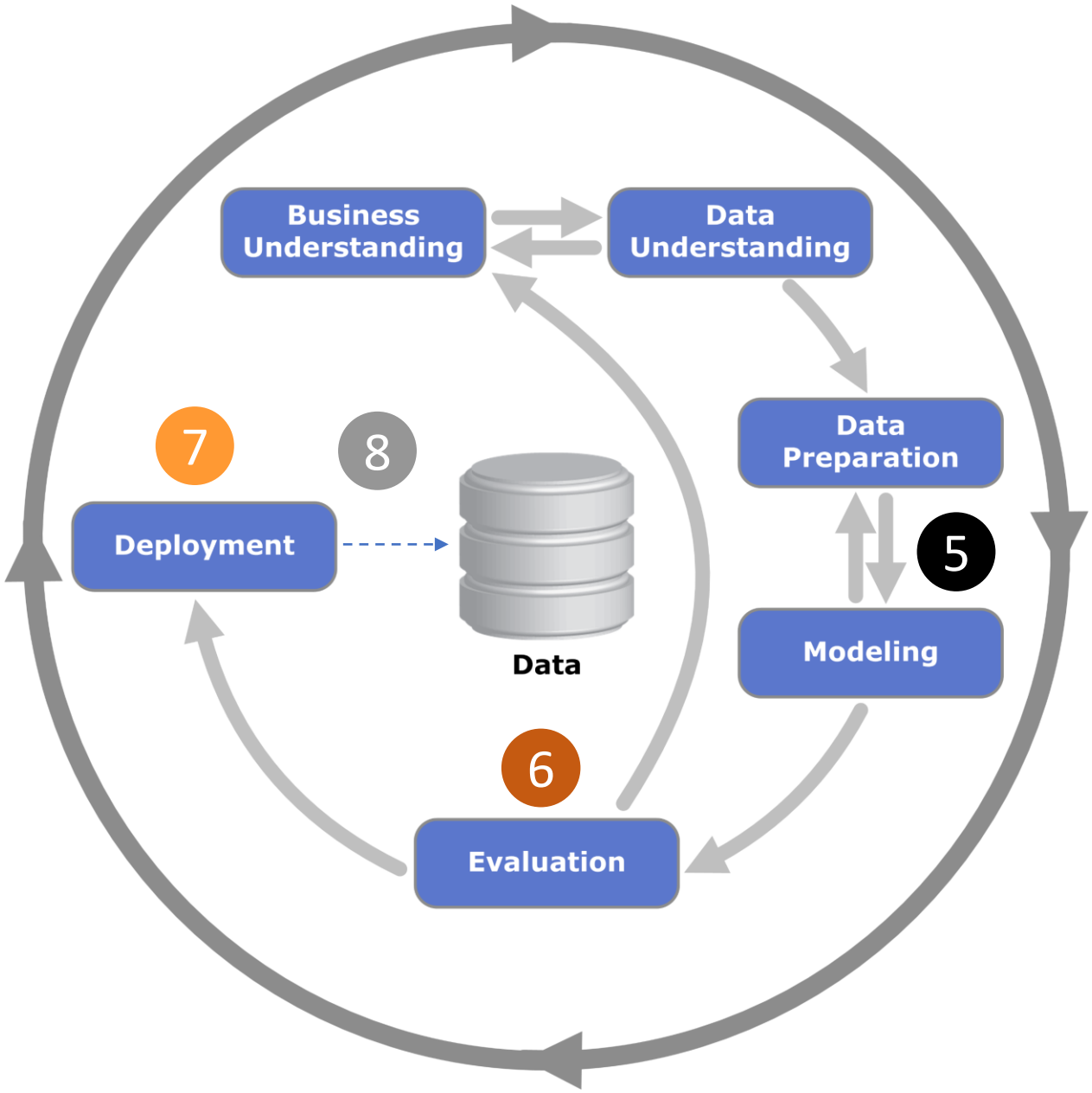

4. **Label Bias**: Labelled data systematically deviate from the underlying truth categories.

What are the different approaches to data labeling?

There are two approaches:

**1.Manual labeling** is the process of assigning labels to data by a human annotator. May introduce errors if the person labeling the data is not careful or is not familiar with the task at hand.

**2.Automated labeling** is the process of using algorithms and software to automatically assign labels to data. This can be faster and more accurate than manual labeling, but it requires training the algorithms on a labeled dataset, which can be labor-intensive and time-consuming. Additionally, automatic labeling may not always be as accurate as manual labeling, especially for complex tasks or data that is difficult to label.




4. **Algorithmic Bias** (*Statistical Bias, Technical Bias*) : Inappropriate technical considerations during modeling lead to systemic deviation of the outcome

5. **Evaluation Bias** (*Observer Bias, Funding Bias*) : A non-representative testing population or inappropriate performance metrics are used to evaluate the ML model

6. **Deployment Bias** (*Cause-Effect Bias*): The ML model is used and interpreted in a different context than it was built for

7. **Feedback Bias** (*Presentation Bias, User Interaction Bias, Popularity Bias, Ranking Bias, Second Order Bias*): The outcome of the ML model influences the training data such that a small bias can be reinforced by a feedback loop.

## AI perpetuates real-world stereotypes and biases

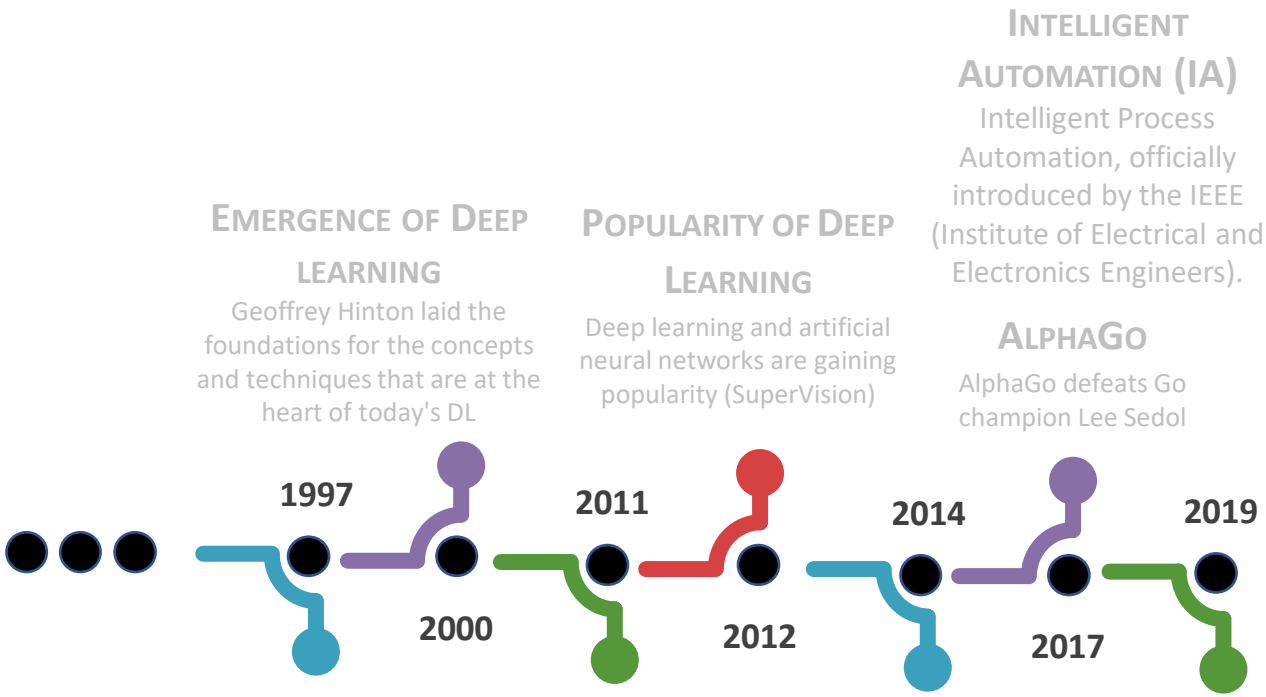
→ Need of Regulatory and Normative Framework for AI

There are several initiatives that emphasize the principles of ethical AI, including:

- **Ethics Guidelines for Trustworthy AI**, 2019, Initiating Organization: European Commission
https://www.aepd.es/sites/default/files/2019-12/ai-ethics-guidelines.pdf
- **IEEE Ethics of AI Principles**, 2016, Initiating Organization: Institute of Electrical and Electronics Engineers (IEEE)
https://ethicsinaction.ieee.org/
- **Asilomar AI Principles**, 2017, Initiating Organization: Future of Life Institute
https://futureoflife.org/ai-principles/
- **UNESCO's Principles for AI**, 2019, Initiating Organization: United Nations Educational, Scientific and Cultural Organization (UNESCO)
https://www.unesco.org/en/artificial-intelligence/recommendation-ethics

The "Ethics Guidelines for Trustworthy AI" is one of the most influential and comprehensive initiatives in the field of AI ethics.

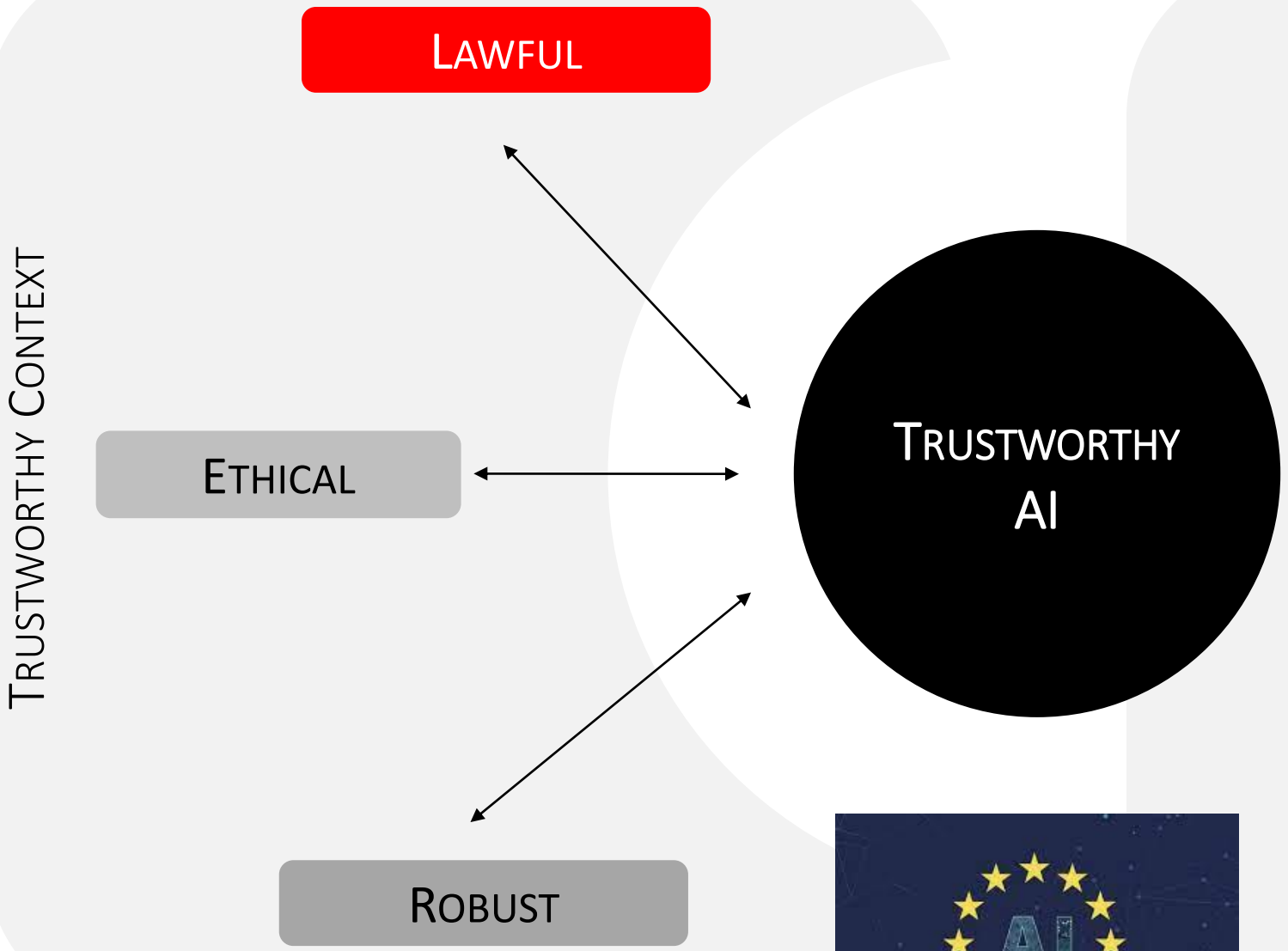
# 70 years of History





- "**Ethics Guidelines for Trustworthy AI**" – established by the European Commission in 2019
- These guidelines were developed by an independent group of experts known as the High-Level Expert Group on Artificial Intelligence (AI HLG).
- The aim was to provide ethical guidance and ensure that AI technologies are developed and used in a responsible and unbiased manner.

# EU’s Ethics Guidelines for Trustworthy AI




The EU's Ethics Guidelines emphasize the importance of AI being human-centric and state that trustworthy AI should be:

- **Lawful**: AI should operate within existing legal frameworks. Many laws, regulations, and fundamental rights that apply to tools, products, and situations also apply to AI.

  Discrimination, for example, is not allowed by law, regardless of whether it originates from a biased AI system or a biased human.

- **Ethical**: Ethical considerations help understand the law and go beyond negative obligations by prescribing positive obligations (Not everything lawful is always ethical).
- **Robust**: Trust in AI systems arises from their robustness. A robust AI system considers both technical aspects and the social environment.

# Law againt descrimination
# Example of Legally Recognized Protected Classes [Boracas & Hardt 2017]

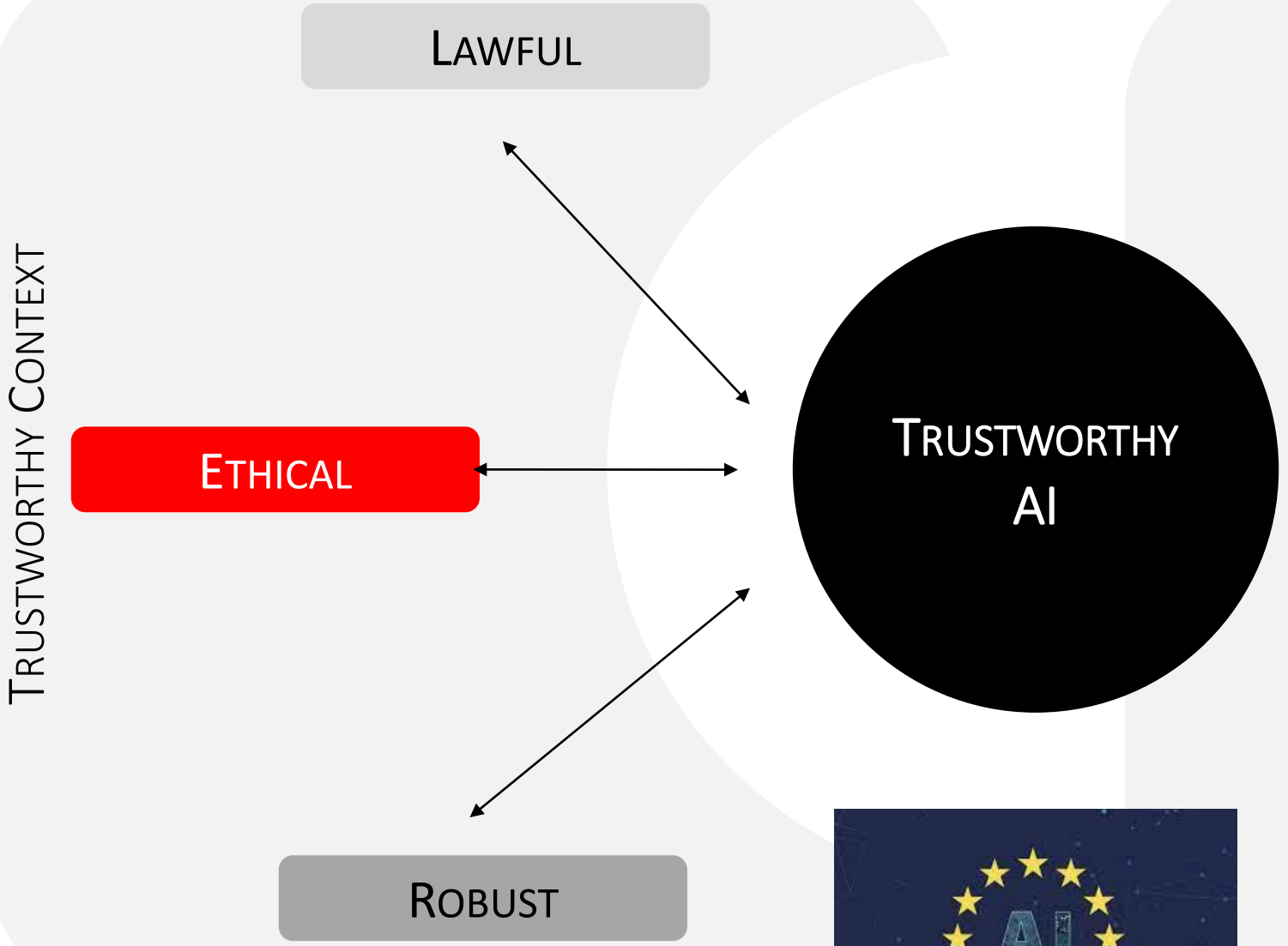
| |
|---|
| Race (Civil Rights Act of 1964); |
| Color (Civil Rights Act of 1964); |
| Sex (Equal Pay Act of 1963; Civil Rights Act of 1964); |
| Religion (Civil Rights Act of 1964); |
| National origin (Civil Rights Act of 1964); |
| Citizenship (Immigration Reform and Control Act); |
| Age (Age Discrimination in Employment Act of 1967); |
| Pregnancy (Pregnancy Discrimination Act); |
| Familial status (Civil Rights Act of 1968); |
| Disability status (Rehabilitation Act of 1973; Americans with Disabilities Act of 1990); |
| Veteran status (Vietnam Era Veterans' Readjustment Assistance Act of 1974; Uniformed Services Employment and Reemployment Rights Act); |
| Genetic information (Genetic Information Nondiscrimination Act) |

And more …

# Regulated Domains [Barocas & Hardt 2017]

- Credit (Equal Credit Opportunity Act)
- Education (Civil Rights Act of 1964; Education Amendments of 1972)
- Employment (Civil Rights Act of 1964)
- Housing (Fair Housing Act)
- Public Accommodation (Civil Rights Act of 1964)

# EU’s Ethics Guidelines for Trustworthy AI





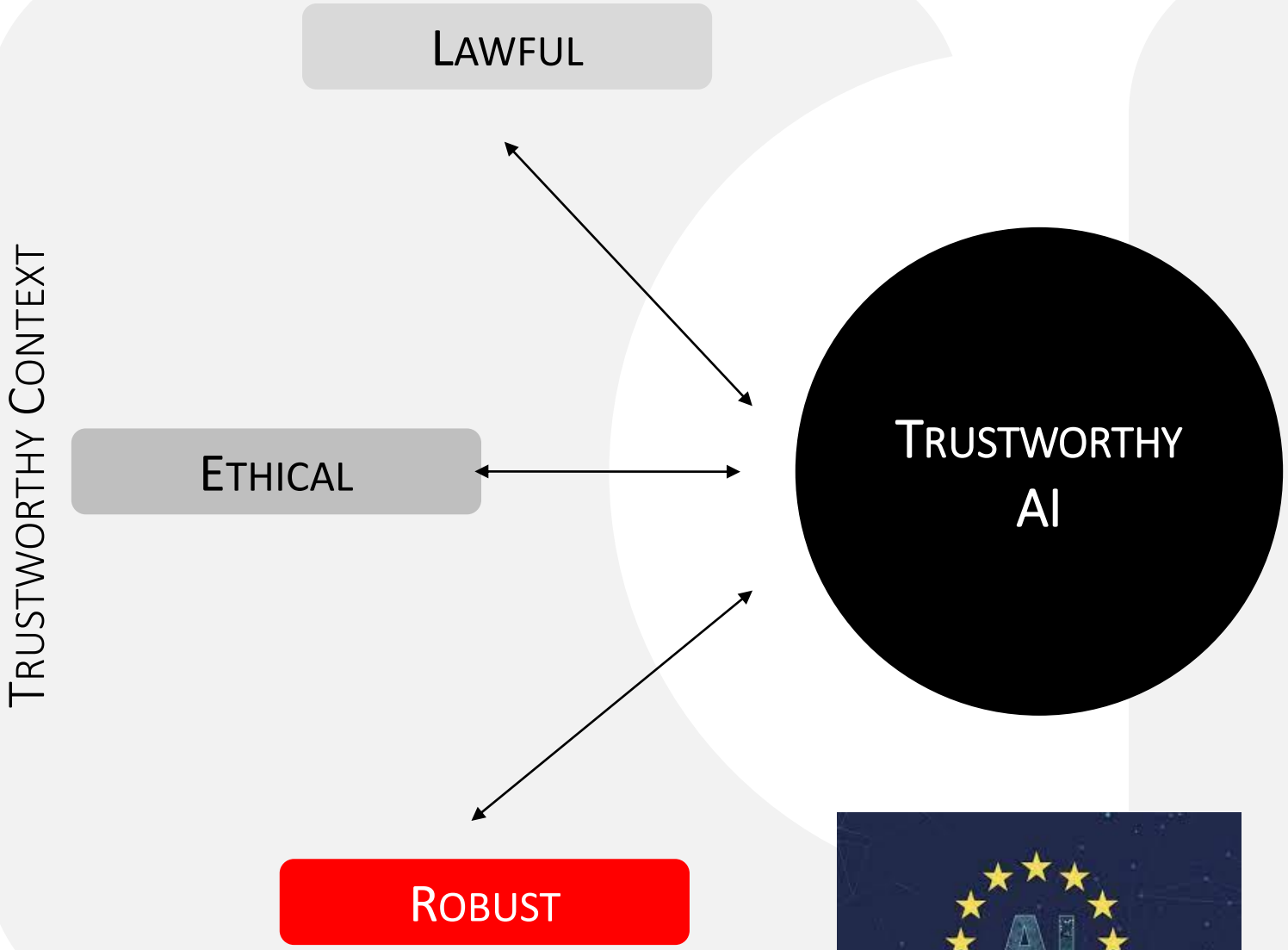


The EU's Ethics Guidelines emphasize the importance of AI being human-centric and state that trustworthy AI should be:

- **Lawful**: AI should operate within existing legal frameworks. Many laws, regulations, and fundamental rights that apply to tools, products, and situations also apply to AI. Discrimination, for example, is not allowed by law, regardless of whether it originates from a biased AI system or a biased human.
- **Ethical**: Ethical considerations help understand the law and go beyond negative obligations by prescribing positive obligations (Not everything lawful is always ethical).

Deriving personal information from large amounts of personal user data without their explicit consent may be legal, but from an ethical standpoint, it could be seen as a violation of individuals' privacy.

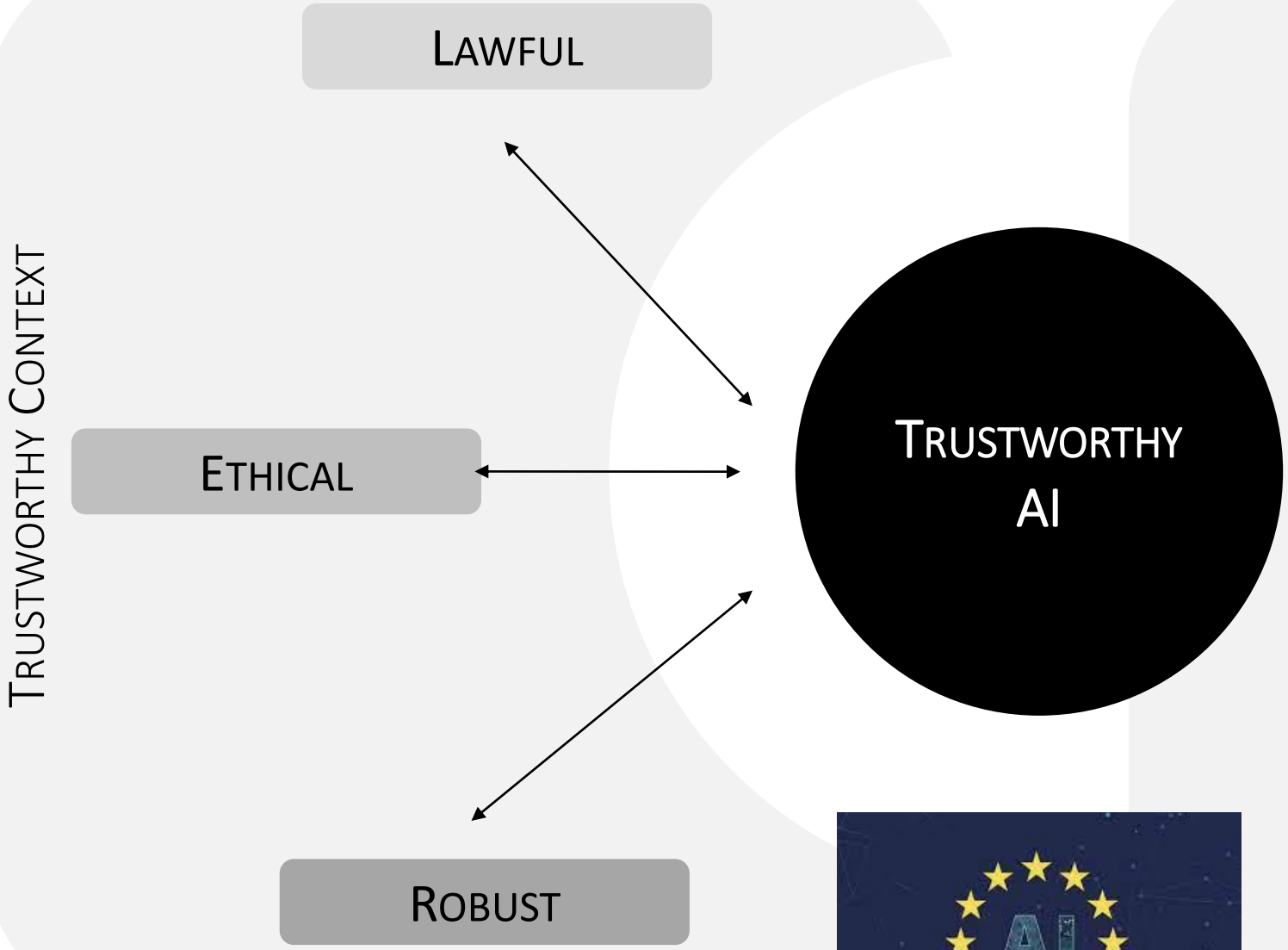
# EU’s Ethics Guidelines for Trustworthy AI




The EU's Ethics Guidelines emphasize the importance of AI being human-centric and state that trustworthy AI should be:

- **Lawful**: AI should operate within existing legal frameworks. Many laws, regulations, and fundamental rights that apply to tools, products, and situations also apply to AI. Discrimination, for example, is not allowed by law, regardless of whether it originates from a biased AI system or a biased human.
- **Ethical**: Ethical considerations help understand the law and go beyond negative obligations by prescribing positive obligations (Not everything lawful is always ethical).
- **Robust**: Trust in AI systems arises from their robustness. A robust AI system considers both **technical** aspects and the **social** environment.

# EU’s Ethics Guidelines for Trustworthy AI



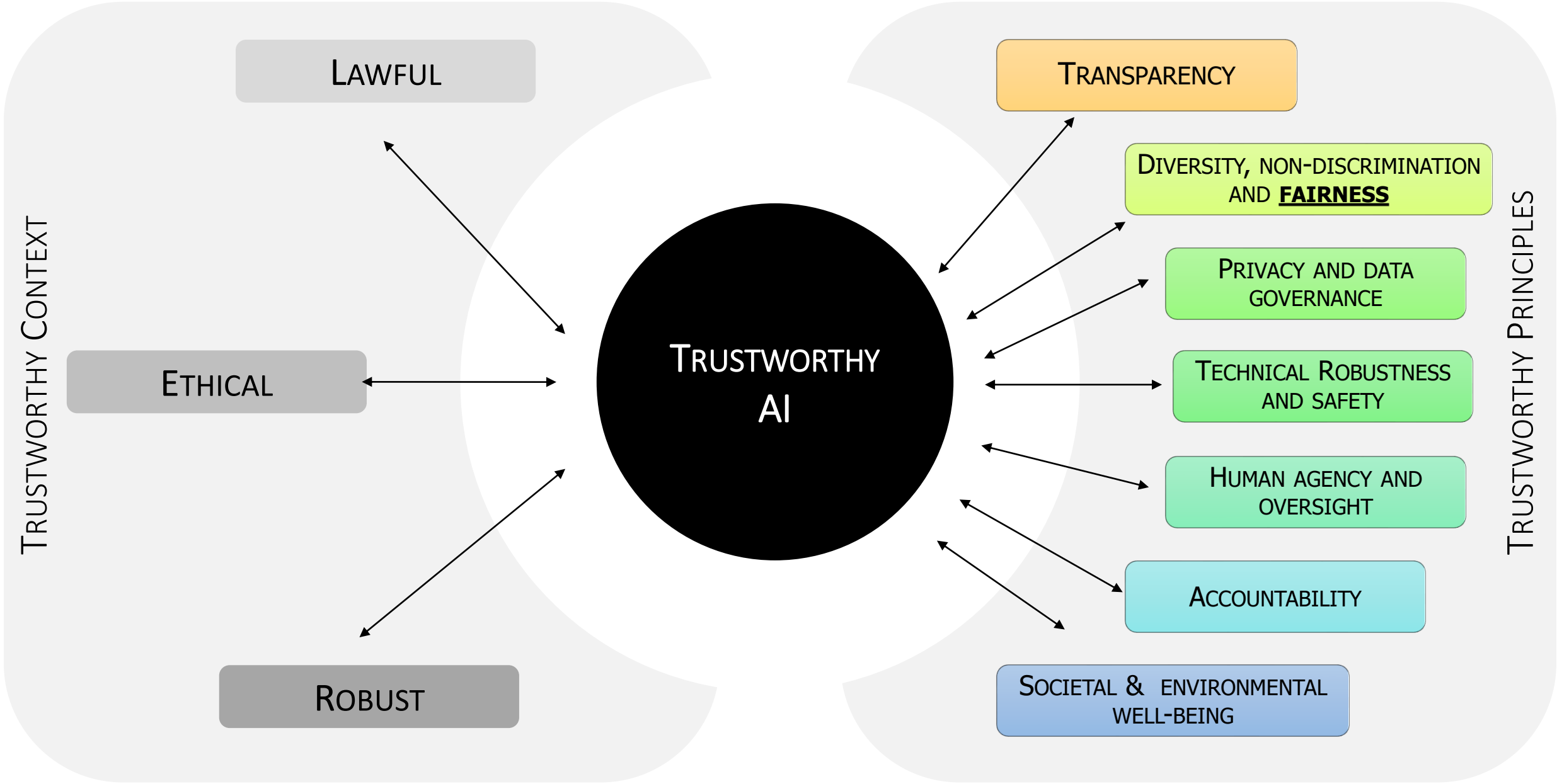

The EU's Ethics Guidelines emphasize the importance of AI being human-centric and state that trustworthy AI should be:

- **Lawful**: AI should operate within existing legal frameworks. Many laws, regulations, and fundamental rights that apply to tools, products, and situations also apply to AI. Discrimination, for example, is not allowed by law, regardless of whether it originates from a biased AI system or a biased human.
- **Ethical**: Ethical considerations help understand the law and go beyond negative obligations by prescribing positive obligations (Not everything lawful is always ethical).
- **Robust**: Trust in AI systems arises from their robustness. A robust AI system considers both **technical** aspects and the **social** environment.

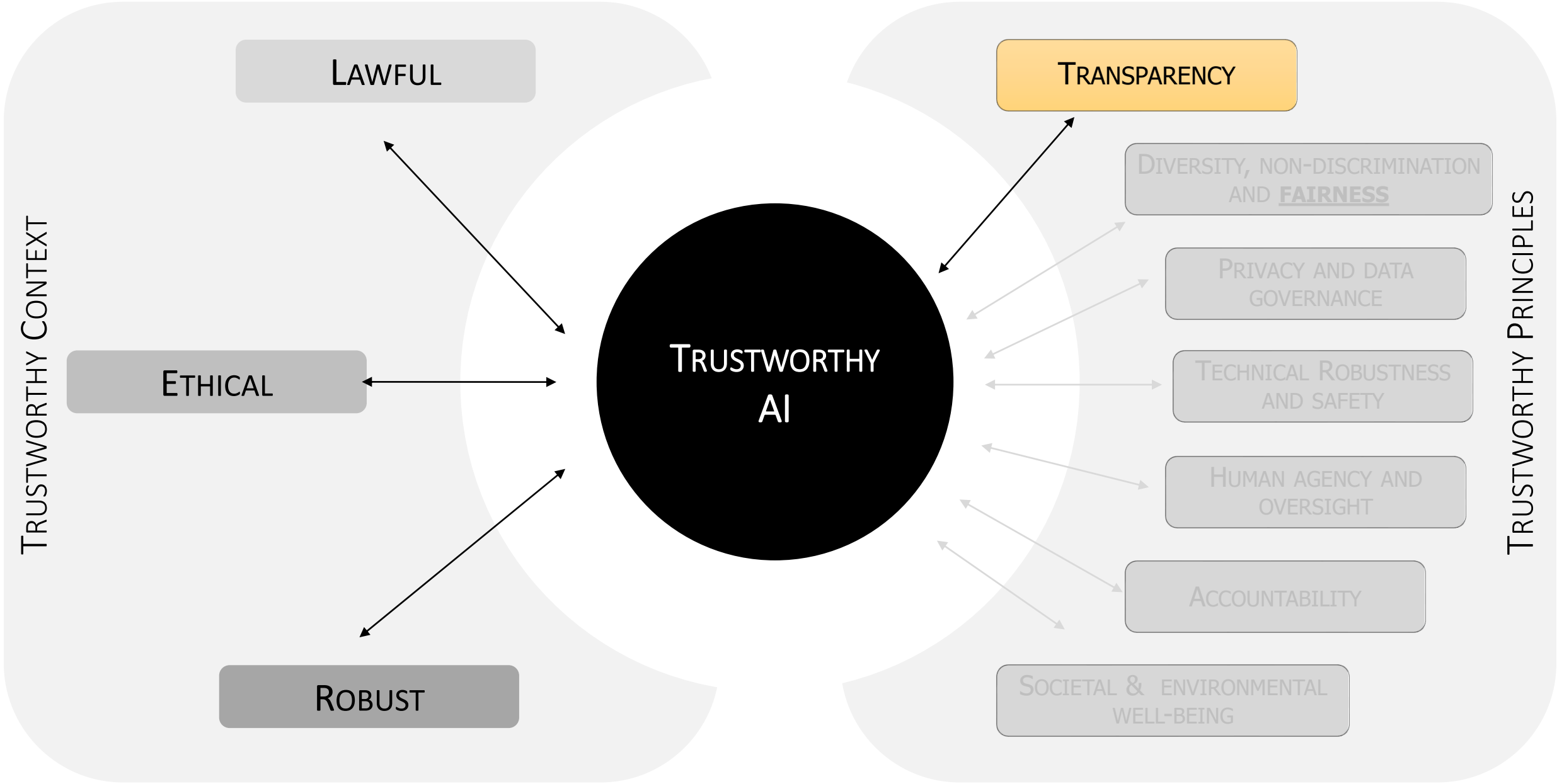



**These guidelines translate into 7 key principles for trustworthy AI**

# EU's Ethics Guidelines for Trustworthy AI

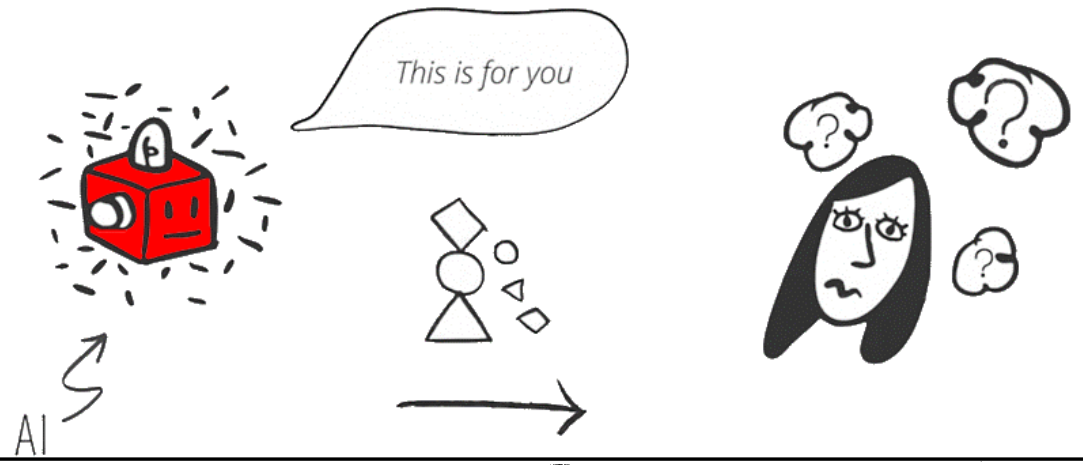


# EU's Ethics Guidelines for Trustworthy AI



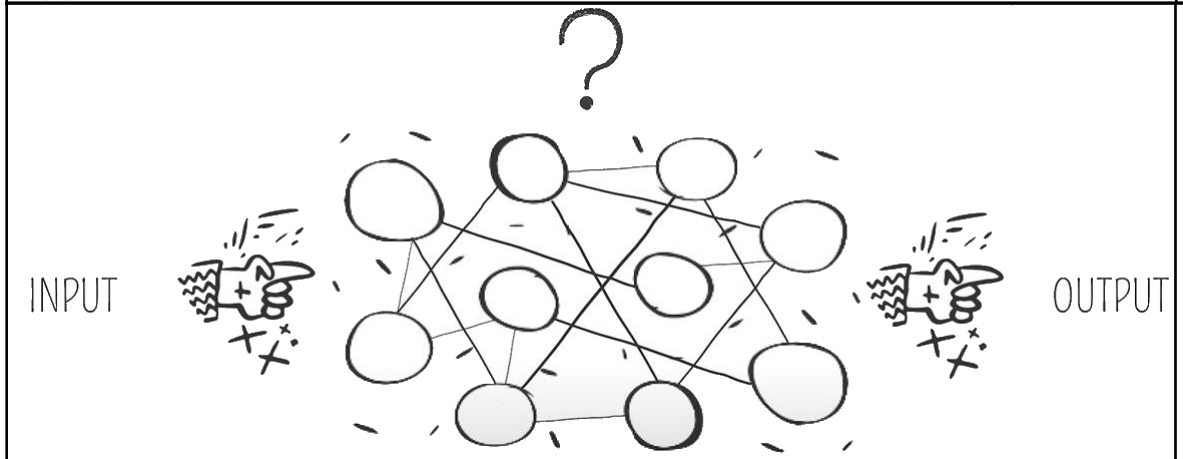
# Transparency (1)

Credit for the illustrations in this section goes to https://www.trustworthyaiproject.eu/teaching-trustworthy-ai-oers/?fbclid=IwAR2p4n_T7VXd45hLdCoevrR2WKR0kPm8OSxEuBpRXVsY2D7tXR4uSfm3zBU




- It is crucial for humans to understand how an AI system makes decisions.
- Therefore, AI systems should be designed to be transparent

IEEE P7001: A Proposed Standard on Transparency

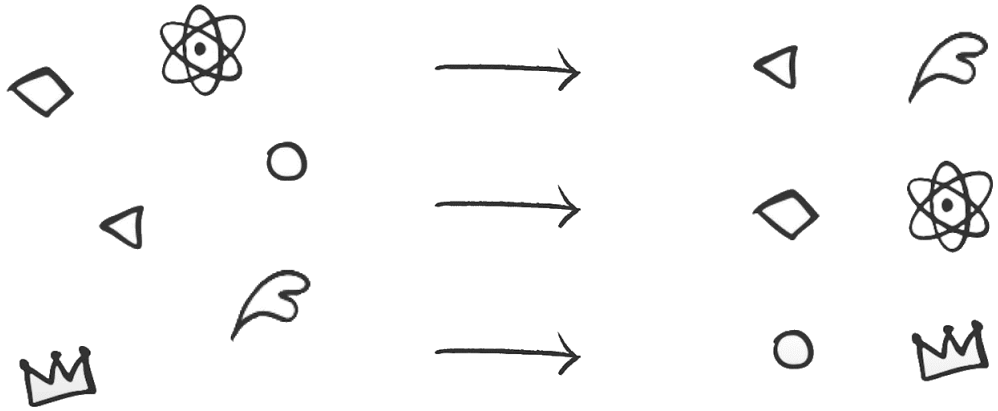



- The need for transparency arises from the "black box" problem.
- Many contemporary AI systems, particularly those based on **deep learning**, learn to recognize patterns independently.
- As these patterns become increasingly complex, humans can only observe the outcomes of the AI system without understanding the internal workings.

# Transparency (1)

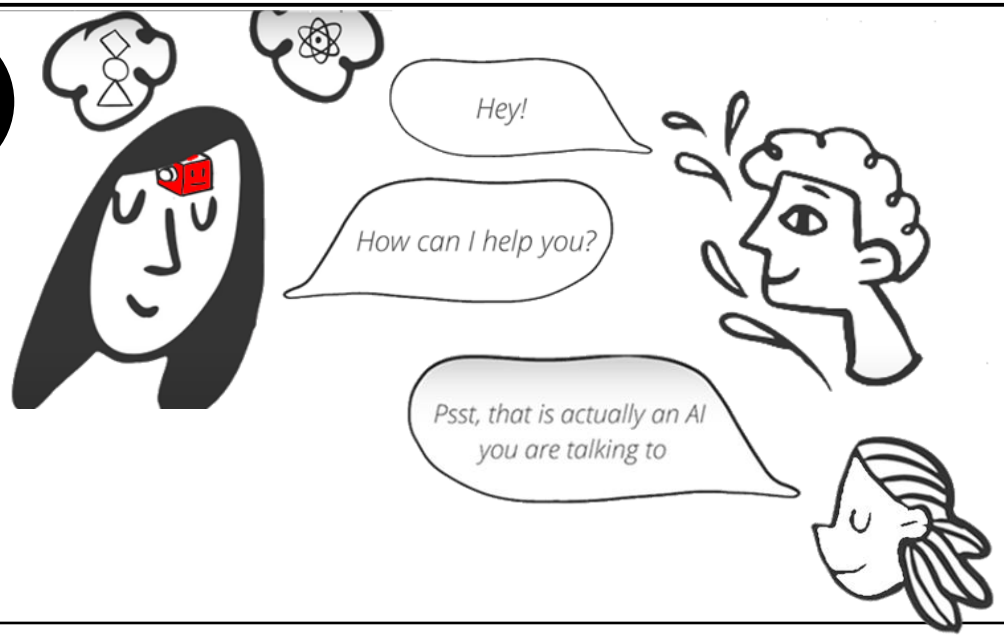
Credit for the illustrations in this section goes to: https://www.trustworthyaiproject.eu/teaching-trustworthy-ai-oers/?fbclid=IwAR2A4wW4_YfMSP6PE1gINU_ZioZyvb1xaSkCjdbpedteuF47eIAI2J6U16s

**1**



- Firstly, transparency means that the AI system should be **traceable**.
- This traceability helps us
  - understand why a decision was made or why the system may have made a mistake or exhibited bias.
  - It also enables us to learn from past mistakes and improve future iterations of the AI system.

**2**




- Secondly, individuals should be able to know whether they are interacting with an AI system or with a human. If they prefer human interaction, the deployer of the AI system should provide that option.
- This helps maintain transparency and allows individuals to make informed choices

# Transparency (2)

3


- Thirdly, it is important for AI users to be able to explain how the system arrives at its decisions
- When an AI system has a significant impact on people's lives, as loan approvals or criminal investigations. they have the right to receive a suitable explanation for the decisions made by the system.

- **Transparency is not just a technical fix; it requires a socio-technical process.**

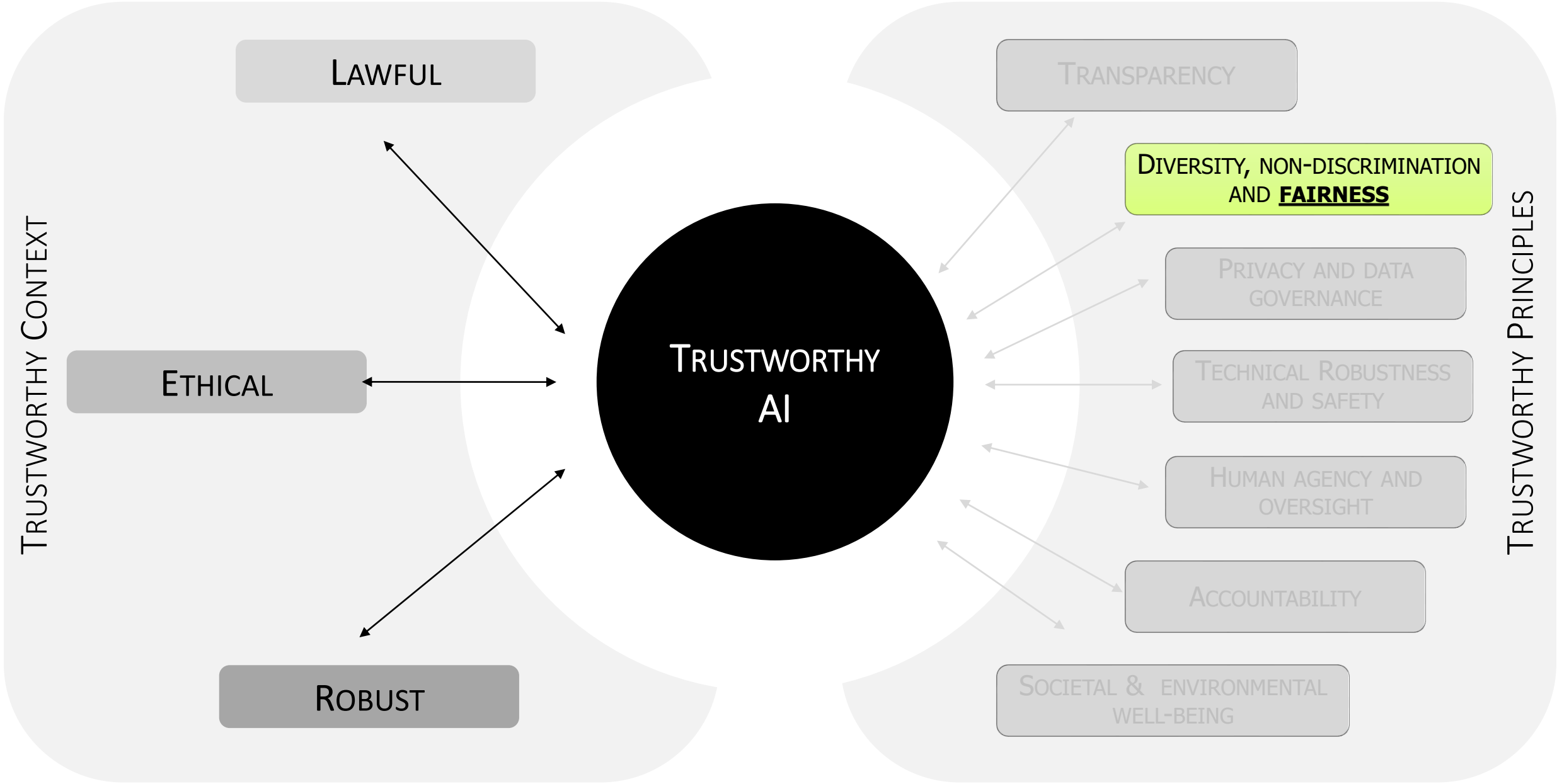
- **Explanations are often required by law and are necessary to ensure fairness and accountability.**

- **AI AUDIT**

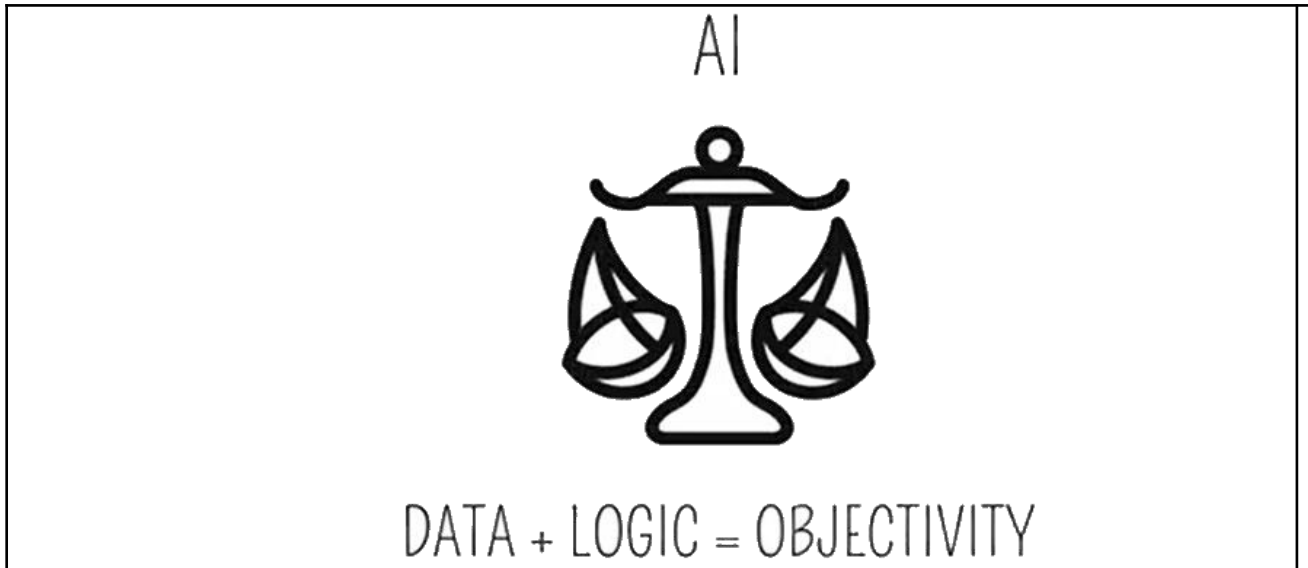
# Transparency (4)

| | Yes | No |
|---|---|---|
| Did you ensure an explanation as to why the system took a certain choice resulting in a certain outcome that all users can understand? | | |
| Why was this particular system deployed in this specific area? | | |
| Did you establish mechanisms to inform (end-users on the reasons and criteria behind the Al system's outcomes? | | |

# EU's Ethics Guidelines for Trustworthy AI



# Diversity, non-discrimination, and fairness (1)

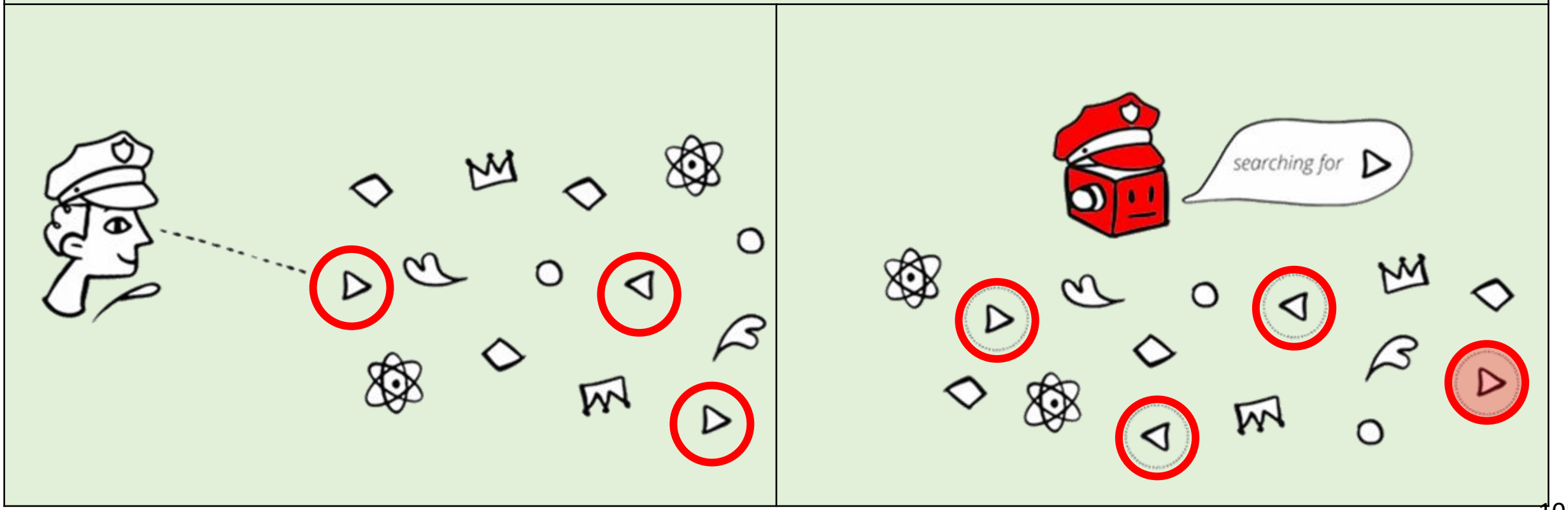



- We usually, tend to think that the AI-systems are **objective** and unbiased as they are based **on data and logic**
- In reality, it is impossible to have completely unbiased data.


- When AI systems are trained on historic data, they tend to perpetuate any biases present in that data.

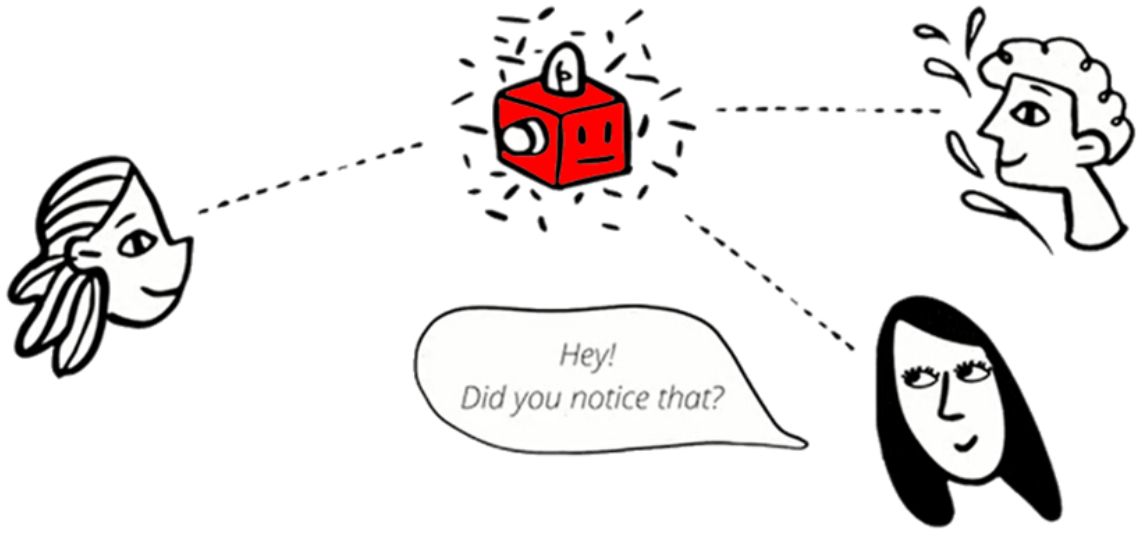
# Diversity, non-discrimination, and fairness (2)

- For example, if the police use historic data for predictive policing, the outcomes will likely be a repetition of the past and create a feedback loop where the same groups are targeted over and over.
- In this process, any biases and discriminatory practices from the past will be repeated, amplified, and systematized towards these groups.
- Groups that are targeted could be targeted not because of an actual suspicion but merely because of a characteristic they happen to share with past criminals.

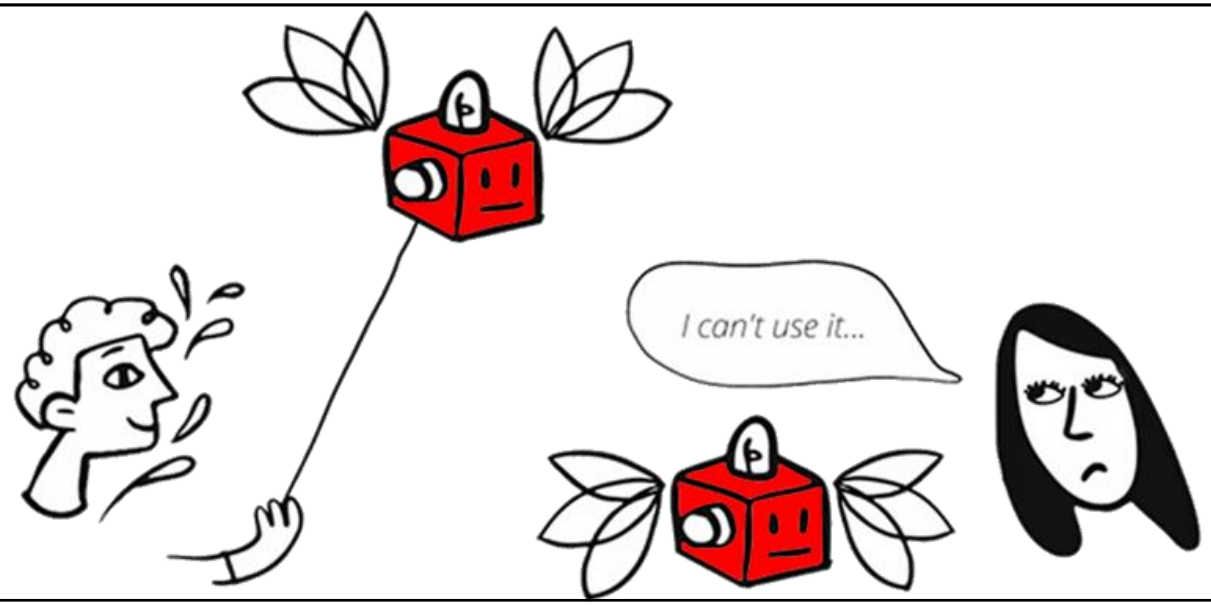


# Diversity, non-discrimination, and fairness (3)

- To address these risks, it is crucial to have an oversight system in place that monitors the purposes, constraints, requirements, and decisions of AI systems.
- Engaging people from diverse backgrounds and disciplines in the development and oversight processes is essential to ensure fairness.
- Moreover, It is important to consult with stakeholders who may be directly or indirectly affected by the system throughout its life cycle.



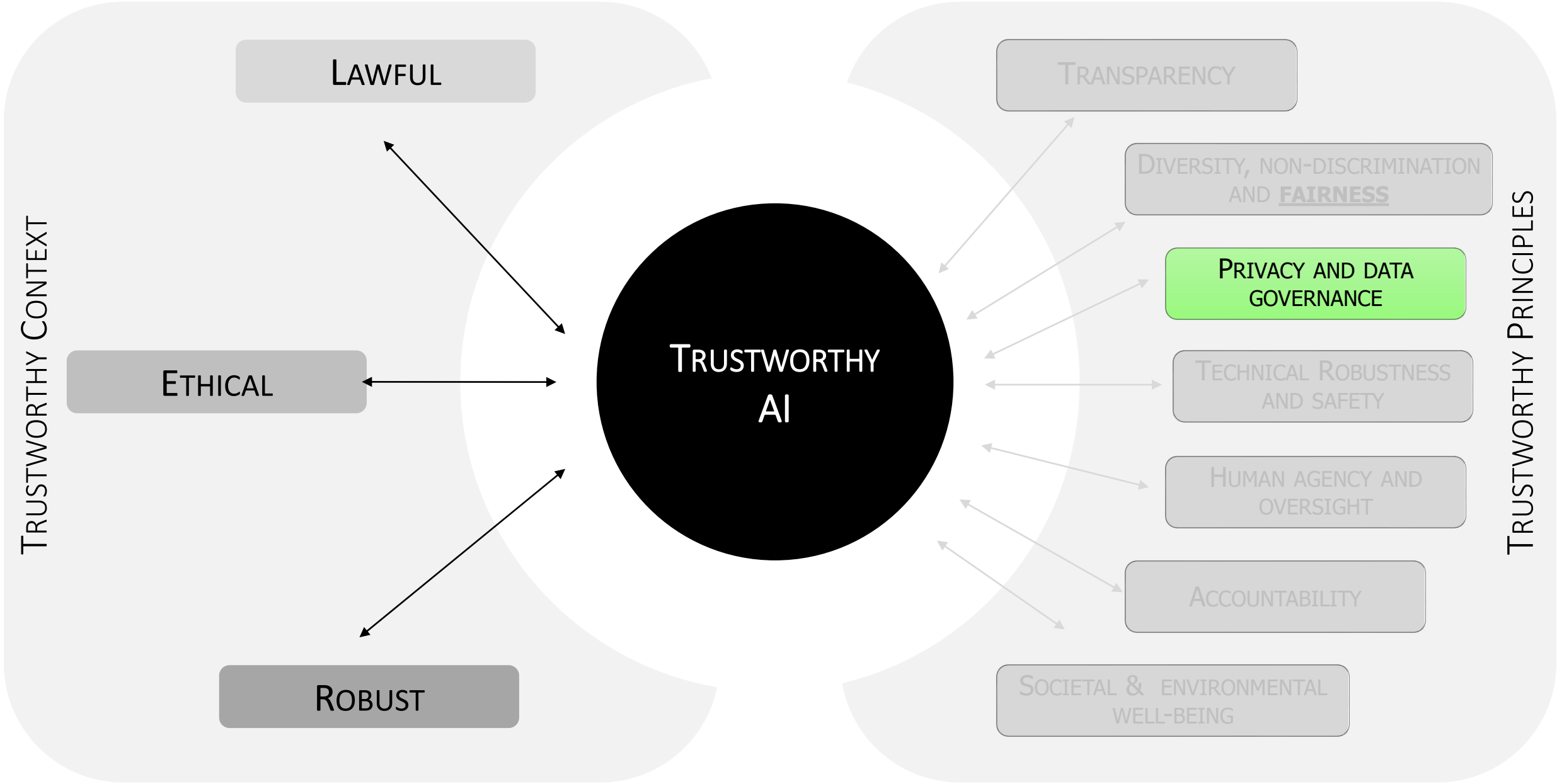

- In addition, AI systems should be designed to be accessible to all individuals, regardless of age, gender, abilities, or characteristics.
- **By considering the widest possible range of users, AI systems can be more <u>inclusive</u> and <u>equitable</u>.**



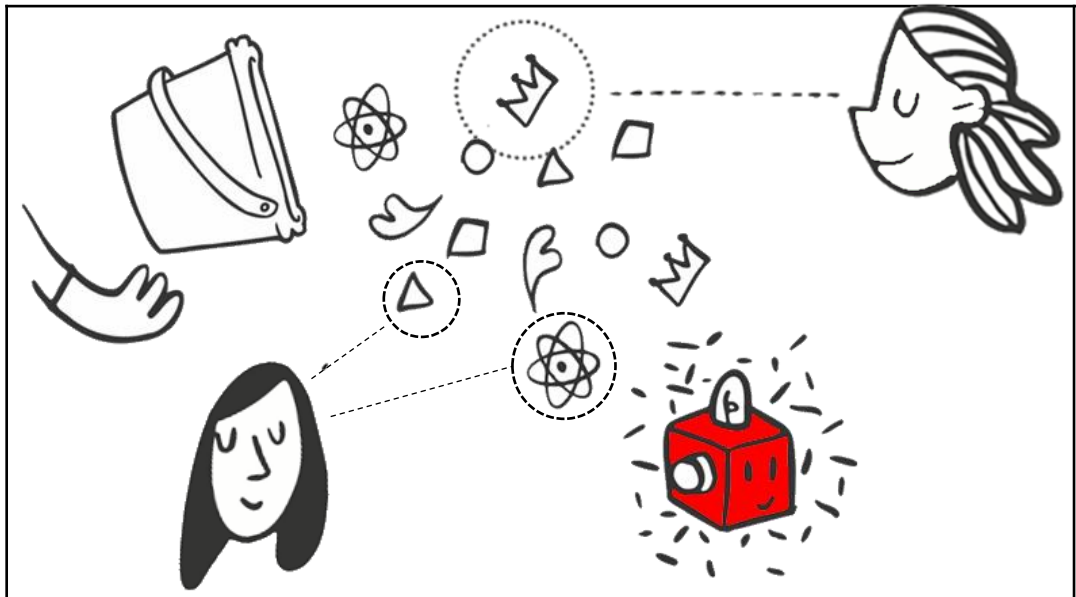
# Diversity, non-discrimination, and fairness (5)

| | Yes | No |
|---|---|---|
| Did you assess and acknowledge the possible limitations stemming from the composition of the used data sets? | | |
| Did you assess whether the Al system usable by those with special needs or disabilities or those at risk of exclusion? How was this designed into the system and how is it verified? | | |
| Did you assess whether there could be persons or groups who might be disproportionately affected by negative implications? | | |
| Did you consider a mechanism to include the participation of different stakeholders in the Al system's development and use? | | |

# EU's Ethics Guidelines for Trustworthy AI

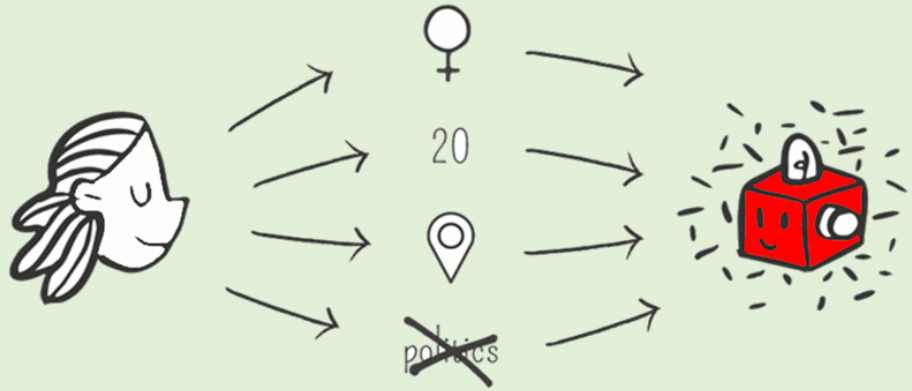


# Privacy and data governance (1)

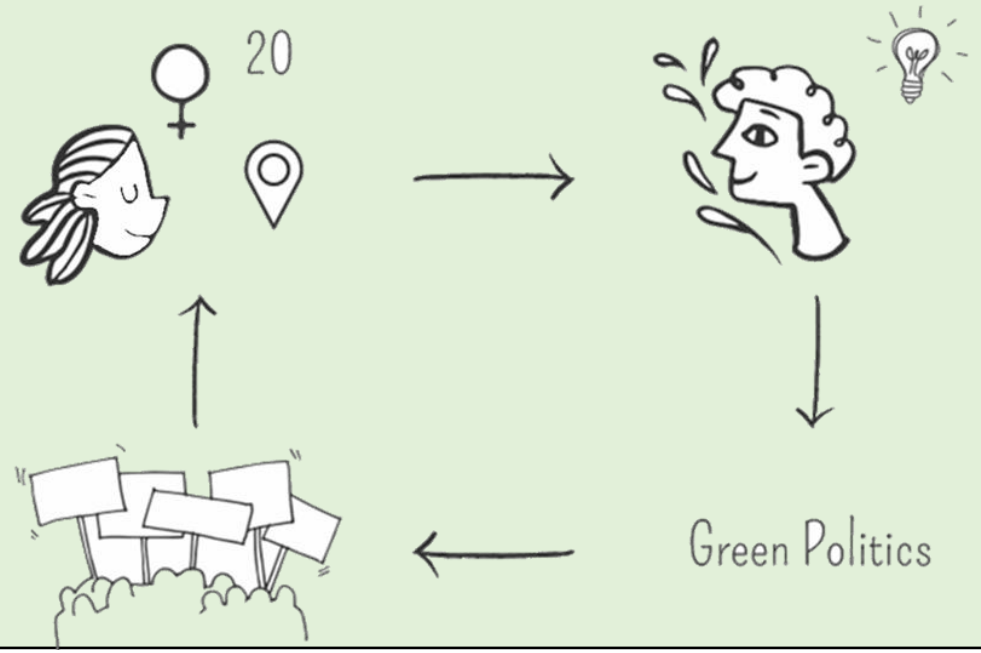


- Data is the fuel of a successful AI system.
- However, we should never forget that as abstract as it seems, data is information about *very real* people, and not taking good care of data has *very real* effects on individuals.
- Therefore, it is essential for deployers of AI systems to adhere to crucial principles regarding privacy and data governance.

# Privacy and data governance (2)

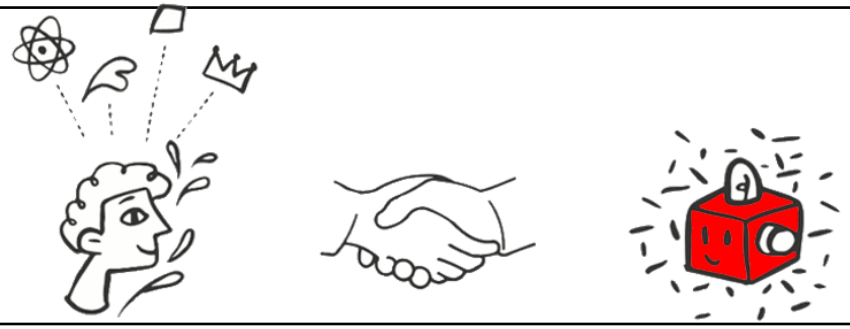



- Consider a situation where Diana, has consented and entrusted her location data, age, gender, and online behavior tracking to an online service provider.
- However, she does not want her data on political opinions to be used or sold to anyone.




- Unfortunately, the service provider combines the data that Diana has consented to and discovers that she is a 20-year-old woman from Paris who loves vegetarian food.
- They sell this data to a third party who deduces that Diana might be interested in green politics and begins targeting her to join a local organization fighting climate change.
- **This example demonstrates a misuse of personal data and a violation of the data subject's trust.**

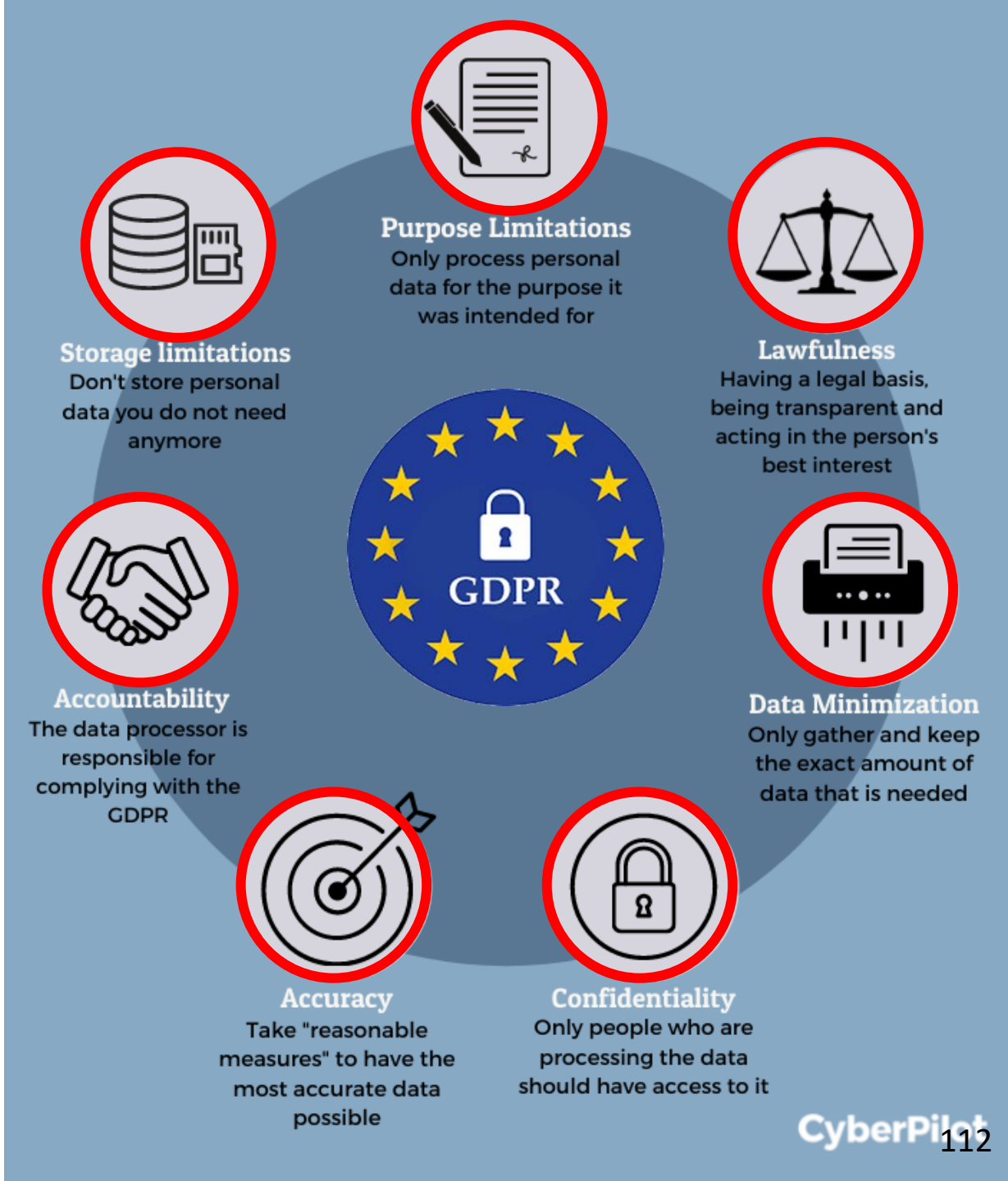
# Privacy and data governance (3)



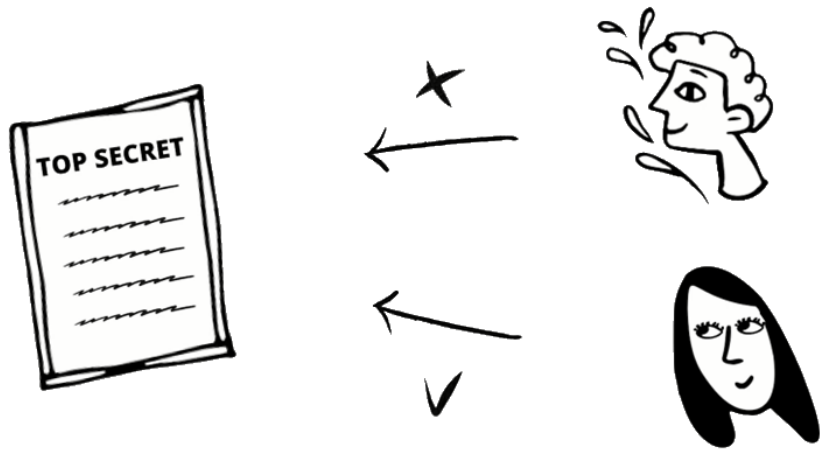
- To ensure privacy and data protection, it is crucial to consciously control and lawfully use the information provided by users or generated about users during their interactions with AI systems.
- The General Data Protection Regulation (GDPR)

# General Data Protection Regulation (GDPR)

- The GDPR (General Data Protection Regulation) is a comprehensive data protection regulation that was implemented by the European Union on May 25, 2018.
- The GDPR introduces strict requirements (7 principles) for organizations handling personal data.
- Non-compliance with the GDPR can result in significant fines and penalties.



# Privacy and data governance (4)

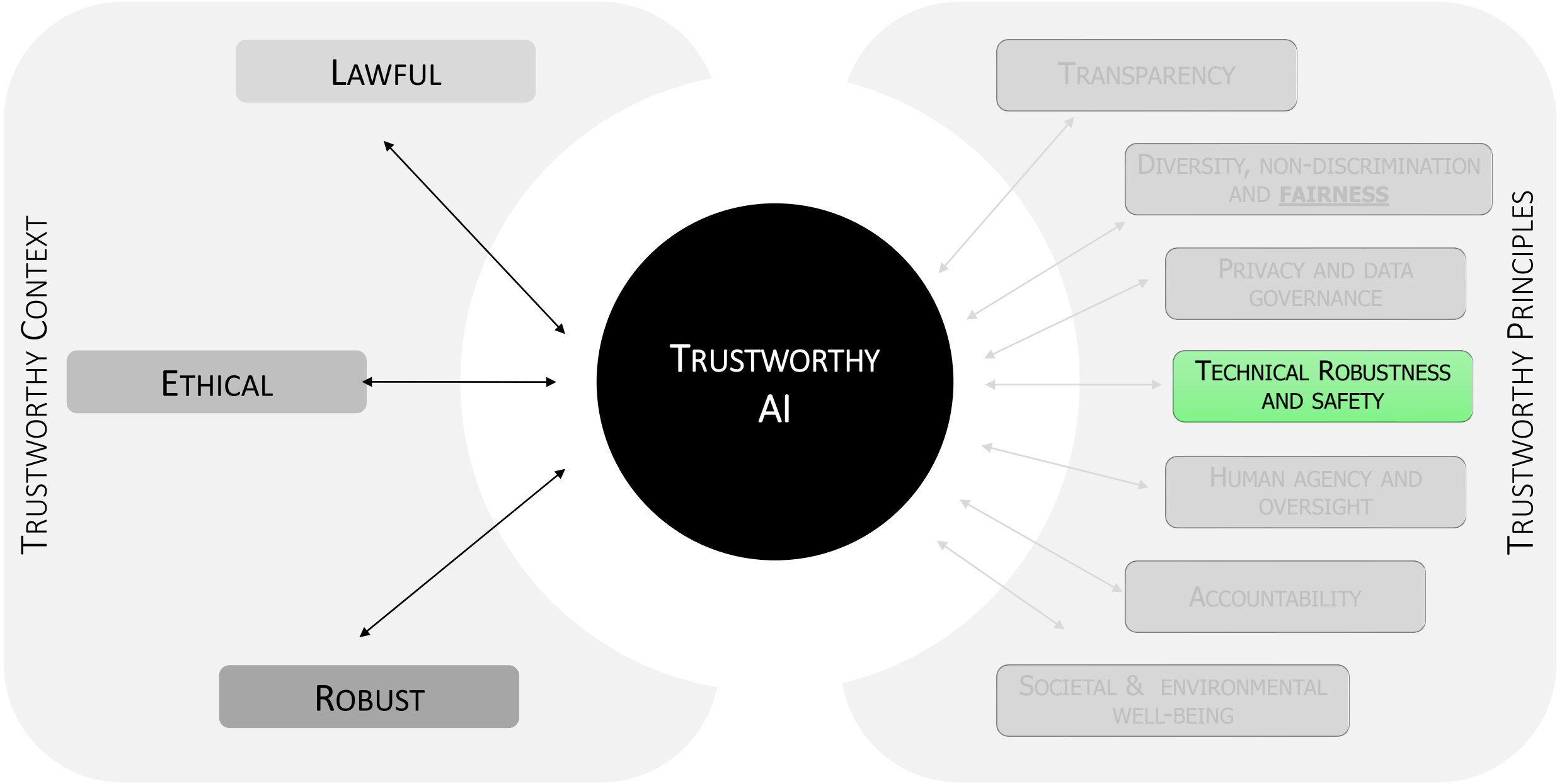



- Solid data protocols should govern data access. These protocols should clearly outline who can access the data and under what circumstances. Only qualified personnel with the necessary competence and a legitimate need to access an individual's data should be granted permission to do so.
- These protocols help ensure that data is handled responsibly and that privacy is protected.

# Privacy and data governance (4)

| | Yes | No |
|---|---|---|
| Did you consider ways to develop the AI system or train the model without or with minimal use of potentially sensitive or personal data? | | |
| Did you take measures to enhance privacy, such as encryption, anonymisation and aggregation? | | |
| Did you establish oversight mechanisms for data collection, storage, processing and use? | | |
| Did you ensure that people working with data are qualified and required to access the data, and that they have the necessary competences to understand the details of data protection policy? | | |
| Did you ensure an oversight mechanism to log when, where, how, by whom, and for what purpose data was accessed? | | |

# Ethics Guidelines for Trustworthy AI



# Technical robustness and safety(1)

- Technical robustness entails developing AI systems with a **preventative** approach to minimize harm.
- This involves considering various components and principles during the development phase to ensure that AI systems behave reliably and as intended in different environments.

1

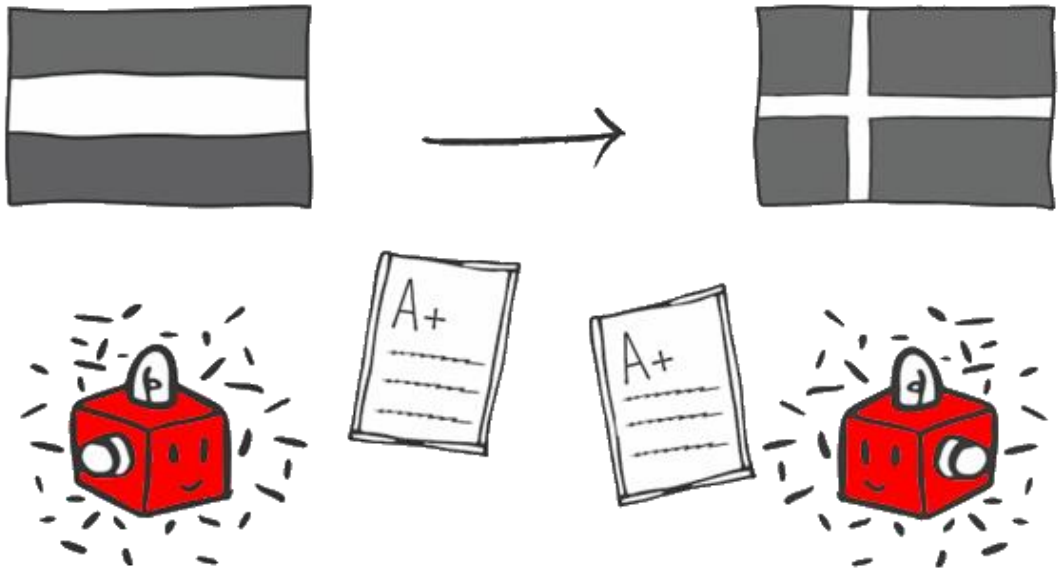
- Firstly, AI systems should be protected against external attacks and have safeguards in place to enable a fallback plan in case of problems or failures

# Technical robustness and safety(2)

2

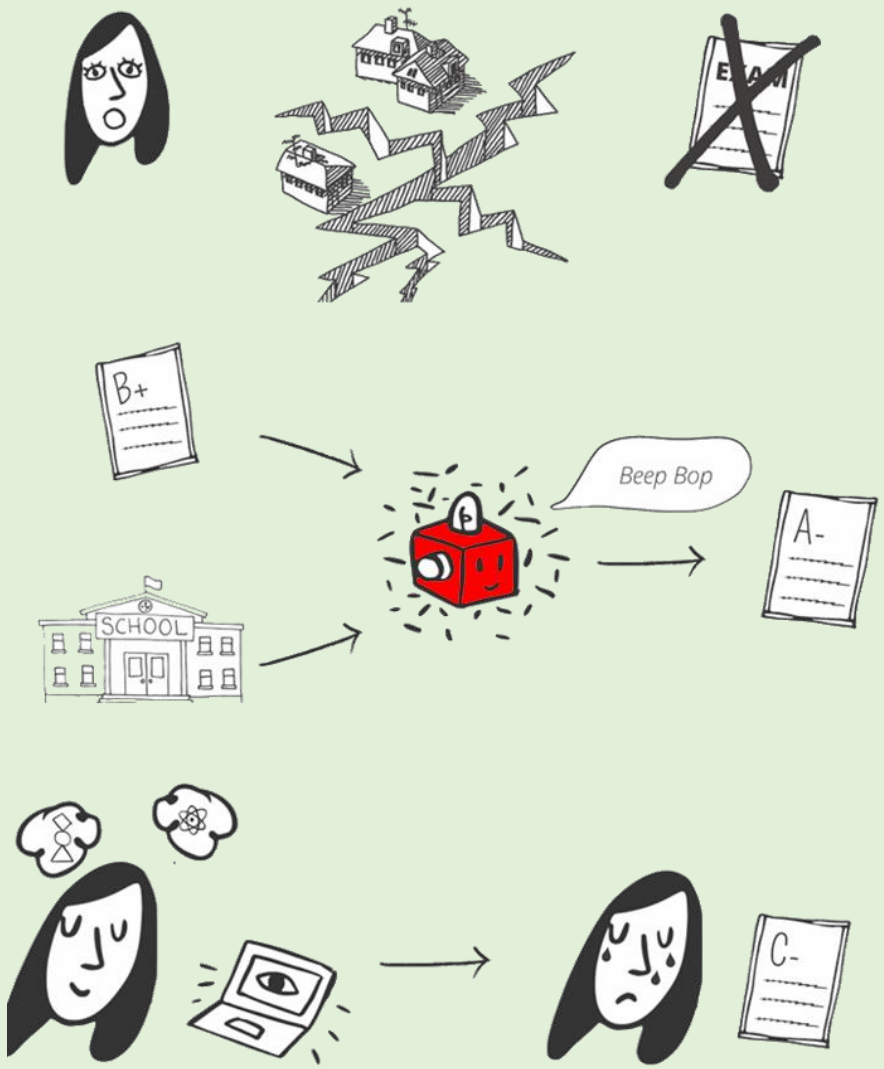

- Secondly, AI systems should focus on **accuracy**.
- They should be capable of making correct judgments, predictions, recommendations, and decisions as they were designed to do.

3




- Thirdly, AI systems should be **reproducible**, meaning that when provided with different data inputs and contexts, they should produce similar results.
- Reproducibility ensures that the system's performance remains consistent and reliable across different scenarios.

# Technical robustness and safety(3)

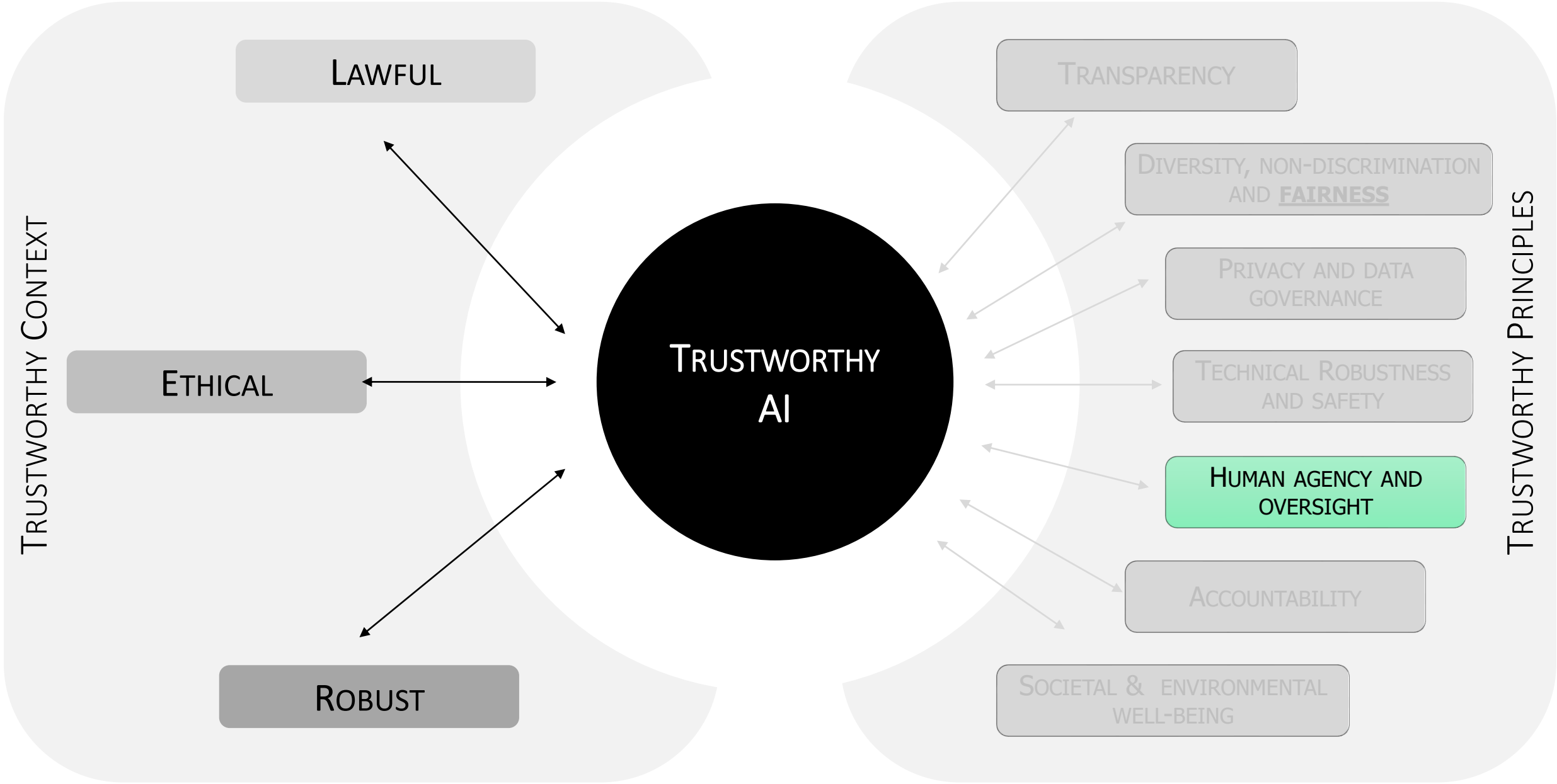



- For example, consider a situation where a high school student named Lizzie is unable to take her final exams due to an earthquake.
- The Ministry of Education decides to use AI to predict the grades Lizzie would have received in the exams based on her past performance and the school's past performance.
- The AI system is tested using these variables and provides good and accurate results.
- However, Lizzie, who has performed well individually but attends an average school, receives significantly lower grades based on the system's variables. This flaw in the system could unjustly deprive Lizzie of access to higher education.
- To prevent such scenarios, developers and deployers of AI systems should ask themselves critical questions during the development phase.
- **These questions help ensure technical robustness and safety and address potential biases, errors, or unintended consequences that may arise**.

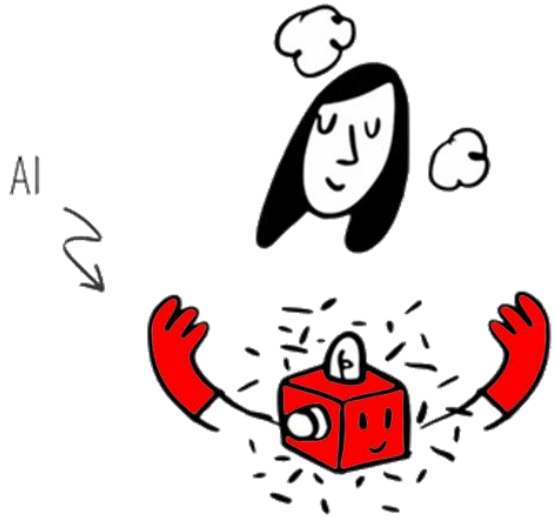
# Technical robustness and safety (4)

| | Yes | No |
|---|---|---|
| Did you verify how your system behaves in unexpected situations and environments? | | |
| Did you ensure that your system has a sufficient fallback plan in the case of adversarial attacks or other unexpected situations? | | |
| Did you assess whether there is a probable chance that the AI system may cause damage or harm to users or third parties? Did you assess the likelihood, potential damage, impacted audience and severity? | | |
| Did you assess what level and definition of accuracy would be required in the context of the AI-system and use case? | | |
| Did you verify what harm would be caused if the AI-system makes inaccurate predictions? | | |

# EU's Ethics Guidelines for Trustworthy AI

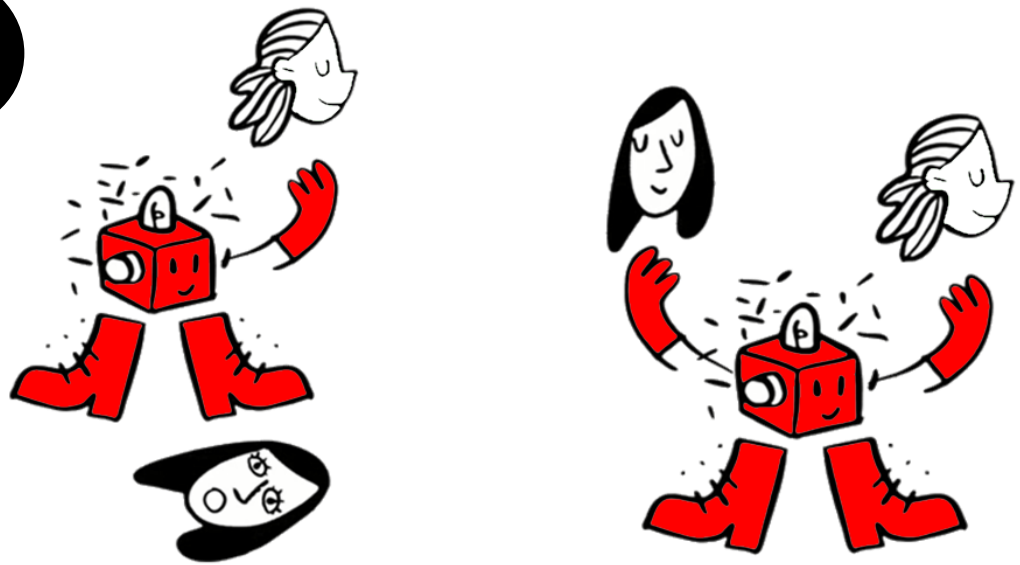


# Human agency and oversight (1)




- AI systems should foster human autonomy and decision-making in three ways.

1



- Firstly, developers should acknowledge and prevent risks that AI systems may pose to fundamental rights.
- They should conduct a human rights impact assessment before deploying AI systems to ensure that they do not violate individuals' rights.

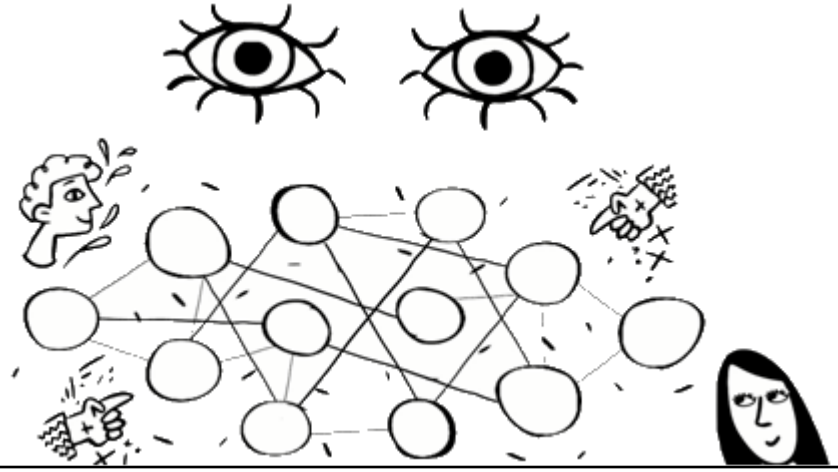
# Human agency and oversight (2)

**2**

- Secondly, individuals should be provided with reasonable tools to understand and interact with AI systems.
- This support helps them make informed choices aligned with their goals and values.

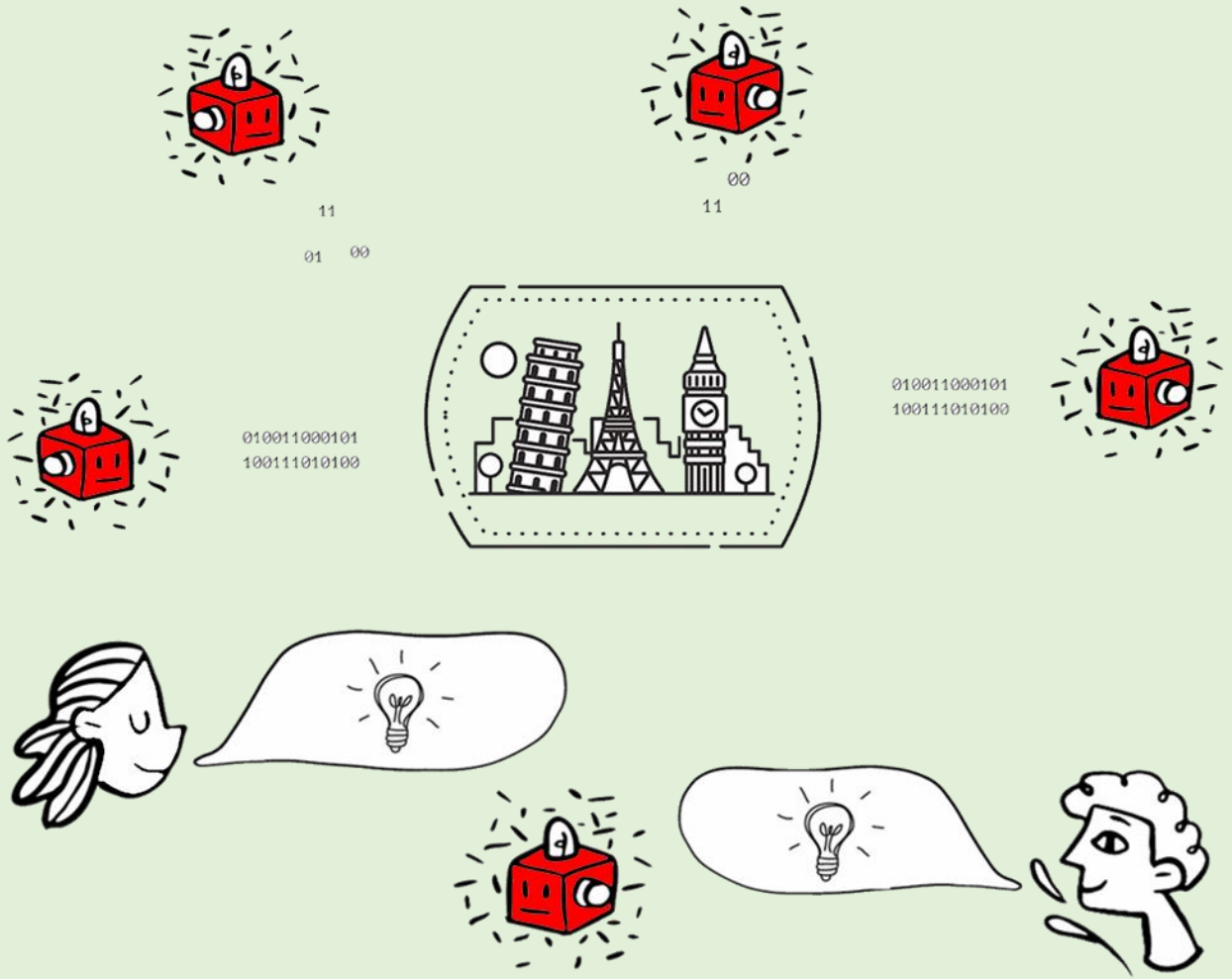


**3**



- Thirdly, to enhance human autonomy, there should be appropriate human oversight over AI systems.
- This means that an actual person should be able to supervise, intervene, or correct automated decisions made by AI systems.

# Human agency and oversight (3)

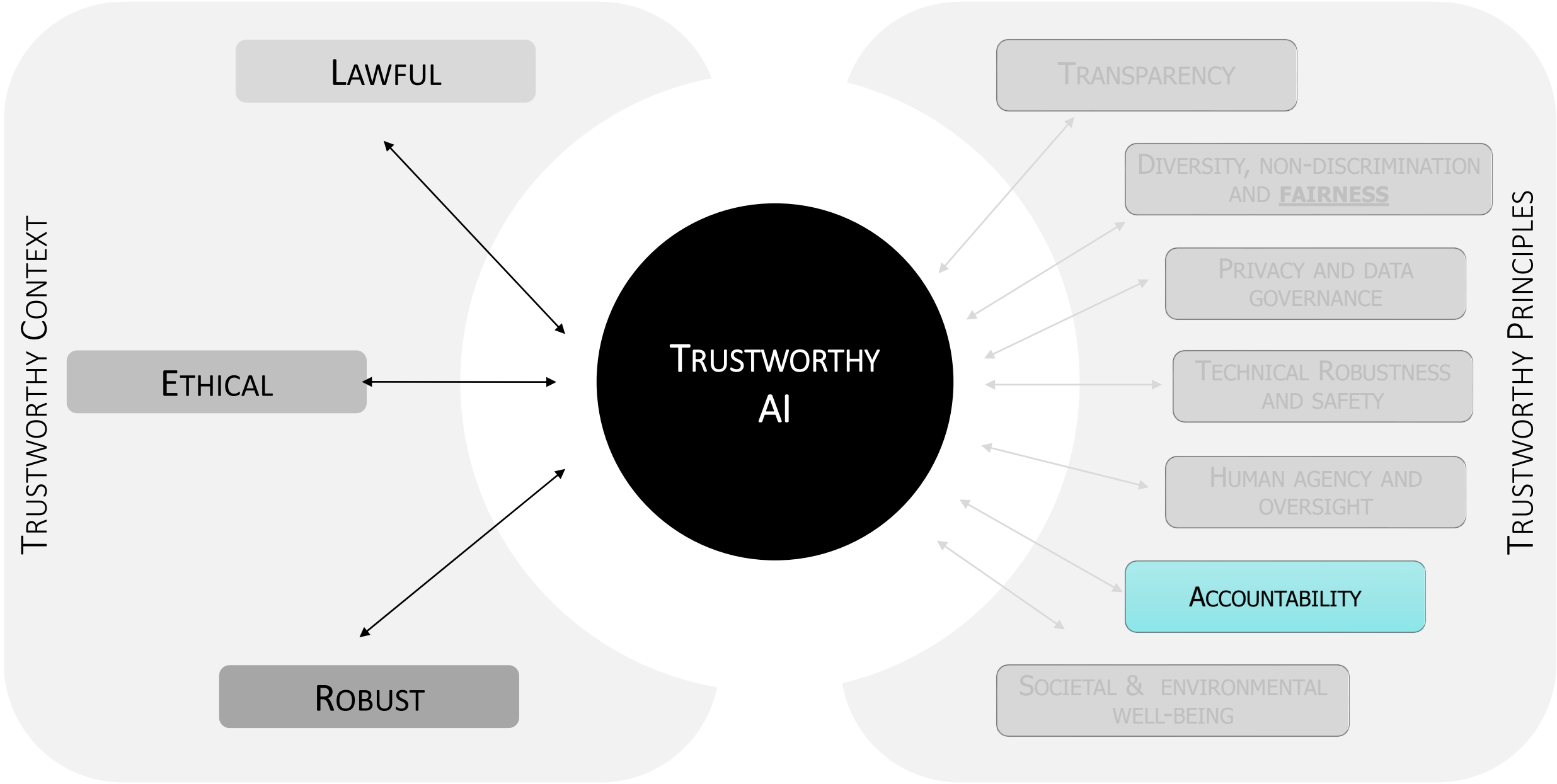



- For example, in smart city projects that employ AI and big data, the goal is to make the city more knowable and controllable, improving the delivery of public services.
- However, such projects can impact the autonomy of citizens if they lead to increased surveillance.
- This may result in a chilling effect, where individuals modify their behavior due to constant monitoring.
- AI that respects human agency would instead empower citizens: For instance, AI could be utilized to involve citizens in policy-making processes or to enable co-designing, aligning services with their needs and desires.

# Human agency and oversight (4)

| | Yes | No |
|---|---|---|
| Did you consider ways to develop the AI system or train the model without or with minimal use of potentially sensitive or personal data? | | |
| Did you take measures to enhance privacy, such as encryption, anonymisation and aggregation? | | |
| Did you establish oversight mechanisms for data collection, storage, processing and use? | | |
| Did you ensure that people working with data are qualified and required to access the data, and that they have the necessary competences to understand the details of data protection policy? | | |
| Did you ensure an oversight mechanism to log when, where, how, by whom, and for what purpose data was accessed? | | |

# EU's Ethics Guidelines for Trustworthy AI

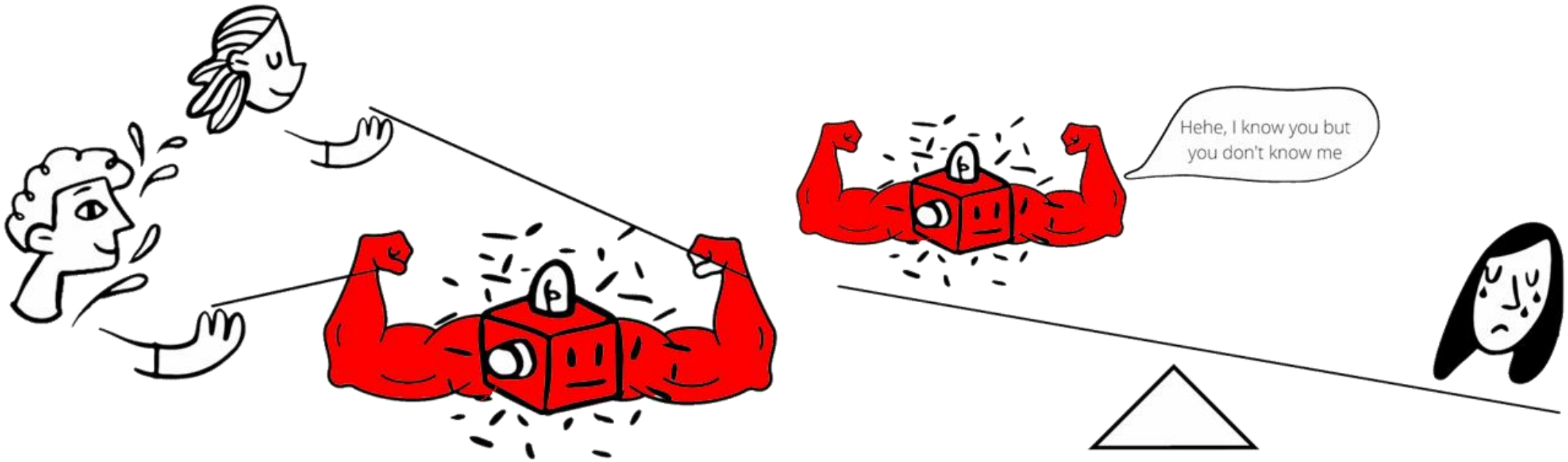


# Accountability (1)

- Umbrella term, covering amongst others:
  1. legal accountability ;
  2. professional accountability;
  3. political accountability;
  4. administrative accountability; and
  5. social accountability.
- Used not only to indicate punishment but also acceptance of responsibility.
- Incidents will occur —and sometimes reoccur : It is essential for **maintaining the public's trust** in the technology.

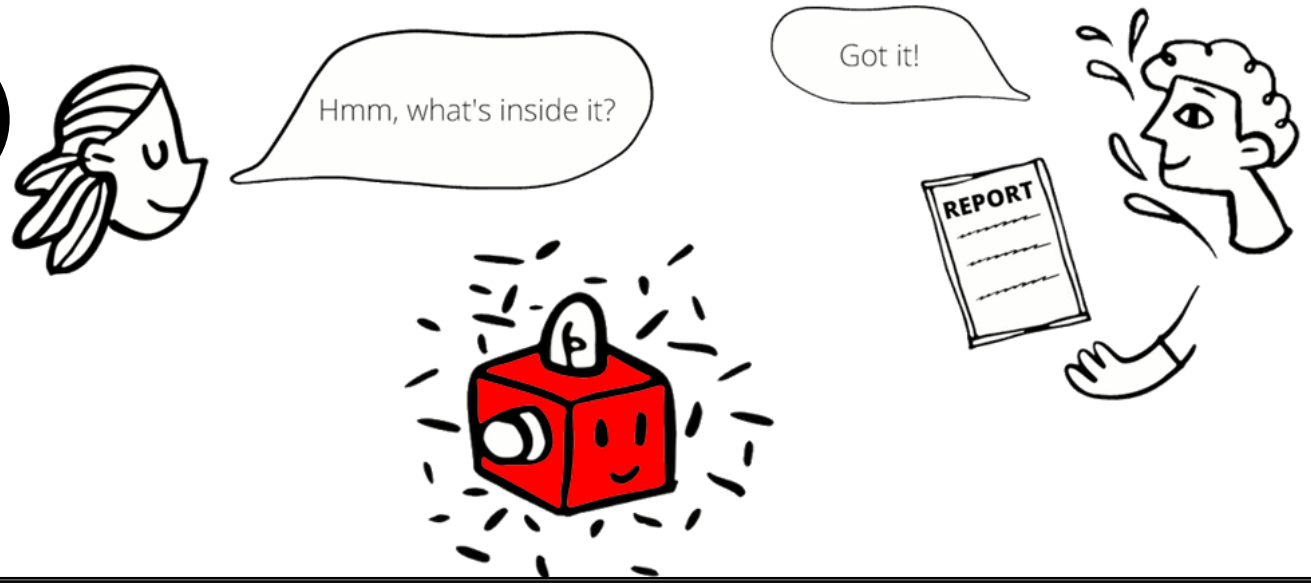
# Accountability (2)

It is crucial to understand that those who wield power through AI systems must take responsibility for their actions and be held accountable.



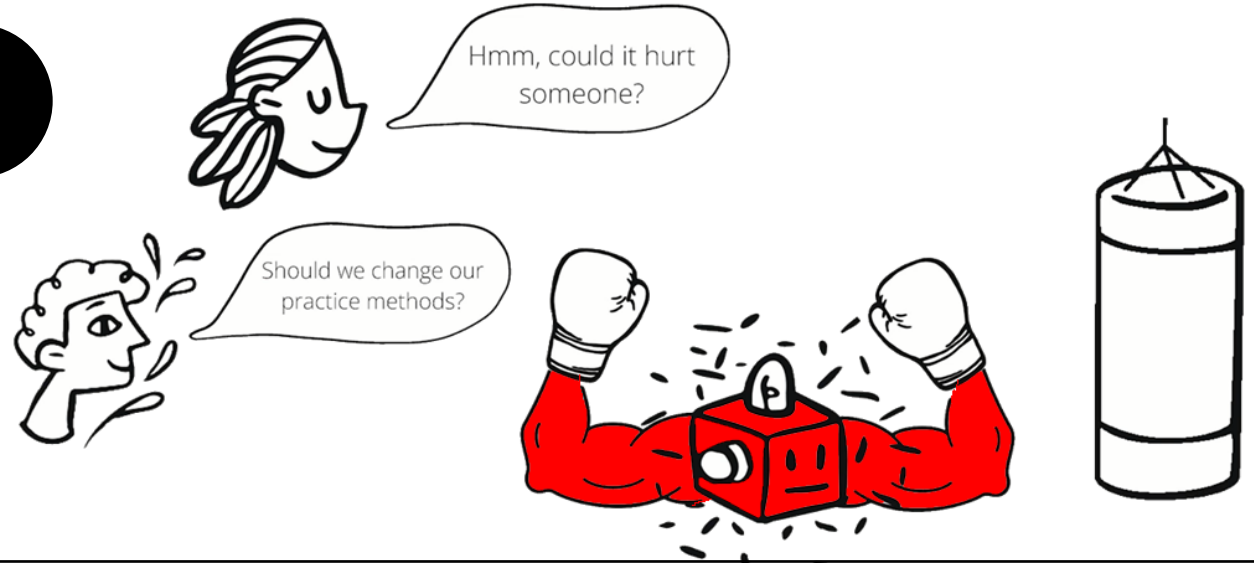
# Accountability (3)

1



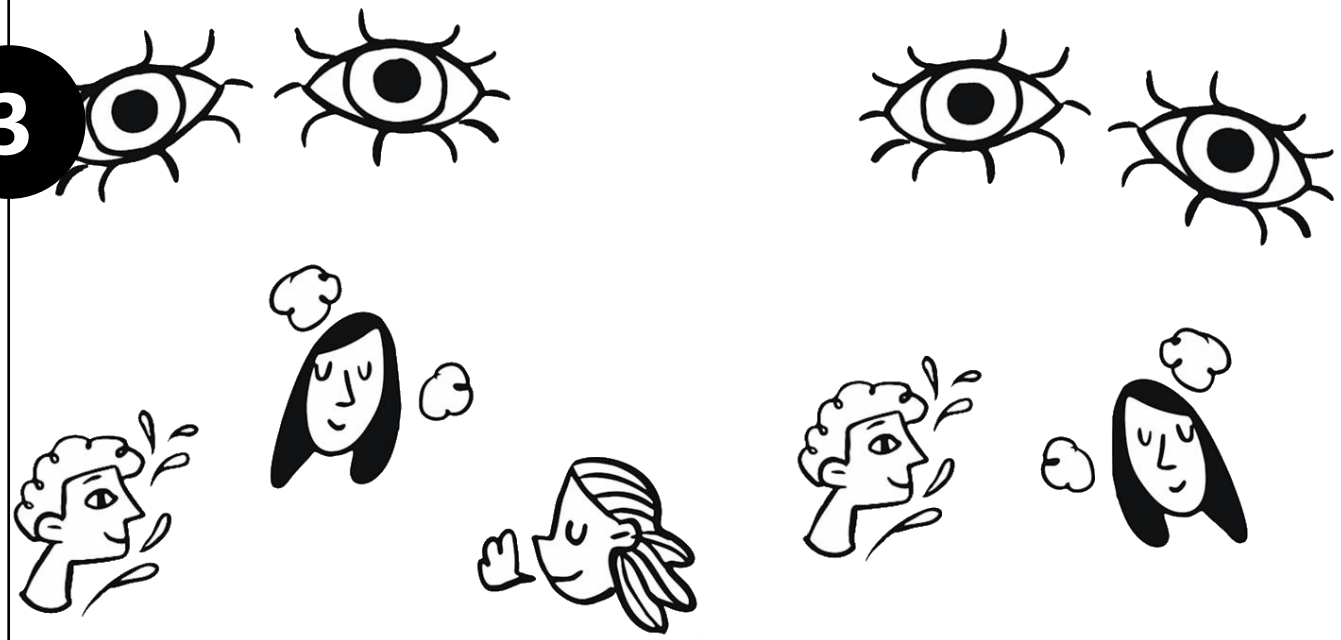

- The first step towards accountability is auditability.
- By allowing internal and independent external auditors to evaluate AI systems and report on their outcomes, transparency is increased, and information asymmetries between deployers and other stakeholders are reduced.

2




- The second step involves minimizing and reporting negative impacts.
- It is important to identify, assess, and document potential negative impacts before, during, and after the development, deployment, and use of AI systems.
- This **proactive** approach helps minimize harm and ensures that accountability is upheld.

# Accountability (4)

| | |
|---|---|
| 3  | • Third, ethical decisions can involve trade-offs between different ethical principles.<br>• For example, many of the AI systems introduced during the COVID-19 pandemic aimed to ensure public health thus supporting the principle of societal well-being however these ai systems compromised people's privacy<br>• It is essential to acknowledge, evaluate, and document these trade-offs to maintain transparency and accountability. |
|  | • Furthermore, if no ethically acceptable trade-offs can be identified, it may be necessary to reconsider the use of the AI system altogether.<br>• This reflects the importance of maintaining ethical standards and prioritizing societal well-being. |

# Accountability (5)

**Accountability vs Responsibility**

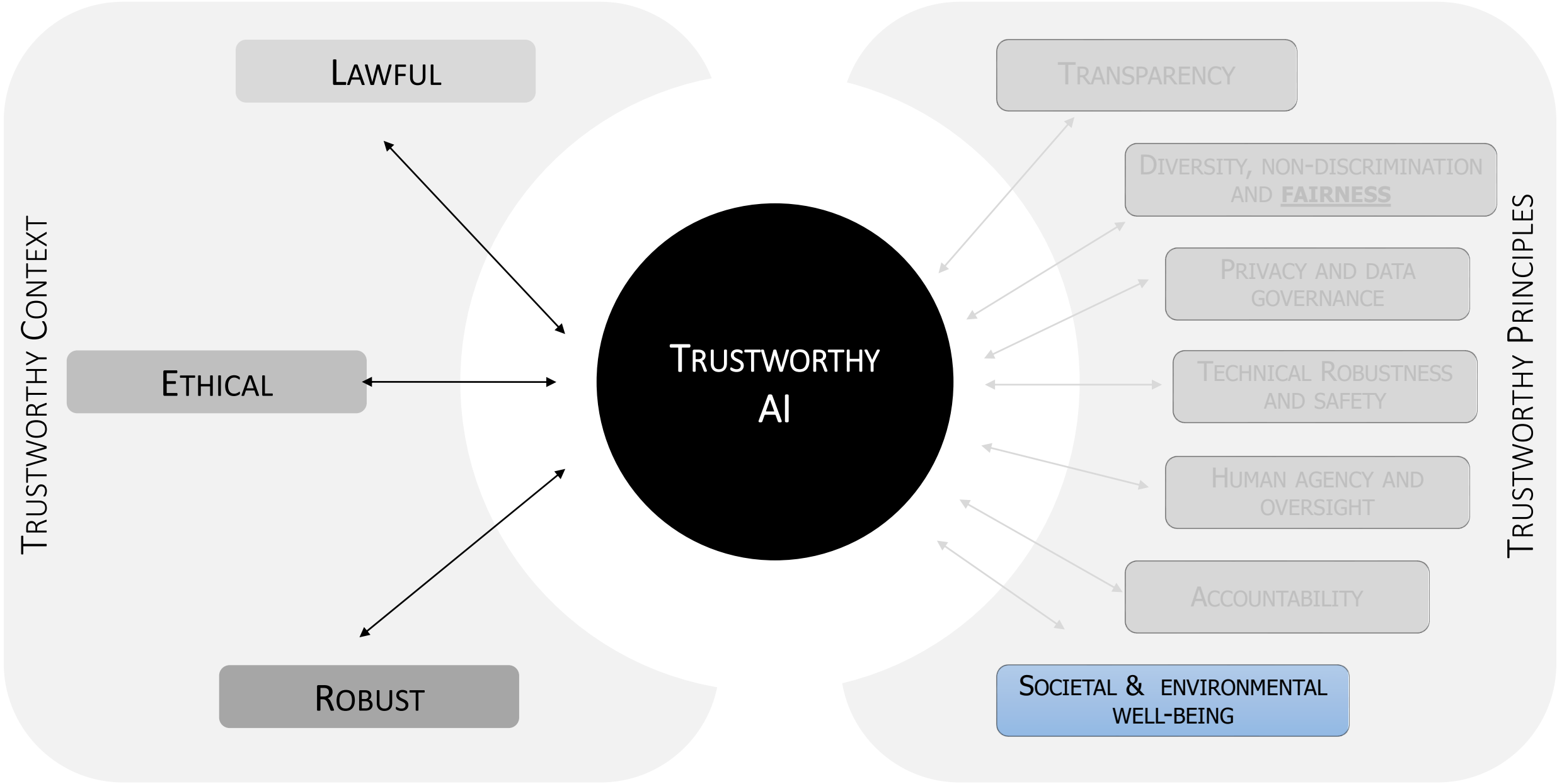
- Accountability is ***backwards thinking,*** provides an account of events after they have occurred.
- Responsibility is ***forwards thinking***, i.e., acting to deter incidents and violations of our ethical and legal values from occurring.

To be held accountable, you need to be held responsible…

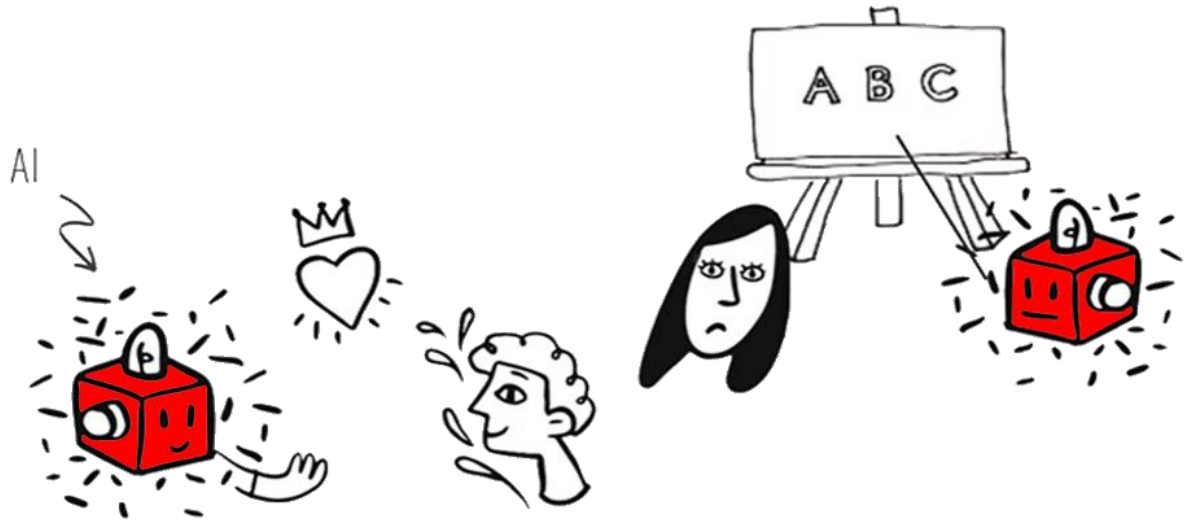
# Accountability (6)

| | Yes | No |
|---|---|---|
| Did you establish mechanisms that facilitate the system's auditability, such as ensuring traceability and logging of the AI-system's processes and outcomes? | | |
| Did you carry out a risk or impact assessment of the AI-system, which takes into account different stakeholders that are (in)directly affected? | | |
| How do you decide on trade-offs between ethical principles? Did you ensure that the trade-off decision was documented? | | |
| Did you establish an adequate set of mechanisms that allows for redress in the case of the occurance of any harm or adverse impact? | | |

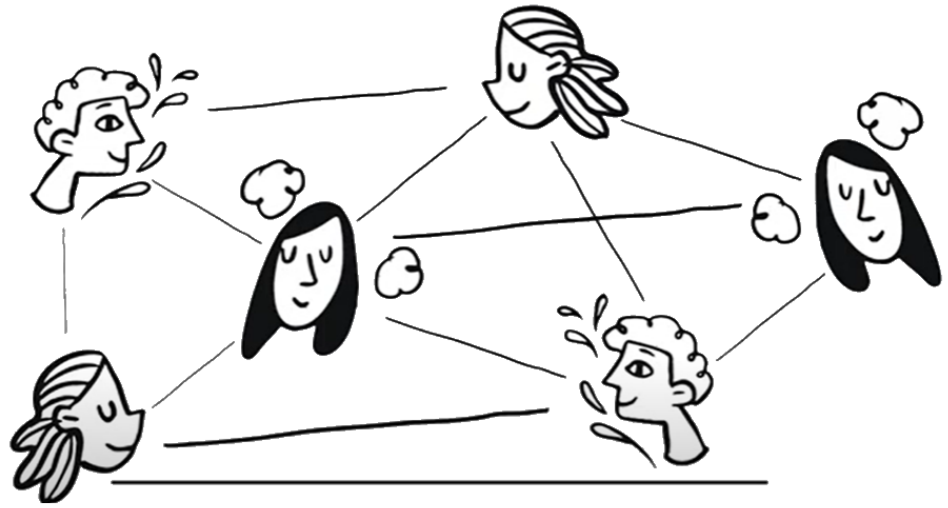
# EU's Ethics Guidelines for Trustworthy AI



# Societal and environmental well-being (1)



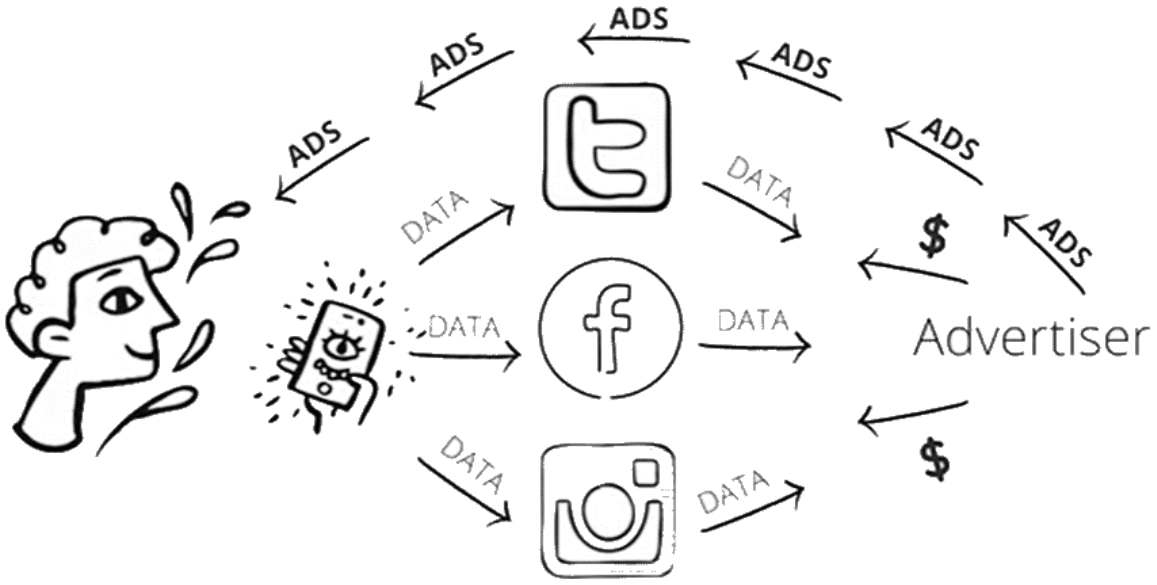

- As technology, including AI, becomes increasingly pervasive in our lives, it is important to consider the broader impacts it has on our well-being and society as a whole.

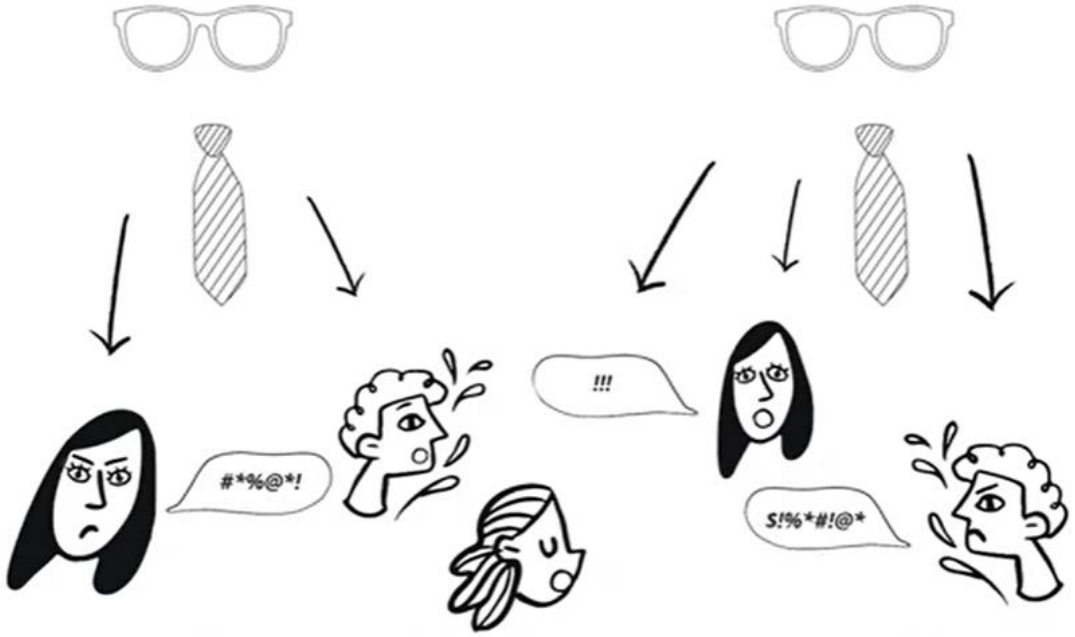


- AI can have both positive and negative effects on our physical and mental well-being, social agency, social skills, and relationships.
- Therefore, it is essential to carefully consider and monitor the effects of AI systems from a societal perspective.
- This includes assessing their impact on individuals, institutions, the rule of law, democracy, and society at large.

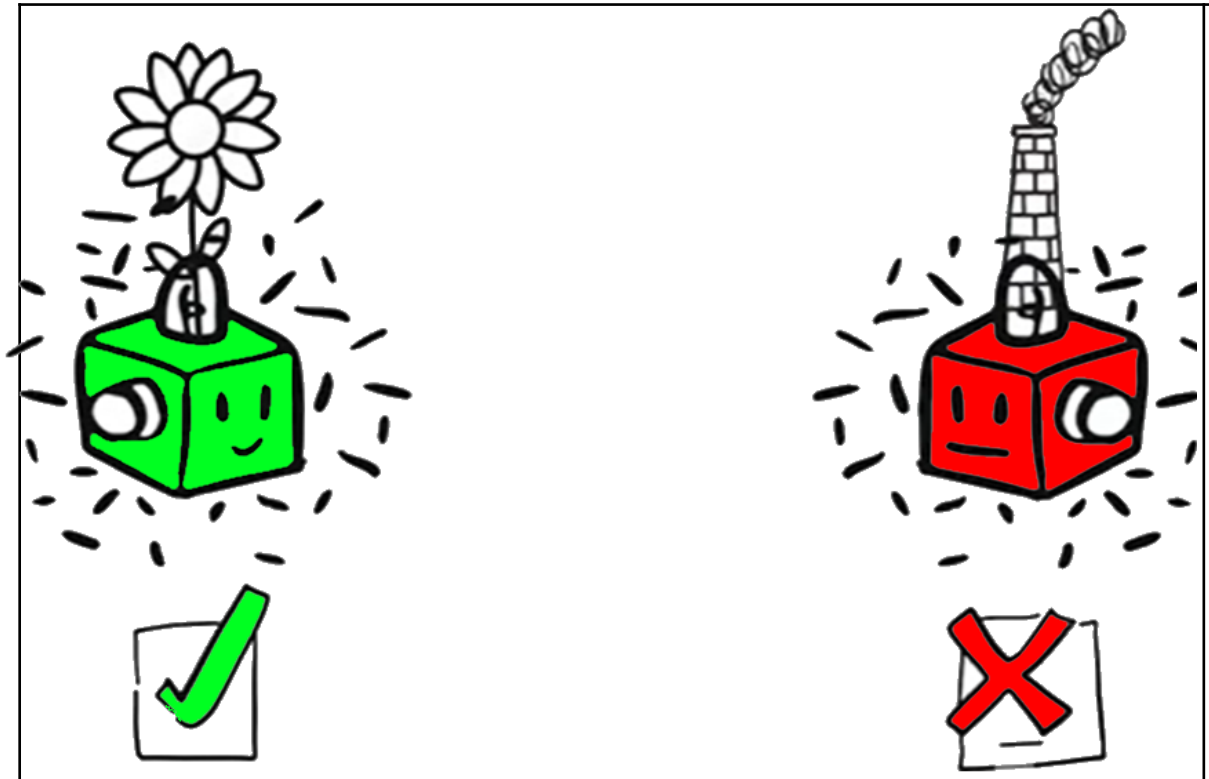
# Societal and environmental well-being (2)

- For example, the algorithms and recommender systems used by social media platforms have a significant impact on the information and suggestions we receive.
- However, these platforms are driven by a business model that aims to capture and maintain users' attention, leading to potential consequences such as data profiling and targeting for commercial purposes.
- When this design logic is applied to sell political ideas or manipulate citizens, it can distort and polarize our democracy.



# Societal and environmental well-being (3)



- Furthermore, it is important to recognize that AI systems have an environmental footprint.
- The training process for AI systems often requires significant amounts of computing power, which consumes a considerable amount of energy.
- Developers and deployers of AI systems should critically assess the environmental impact of their systems and strive to make choices that minimize harm to the environment.

# Societal and environmental well-being (4)

| | Yes | No |
|---|---|---|
| Did you assess whether the AI-system encourages humans to develop attachment and empathy or vice versa? | | |
| Did you assess whether the logic of the AI-system might simplify and polarize public discourse? | | |
| Did you assess whether the AI-system could be used to manipulate or confuse people? | | |
| Did you establish mechanisms to measure the environmental impact of the AI-system's development, deployment and use (for example the type of energy used by the data centers)? | | |
| Did you ensure measures to reduce the environmental impact of your AI-system's life cycle? | | |

# To do: Ethical Implications and Trustworthy Use of AI

As a team of 3-4 students, your task is to choose one of the following three situations to work on (each situation should be discussed by a unique group). Your objective is to imagine and analyze the ethical implications of using AI in the chosen situation, guided by the "EU's Ethics Guidelines for Trustworthy AI." Additionally, you should provide recommendations for ensuring the trustworthiness and ethics of AI use in the given situation.

- **Case study 1**: Digital Transformation of a Retail Company - High-end Fashion: Imagine a high-end fashion retail company that wants to embark on a digital transformation by integrating AI technologies to enhance its operations. Your group will explore the various potential applications of AI in this company, such as inventory optimization, personalized style recommendations, and improving the online customer experience. Analyze the ethical implications specific to this domain, such as responsibility in style recommendations, customer data protection, and fairness in access to fashion products.

- **Case study 2**: Digital Transformation of an Educational Institution – University : Imagine a university that aims to leverage AI to enhance the student experience and optimize learning processes. Your group will explore different ways AI could be integrated into this institution, such as offering personalized guidance to students and automating certain administrative tasks. Analyze the ethical implications specific to education, such as equity in access to education, student data privacy, and integrity of automated assessments.

- **Case study 3** : Digital Transformation of a Financial Institution - Bank or Investment Management Company: Imagine a financial institution, such as a bank or investment management company, looking to integrate AI to improve its processes, including loan and credit evaluation, fraud detection, and investment management. Your group will explore the various potential applications of AI in this financial institution and analyze the ethical implications specific to this domain, such as fairness in lending decisions, financial data confidentiality, and transparency of the algorithms used.

**Oral Presentation**:

- Each group is encouraged to provide a detailed description of the specific contexts within the chosen situation, such as the types of products involved, the characteristics of the target clients, the technical aspects of website development, or any other relevant details that can enhance the analysis and recommendations.
- Each group should prepare an oral presentation to share the findings of their analysis. The duration of each oral presentation should be approximately 15 minutes. In your presentation, please make sure to provide a comprehensive analysis of the ethical implications specific to your chosen situation and offer well-supported recommendations for ensuring the trustworthy use of AI.

# Best practices for Bias avoidance/mitigation

1. Practices related to Implementation, monitoring, and awareness
2. Practices related to data preparation and modeling

# Best practices for Bias avoidance/mitigation

## Practices related to Implementation, monitoring, and awareness

**Main Actors** :
Human resources managers, Project managers, Training teams, Customer support teams, External auditors

1. Consider team composition for diversity of thought, background, and experiences.
2. Understand the task, stakeholders, and potential for errors and harm.
3. **Post-Deployment**:
   a) Ensure optimization and guardrail metrics consistent with responsible practices and avoid harms.
   b) Continual monitoring, including customer feedback.
   c) Have a plan to identify and respond to failures and harms as they occur.
4. **Conduct external audits or third-party evaluations**: Seek external assessments or audits of the AI system to gain independent insights into potential biases and ensure compliance with ethical and legal standards.
5. **Foster ongoing education and awareness**: Promote awareness and understanding of AI biases among team members, stakeholders, and users. Encourage ongoing education and training on responsible AI practices and the potential impacts of biases.

# Best practices for Bias avoidance/mitigation

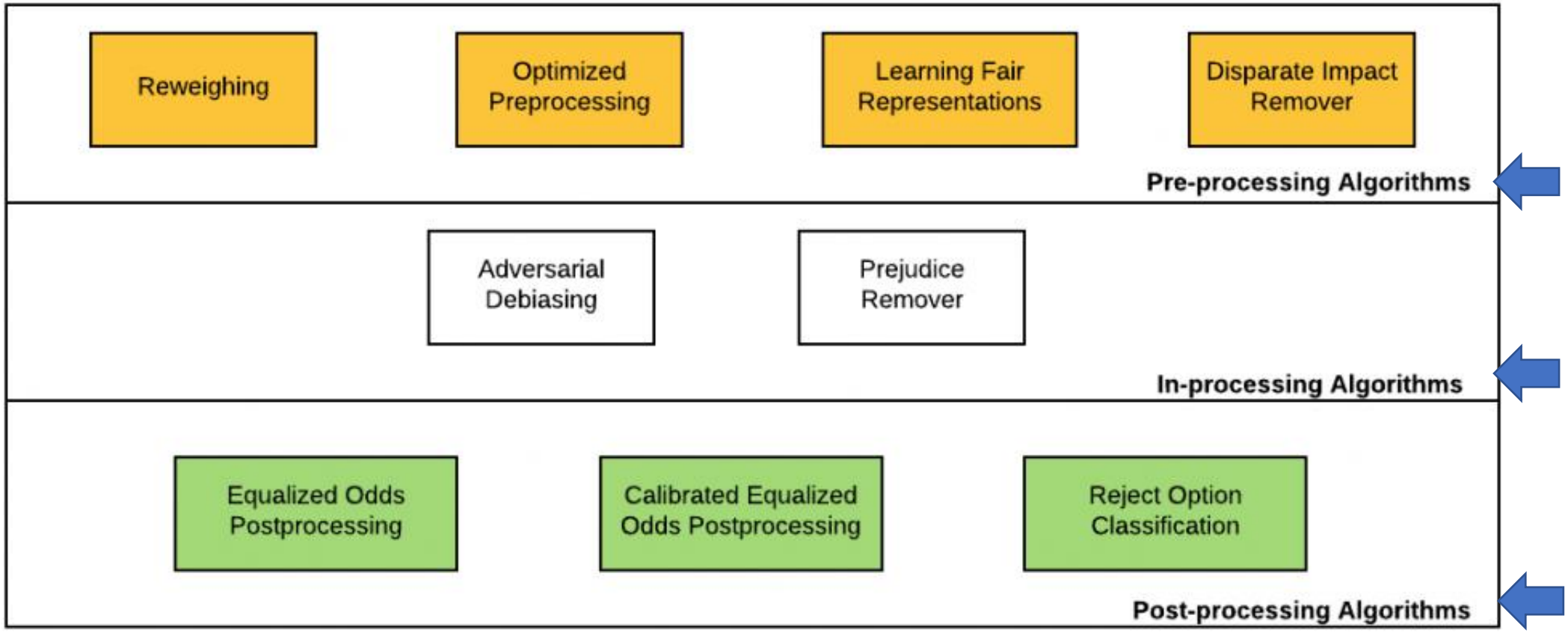
## Practices related to data preparation and modeling

**Main Actors** : Data scientists, Machine learning engineers, Domain experts

1. **Check data sets:**
    a) Consider data provenance.
    b) Verify the data using qualitative, experimental, survey, and other relevant methods.
2. **Apply Bias Mitigation Strategies for ML models**
    a) Pre-processing algorithms.
    b) In-processing algorithms.
    c) Post-processing algorithms.

# Bias Mitigation Strategies for ML models

Some practices related to data preparation and modeling :




https://dzone.com/articles/machine-learning-models-bias-mitigation-strategies
https://towardsdatascience.com/approaches-for-addressing-unfairness-in-machine-learning-a31f9807cf31

# Bias Mitigation Strategies for ML models

**Pre-Processing Algorithms**

- **Reweighing**: Reweighing is a data preprocessing technique that recommends generating weights for the training examples in each (group, label) combination differently to ensure fairness before classification. The idea is to apply appropriate weights to different tuples in the training dataset to make the training dataset discrimination free with respect to the sensitive attributes.
- **Optimized preprocessing**: The idea is to learn a probabilistic transformation that edits the features and labels in the data with group fairness, individual distortion, and data fidelity constraints and objectives.
- **Learning fair representations**: The idea is to find a latent representation that encodes the data well while hiding information about protected attributes. In the context of hiring, the model is trained using a dataset that includes candidate information, such as gender or ethnicity, but once it learns to generate the latent representation, it does not directly consider the protected attributes when making predictions on new candidates, aiming to ensure fairness and reduce bias. For example, if a hiring model learns fair representations, it may focus on relevant factors like skills, experience, and qualifications rather than relying on gender or ethnicity to make hiring decisions.
- **Disparate impact remover**: Feature values are appropriately edited to increase group fairness while preserving rank-ordering within groups.

# Bias Mitigation Strategies for ML models

## In-Processing Algorithms

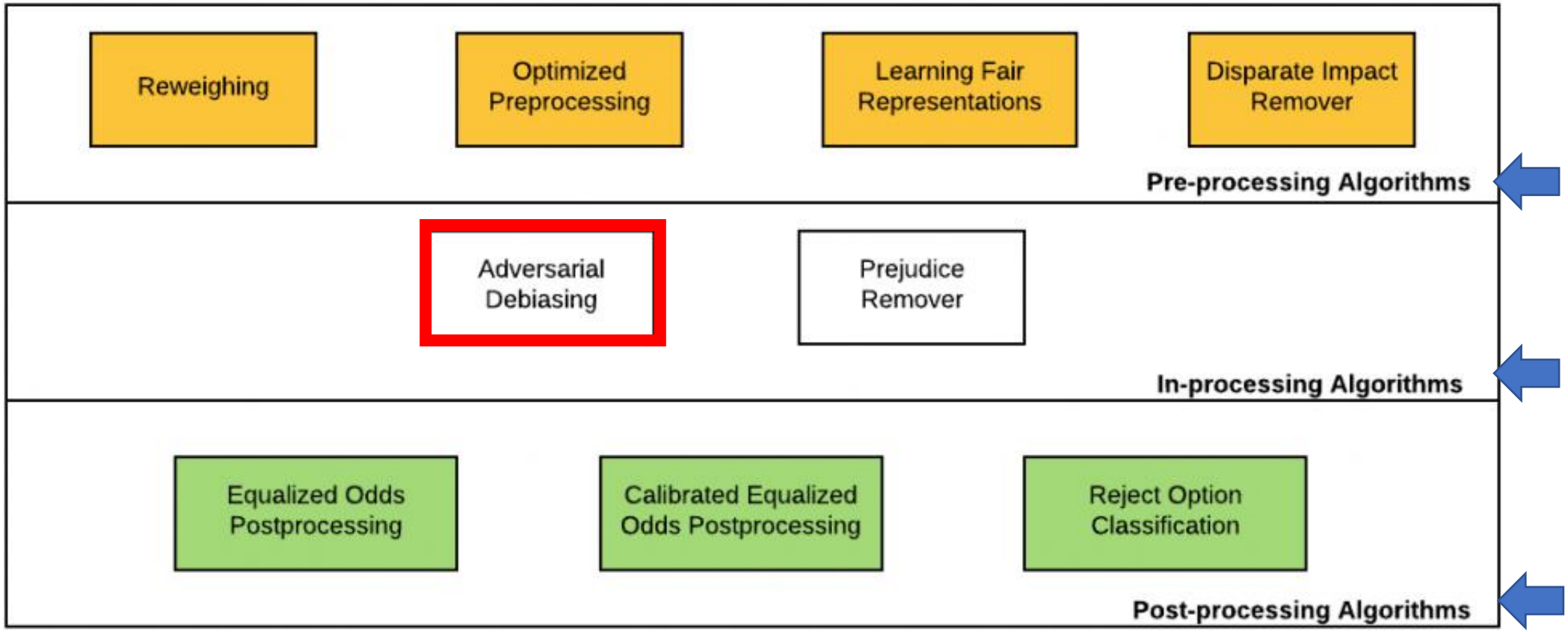
- **Adversarial Debiasing**: A classifier model is learned to maximize prediction accuracy and simultaneously reduce an adversary's ability to determine the protected attribute from the predictions [**to be developed**]
- **Prejudice remover**: The idea is to add a discrimination-aware regularization term to the learning objective.

## Post-Processing Algorithms

- **Equalized odds postprocessing**: The algorithm solves a linear program to find probabilities with which to change output labels to optimize equalized odds.
- **Calibrated equalized odds postprocessing**: The algorithm optimizes over calibrated classifier score outputs to find probabilities with which to change output labels with an equalized odds objective.
- **Reject option classification**: The idea is to give favorable outcomes to unprivileged groups and unfavorable outcomes to privileged groups in a confidence band around the decision boundary

# Bias Mitigation Strategies for ML models

Some of the bias mitigation strategies that can be applied in ML Model Development lifecycle (MDLC) to achieve discrimination-aware Machine Learning models:



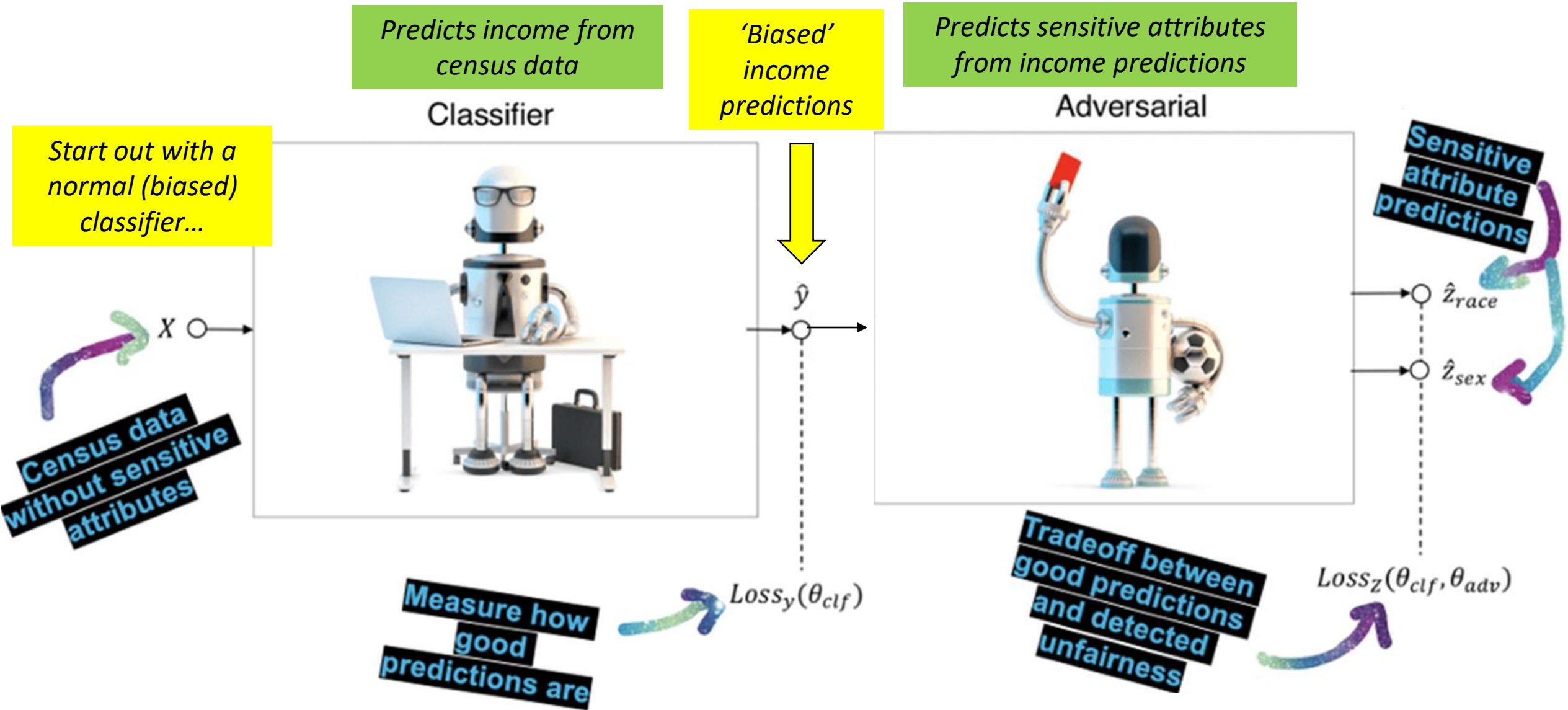

https://dzone.com/articles/machine-learning-models-bias-mitigation-strategies

Exemple of how make fair machine learning models

**Adversarial Debiasing**

# Training for fairness : Adversarial training procedure

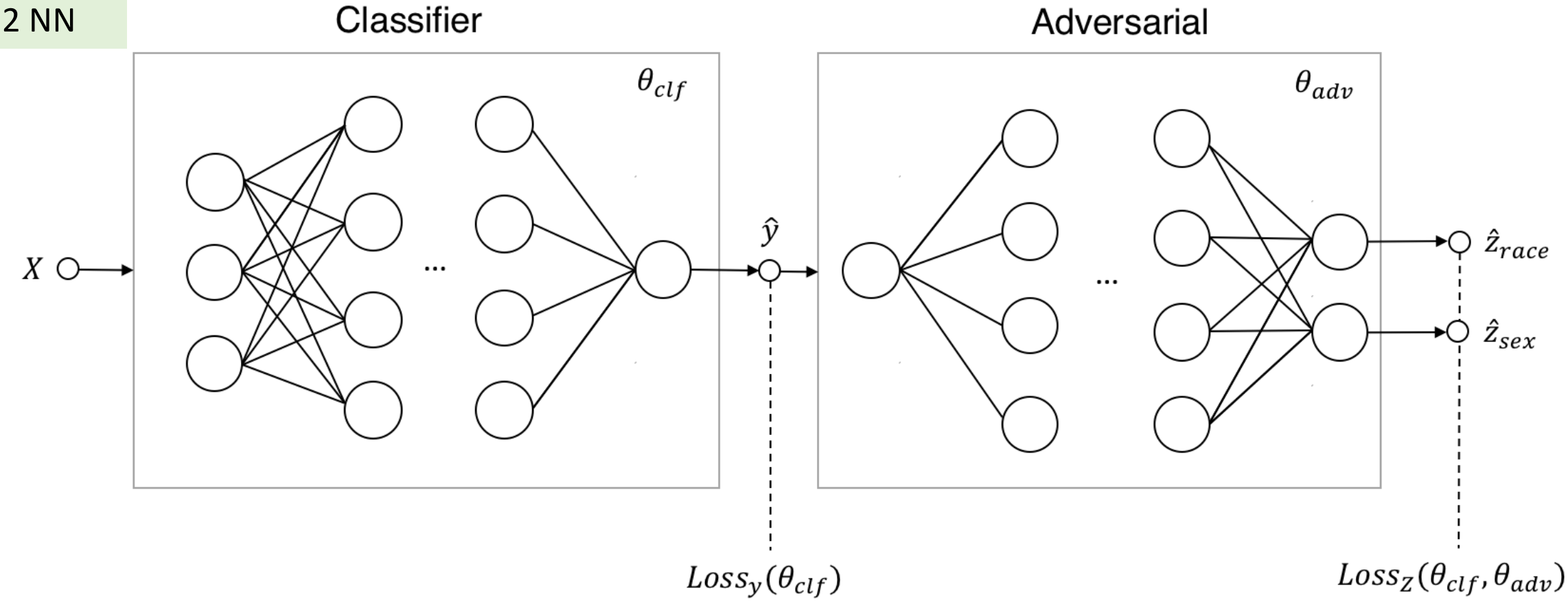
- **Principle**: Enforce fairness by adding an adversarial classifier as a fairness referee to a normal (biased) classifier
- This referee tries to reconstruct bias from the predictions and penalizes the classifier if it can find any unfairness.



# Training for fairness : Adversarial training procedure

- Principle: Enforce fairness by adding an adversarial classifier as a fairness referee to a normal (biased) classifier
- This referee tries to reconstruct bias from the predictions and penalizes the classifier if it can find any unfairness.

use **PyTorch** with 2 NN



# Example: "Census Income"

fairness-in-torch.ipynb

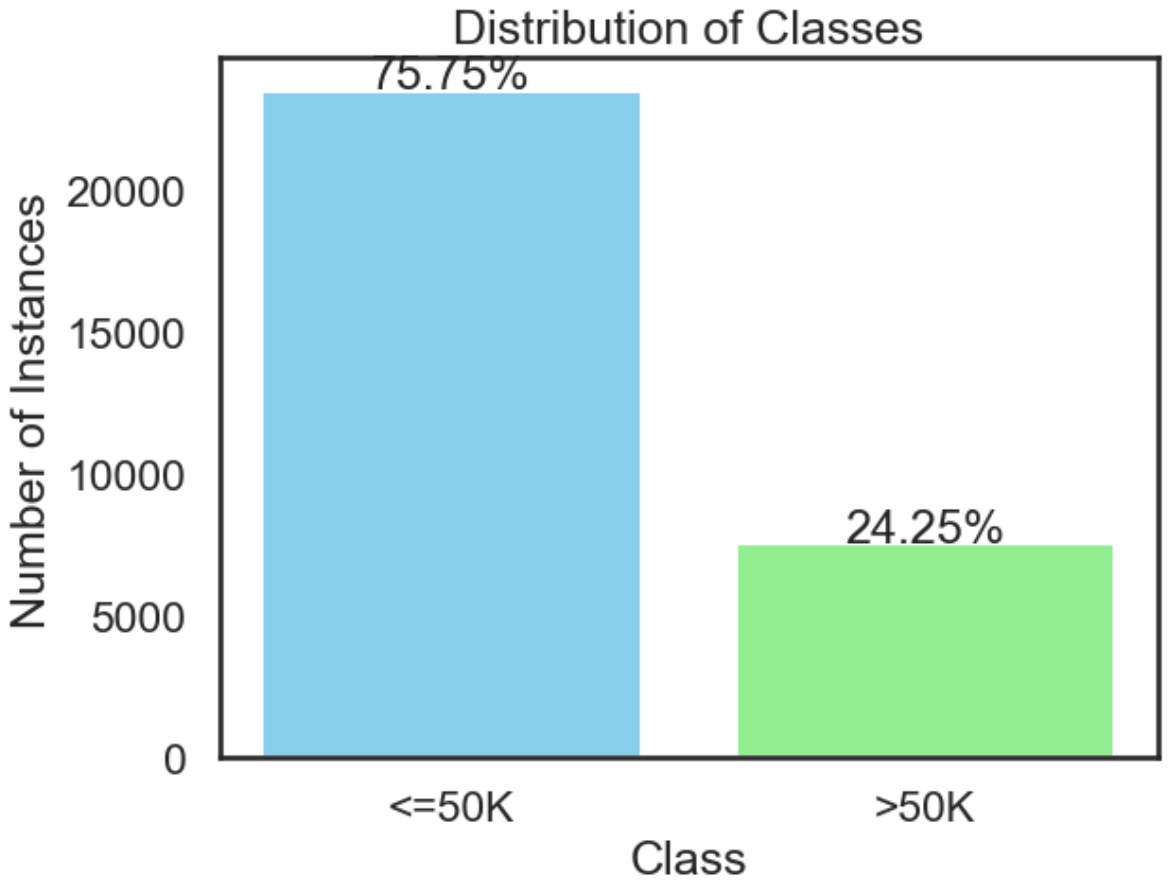
- Predict **income level** using **Adult dataset** (https://archive.ics.uci.edu/ml/datasets/Adult) that involves predicting personal income levels as above or below $50,000 per year based on personal details

  **# Instances** 48842 **# Attributes** 14
- The used approach is based on the 2017 NIPS paper "Learning to Pivot with Adversarial Networks" by Louppe et al.

| age | workclass | fnlwgt | education | education-num | marital-status | occupation | relationship | race | sex | capital-gain | capital-loss | hours-per-week | native-country | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 39 | State-gov | 77516 | Bachelors | 13 | Never-married | Adm-clerical | Not-in-family | White | Male | 2174 | 0 | 40 | United-States | <=50K |
| 50 | Self-emp-not | 83311 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 13 | United-States | <=50K |
| 38 | Private | 215646 | HS-grad | 9 | Divorced | Handlers-cleaners | Not-in-family | White | Male | 0 | 0 | 40 | United-States | <=50K |
| 53 | Private | 234721 | 11th | 7 | Married-civ-spouse | Handlers-cleaners | Husband | Black | Male | 0 | 0 | 40 | United-States | <=50K |
| 28 | Private | 338409 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Wife | Black | Female | 0 | 0 | 40 | Cuba | <=50K |
| 37 | Private | 284582 | Masters | 14 | Married-civ-spouse | Exec-managerial | Wife | White | Female | 0 | 0 | 40 | United-States | <=50K |
| 49 | Private | 160187 | 9th | 5 | Married-spouse-abs | Other-service | Not-in-family | Black | Female | 0 | 0 | 16 | Jamaica | <=50K |
| 52 | Self-emp-not | 209642 | HS-grad | 9 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 45 | United-States | >50K |
| 31 | Private | 45781 | Masters | 14 | Never-married | Prof-specialty | Not-in-family | White | Female | 14084 | 0 | 50 | United-States | >50K |
| 42 | Private | 159449 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 5178 | 0 | 40 | United-States | >50K |
| 37 | Private | 280464 | Some-college | 10 | Married-civ-spouse | Exec-managerial | Husband | Black | Male | 0 | 0 | 80 | United-States | >50K |
| 30 | State-gov | 141297 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Husband | Asian-Pac-Isl | Male | 0 | 0 | 40 | India | >50K |
| 23 | Private | 122272 | Bachelors | 13 | Never-married | Adm-clerical | Own-child | White | Female | 0 | 0 | 30 | United-States | <=50K |
| 32 | Private | 205019 | Assoc-acdm | 12 | Never-married | Sales | Not-in-family | Black | Male | 0 | 0 | 50 | United-States | <=50K |
| 40 | Private | 121772 | Assoc-voc | 11 | Married-civ-spouse | Craft-repair | Husband | Asian-Pac-Isl | Male | 0 | 0 | 40 | ? | >50K |

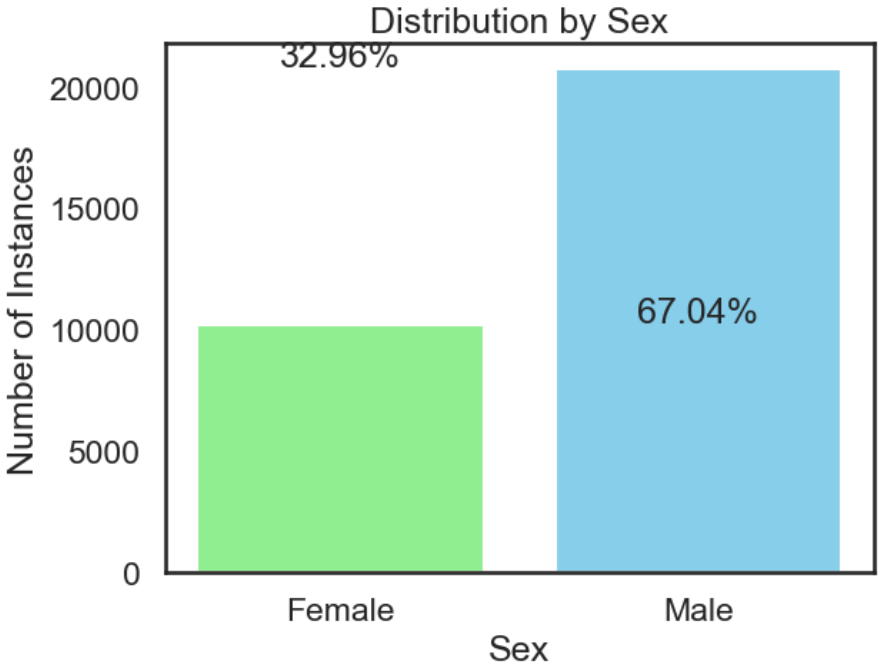
1. **The set of features** contains the input attributes that the model uses for making the predictions, with attributes like age, education level and occupation.
2. **The target** contains the binary class that the model needs to predict (above or below $50,000).

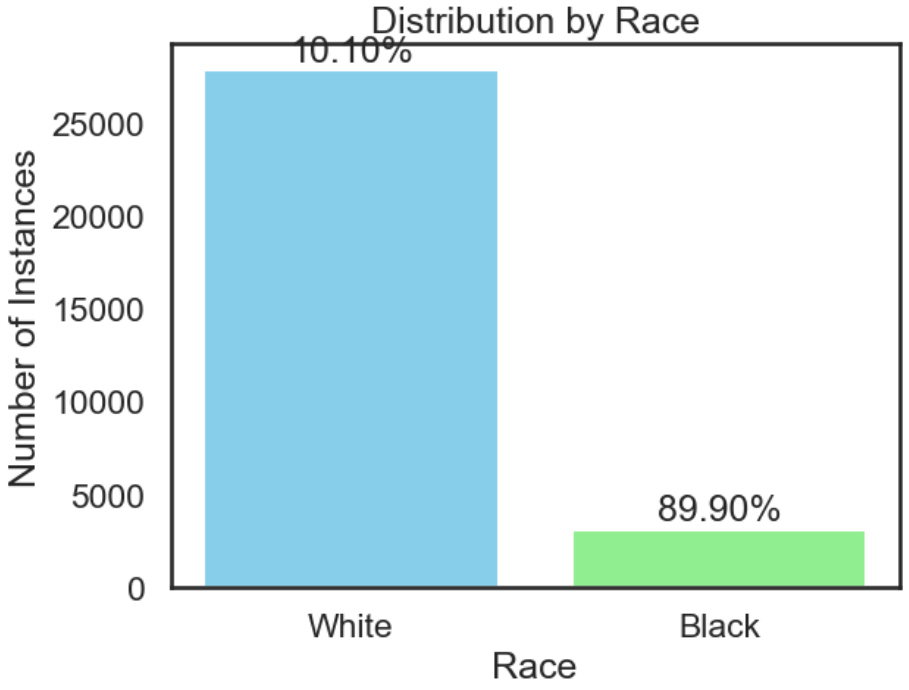
# Take a look on the data…




- Initial observation: The dataset exhibits class **imbalance**, meaning that the distribution of class labels is skewed, with one class (low-income class) having significantly more instances than the other.
- However, addressing this class imbalance is not our current focus.

# Take a look on the data…




- We consider two sensitive attributes, **Sex** and **Race**, have imbalanced distributions that can lead to unfair decisions
- How to measure fairness and how make fair machine learning models ?

# Example: "Census Income" dataset

**We dispatch the data into 3 subsets:**

1. **The set of features** $X$ contains the input attributes that the model uses for making the predictions, with attributes like age, education level and occupation.
2. **The targets** $y$ contain the binary class labels that the model needs to predict. These labels are $y \in \{income > 50K, income \leq 50K\}$
3. **The set of sensitive attributes** $Z$ contains the attributes for which we want the prediction to be fair. These are $zrace \in \{black, white\}$ and $zsex \in \{male, female\}$

$y$

$X$ $Z$ $X$

| age | workclass | fnlwgt | education | education-num | marital-status | occupation | relationship | race | sex | capital-gain | capital-loss | hours-per-week | native-country | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 39 | State-gov | 77516 | Bachelors | 13 | Never-married | Adm-clerical | Not-in-famil | White | Male | 2174 | 0 | 40 | United-States | <=50K |
| 50 | Self-emp-not | 83311 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 13 | United-States | <=50K |
| 38 | Private | 215646 | HS-grad | 9 | Divorced | Handlers-cleaners | Not-in-famil | White | Male | 0 | 0 | 40 | United-States | <=50K |
| 53 | Private | 234721 | 11th | 7 | Married-civ-spouse | Handlers-cleaners | Husband | Black | Male | 0 | 0 | 40 | United-States | <=50K |
| 28 | Private | 338409 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Wife | Black | Female | 0 | 0 | 40 | Cuba | <=50K |
| 37 | Private | 284582 | Masters | 14 | Married-civ-spouse | Exec-managerial | Wife | White | Female | 0 | 0 | 40 | United-States | <=50K |
| 49 | Private | 160187 | 9th | 5 | Married-spouse-abs | Other-service | Not-in-famil | Black | Female | 0 | 0 | 16 | Jamaica | <=50K |
| 52 | Self-emp-not | 209642 | HS-grad | 9 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 0 | 0 | 45 | United-States | >50K |
| 31 | Private | 45781 | Masters | 14 | Never-married | Prof-specialty | Not-in-famil | White | Female | 14084 | 0 | 50 | United-States | >50K |
| 42 | Private | 159449 | Bachelors | 13 | Married-civ-spouse | Exec-managerial | Husband | White | Male | 5178 | 0 | 40 | United-States | >50K |
| 37 | Private | 280464 | Some-college | 10 | Married-civ-spouse | Exec-managerial | Husband | Black | Male | 0 | 0 | 80 | United-States | >50K |
| 30 | State-gov | 141297 | Bachelors | 13 | Married-civ-spouse | Prof-specialty | Husband | Asian-Pac-Isl | Male | 0 | 0 | 40 | India | >50K |
| 23 | Private | 122272 | Bachelors | 13 | Never-married | Adm-clerical | Own-child | White | Female | 0 | 0 | 30 | United-States | <=50K |
| 32 | Private | 205019 | Assoc-acdm | 12 | Never-married | Sales | Not-in-famil | Black | Male | 0 | 0 | 50 | United-States | <=50K |
| 40 | Private | 121772 | Assoc-voc | 11 | Married-civ-spouse | Craft-repair | Husband | Asian-Pac-Isl | Male | 0 | 0 | 40 | ? | >50K |

# Example: "Census Income" dataset

…/fairness-in-torch.ipynb

```
 load ICU data set
X, y, Z = load_ICU_data('data/adult.data')
```

…/fairness/helpers.py

```
Z = (input_data.loc[:, sensitive_attribs]
   .assign(race=lambda df: (df['race'] == 'White').astype(int),
    sex=lambda df: (df['sex'] == 'Male').astype(int)))

   # targets; 1 when someone makes over 50k , otherwise 0
   y = (input_data['target'] == '>50K').astype(int)

   # features; 'target' and sentive attribute columns are dropped
   X = (input_data
  .drop(columns=['target', 'race', 'sex', 'fnlwgt'])
  .fillna('Unknown')
  .pipe(pd.get_dummies, drop_first=True))
```

**One-hot encoding** : converts categorical variables into distinct binary variables, indicating the presence or absence of each category – thus we pass from 14 to 93 attributes

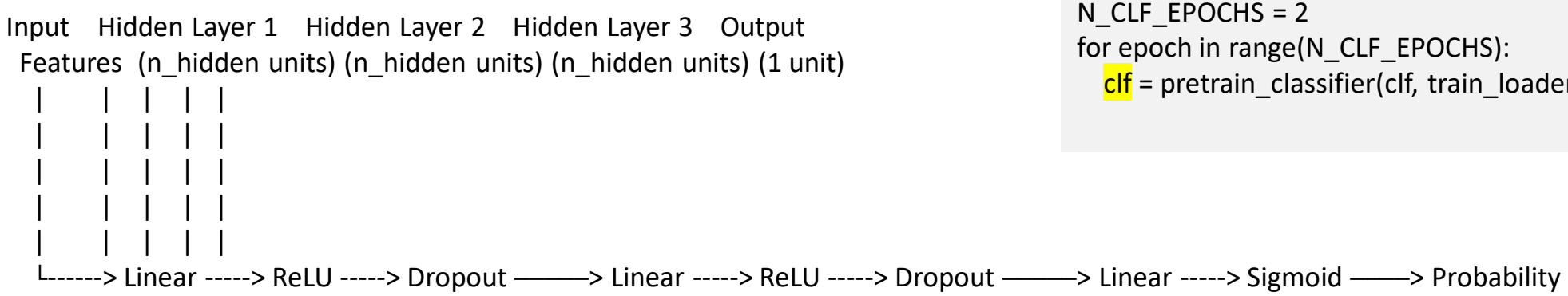
```
features X: 30940 samples, 93 attributes
targets y: 30940 samples
sensitives Z: 30940 samples, 2 attributes
```

- It is important to note that datasets are non-overlapping, so the sensitive attributes **race** and **sex** are not part of the features used for training the model.

# Classifier (1)

1. We train a basic income level predictor using **PyTorch**.
2. The network consists of **three** sequential hidden layers with ReLu activation and dropout. The sigmoid layer turns these activations into a probability for the income class (i.e. probability of income being greater than 50K).

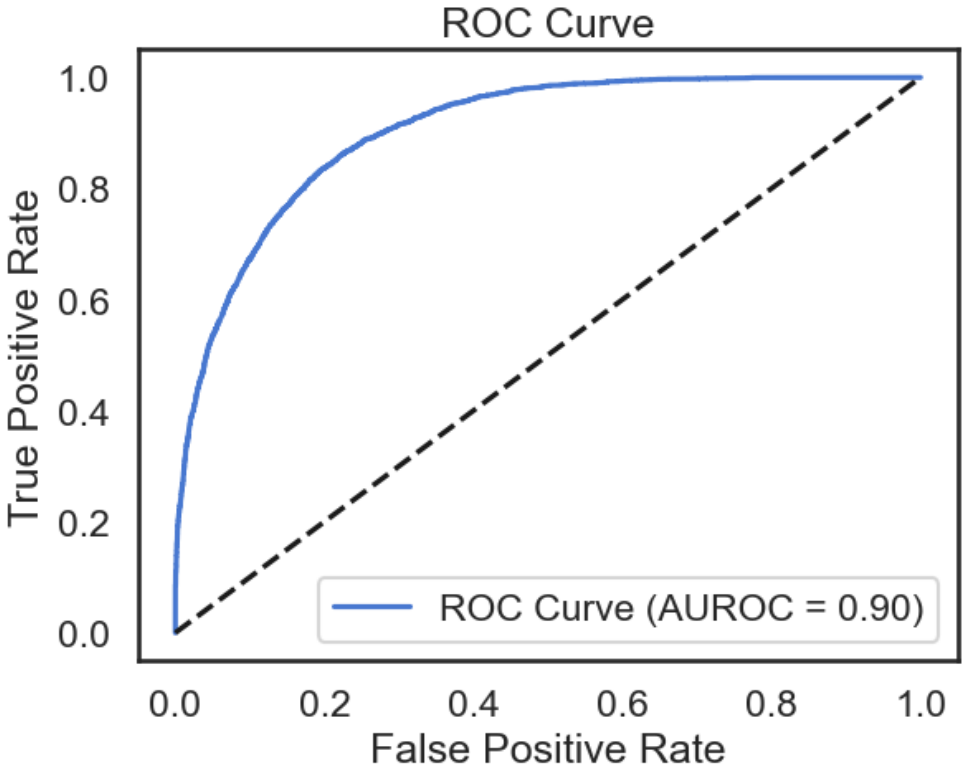
```
class Classifier(nn.Module):

  def __init__(self, n_features, n_hidden=32, p_dropout=0.2):
 super(Classifier, self).__init__()
 self.network = nn.Sequential(
    nn.Linear(n_features, n_hidden),
    nn.ReLU(),
    nn.Dropout(p_dropout),
    nn.Linear(n_hidden, n_hidden),
    nn.ReLU(),
    nn.Dropout(p_dropout),
    nn.Linear(n_hidden, n_hidden),
    nn.ReLU(),
    nn.Dropout(p_dropout),
    nn.Linear(n_hidden, 1),
 )

  def forward(self, x):
 return torch.sigmoid(self.network(x))
…

N_CLF_EPOCHS = 2
for epoch in range(N_CLF_EPOCHS):
  clf = pretrain_classifier(clf, train_loader, clf_optimizer, clf_criterion)
```



# Classifier (2)

```
#Evaluate the accuracy of the classifier on the test data
test_accuracy = evaluate_classifier(clf, test_data)
print("Accuracy on test data:", test_accuracy)
```

```
Accuracy on test data: 0.85%
```




- Area Under the Receiver Operating Characteristic Curve (AUROC): metric used to evaluate the performance of a binary classification model.
- It measures the ability of the model to distinguish between positive and negative classes by calculating the area under the (ROC) curve.

With a ROC AUC larger than 0.9 and a prediction accuracy of 85% we can say that the basic classifier performs pretty well!

**However, if it is also fair in its predictions, that remains to be seen…**

# How to measure fairness ?

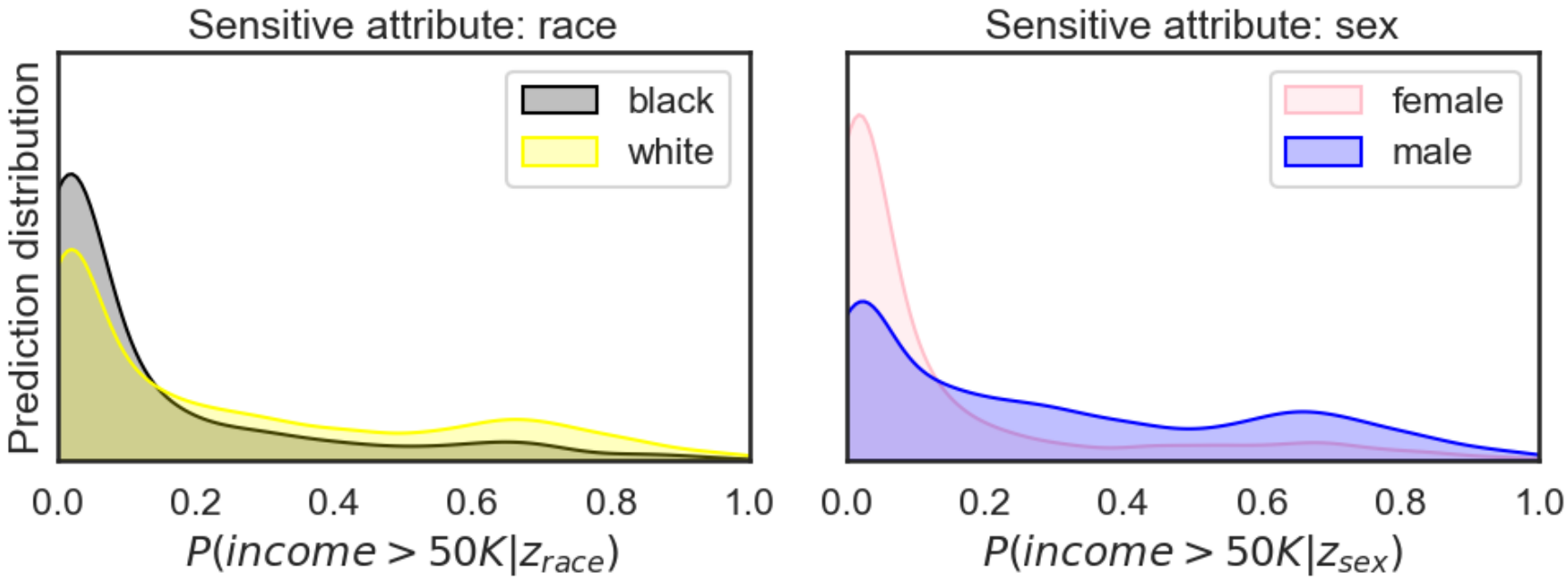
- Qualitative model
- Quantitative model

## Qualitative model fairness

- We start the investigation into the fairness of our classifier by analysing the predictions it made on the test set.
- The plots in the figure below show the distributions of the predicted $P(income>50K)$ given the sensitive attributes.

```
with torch.no_grad():
    pre_clf_test = clf(test_data.tensors[0])
```

- **pre_clf_test** stores the raw predictions of the clf model on the test data (transformed into y_pre_clf in the code)




Individuals who are classified as black and/or female have a significantly higher likelihood of being predicted to have an income below 50K compared to those who are white and/or male.

- **The predictions are biased when considered in the context of race and sex.**
- The model tends to favor white males when it comes to assigning high-income levels

## Quantitative model fairness

In order to get a 'quantitative' measure of how fair our classifier is, we use the **Demographic Parity** (p%-rule ), also called Independence, Statistical Parity, which is one of the most well-known criteria for fairness:

- This rule measures demographic parity by quantifying the disparate impact on a group of people.
- The p%-rule is defined as the ratio of:
  probability of a positive outcome given the sensitive attribute being true;
  probability of a positive outcome given the sensitive attribute being false;

- A classifier that makes a binary class prediction $\hat{y} \in \{0,1\}$ given a binary sensitive attribute $z \in \{0,1\}$ satisfies the p%-rule if the following inequality holds.

$$\min\left(\frac{P(\hat{y}=1|z=1)}{P(\hat{y}=1|z=0)}, \frac{P(\hat{y}=1|z=0)}{P(\hat{y}=1|z=1)}\right) \geq \frac{p}{100}$$

- ( p is often set at 80%)
- In determining the fairness we follow the EEOC and say that a model is fair when it satisfies at least an 80%-rule

- When a classifier is completely fair it will satisfy a 100%-rule. In contrast, when it is completely unfair it satisfies a %0-rule.

…/fairness/helpers.py

```
def p_rule(y_pred, z_values, threshold=0.5):
    y_z_1 = y_pred[z_values == 1] > threshold if threshold else y_pred[z_values == 1]
    y_z_0 = y_pred[z_values == 0] > threshold if threshold else y_pred[z_values == 0]
    odds = y_z_1.mean() / y_z_0.mean()
    return np.min([odds, 1/odds]) * 100
```

1. **y_pred** predicted values
2. **z_values** is a binary vector that divides the data into two groups.
3. **threshold** specifies the threshold to use for binary classification (to transform prob into binary values).
4. **y_z_1** examples where `z_values` is equal to 1.
5. **y_z_0** examples where `z_values` is equal to 0.
6. **odds** calculates the ratio between the average of positive predictions in the `z_values == 1` group and the average of positive predictions in the `z_values == 0` group. This represents the disparity of probabilities between the two groups.

The function returns the **minimum** value between `odds` and `1/odds`, multiplied by 100. This normalizes the disparity to express it as a percentage, limiting its maximum value to 100.

…/fairness/helpers.py

```
p_rules = {'race': p_rule(y_pred, Z_test['race']),
      'sex' : p_rule(y_pred, Z_test['sex']),}
```

Satisfied p%-rules:
- race: 42%-rule
- sex: 39%-rule

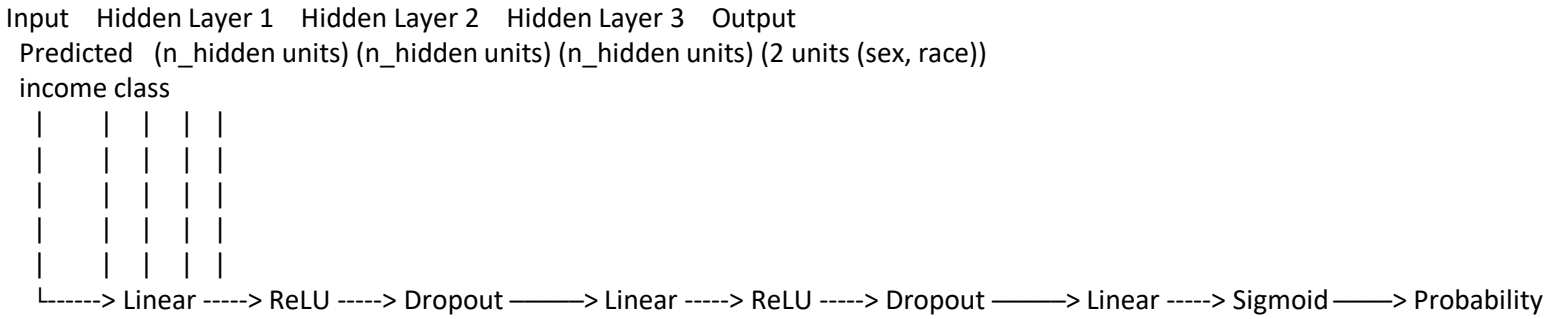
- 42% p-rule for the 'race' attribute means that there is a 42% disparity between the predicted probability rates for different races.
- 39% p-rule for the 'sex' attribute indicates a 39% disparity between the predicted probability rates for male and female sexes.

- For both sensitive attributes the classifier satisfies a p%-rule that is significantly lower than 80%.
- This supports our earlier conclusion that the trained classifier is unfair in making its predictions.

# Adversary

```
class Adversary(nn.Module):

   def __init__(self, n_sensitive, n_hidden=32):
 super(Adversary, self).__init__()
 self.network = nn.Sequential(
    nn.Linear(1, n_hidden),
    nn.ReLU(),
    nn.Linear(n_hidden, n_hidden),
    nn.ReLU(),
    nn.Linear(n_hidden, n_hidden),
    nn.ReLU(),
    nn.Linear(n_hidden, n_sensitive),
 )

   def forward(self, x):
 return torch.sigmoid(self.network(x))
```

- The adversary (adv) has the same structure than the classifier (clf)
- The input comes from a single class (the predicted income class) and the output consists of two sensitive classes (sex and race).




```
…
N_ADV_EPOCHS = 5

for epoch in range(N_ADV_EPOCHS):
   adv= pretrain_adversary(adv, clf, train_loader, adv_optimizer,  adv_criterion)
…
with torch.no_grad():
   pre_adv_test = adv(pre_clf_test)
```

Adversary performance:
- ROC AUC: 0.66

- **pre_adv_test** stores the raw predictions of the adv model on the test data (transformed into y_pre_adv in the code)

## Training for fairness

Now that we have an unfair classifier and an adversary that is able to pick up on unfairness, we can engage them in the **zero-sum game** to make the classifier fair.

- The zero-sum game consists of a competition between the classifier and the adversary, where both parties are in conflict.
  - The classifier aims to make accurate predictions on the data,
  - while the adversary seeks to detect unfair decisions made by the classifier regarding sensitive attributes.

- The classifier (clf) aims to minimize its prediction losses ($Loss_y$) while also minimizing the ability of the adversary to detect unfair decisions ($Loss_Z$) by adjusting its parameters ($\theta_{clf}$). The objective function for the classifier is:

$$\min_{\theta_{clf}}[Loss_y(\theta_{clf}) - \lambda\, Loss_Z(\theta_{clf},\theta_{adv})]$$

$\lambda$ represents a scalar parameter that allows adjusting the relative importance of the adversary's loss compared to the classifier's loss in the optimization problem.

- The adversary (adv) aims to maximize its ability to detect instances where the classifier's decisions are unfair with respect to sensitive attributes. The loss function captures the discrepancy between the predictions of the classifier (based on $\theta_{clf}$) and the adversary (based on $\theta_{adv}$) regarding unfair decisions. The objective function of the adversary is to minimize the loss function $Loss_Z(\theta_{clf},\theta_{adv})$, which measures its ability to detect unfair decisions made by the classifier.

$$\min_{\theta_{adv}}[Loss_Z(\theta_{clf},\theta_{adv})]$$

## Training for fairness

We can summarize this procedure in the following 3 steps:

1. Pre-train the classifier on the full data set. (DONE)
2. Pre-train the adversarial on the predictions of the pre-trained classifier. (DONE)
3. During $T$ iterations simultaneously train the adversarial and classifier networks:

first train the adversarial for a single epoch while keeping the classifier fixed
then train the classifier on the full dataset for several epochs while keeping the adversarial fixed.
(Note that originally training was done on a single random minibatch, because this approach greatly speeds up the training procedure.)

The actual adversarial training starts only after the first two pre-training steps. It is then that the training procedure mimics the zero-sum game during which our classifier will (hopefully) learn how make predictions that are both accurate and fair.

```
def train(clf, adv, data_loader, clf_criterion, adv_criterion, clf_optimizer, adv_optimizer, lambdas):

    # Train adversary
    for x, y, z in data_loader:
  p_y = clf(x)
  adv.zero_grad()
  p_z = adv(p_y)
  loss_adv = (adv_criterion(p_z, z) * lambdas).mean()
  loss_adv.backward()
  adv_optimizer.step()

    # Train classifier on single batch (pass in for)
    for x, y, z in data_loader:
  pass
    p_y = clf(x)
    p_z = adv(p_y)
    clf.zero_grad()
    loss_adv = (adv_criterion(p_z, z) * lambdas).mean()
    clf_loss = clf_criterion(p_y, y) - (adv_criterion(adv(p_y), z) * lambdas).mean()
    clf_loss.backward()
    clf_optimizer.step()

    return clf, adv
```

## Training for fairness

```
  # Train adversary
  for x, y, z in data_loader:
p_y = clf(x)
adv.zero_grad()
p_z = adv(p_y)
loss_adv = (adv_criterion(p_z, z) * lambdas).mean()
loss_adv.backward()
adv_optimizer.step()
```

**lambdas = torch.Tensor([130, 30])**

- 130 : optimization process will give higher importance to minimizing the adversary's loss.
- 30: optimization process will assign less importance to minimizing the classifier's loss.

1. **p_y = clf(x)**: The classifier model (**clf**) predicts the income (**p_y**) based on the input **x**.
2. **adv.zero_grad()**: The gradients of the adversary model (**adv**) are reset to zero.
3. **p_z = adv(p_y)**: The adversary model makes predictions (**p_z**) based on the income predictions (**p_y**) from the classifier model.
4. **loss_adv =** The loss between the adversary's predictions (**p_z**) and the true labels (**z**) is calculated using the adversary criterion (**adv_criterion**). The loss is then multiplied by a scalar value **lambdas** and averaged across the batch.
5. **loss_adv.backward()**: The gradients of the loss with respect to the adversary model's parameters are computed.
6. **adv_optimizer.step()**: update the parameters of the adversary model (**θ_adv**).

```
def train(clf, adv, data_loader, clf_criterion, adv_criterion, clf_optimizer, adv_optimizer, lambdas):

    # Train adversary
    for x, y, z in data_loader:
  p_y = clf(x)
  adv.zero_grad()
  p_z = adv(p_y)
  loss_adv = (adv_criterion(p_z, z) * lambdas).mean()
  loss_adv.backward()
  adv_optimizer.step()

    # Train classifier on single batch (pass in for)
    for x, y, z in data_loader:
  pass
    p_y = clf(x)
    p_z = adv(p_y)
    clf.zero_grad()
    loss_adv = (adv_criterion(p_z, z) * lambdas).mean()
    clf_loss = clf_criterion(p_y, y) - (adv_criterion(adv(p_y), z) * lambdas).mean()
    clf_loss.backward()
    clf_optimizer.step()

    return clf, adv
```

## Training for fairness

```
# Train classifier on single batch
  for x, y, z in data_loader:
pass
  p_y = clf(x)
  p_z = adv(p_y)
  clf.zero_grad()
  loss_adv = (adv_criterion(p_z, z) * lambdas).mean()
  clf_loss = clf_criterion(p_y, y) - (adv_criterion(adv(p_y), z) * lambdas).mean()
  clf_loss.backward()
  clf_optimizer.step()
```

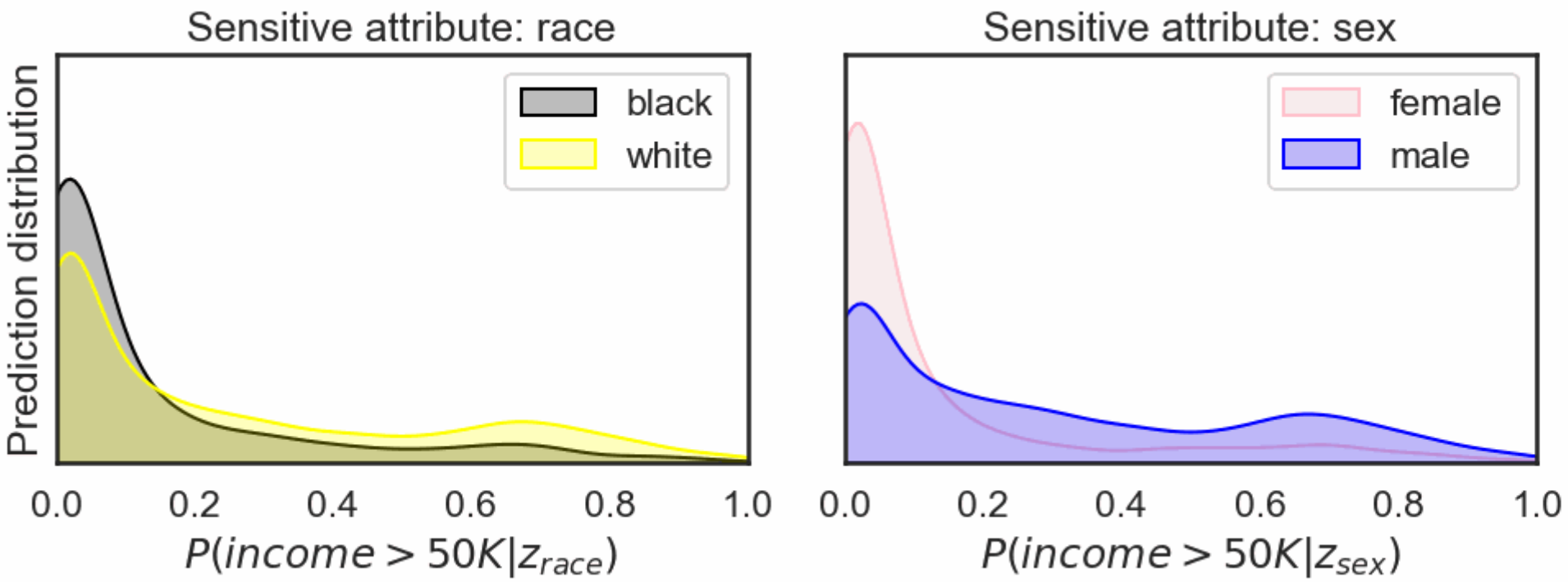
$$Loss_y(\theta_{clf}) - \lambda\, Loss_Z(\theta_{clf}, \theta_{adv})$$

- **clf_criterion(p_y, y)=** $Loss_y(\theta_{clf})$ represents the loss of the classifier (clf) calculated between the income predictions (p_y) and the true income labels (y).
- **adv_criterion(adv(p_y), z)** = $Loss_Z(\theta_{clf}, \theta_{adv})$] corresponds to the loss of the adversary (adv) calculated between the adversary predictions (adv(p_y)) and the true adversary labels (z).

1. **p_y = clf(x)**: The classifier model (**clf**) predicts the income (**p_y**) based on the input **x**.
2. **p_z = adv(p_y)**: The adversary model (**adv**) makes predictions (**p_z**) based on the income predictions (**p_y**) from the classifier model.
3. **clf.zero_grad()**: The gradients of the classifier model (**clf**) are reset to zero.
4. **loss_adv** : The loss between the adversary's predictions (**p_z**) and the true labels (**z**) is calculated and multiplied by a scalar value **lambdas** and averaged across the batch.
5. **clf_loss** : The objective function for training the classifier.
6. **clf_loss.backward()**: The gradients of the classifier loss with respect to the classifier model's parameters are computed.
7. **clf_optimizer.step()**: update the parameters of the classifier model (θ_clf).

# Training for fairness

```
N_EPOCH_COMBINED = 165

for epoch in range(1, N_EPOCH_COMBINED):

    clf, adv = train(clf, adv, train_loader, clf_criterion, adv_criterion, clf_optimizer, adv_optimizer, lambdas)
```



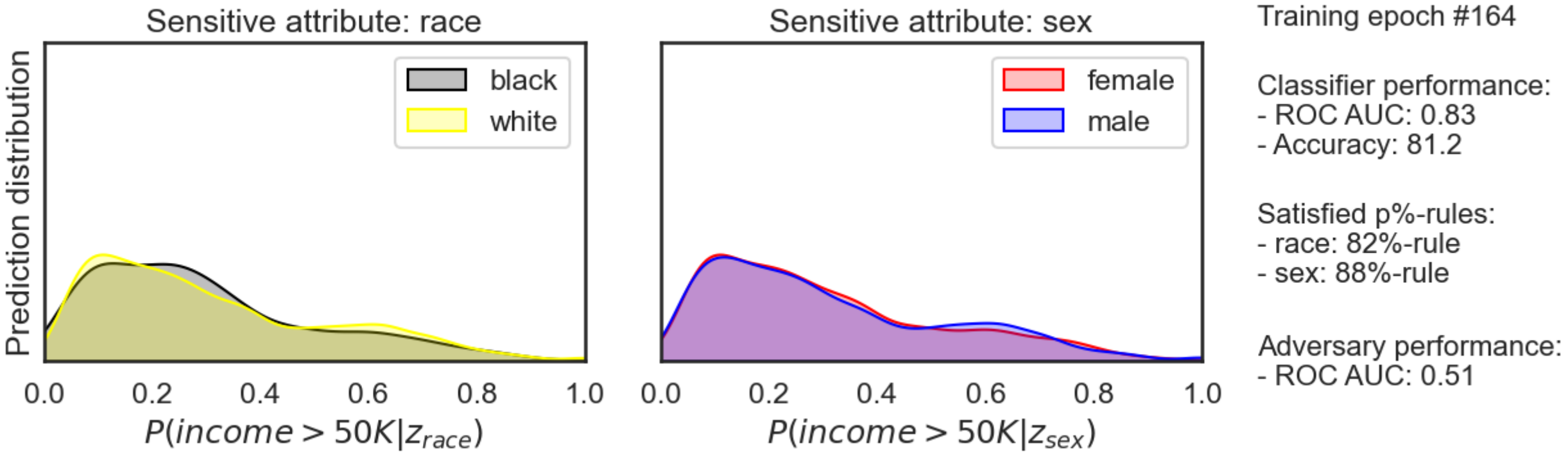

Training epoch #1 65

Classifier performance:
- ROC AUC: 0.90
- Accuracy: 84.8

Satisfied p%-rules:
- race: 43%-rule
- sex: 39%-rule

Adversary performance:
- ROC AUC: 0.66

## Training for fairness




- We've successfully used an adversarial neural network to make our classifier fair!
- The expected outcome of this zero-sum game is a fair classifier that makes accurate predictions without discriminating against or being unfair to certain groups. By combining the predictions of the classifier and the adversary, the goal is to obtain a model that is both high-performing and equitable.

# Exercice

Rank the above questions according to the stage of the machine learning lifecycle to ensure fairness.

1. Was our historical data generated by a biased process that we reify?
2. Can we collect more data or reweight?
3. Is the data skewed?
4. Do our proxies really measure what we think they do?
5. Does our data include enough minority samples?
6. Is the objective function in line with ethics?
7. Do we need to apply debiasing algorithms to preprocess our data?
8. Do our labels reinforce stereotypes?
9. Are there missing/biased features?
10. Do we need to include fairness constraints in the function?
11. Do we need to model minority populations separately?
12. Have we evaluated the model using relevant fairness metrics?
13. Can we evaluate the model on other datasets beyond the test set?
14. Are we deploying our model on a population that we did not train/test on?
15. Does the model encourage feedback loops that can produce increasingly unfair outcomes?
16. Is an algorithm an ethical solution to our problem?
17. Is the algorithm misusable in other contexts?