UNIVERSITÄT DUISBURG ESSEN

Open-Minded

Machine Learning Basics

Neuroinformatics Tutorial 1

Duc Duy Pham[1]

[1]Intelligent Systems, Faculty of Engineering,
University of Duisburg-Essen, Germany

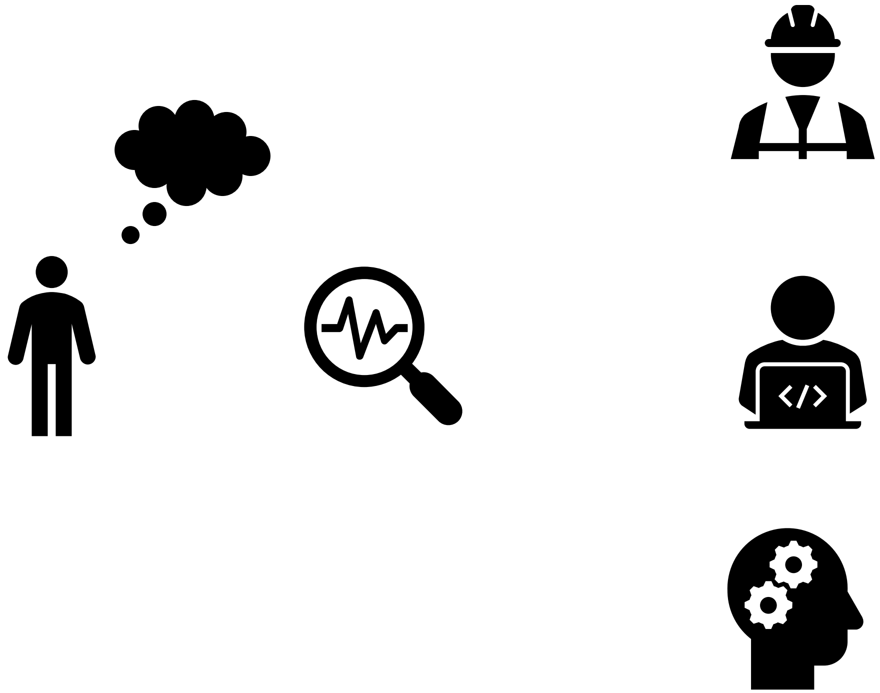
Content

- Motivation
- When does a Machine learn?
- Machine Learning tasks
- Broad types of Machine Learning Algorithms
- How do input/output look like?
- Data partitioning

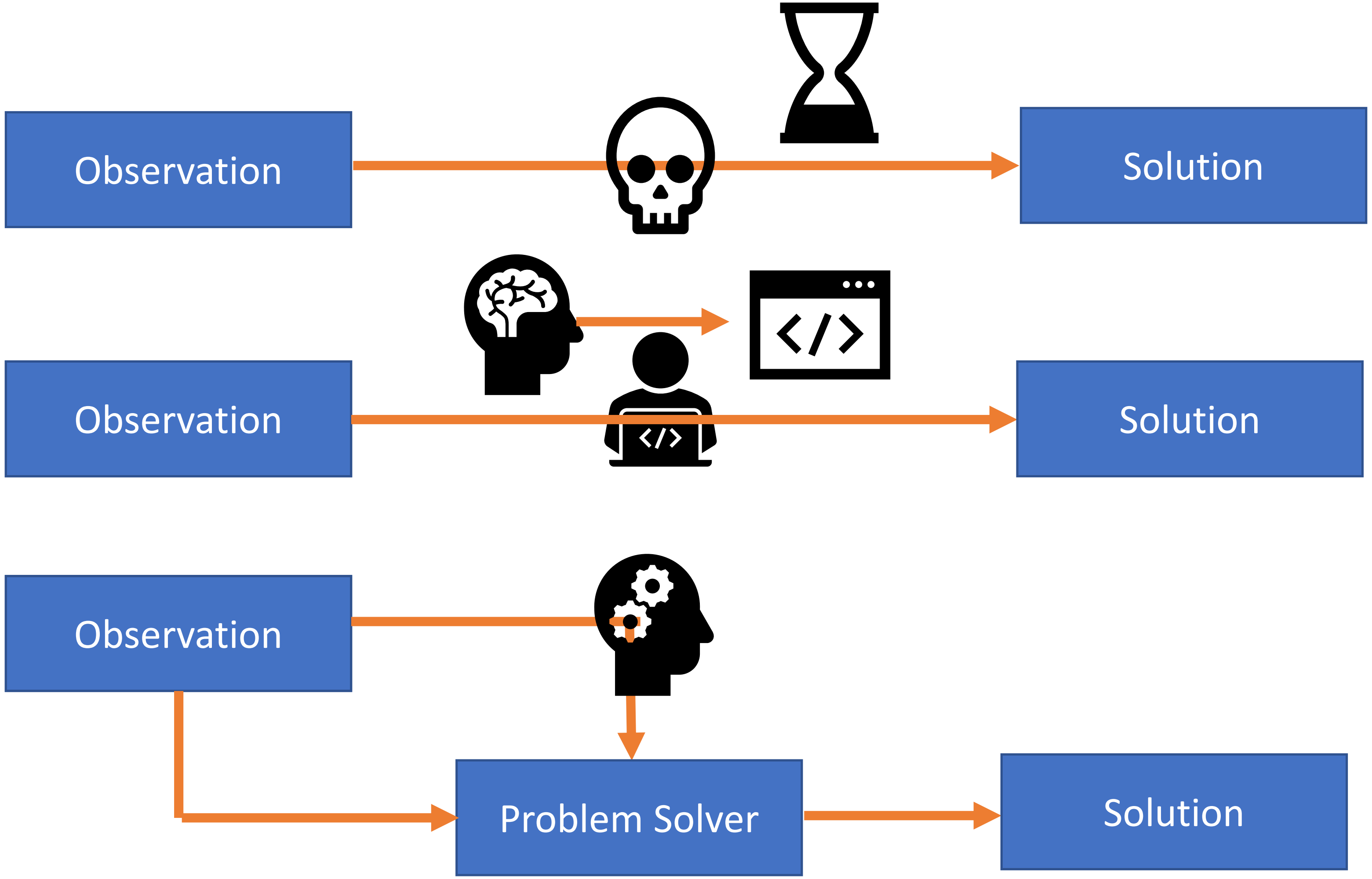
Content

- <u>Motivation</u>
- When does a Machine learn?
- Machine Learning tasks
- Broad types of Machine Learning Algorithms
- How do input/output look like?
- Data partitioning

Motivation



Motivation

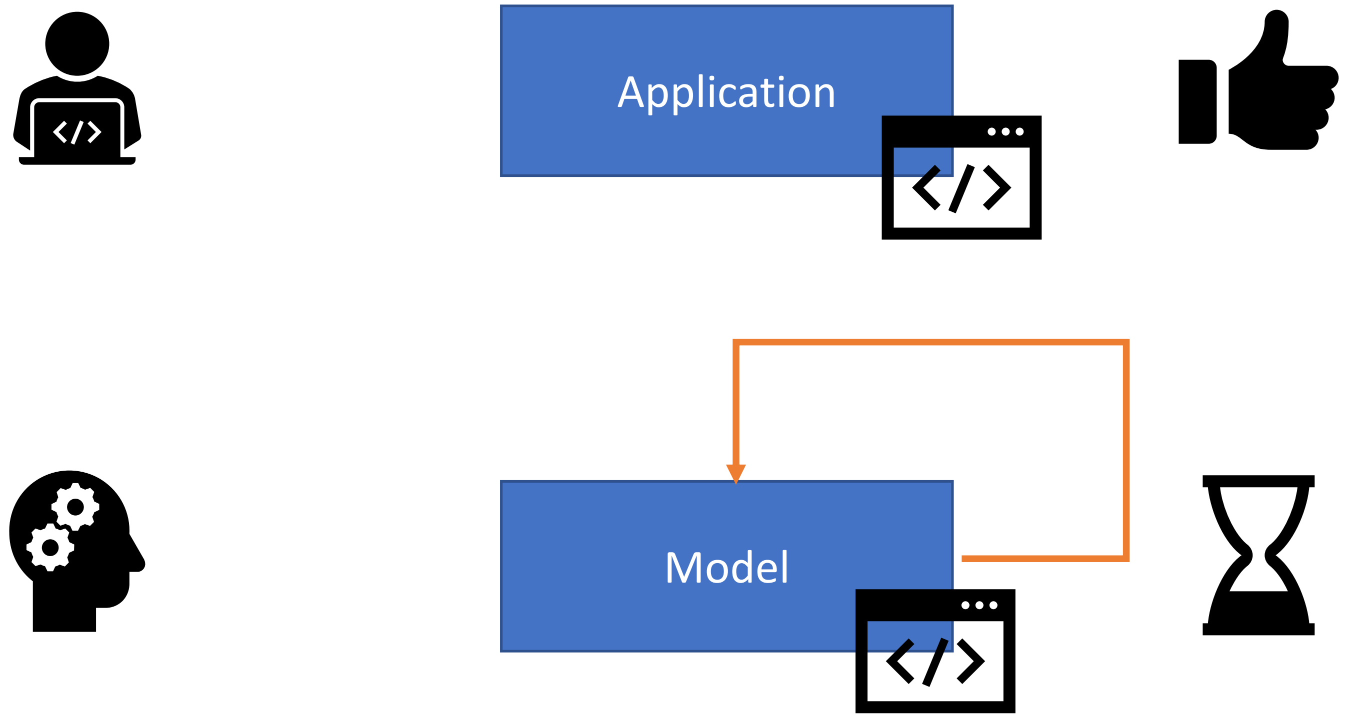


Can be very time consuming!

May lack generalizability

May also lack generalizability

Difference Coding and ML?



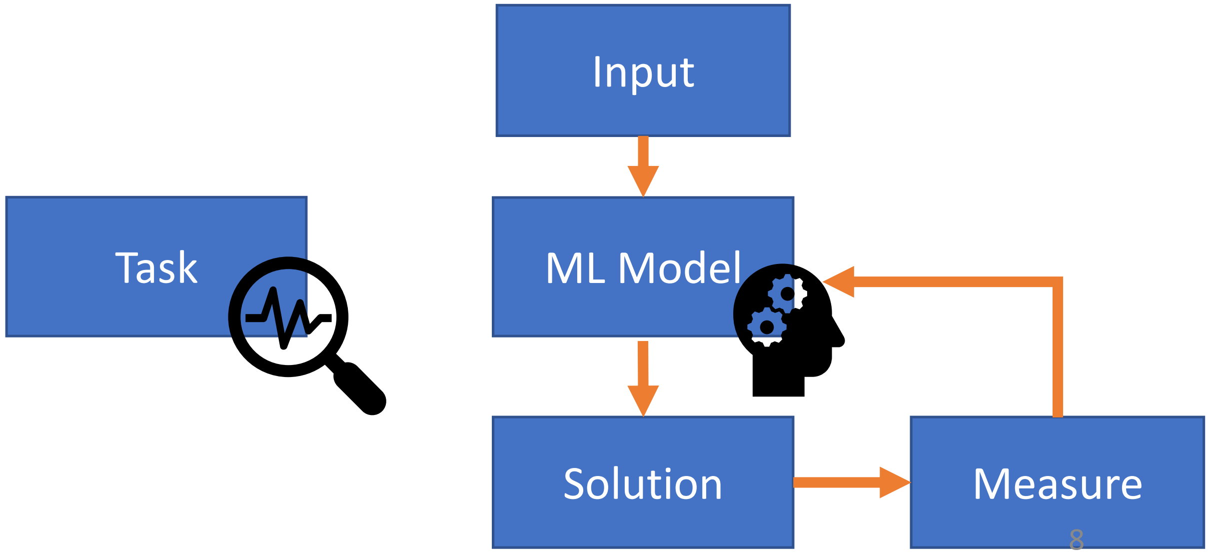
Content

- Motivation
- <u>When does a Machine learn?</u>
- Machine Learning tasks
- Broad types of Machine Learning Algorithms
- How do input/output look like?
- Data partitioning

When does a Machine learn?

Mitchell (1997):
„A computer is said to learn from experience E*,*
if its performance at tasks in T*,*
as measured by P*,*
improves with experience E*"*

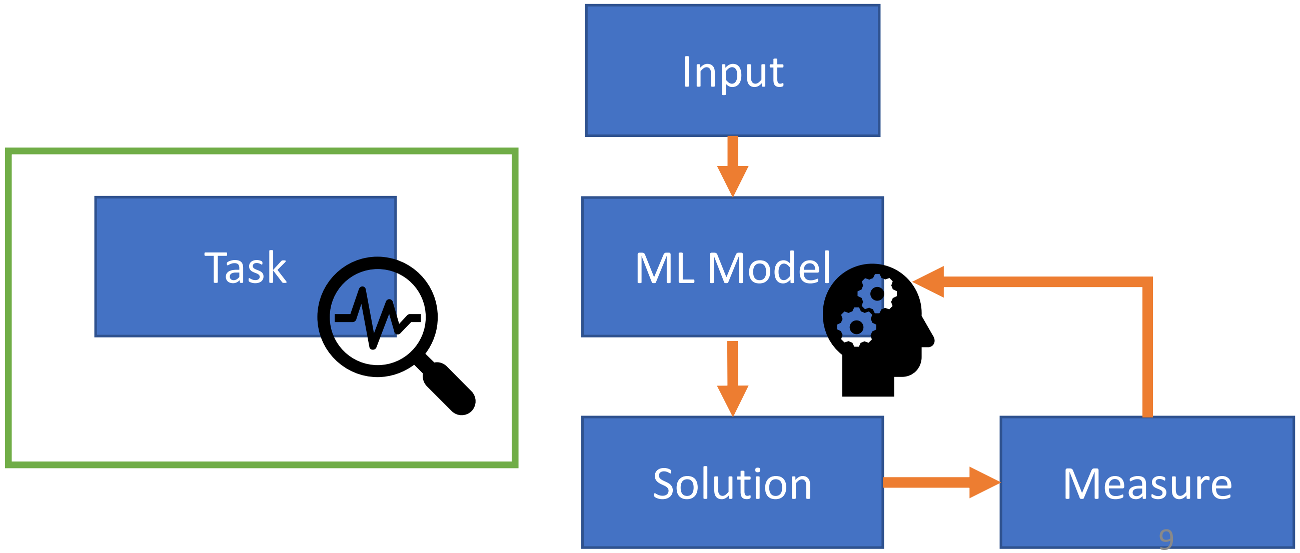
0. Define task T
1. Try to solve task T with your Algorithm
2. Measure Algorithm performance by P
3. Gain experience E by doing so
4. Go to 1.



When does a Machine learn?

Mitchell (1997):
„A computer is said to learn from experience E,
if its performance at tasks in T,
as measured by P,
improves with experience E*"*

0. Define task T
1. Try to solve task T with your Algorithm
2. Measure Algorithm performance by P
3. Gain experience E by doing so
4. Go to 1.



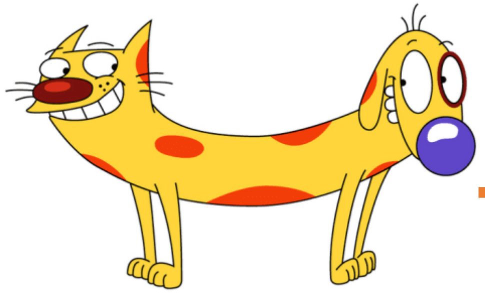
Content

- Motivation
- When does a Machine learn?
- <u>Machine Learning tasks</u>
- Broad types of Machine Learning Algorithms
- How do input/output look like?
- Data partitioning

Tasks

- Typical tasks:
  - Classification

Classification



ML Model

?

Cat

Dog

Bird

Categorical Output!

https://www.buzzfeed.com/lyapalater/how-does-catdog-poop

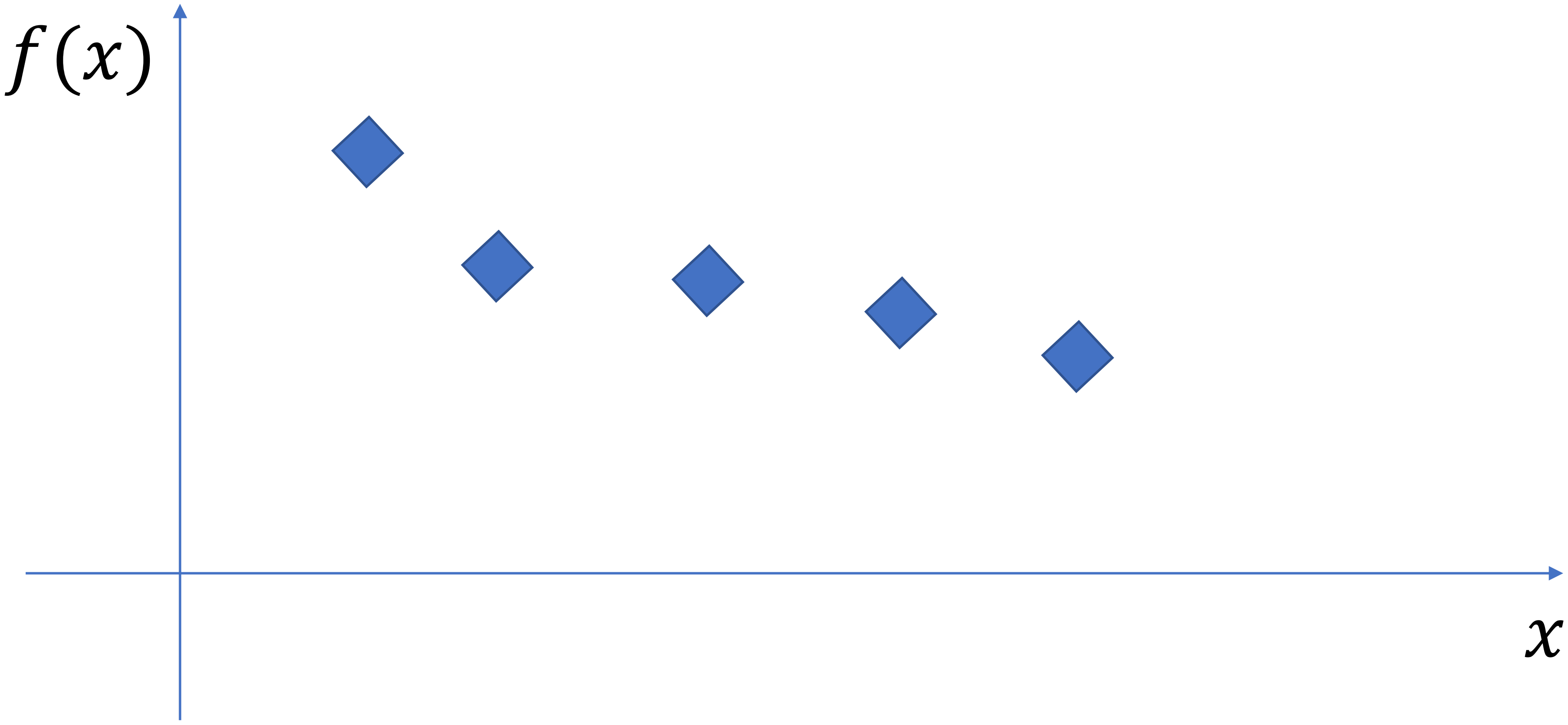
Tasks

- Typical tasks:
  - Classification
  - Regression

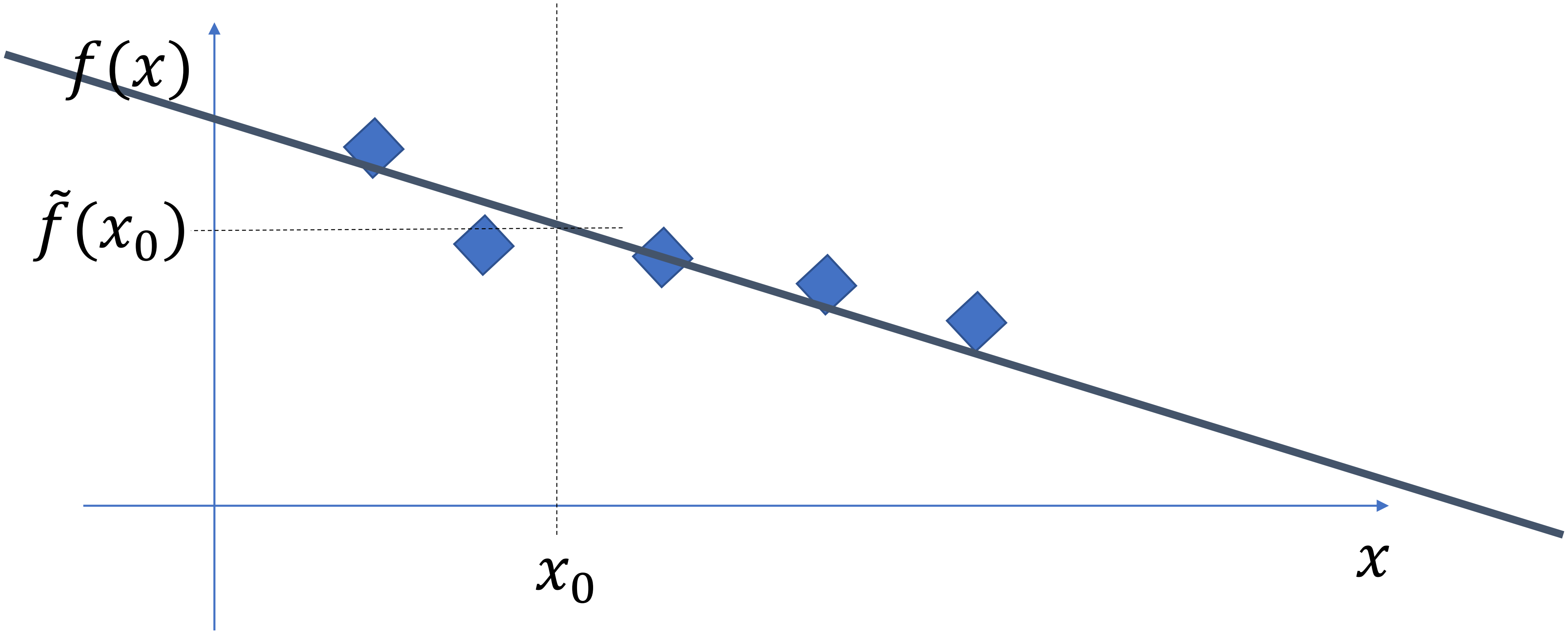
Regression

Continuous Output!

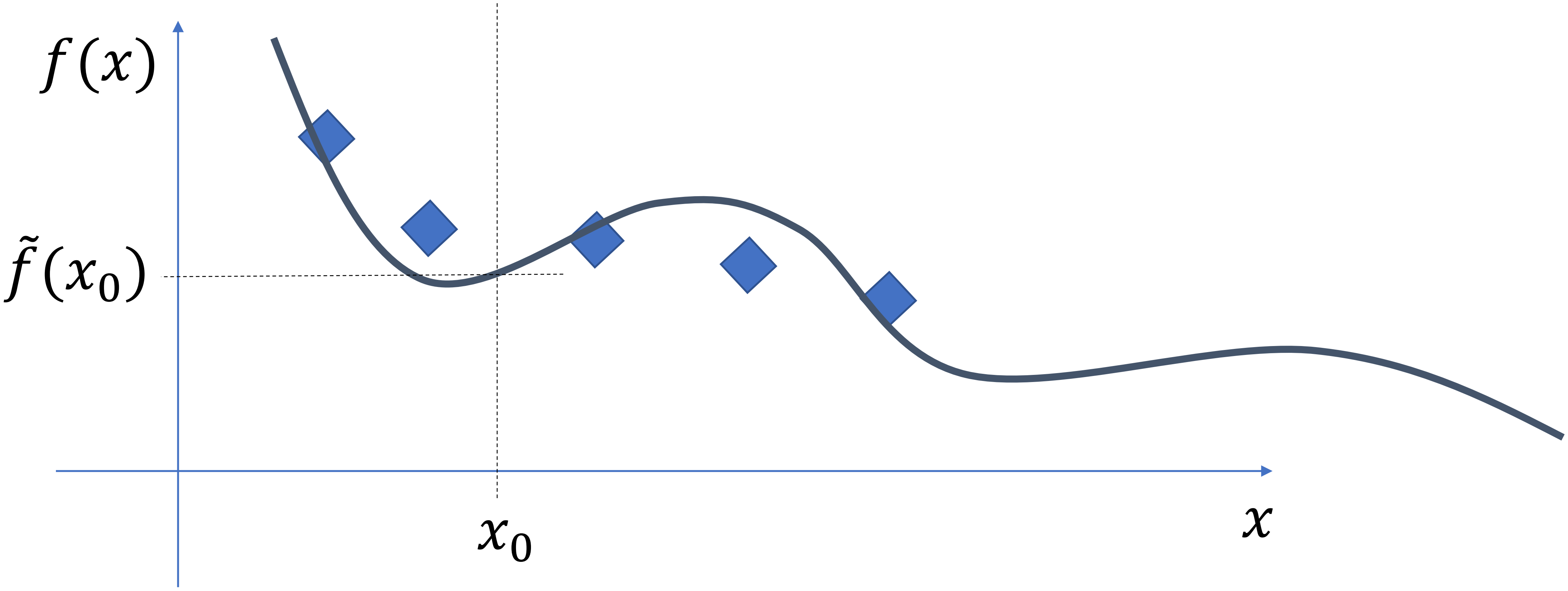
Regression



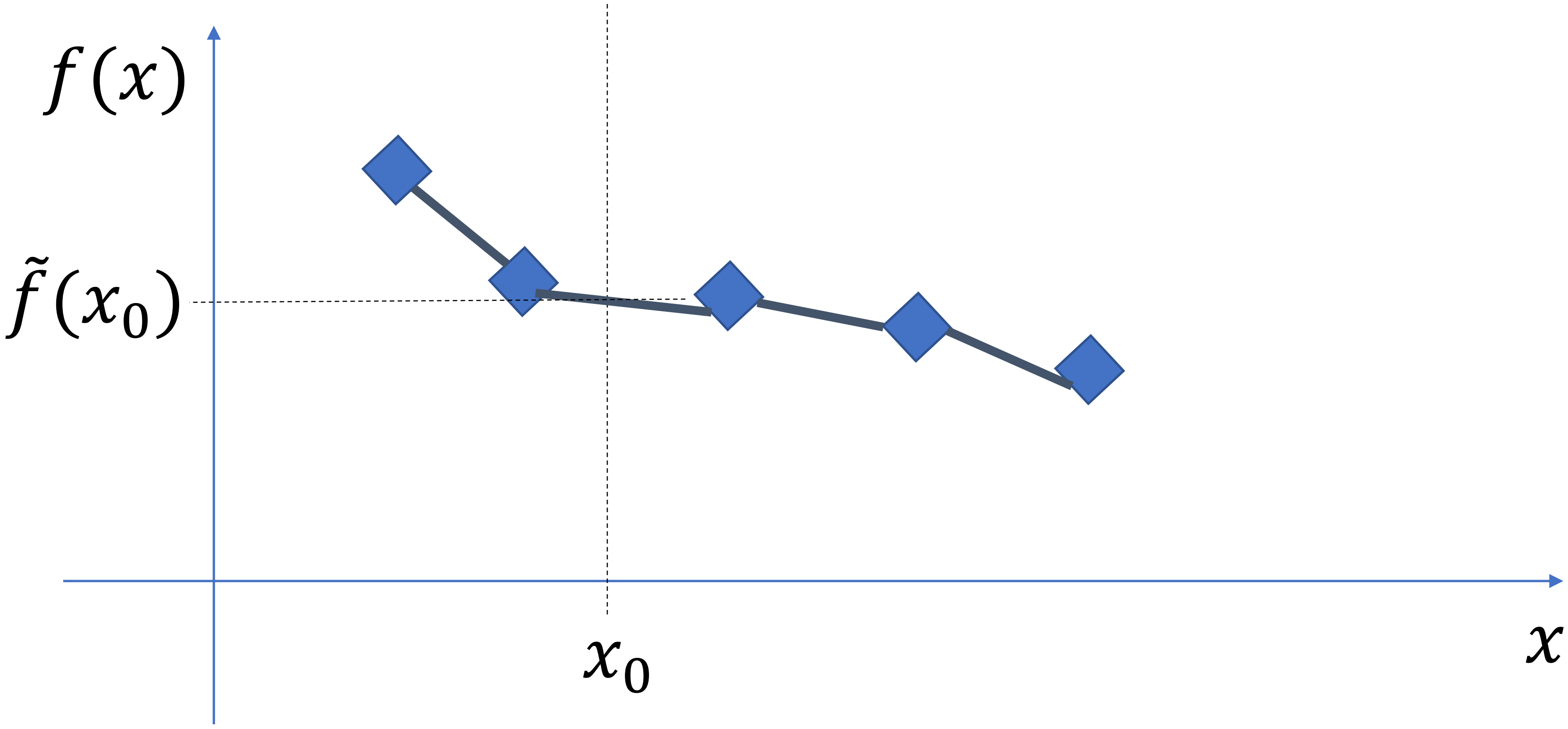
Regression



Regression



Interpolation



Tasks

- Typical tasks:
  - Classification
  - Regression (not the same as interpolation!)
- Further tasks:
  - Transcription

Transcription

- Speech to text

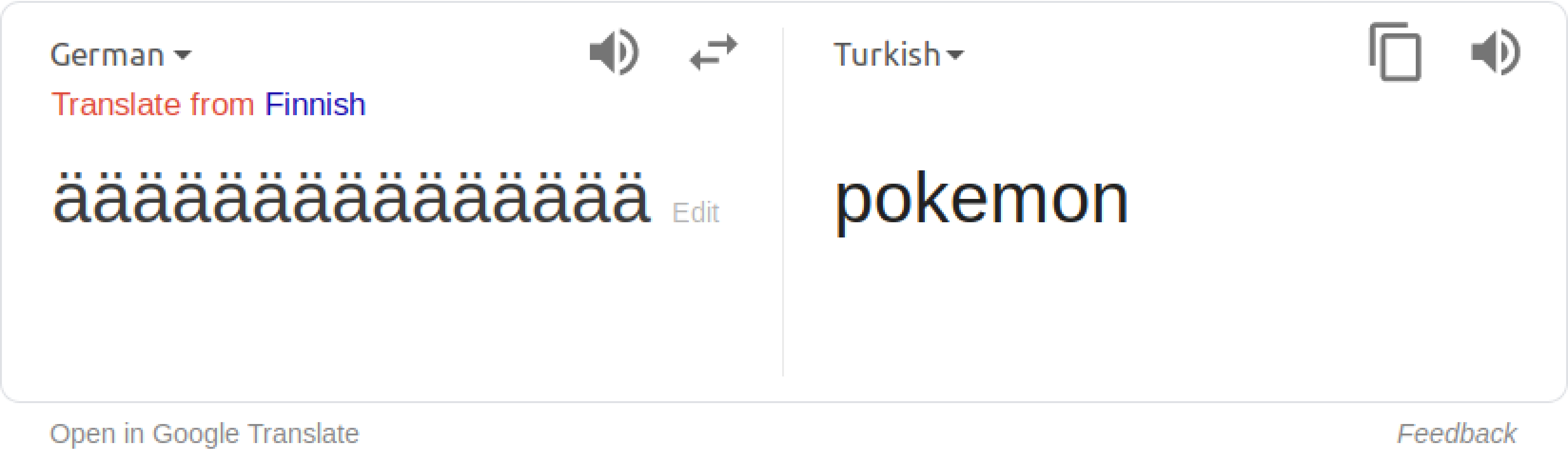
- Image to text

Vinyals, Oriol, et al. "Show and tell: A neural image caption generator."
Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

Tasks

- Typical tasks:
  - Classification
  - Regression (not the same as interpolation!)
- Further tasks:
  - Transcription
  - Machine Translation

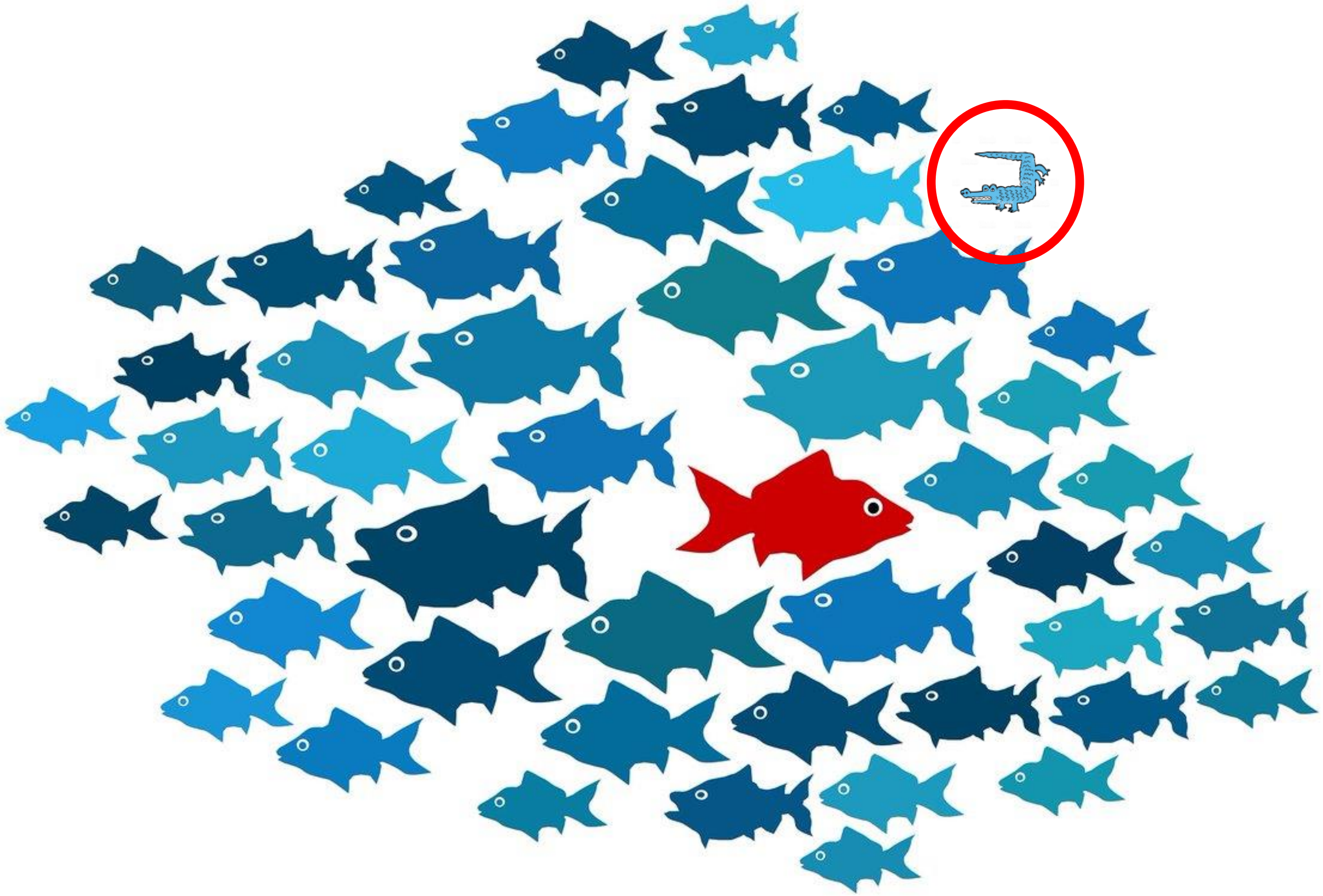
Machine Translation



Tasks

- Typical tasks:
  - Classification
  - Regression (not the same as interpolation!)
- Further tasks:
  - Transcription
  - Machine Translation
  - Anomaly Detection

Anomaly Detection

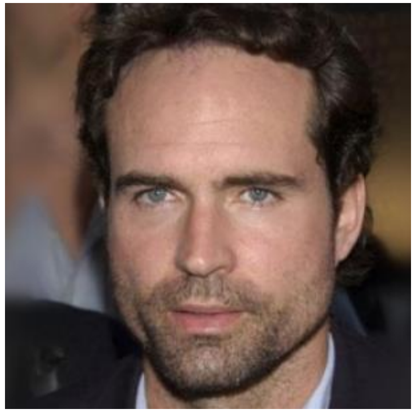


https://towardsdatascience.com/a-brief-overview-of-outlier-detection-techniques-1e0b2c19e561

Tasks

- Typical tasks:
  - Classification
  - Regression (not the same as interpolation!)
- Further tasks:
  - Transcription
  - Machine Translation
  - Anomaly Detection
  - Synthesis

Image Synthesis

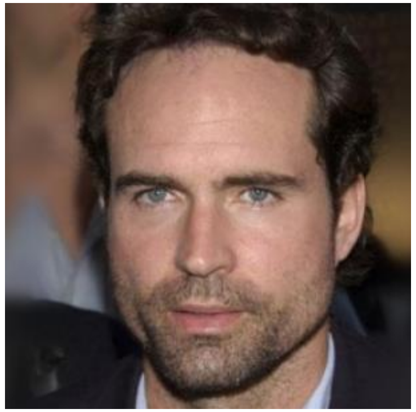


A: Real

B: Fake

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

Image Synthesis



Real

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

Image Synthesis

A: Real

B: Fake

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

Image Synthesis

Fake

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

Image Synthesis

A: Real

B: Fake

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

Image Synthesis

Fake

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

Image Synthesis

A: Real

B: Fake

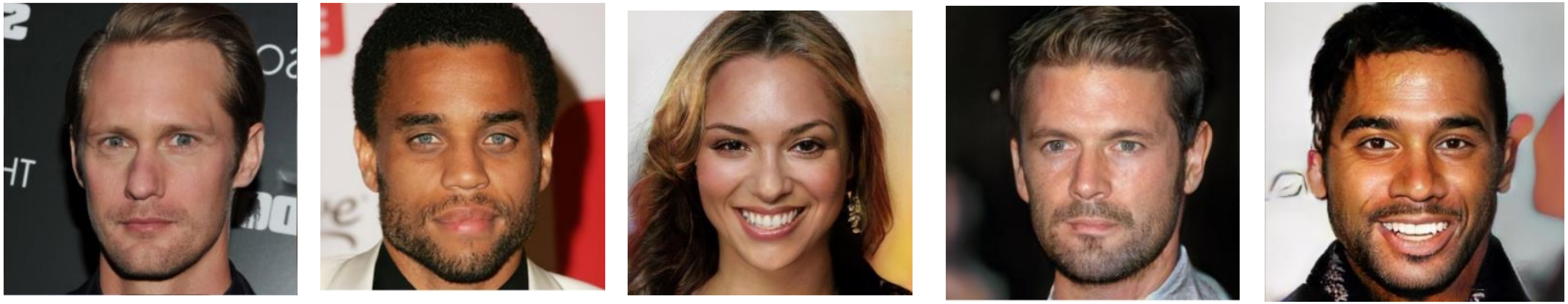
Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

Image Synthesis

Fake

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

Image Synthesis

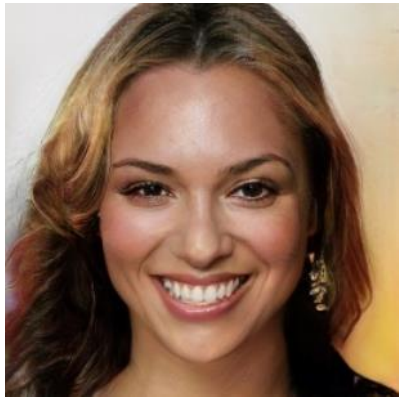


Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

Image Synthesis

Real

Real



Fake

Fake

Fake

Karras, Tero, et al. "Progressive growing of gans for improved quality, stability, and variation." arXiv preprint arXiv:1710.10196 (2017).

Tasks

- Typical tasks:
  - Classification
  - Regression (not the same as interpolation!)
- Further tasks:
  - Transcription
  - Machine Translation
  - Anomaly Detection
  - Synthesis
  - Denoising
  - Imputation of missing values
  - Etc …

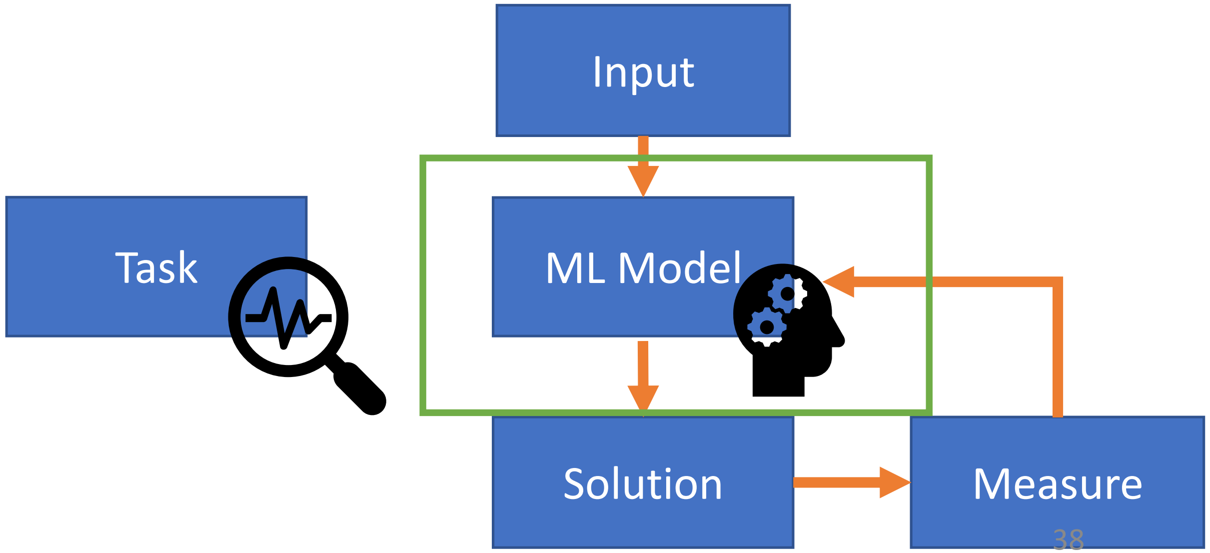
Scope of this course

- Basic/Fundamental Machine Learning methods and algorithms
- Most tasks will be **classification** tasks

When does a Machine learn?

Mitchell (1997):
„A computer is said to learn from experience E*,*
if its performance at tasks in T*,*
as measured by P*,*
improves with experience E*"*

0. Define task T
1. Try to solve task T with your Algorithm
2. Measure Algorithm performance by P
3. Gain experience E by doing so
4. Go to 1.



Content

- Motivation
- When does a Machine learn?
- Machine Learning tasks
- <u>Broad types of Machine Learning Algorithms</u>
- How do input/output look like?
- Data partitioning

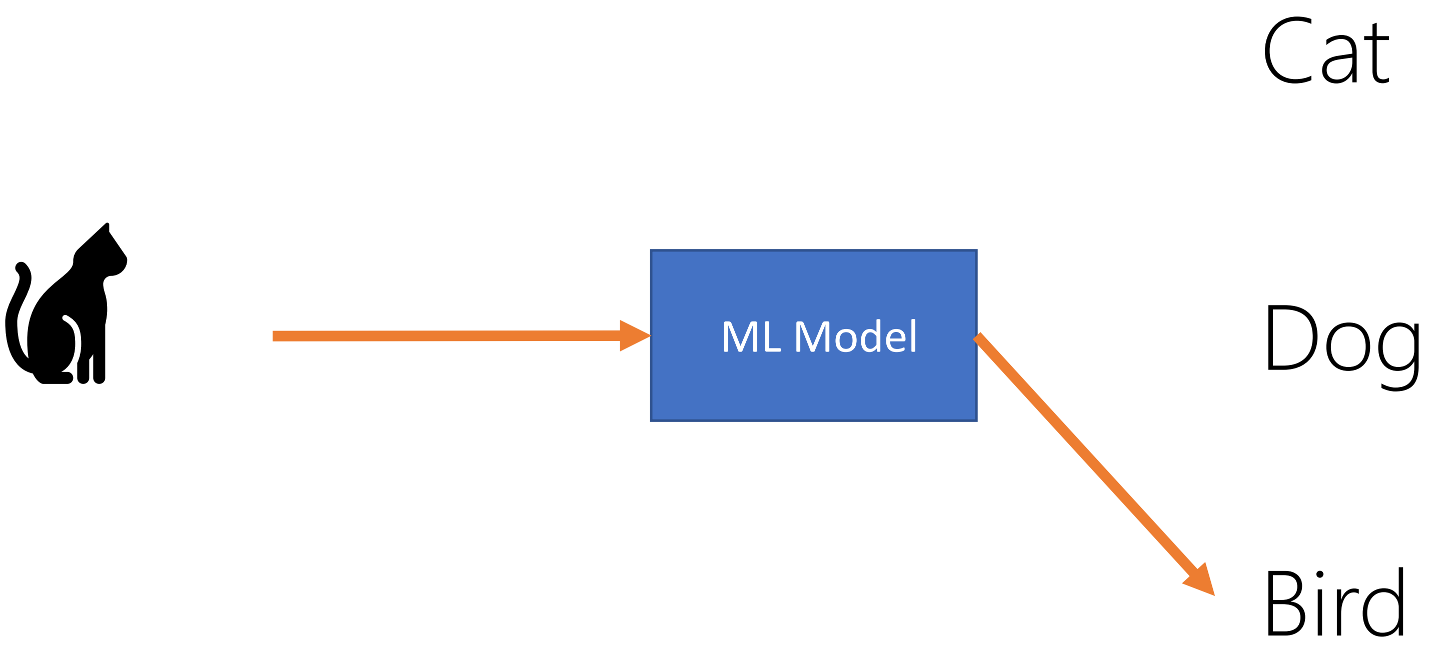
Broad types of ML Algorithms

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

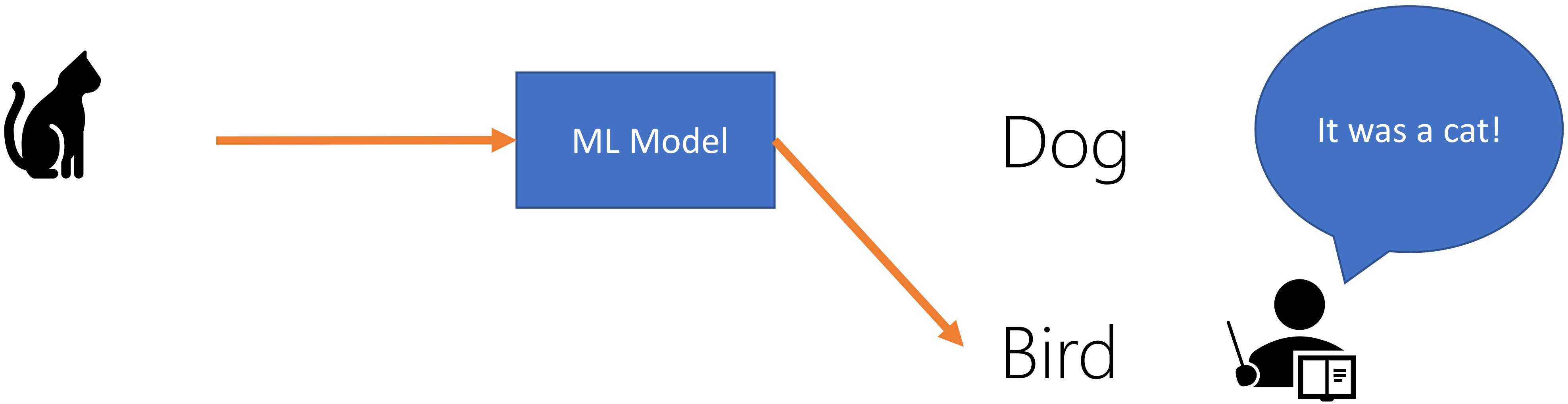
Supervised Learning

- Desired output is known!
- „teacher tells us right solution“

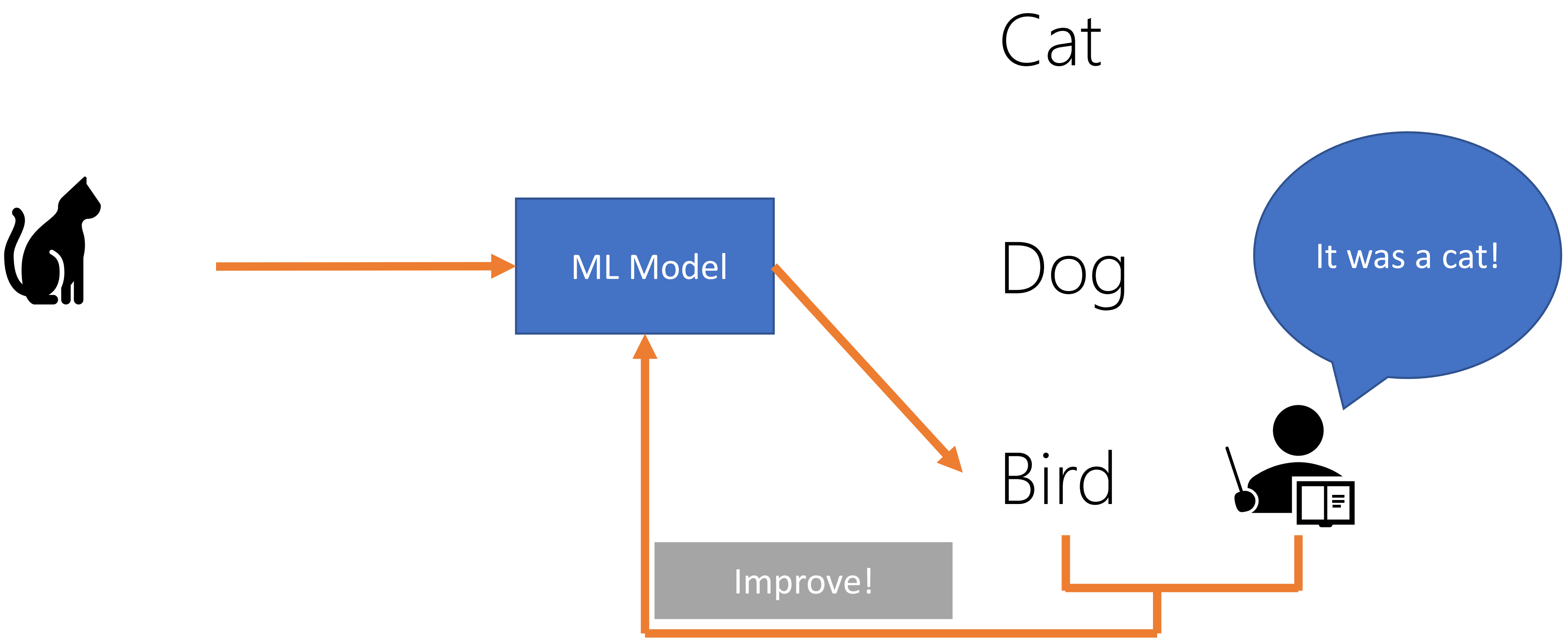
Classification



Classification

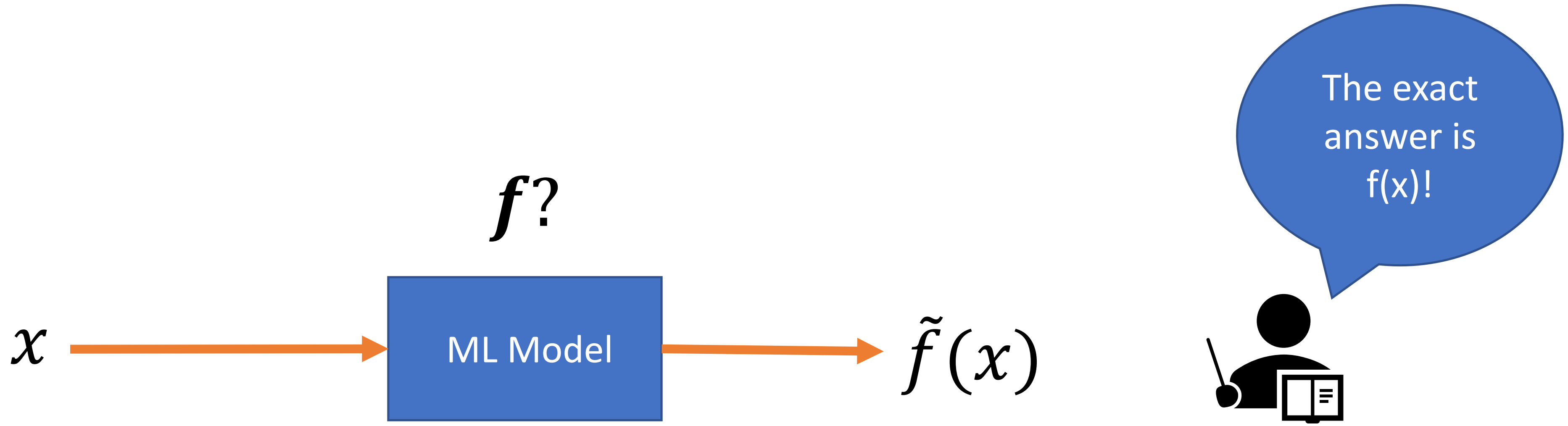


Classification

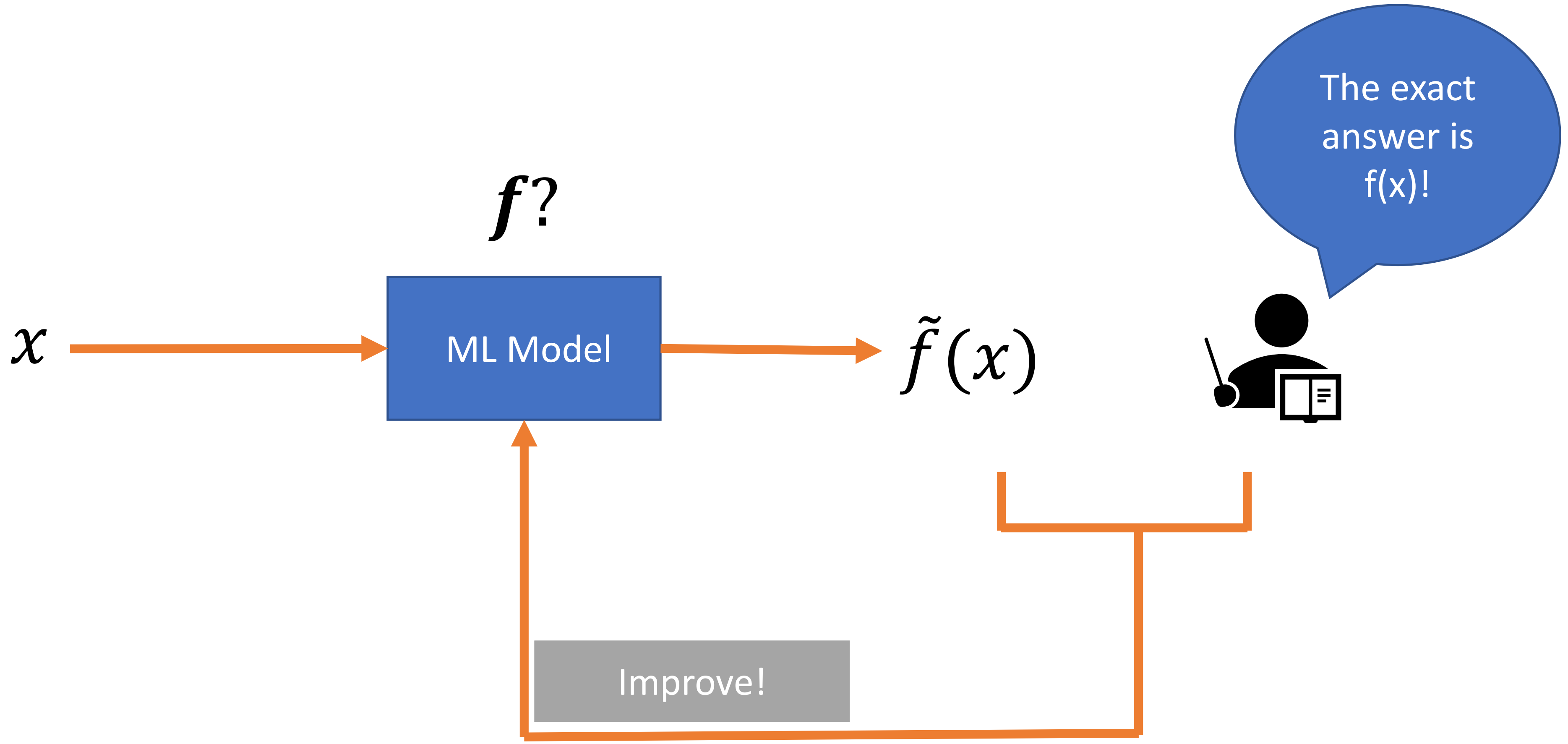


Regression

Regression



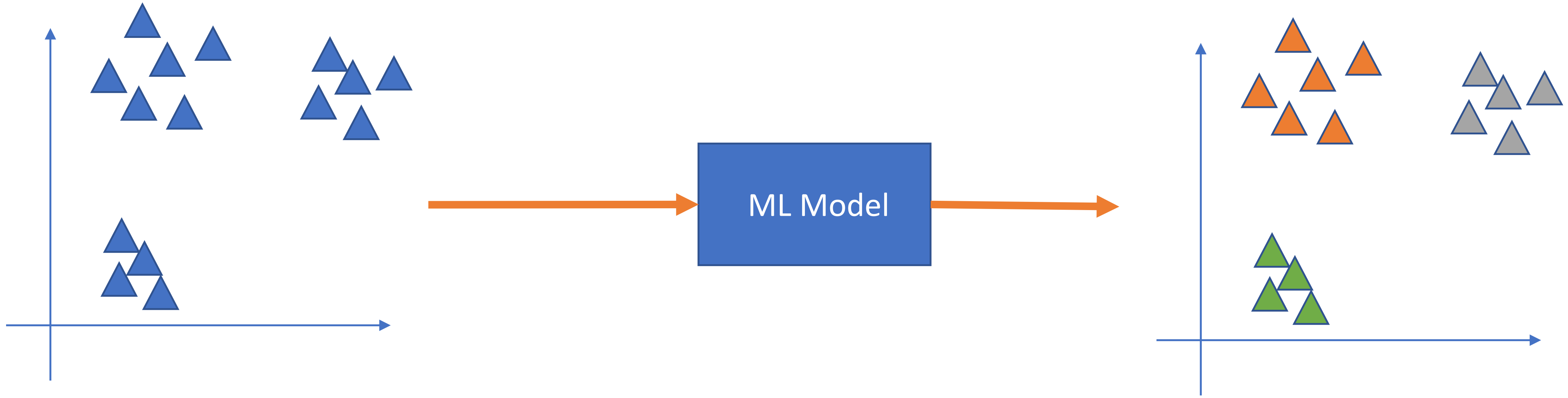
Regression



Unsupervised Learning

- Try to find useful properties/structures in example data set
- „no teacher   "

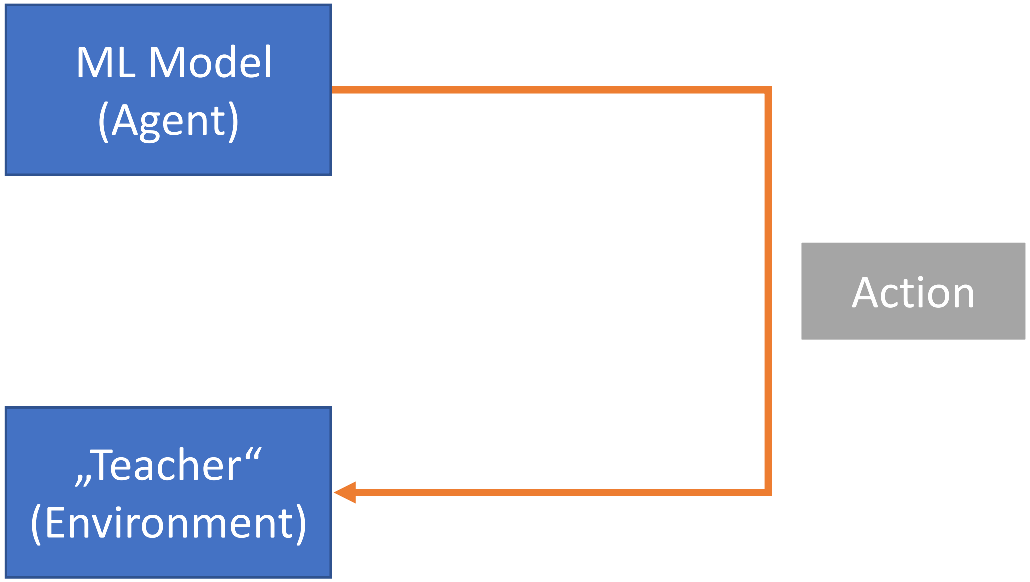
Unsupervised learning



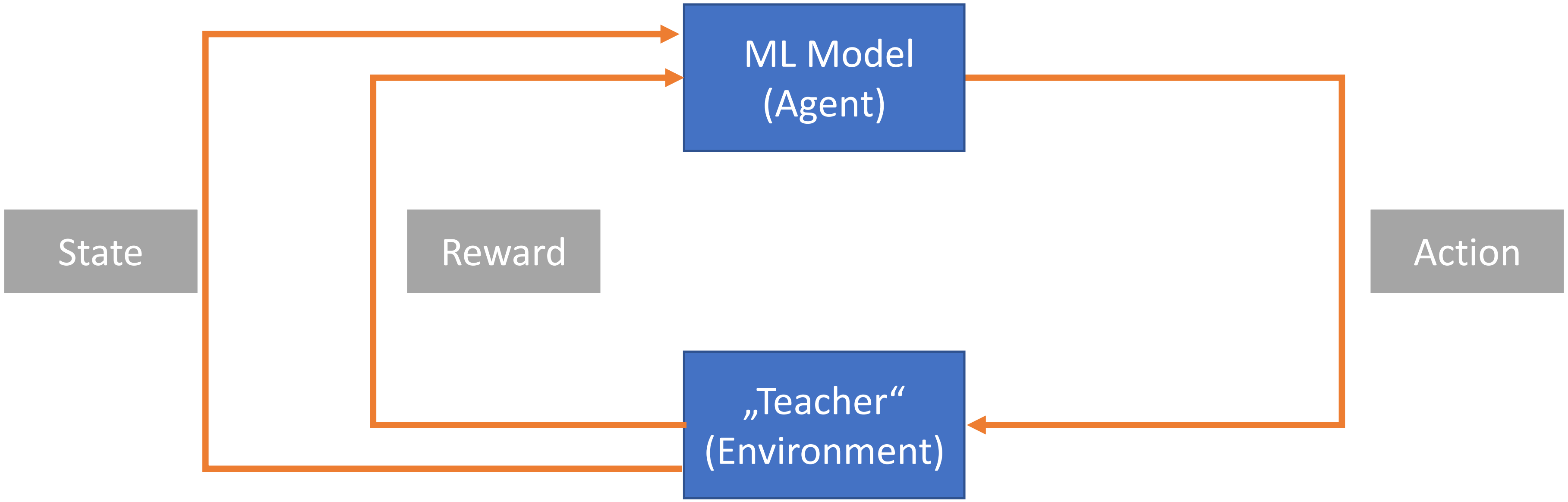
Reinforcement Learning

- Learning by punishment / reward
- Feedback loop between learning system and environment
- „teacher points to right direction“

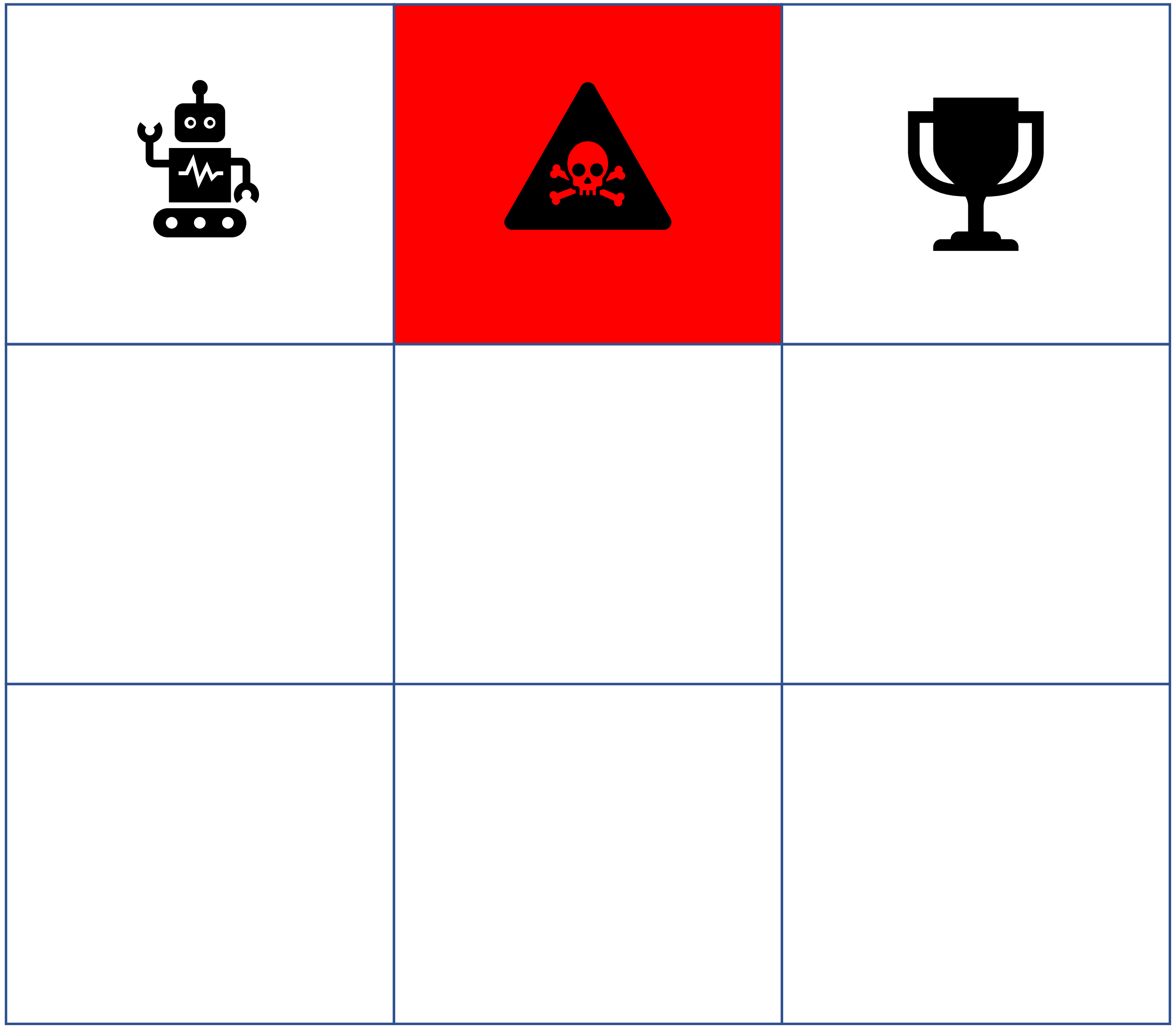
Reinforcement Learning



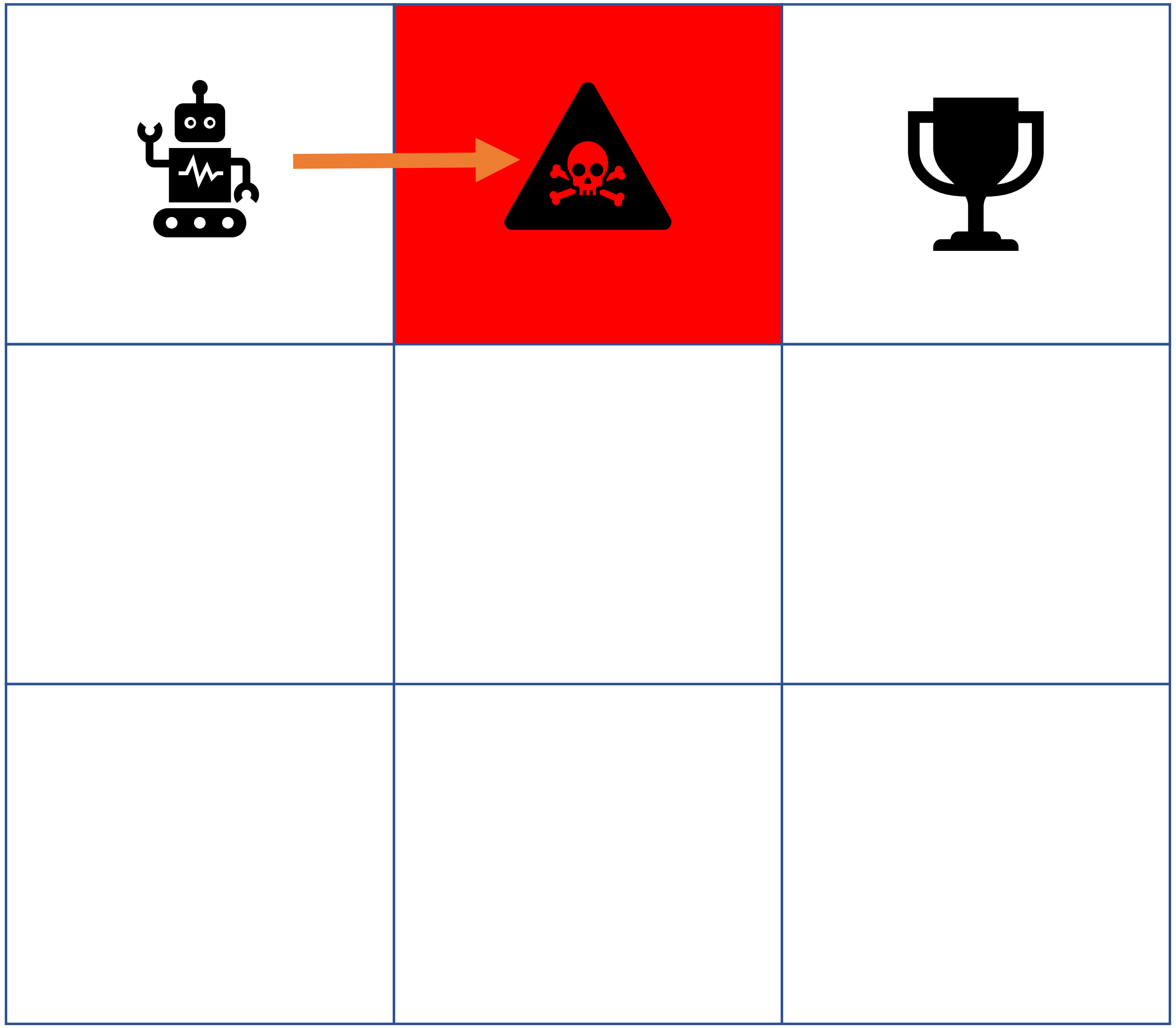
Reinforcement Learning



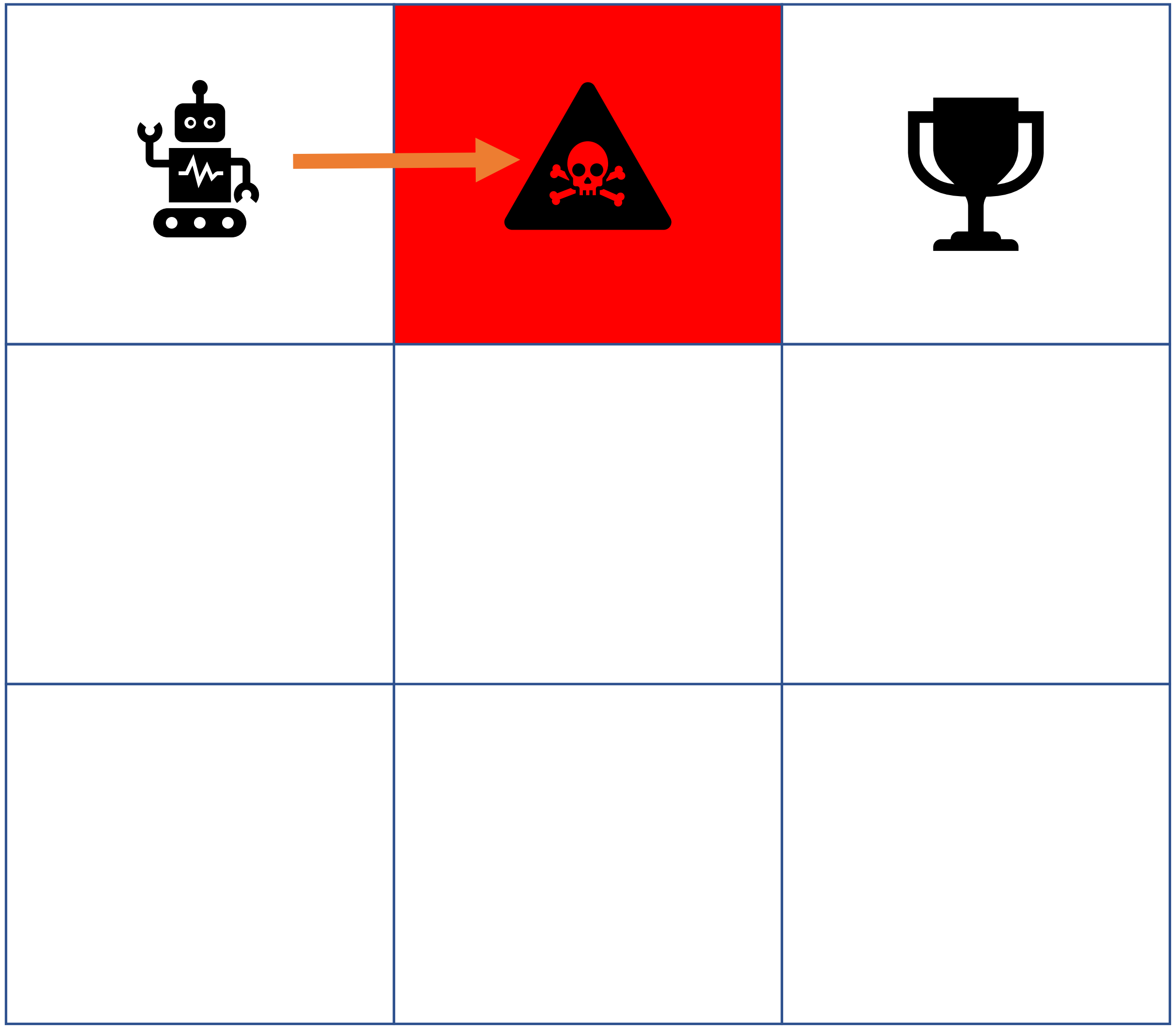
Reinforcement Learning

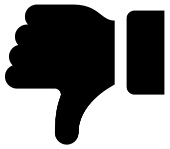


Reinforcement Learning

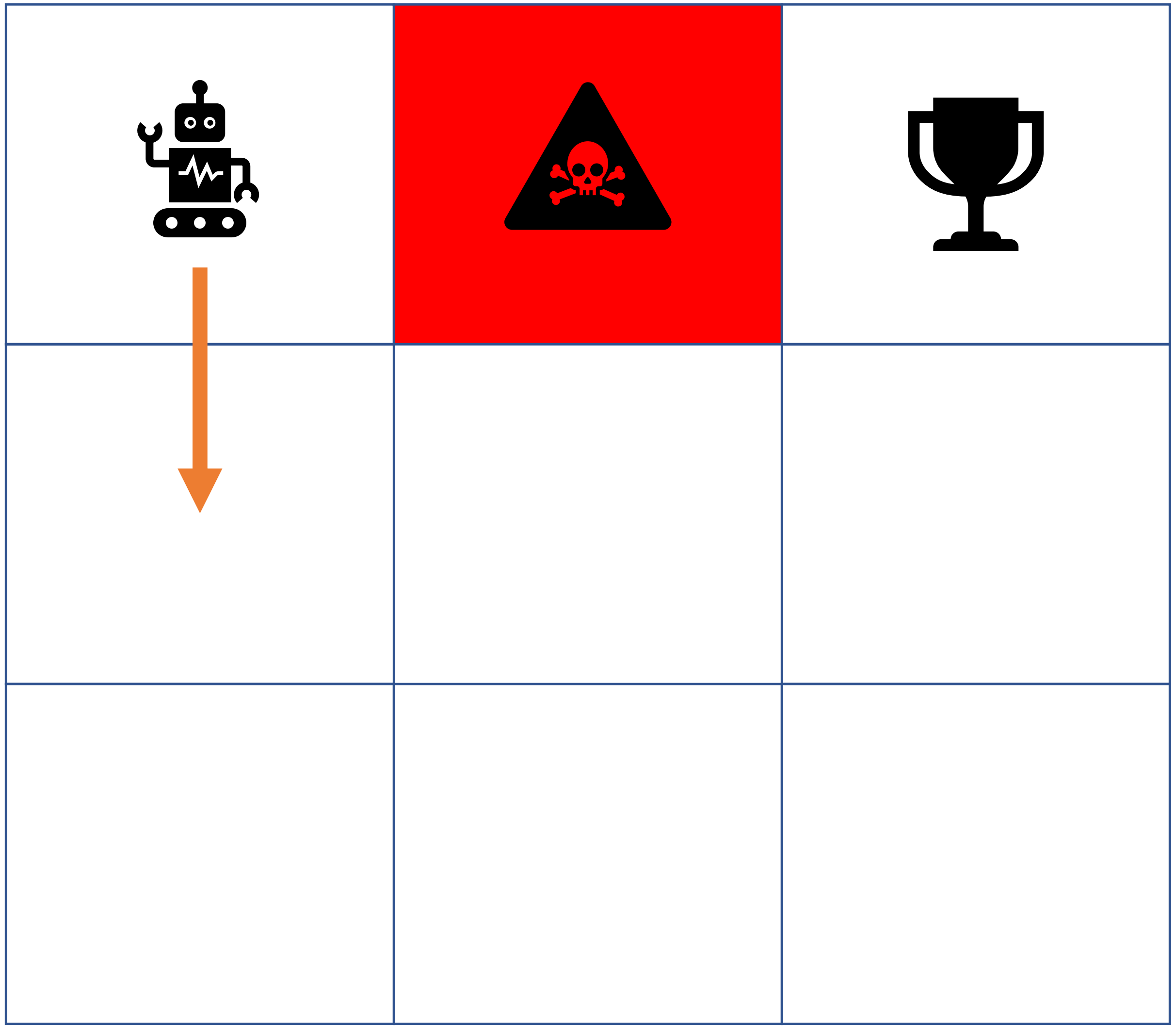


Reinforcement Learning

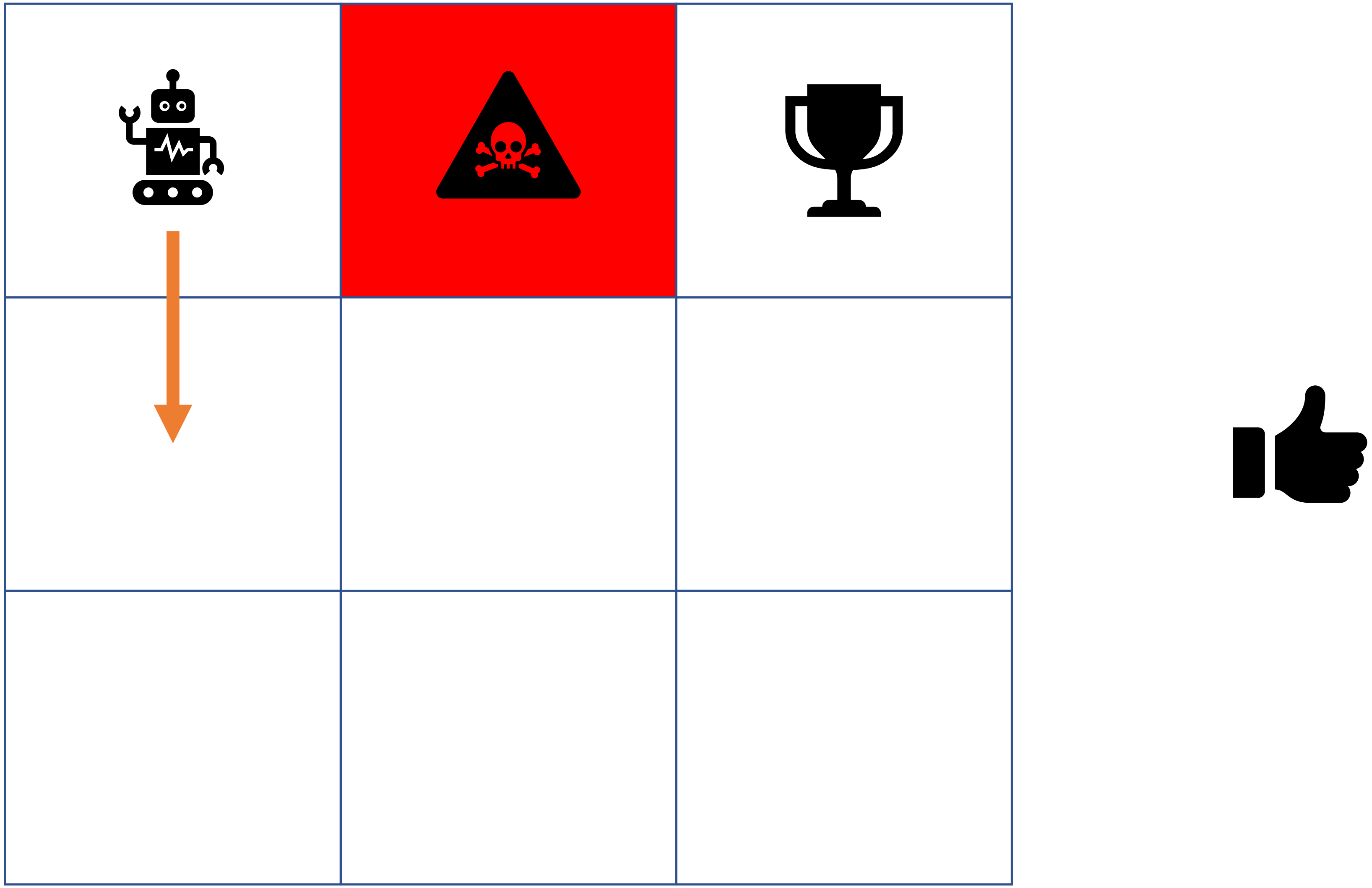




Reinforcement Learning



Reinforcement Learning



Reinforcement Learning

- Learning by punishment / reward
- Feedback loop between learning system and environment
- „teacher points to right direction“
- A little more complicated than illustration ;)
- More details in Computer Robot Systems

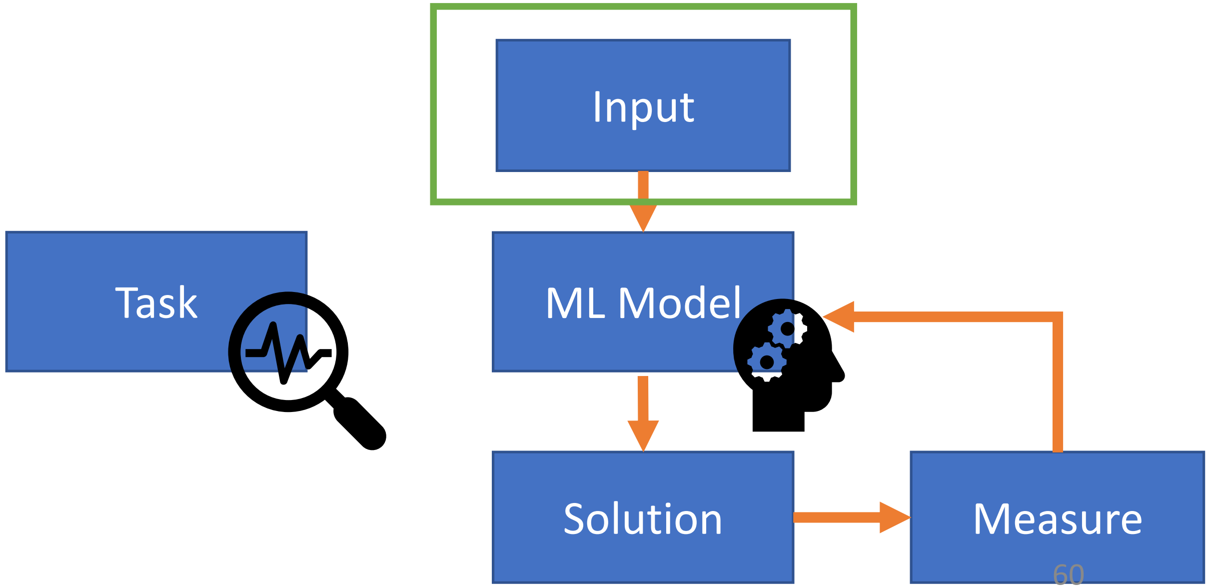
Content

- Motivation
- When does a Machine learn?
- Machine Learning tasks
- Broad types of Machine Learning Algorithms
- <u>How do input/output look like?</u>
- Data partitioning

When does a Machine learn?

Mitchell (1997):
„A computer is said to learn from experience E,
if its performance at tasks in T,
as measured by P,
improves with experience E*"*

0. Define task T
1. Try to solve task T with your Algorithm
2. Measure Algorithm performance by P
3. Gain experience E by doing so
4. Go to 1.



What does the input look like?

- Usually a vector $x \epsilon R^N$
- Either raw observation vector
- or feature vector,
  where each component may represent a specific feature

What are features?

- Salient properties of observation
- In generable measurable
- Sometimes needs to be extracted from observation

Example: Disease diagnosis

- Possible features (can be observed):
  - Oxygen partial pressure in blood
  - Carbon dioxide partial pressure
  - Heart rate
  - Etc …

Example: Disease diagnosis

- Possible features (can be observed):
  - Oxygen partial pressure in blood
  - Carbon dioxide partial pressure
  - Heart rate
  - Etc ...

$$x = \begin{pmatrix} heart\,rate \\ blood\,pressure \\ \cdots \\ nose\,length \end{pmatrix}$$

Example: Iris classification

- Raw observation: image

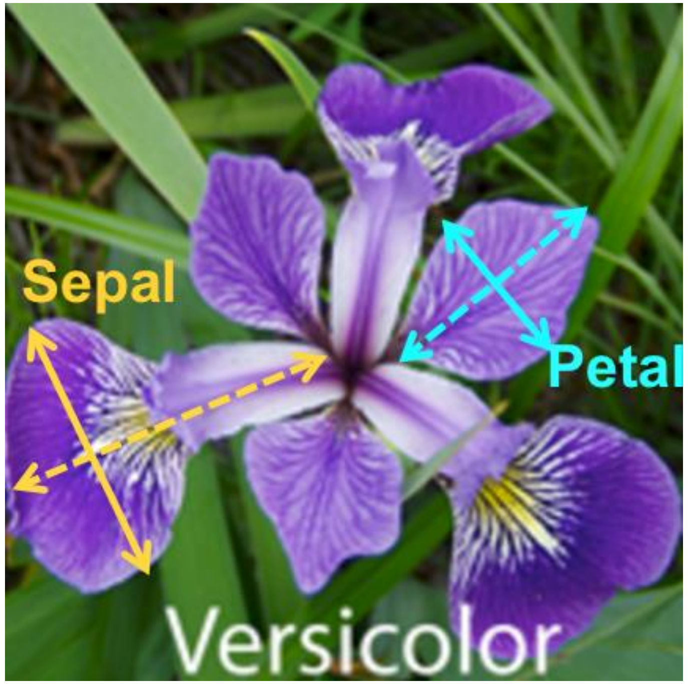
Iris setosa

Iris versicolor

Iris virginica

Example: Iris classification

- (Manually) extracted features from image:



$$x = \begin{pmatrix} sepal\ length \\ \vdots \\ petal\ width \end{pmatrix}$$

Example: Iris classification

- Often samples are stored in arrays!
- Be sure you know how the data is structured!
- Example:

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

Example: Iris classification

- Often samples are stored in arrays!
- Be sure you know how the data is structured!
- Example:

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

First sample

Example: Iris classification

- Often samples are stored in arrays!
- Be sure you know how the data is structured!
- Example:

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

Second sample

Example: Iris classification

- Often samples are stored in arrays!
- Be sure you know how the data is structured!
- Example:

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

Third sample

Example: Iris classification

- Often samples are stored in arrays!
- Be sure you know how the data is structured!
- Example:

Sepal length

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

Example: Iris classification

- Often samples are stored in arrays!
- Be sure you know how the data is structured!
- Example:

Sepal width

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

Example: Iris classification

- Often samples are stored in arrays!
- Be sure you know how the data is structured!
- Example:

Petal length

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

Example: Iris classification

- Often samples are stored in arrays!
- Be sure you know how the data is structured!
- Example:

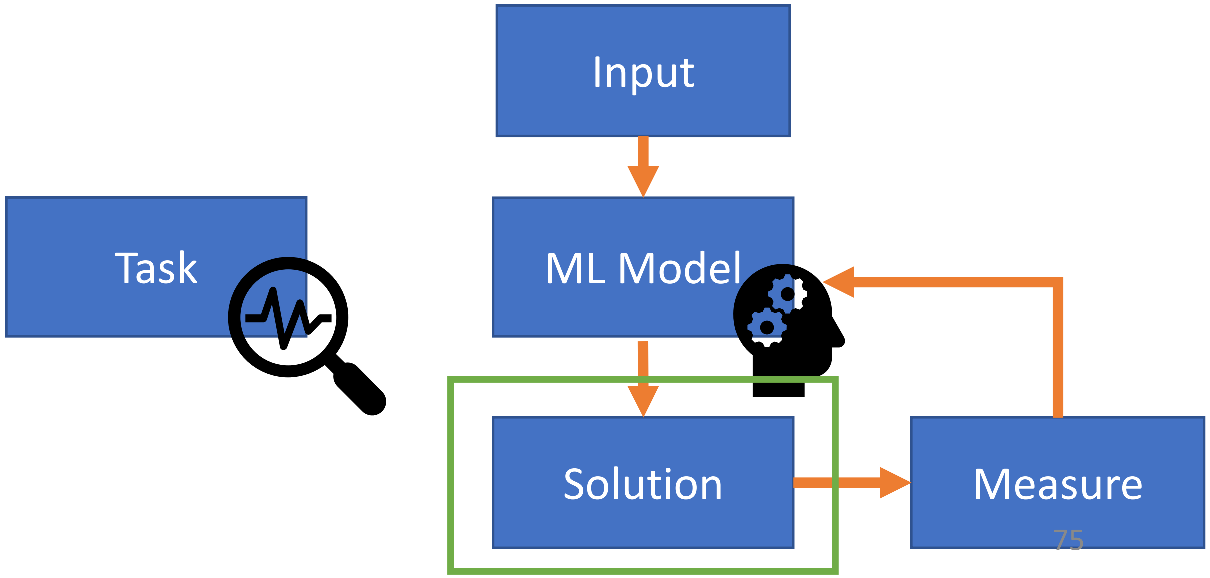
Petal width

```
[[5.1 3.5 1.4 0.2]
 [4.9 3.  1.4 0.2]
 [4.7 3.2 1.3 0.2]
 [4.6 3.1 1.5 0.2]
 [5.  3.6 1.4 0.2]
 [5.4 3.9 1.7 0.4]
 [4.6 3.4 1.4 0.3]
 [5.  3.4 1.5 0.2]
 [4.4 2.9 1.4 0.2]
 [4.9 3.1 1.5 0.1]]
```

When does a Machine learn?

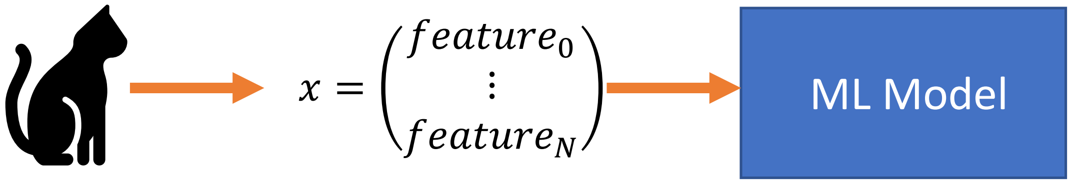
Mitchell (1997):
„A computer is said to learn from experience E*,*
if its performance at tasks in T*,*
as measured by P*,*
improves with experience E*"*

0. Define task T
1. Try to solve task T with your Algorithm
2. Measure Algorithm performance by P
3. Gain experience E by doing so
4. Go to 1.



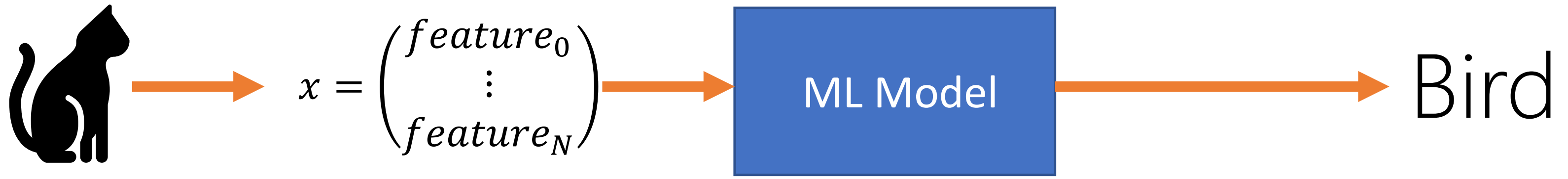
How does the output look like?

- For supervised classification (categorical output):



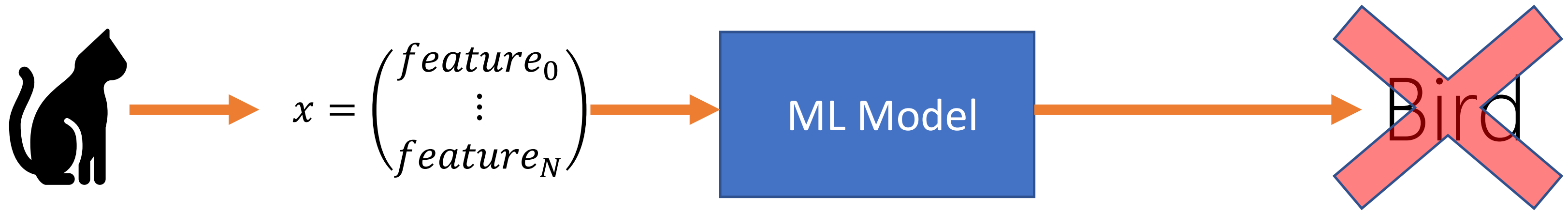
How does the output look like?

- For supervised classification (categorical output):



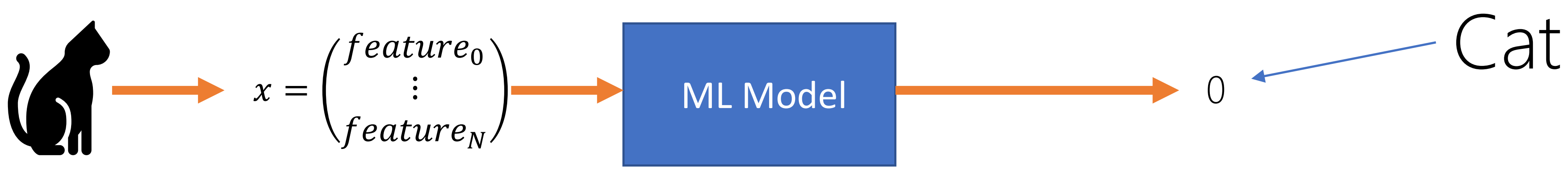
How does the output look like?

- For supervised classification (categorical output)



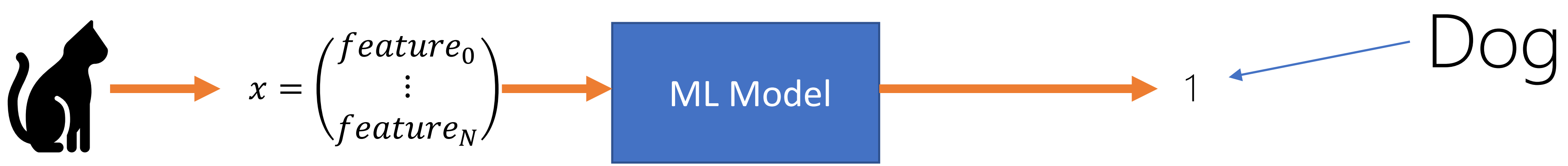
How does the output look like?

- For supervised classification (categorical output):
- Numerical class labels!



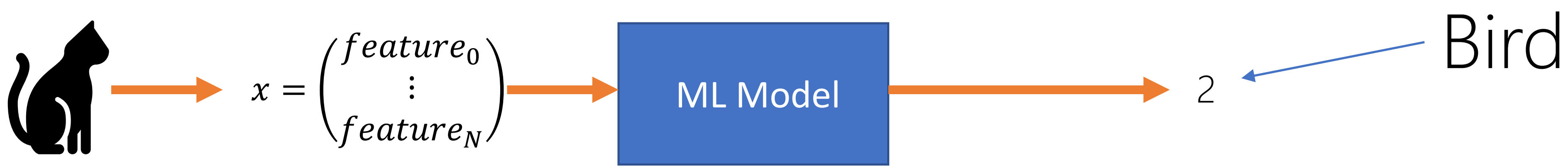
How does the output look like?

- For supervised classification (categorical output):
- Numerical class labels!



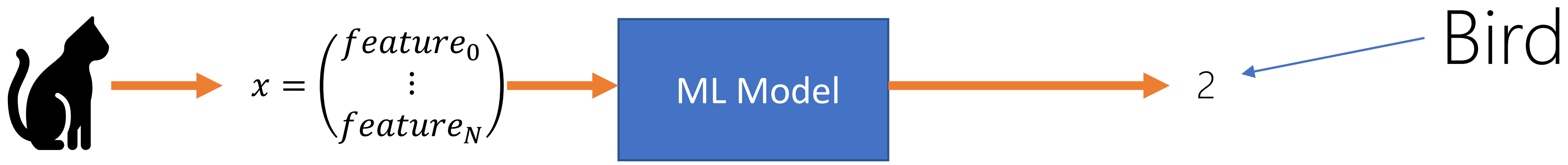
How does the output look like?

- For supervised classification (categorical output):
- Numerical class labels!



How does the output look like?

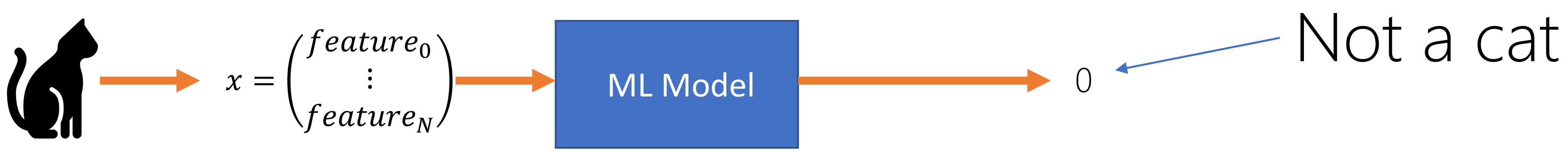
- For supervised classification (categorical output):
- Numerical class labels!



Rather uncommon for multi-class tasks!

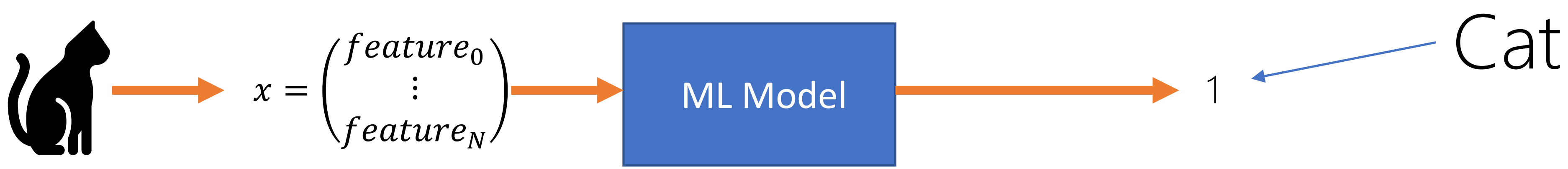
How does the output look like?

- For supervised classification (categorical output):
- Numerical class labels!
- More common for binary classification



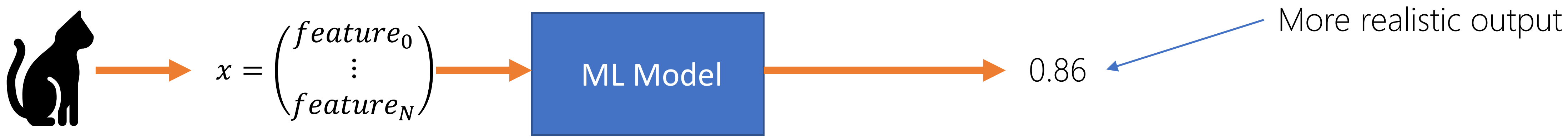
How does the output look like?

- For supervised classification (categorical output):
- Numerical class labels!
- More common for binary classification



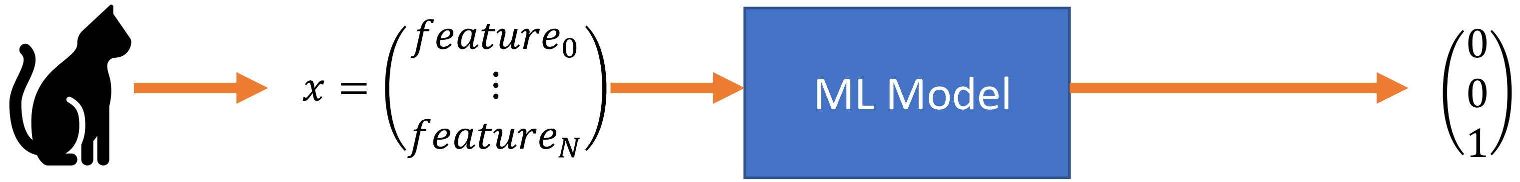
How does the output look like?

- For supervised classification (categorical output):
- Numerical class labels!
- More common for binary classification



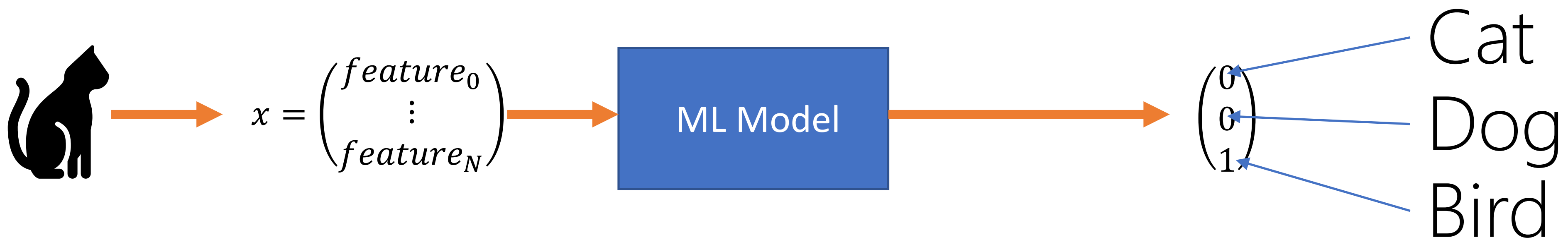
How does the output look like?

- For supervised classification (categorical output):
- One-hot-Encoding!



How does the output look like?

- For supervised classification (categorical output):
- One-hot-Encoding!



How does the output look like?

- For supervised classification (categorical output):
- One-hot-Encoding!

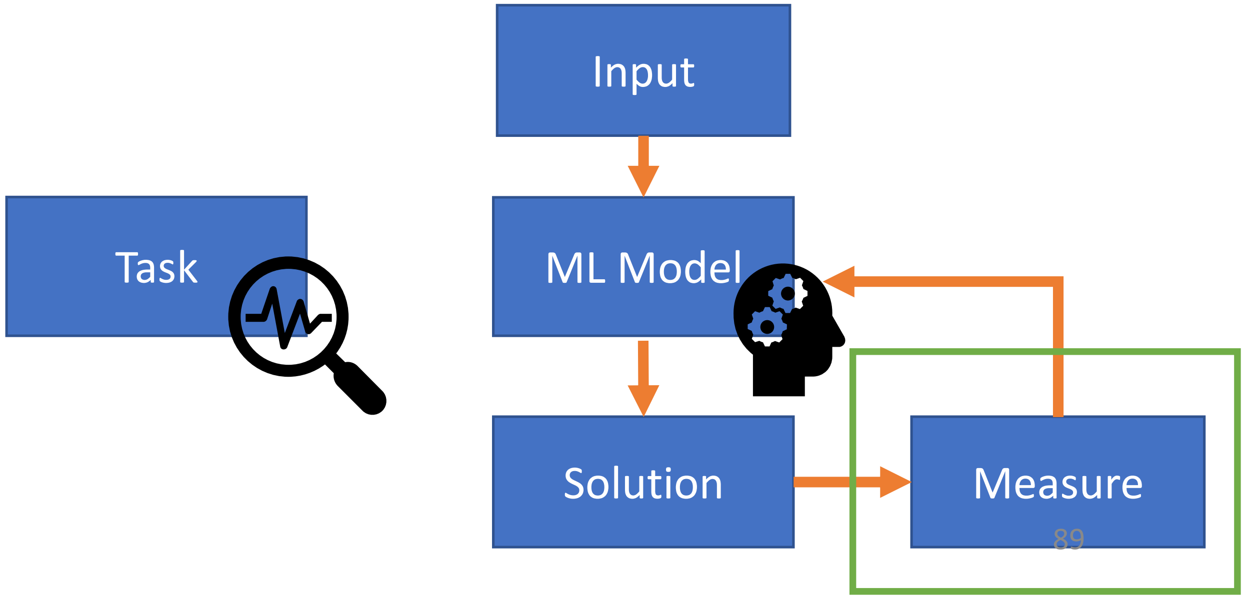
$$x = \begin{pmatrix} feature_0 \\ \vdots \\ feature_N \end{pmatrix} \rightarrow \text{ML Model} \rightarrow \begin{pmatrix} 0.33 \\ 0.01 \\ 0.66 \end{pmatrix}$$

More realistic output

When does a Machine learn?

Mitchell (1997):
„A computer is said to learn from experience E*,*
if its performance at tasks in T*,*
as measured by P*,*
improves with experience E*"*

0. Define task T
1. Try to solve task T with your Algorithm
2. Measure Algorithm performance by P
3. Gain experience E by doing so
4. Go to 1.



Content

- Motivation
- When does a Machine learn?
- Machine Learning tasks
- Broad types of Machine Learning Algorithms
- How do input/output look like?
- <u>Data partitioning</u>

Caution!

- Differentiate between performance measure:
  - During learning phase (training)
    -> to improve ML model
  - After learning (testing)
    -> to estimate how good your model is on unseen data

Data partitioning

- Split your data into
  - Training data
    -> use this data to improve model during learning phase

  - Validation data
    -> **do not** use this data during learning!
    -> use it to measure model performance on unseen data
    -> use measurement for hyperparamter tuning!

  - Testing data
    -> **do not** use this data during learning!
    -> use it to measure model performance on unseen data
    -> **do not** use measurement for hyperparamter tuning!

Summary

- Motivation
- When does a Machine learn?
- Machine Learning tasks
- Broad types of Machine Learning Algorithms
- How do input/output look like?
- Data partitioning

Outlook – Biologically inspired

- McCulloch-Pitts Cell [Tutorial]
- Perceptron [Lecture + Tutorial]
- AdaLine [Lecture + Tutorial]
- Multilayer-Perceptron (MLP) [Lecture + Tutorial]
- Convolutional Neural Networks (CNN) [Tutorial]
- Radial Basis Function-Networks (RBF-Network) [Lecture + Tutorial]

Outlook – Non-Biologically inspired

- Naive Bayes Classifier [Tutorial]
- K-Means Clustering [Lecture]
- Support Vector Machines (SVM) [Lecture + Tutorial]

Further interesting ML algorithms

- Neural Gas
- Self Organizing Maps (SOMs)
- Random Forest
- AdaBoost
- Deep Learning in general
  (we only shortly cover CNNs)

Relation to AI