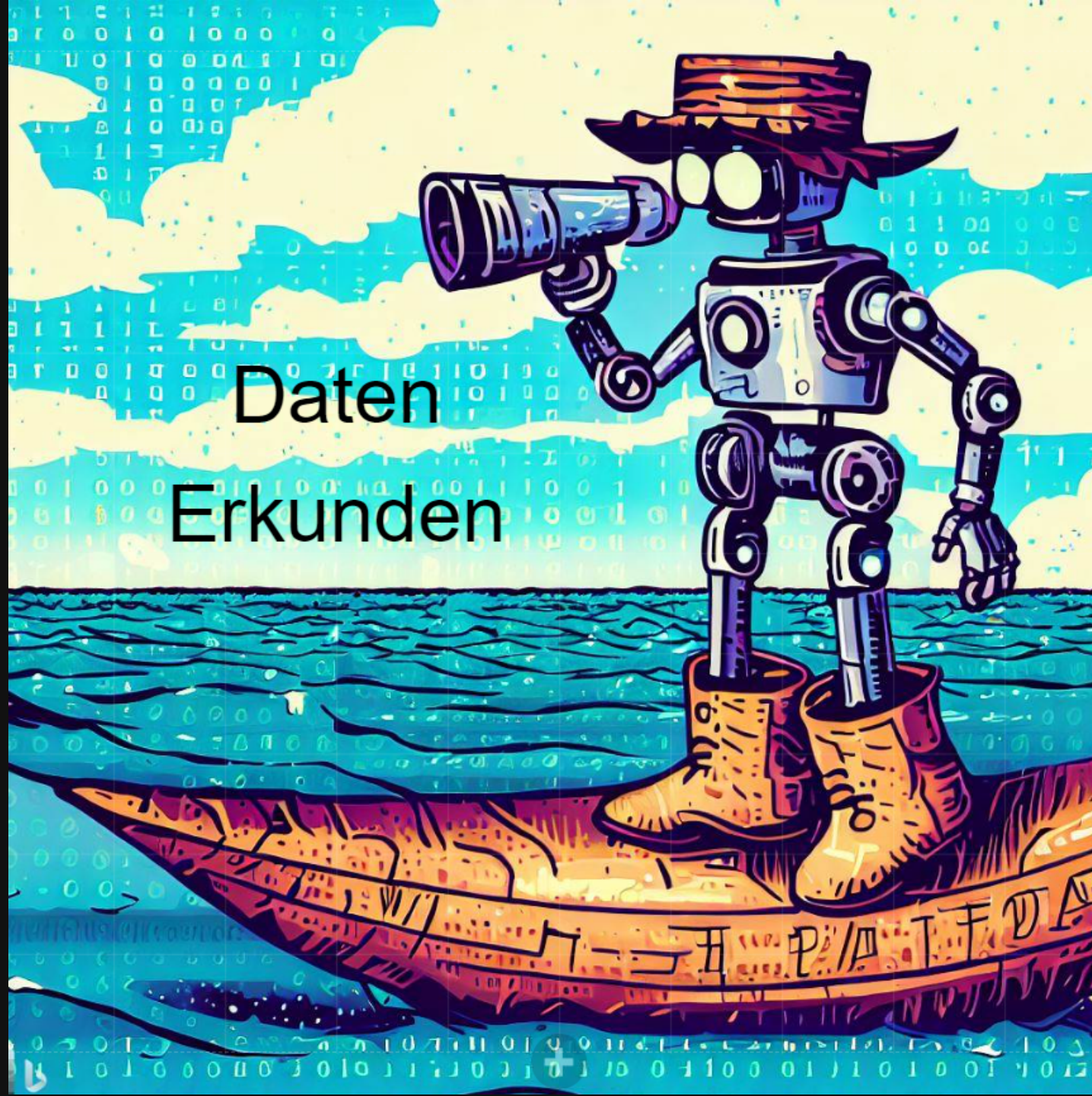


Daten Erkunden



<https://www.youtube.com/embed/KrW3vneNe2U?enablejsapi=1>

SUPERVISED LEARNING



UNSUPERVISED LEARNING



REINFORCEMENT LEARNING

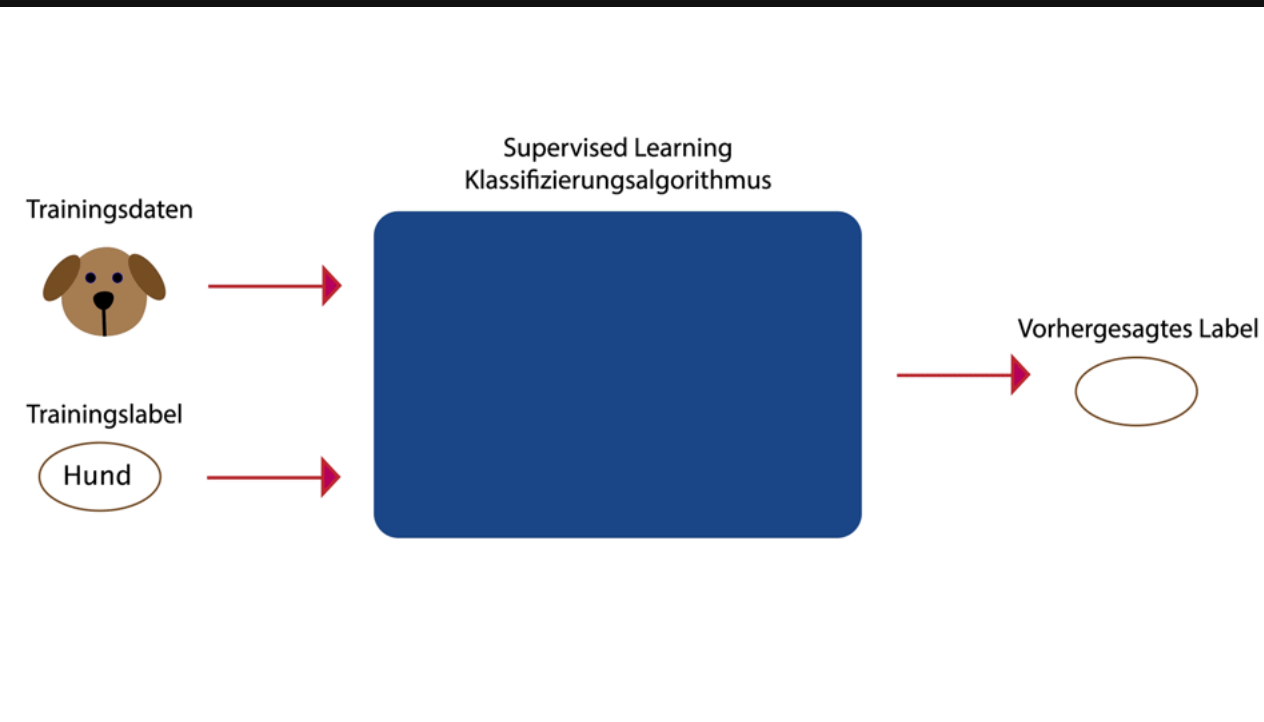


Mehr zum Thema

Supervised

Lernen mit Zieldaten (Label / Target)

<https://www.youtube.com/embed/vWijy-YZxu0?enablejsapi=1>

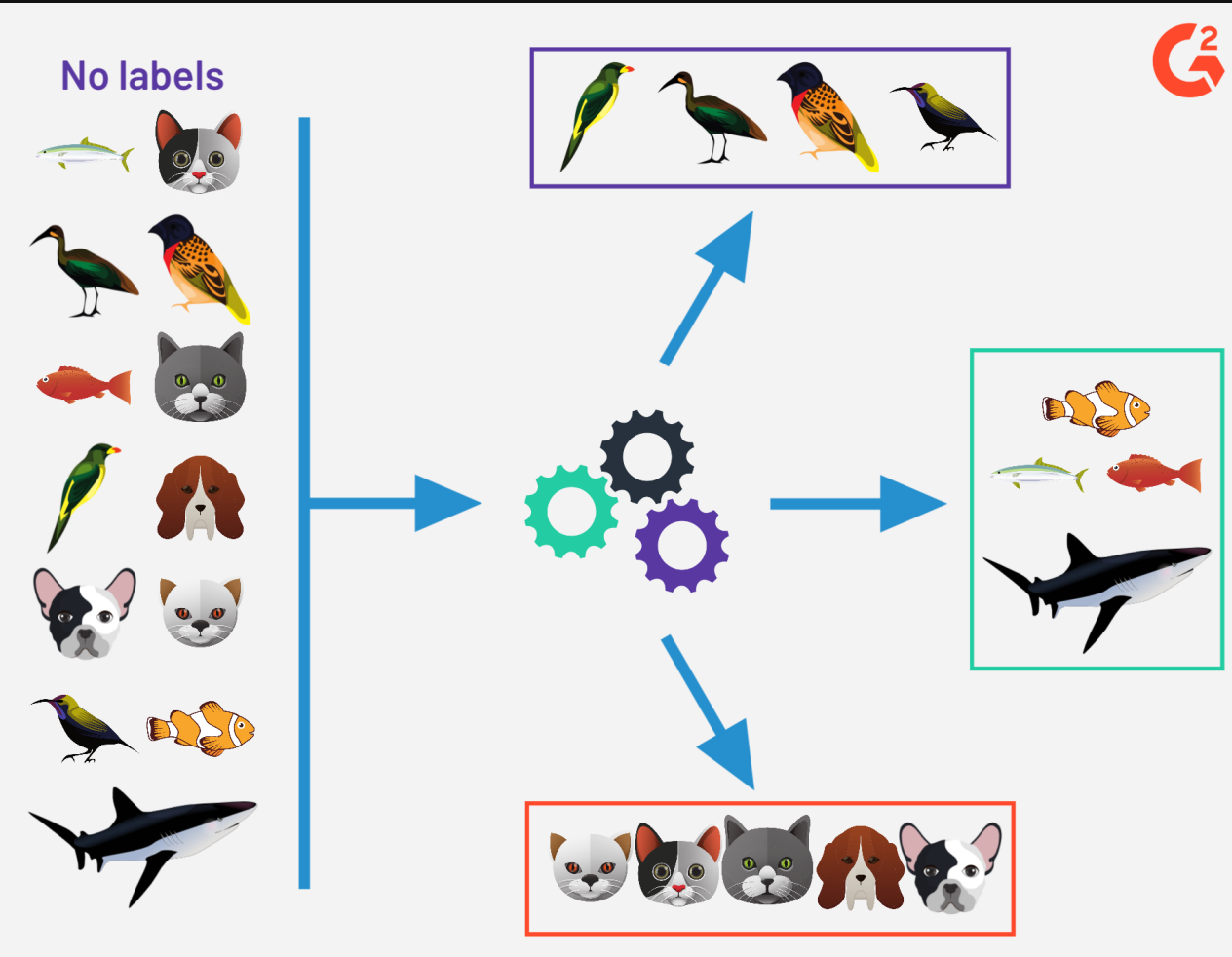


- Target fuer jeden Datensatz
- Training durch vergleich Output = Target
- Klassifizierung, Spracherkennung

Unsupervised

Lernen ohne Zieldaten

<https://www.youtube.com/embed/p3RfGkC-pHU?enablejsapi=1>



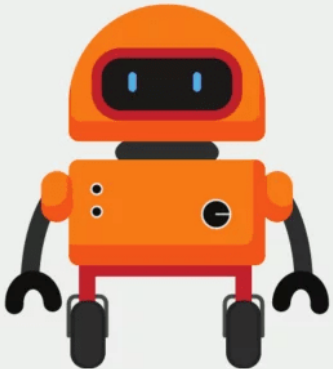
- Kein Target pro Datensatz
- Muster & Strukturen erkennen
- Clustering, Dimensionsreduktion

Reinforcement

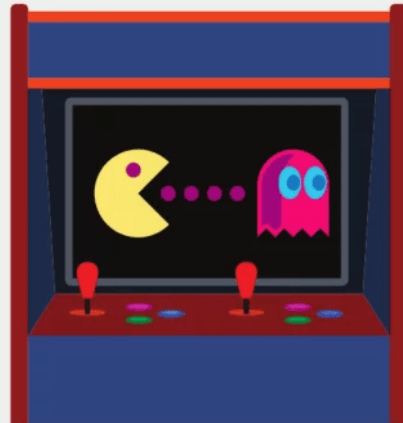
Interaktion mit Umgebung

https://www.youtube.com/embed/ICX-L_H5BKA?enablejsapi=1

Agent



Environment



Actions



Rewards



Observations



- Lernen durch Interaktion
- Belohnung & Bestrafung
- Spiele, Roboter, Autonome Systeme

Daten Erkunden

https://www.youtube.com/embed/Sk_cPEWDKa4?enablejsapi=1

- Verstehen: Inhalt, Format, Label, Metadaten
- Visualisieren: Verteilung, Korrelation, Ausreisser
- Preprocessing: normalisieren, skalieren, enkodieren, balancieren

Daten Erkunden

<https://www.youtube.com/embed/T3MX94rx71o?enablejsapi=1>

Hands-On: MNIST Datensatz

Welche Daten enthält der Datensatz?

Die shape eines Datensatzes zeigt die Anzahl der Elemente (N_data) sowie das format der einzelnen Elemente (size_input)

```
1 data.shape -> (N_data, size_input)
```

Daten Erkunden

Hands-On: MNIST Datensatz
Welches Format haben die Daten?

<https://www.youtube.com/embed/c-gqSSRz8Jk?enablejsapi=1>

Die `type(x)` Funktion gibt die Klasse von `x` an

```
1 type(data[0]) -> class
```

Die built-in Funktion `x.dtype` gibt den Datentyp von `x` an

```
1 data.dtype -> data_type
```


Daten Erkunden

<https://www.youtube.com/embed/swzmrTXIfkU?enablejsapi=1>

Hands-On: MNIST Datensatz

Welche Klassen gibt es und wie sind diese verteilt?

`numpy.unique(x)` liefert eine liste aller Elemente die in x vorkommen

```
1 labels = np.unique(target)
```

`numpy.bincount(x)` liefert die Anzahl von Integerwerten in x, geordnet nach Zahlenwert der Integer

```
1 counts = np.bincount(target.astype(int))
```

Daten Erkunden

<https://www.youtube.com/embed/6YUPtt1gLBE?enablejsapi=1>

Hands-On: MNIST Datensatz

Wie machen wir die Klassen dem Modell verständlich?

Um Stringlabel in für das Modell verständliche Floats zu verwandeln
nutzen wir One-Hot-Encoding

```
1 # zB "3" -> [0,0,0,1,0,0,0,0,0]
2 #
3 from sklearn.preprocessing import OneHotEncoder
4 encoder = OneHotEncoder()
5 labels = encoder.fit_transform(target)
```

Achtung: Die letzte Zeile erwartet hier target mit shape (N,1)

Daten Erkunden

<https://www.youtube.com/embed/PG53gcP159s?enablejsapi=1>

Hands-On: MNIST Datensatz

Welche Skalierung der Daten ist sinnvoll?

- Knoten: Summe vieler Werte -> sehr grosse Werte
- Features: grösseres Wertintervall ~ stärkeres Gewicht.
- => Input-Daten auf [-1,1] skaliert.
- idR mit Min-Max scaling

$$\frac{\text{data} - \min(\text{data})}{\max(\text{data}) - \min(\text{data})} \cdot 2 - 1$$

```
1 scaled_data = (data - np.min(data)) / (np.max(data) - np.min(data)) * 2 - 1
```

Daten Erkunden

Hands-On: MNIST Datensatz

<https://www.youtube.com/embed/PRY5MWJky-w?enablejsapi=1>

Öffnen sie [dieses Notebook](#) und bearbeiten Sie die Aufgaben.

Beantworten Sie so folgende Fragen:

1. Welche Daten enthält der Datensatz?
2. Welches Format haben die Daten?
3. Welche Klassen gibt es und wie sind diese verteilt?
4. Wie machen wir die Klassen dem Modell verständlich?
5. Welche Skalierung der Daten ist sinnvoll?

Die Lösung finden Sie in [diesem Notebook](#)