

Künstliche Intelligenz kapieren und programmieren

Teil 4: Lernen durch Beobachten

Michael Weigend
Universität Münster



mw@creative-informatics.de
www.creative-informatics.de
2024



Foto: cc-by 2008 Steve Lodefink, in Wikimedia Commons

Materialien bei GitHub:
<https://github.com/mweigend/ki-workshop>

Tag 1

Zeit	Thema	Inhalte
9.00	Denkende Maschinen	Pädagogische Konzepte, Einstieg in Python, Chatbots und Assistenzsysteme
11.15	KI als Spielgegner	Modellieren mit Listen, Nim-Spiel mit KI als Gegner
12.45	<i>Mittagspause</i>	
13.45	Klassifizieren	Entscheidungsbaum, k-Means-Clustern
15.00	Lernen durch Beobachten	Lernfähiger Währungsrechner, Fußgänger erkennen
16.00	<i>Ende</i>	

Wie lernt man Boules spielen?




In einer fremden Währung bezahlen



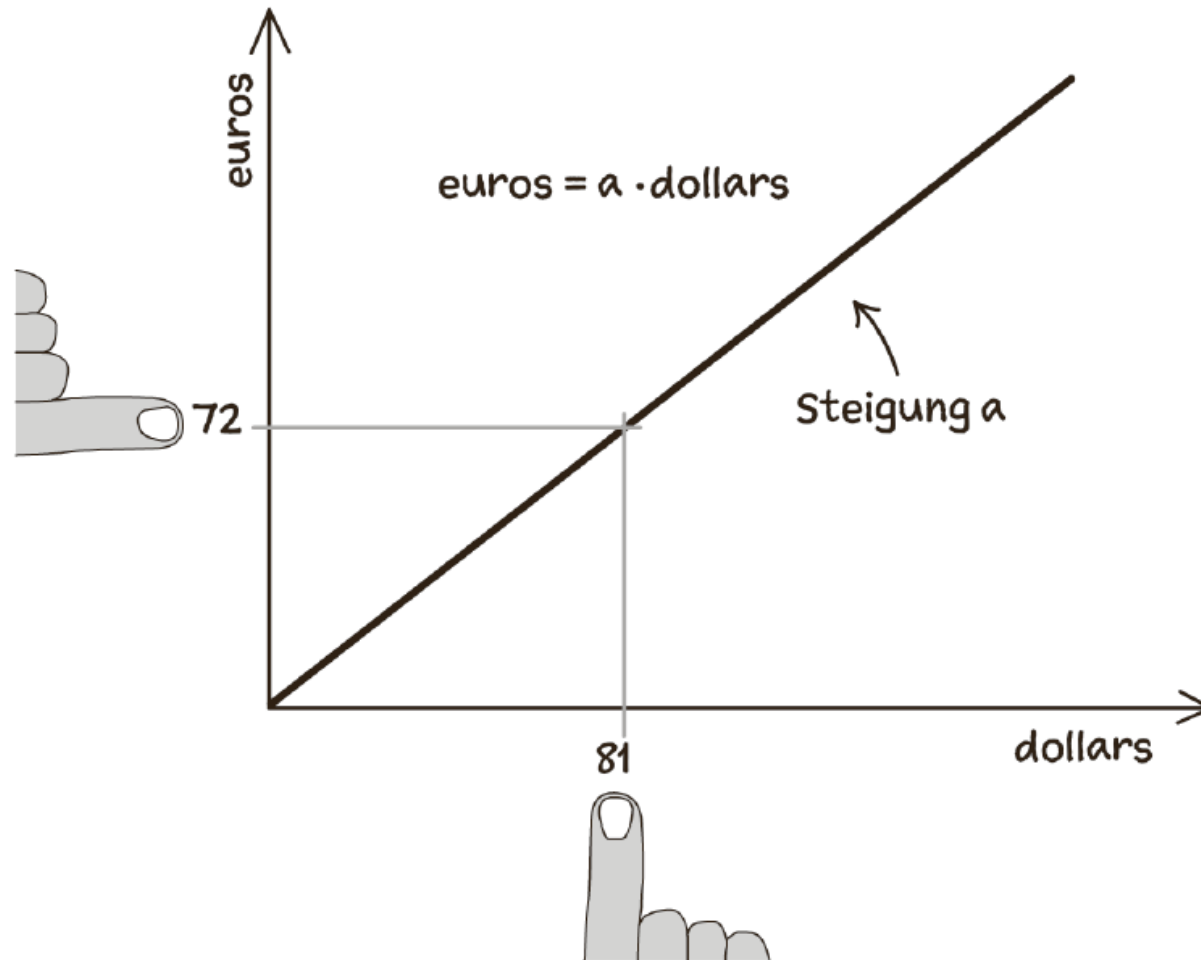
Projekt: Währungsrechner, der aus Beobachtungen lernt

Produkt	Preis in Dollar	Preis in Euro
Jeans	81	72
Turnschuhe	61.99	57.71
T-Shirt	9.50	9.00
Tablet-Hülle	23.60	20.77



Programm
training.py
ausprobieren

Lineare Beziehung



Lernender Währungsrechner

```
# training.py
a = 1
eingabeDollars = input('Betrag in Dollar: ')
while eingabeDollars != '':
    dollars = float(eingabeDollars)
    vorhersageEuros = a * dollars
    eingabeEuros = input('Tatsächlicher Betrag in Euro: ')
    euros = float(eingabeEuros)
    fehler = euros - vorhersageEuros

    a += 0.5 * fehler/dollars

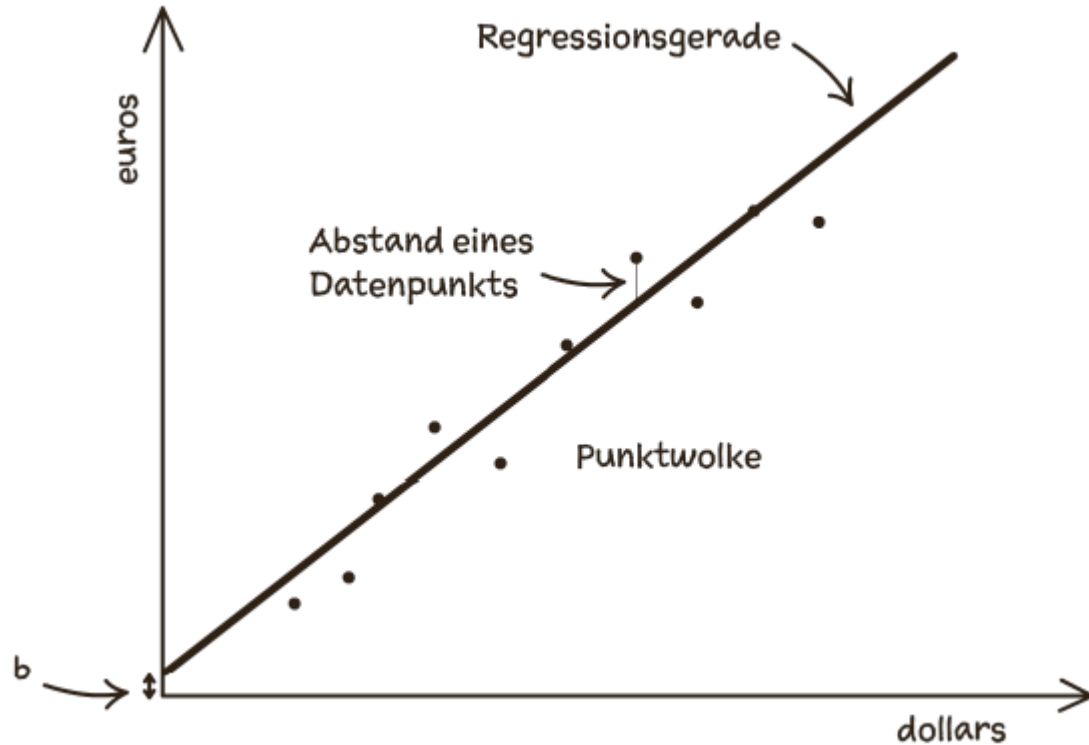
    print('Vorhersage: ', round(vorhersageEuros, 2), '€')
    print('Fehler:', fehler)
    print('Neuer Wechselkurs a:', round(a, 4))
    eingabeDollars = input('Betrag in Dollar: ')
print('Auf Wiedersehen!')
input()
```

Warum dieser
Faktor?

Absoluter Fehler

Fehler der
Steigung a

Regressionsgerade



Liefert das Training eine Regressionsgerade?

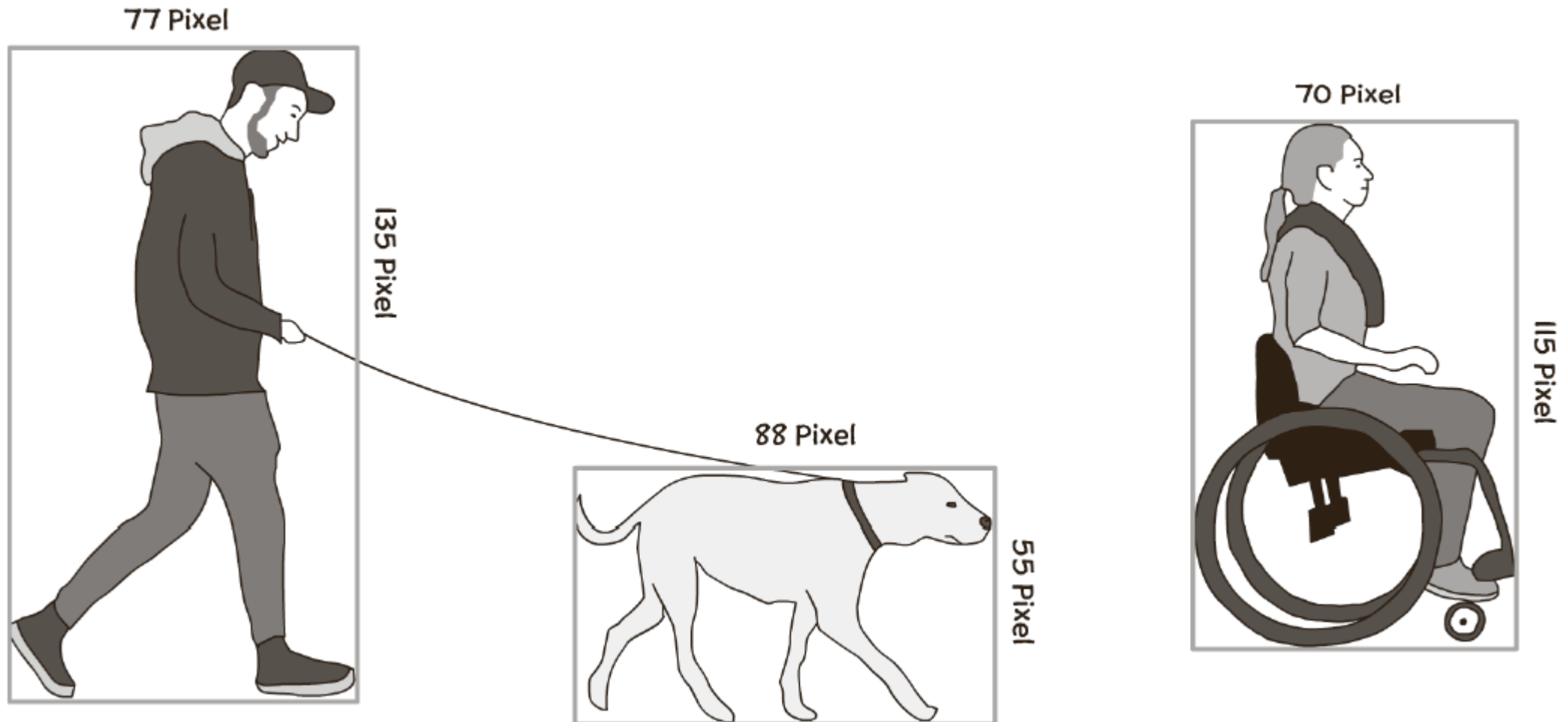
$$\text{euros} = a \cdot \text{dollars} + b$$

$$a = \frac{\sum_{i=1}^n (x_i - \bar{x}) \cdot (y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

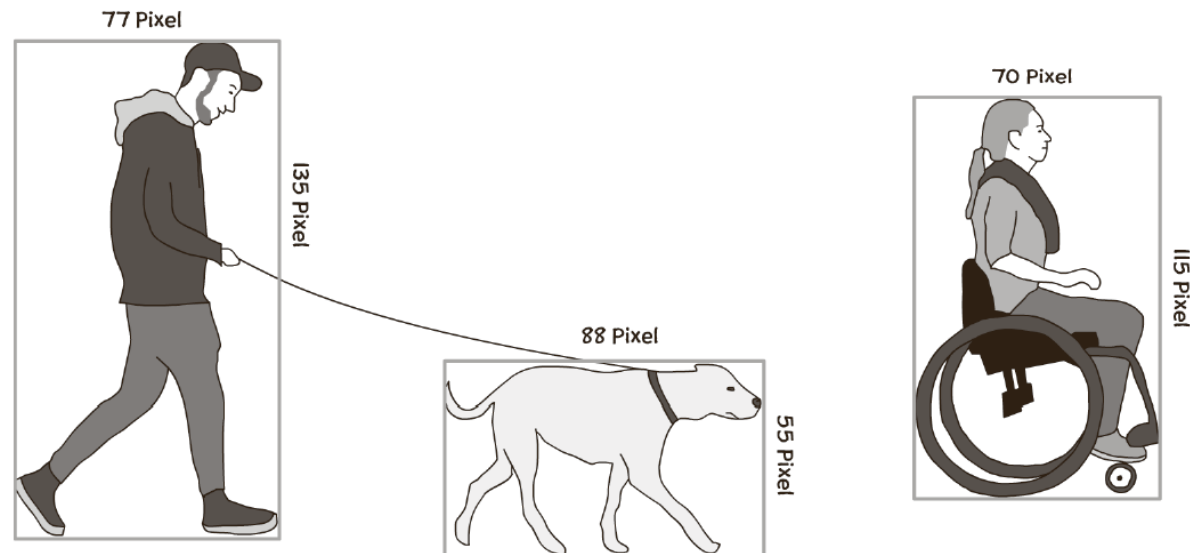
$$b = \bar{y} - a \cdot \bar{x}$$

Datenpunkte haben einen möglichst geringen Abstand von der Regressionsgeraden.

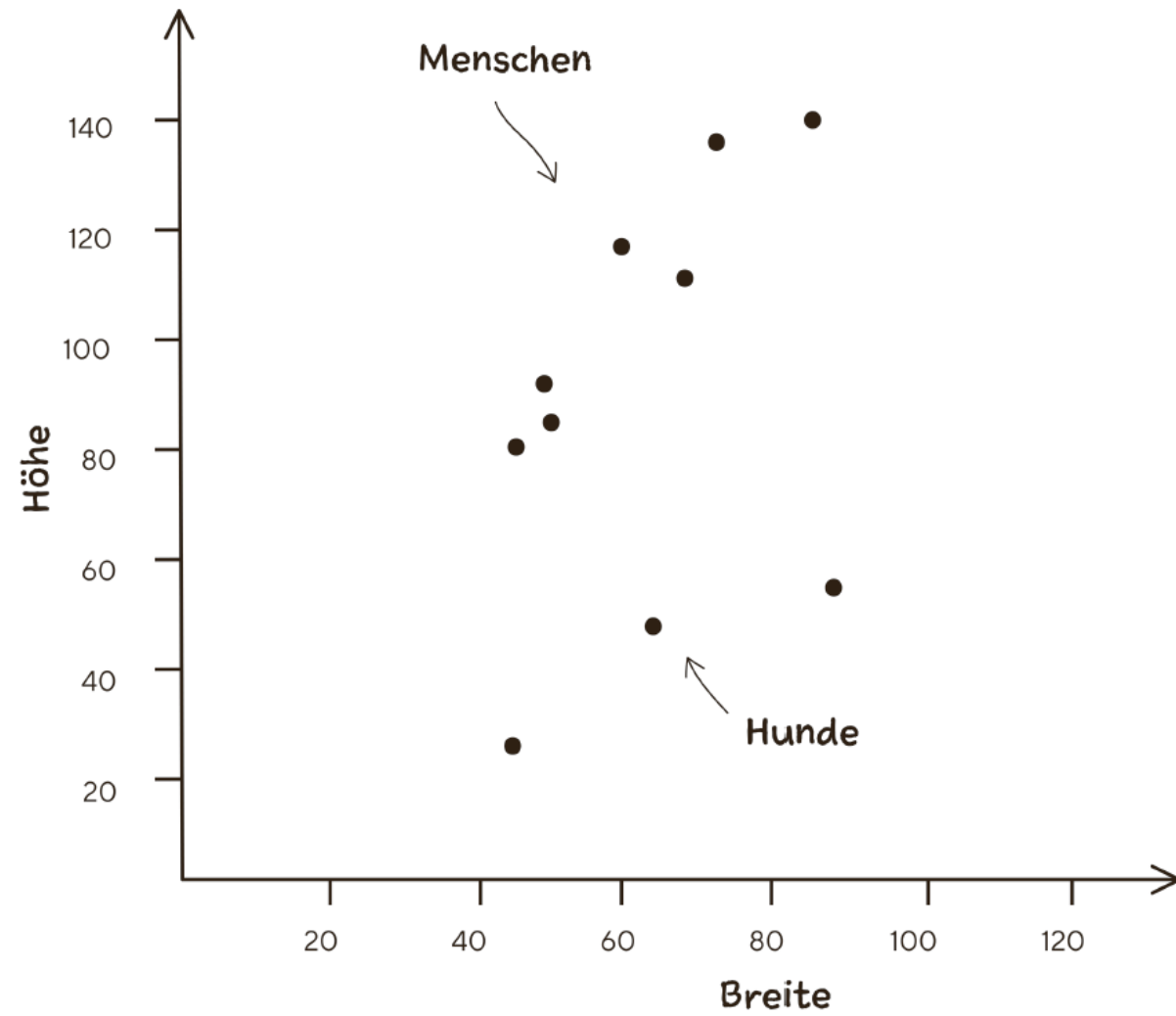
Daten klassifizieren und Objekte erkennen



Rechteckige Bildausschnitte



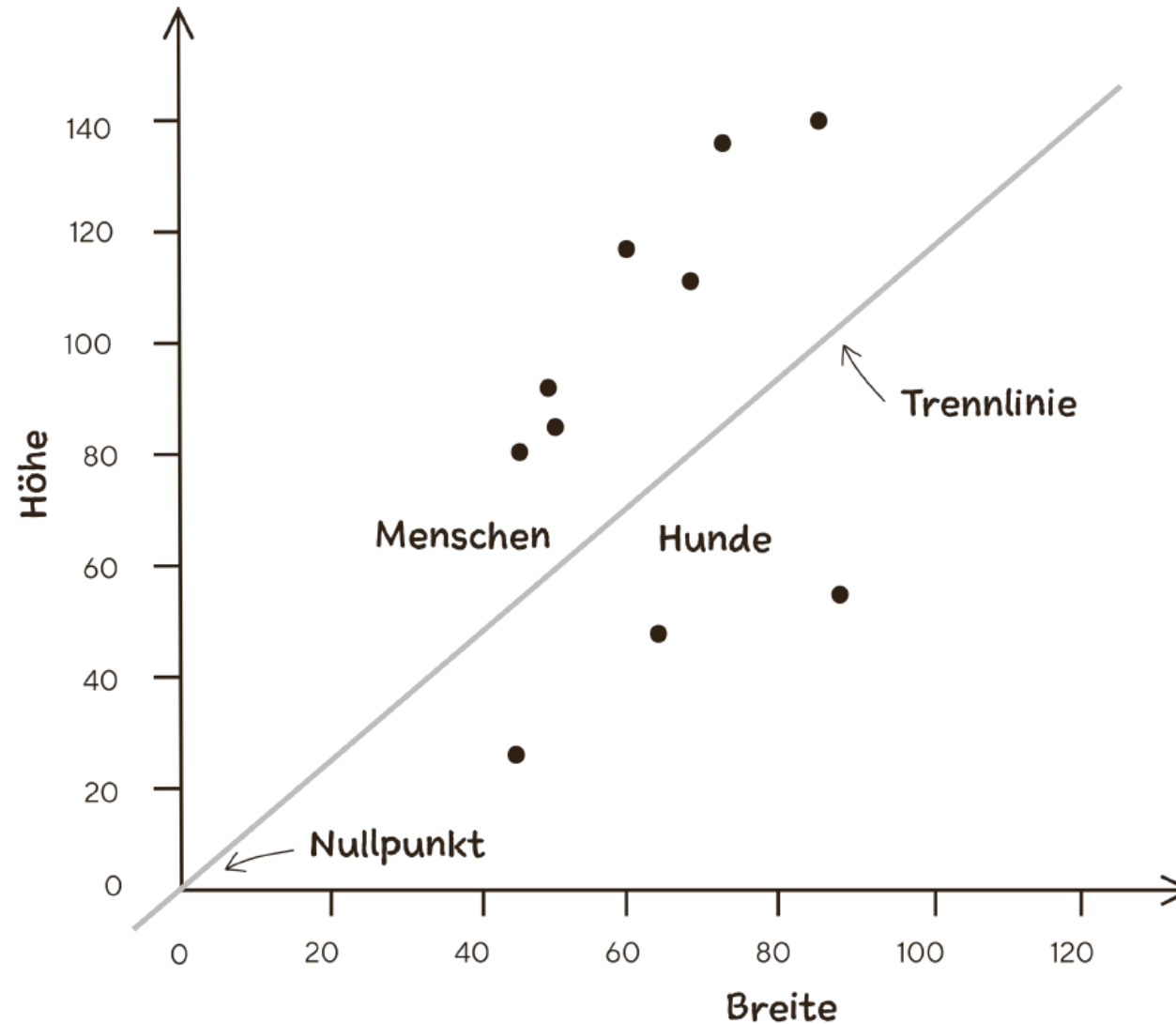
[(77, 135), (88, 55), (70, 115)]



Klassifizierer

$$y = a \cdot x$$

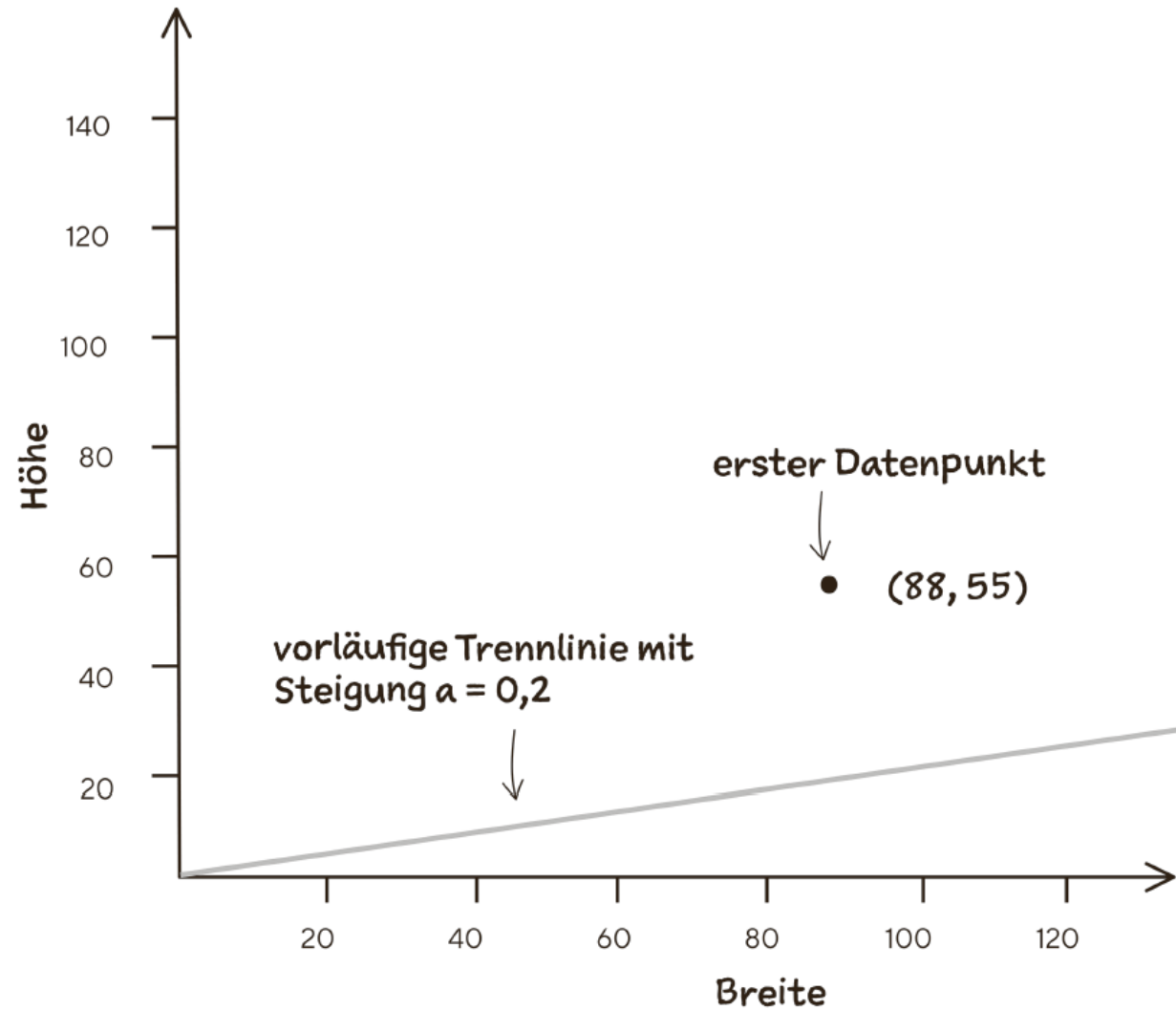
Gesucht ist die Steigung a
der Trennlinie zwischen
Menschen und Hunden



Mit etikettierten Daten lernen

Breite	Höhe	Etikett
88	55	Hund
77	135	Mensch
70	115	Mensch

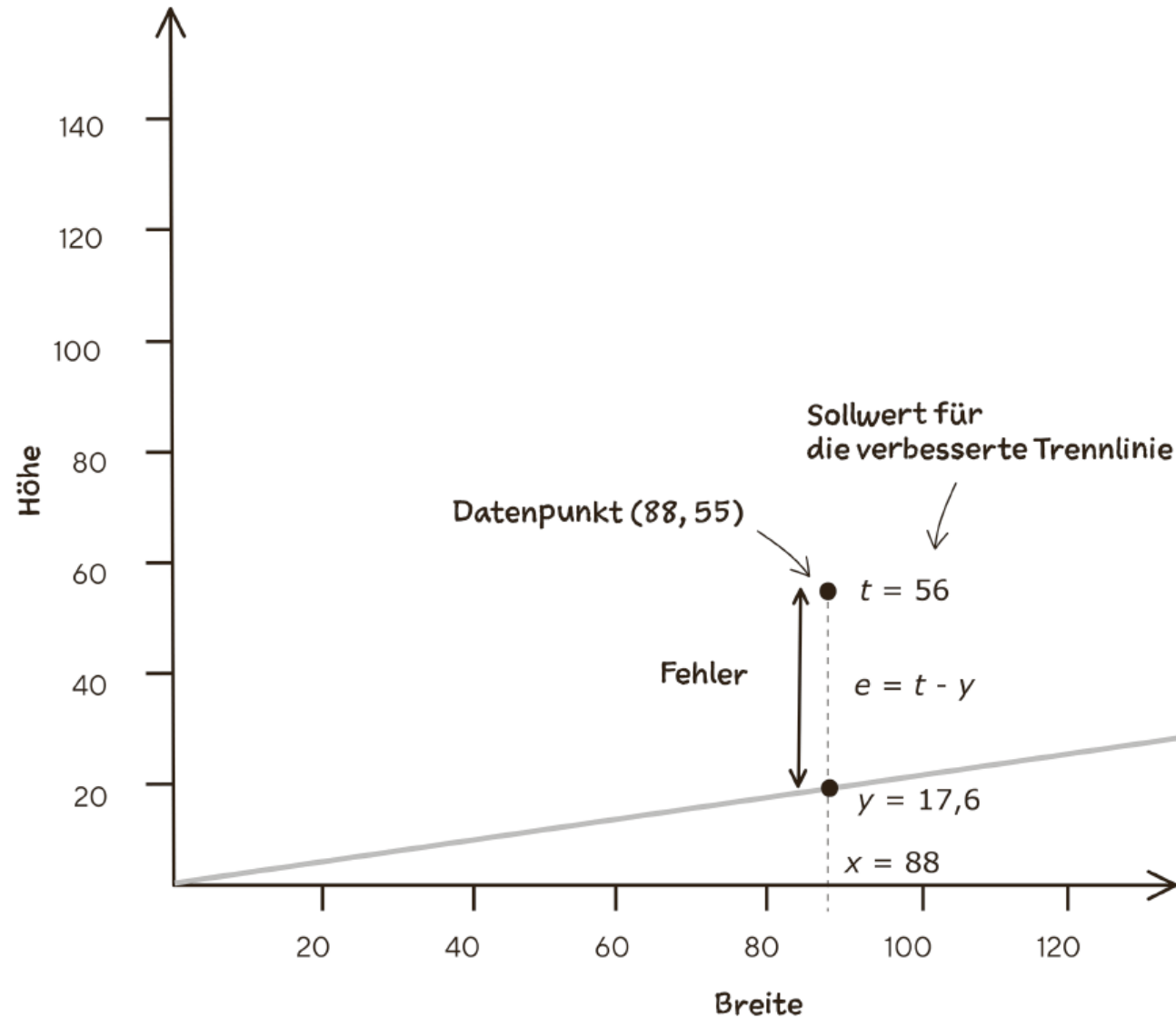
Lernschritt 1



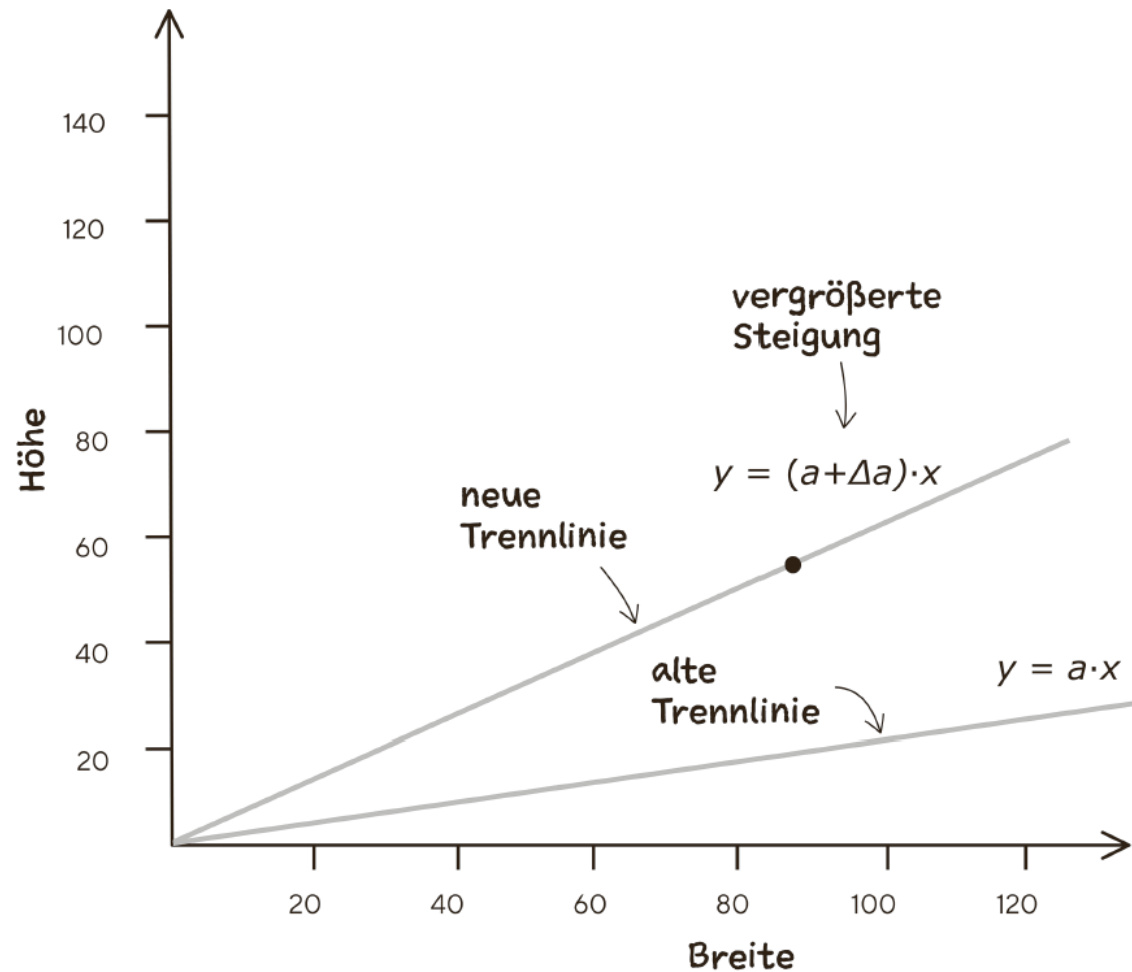
Breite	Höhe	Etikett
88	55	Hund
77	135	Mensch
70	115	Mensch

Gesucht: Bessere Trennlinie

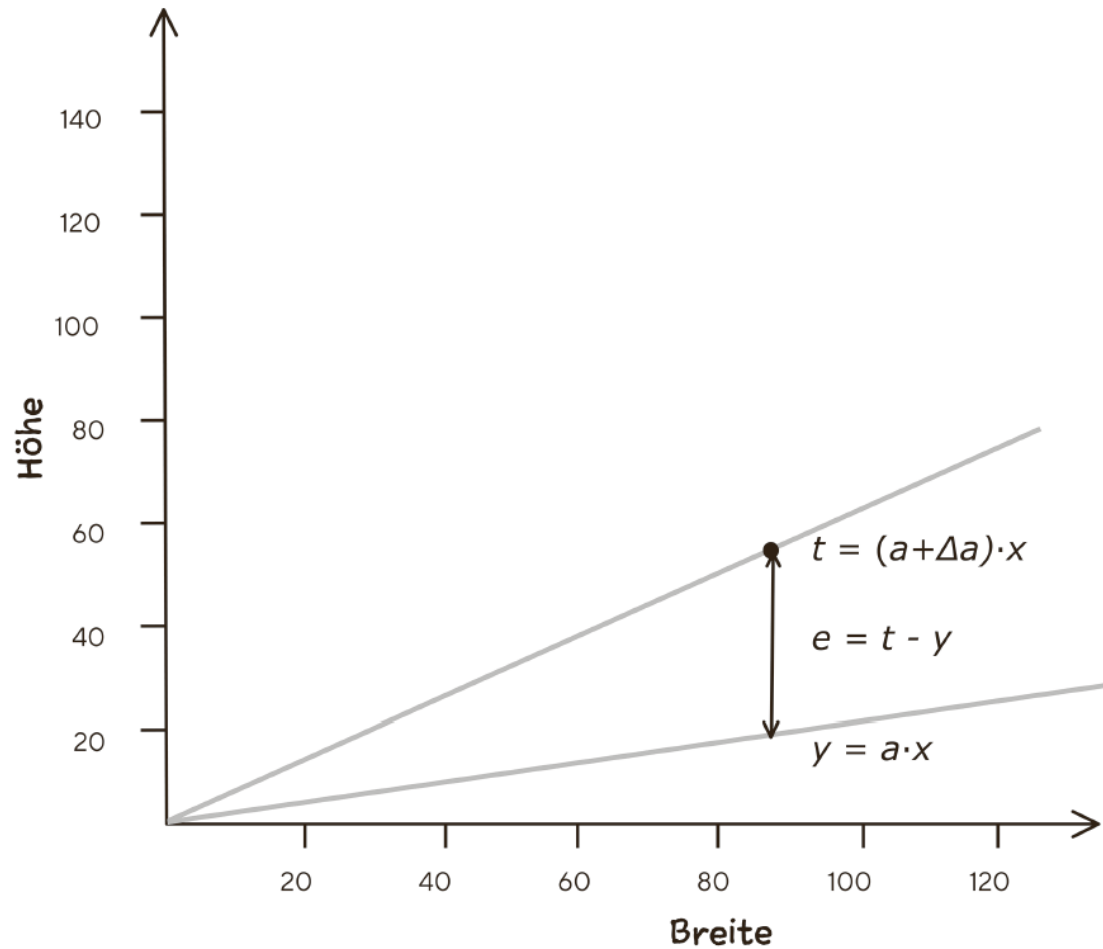
Breite	Höhe	Etikett
88	55	Hund
77	135	Mensch
70	115	Mensch



Gesucht: Bessere Trennlinie



Gesucht: Bessere Trennlinie



$$e = t - y$$

$(a + \Delta a) \cdot x$ $a \cdot x$

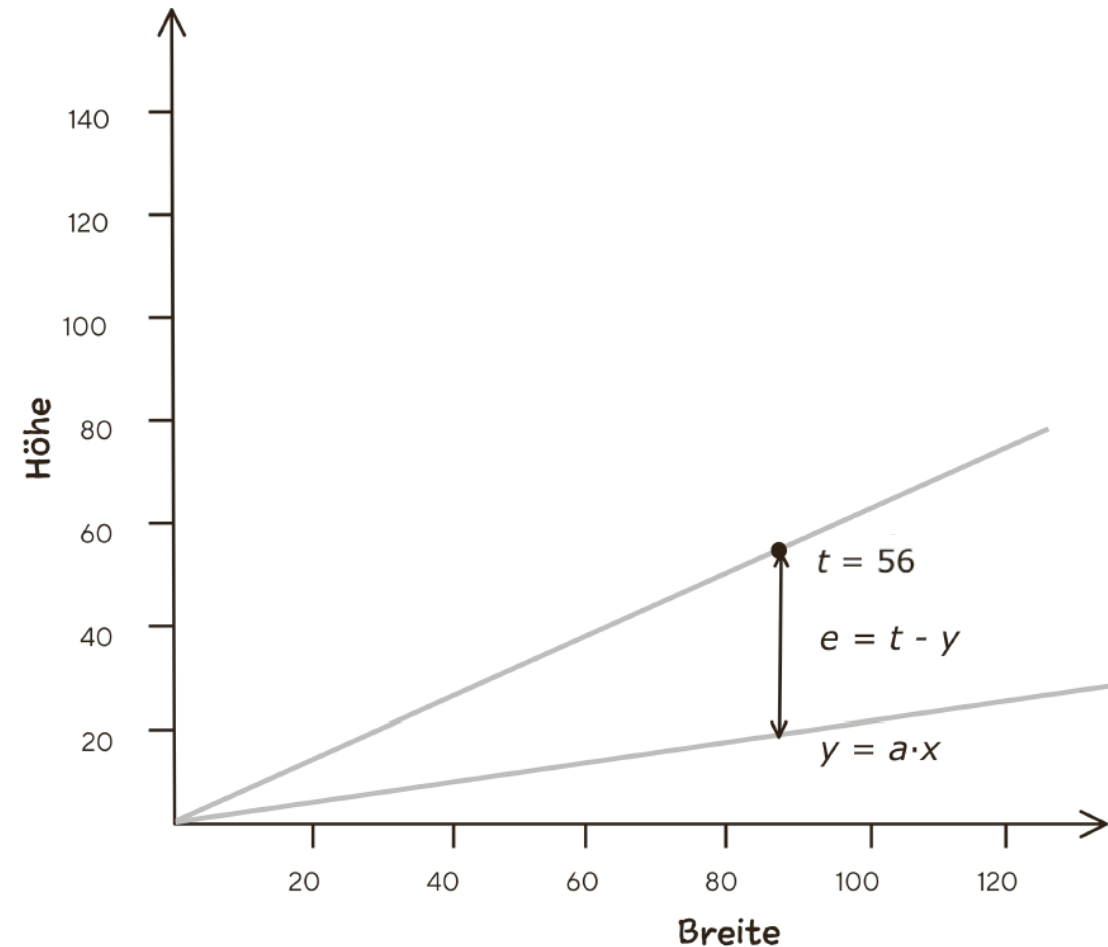
$$e = (a + \Delta a) \cdot x - a \cdot x$$

$$e = a \cdot x + \Delta a \cdot x - a \cdot x$$

$$e = \Delta a \cdot x$$

$$\Delta a = \frac{e}{x}$$

Lernschritt 1



$$\Delta a = \frac{e}{x}$$

Breite	Höhe	Etikett
88	55	Hund
77	135	Mensch
70	115	Mensch

$$y = 0,2 \cdot x$$

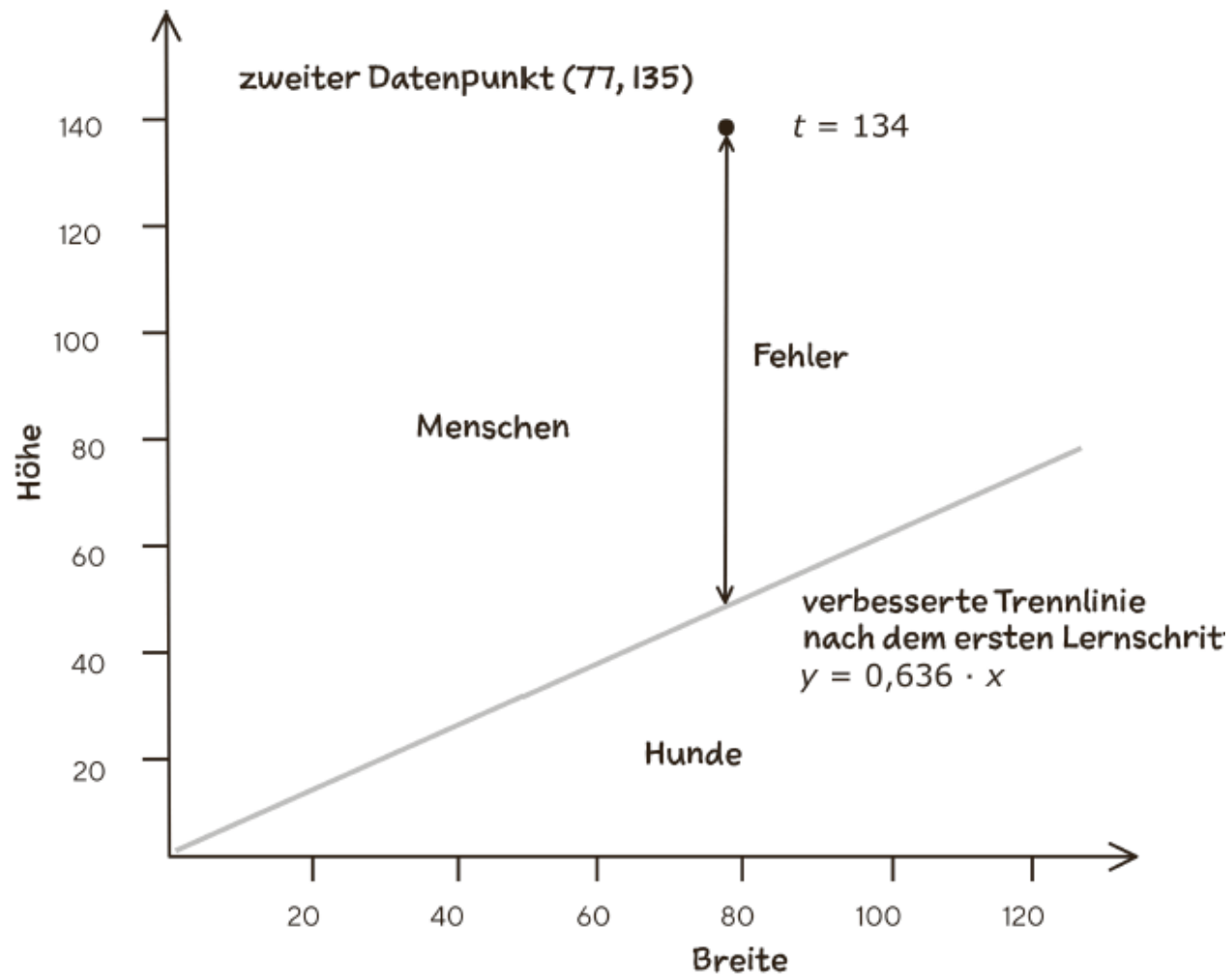
$$e = t - y = 56 - 0,2 \cdot 88 = 56 - 17,6 = 38,4$$

$$\Delta a = \frac{e}{x} = \frac{38,4}{88} \approx 0,436$$

$$a_{neu} = a_{alt} + \Delta a$$

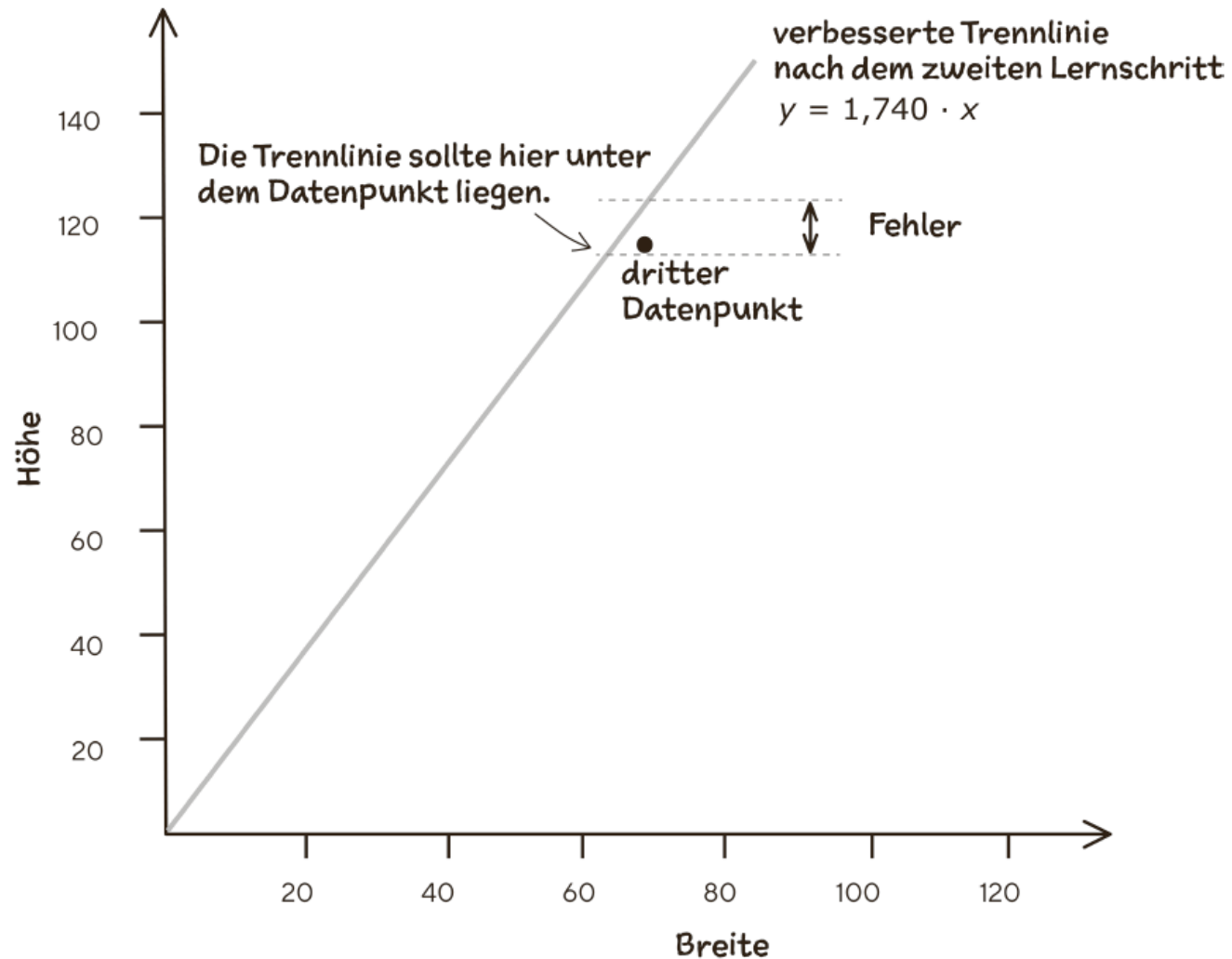
$$a = 0,2 + 0,436 = 0,636$$

Lernschritt 2



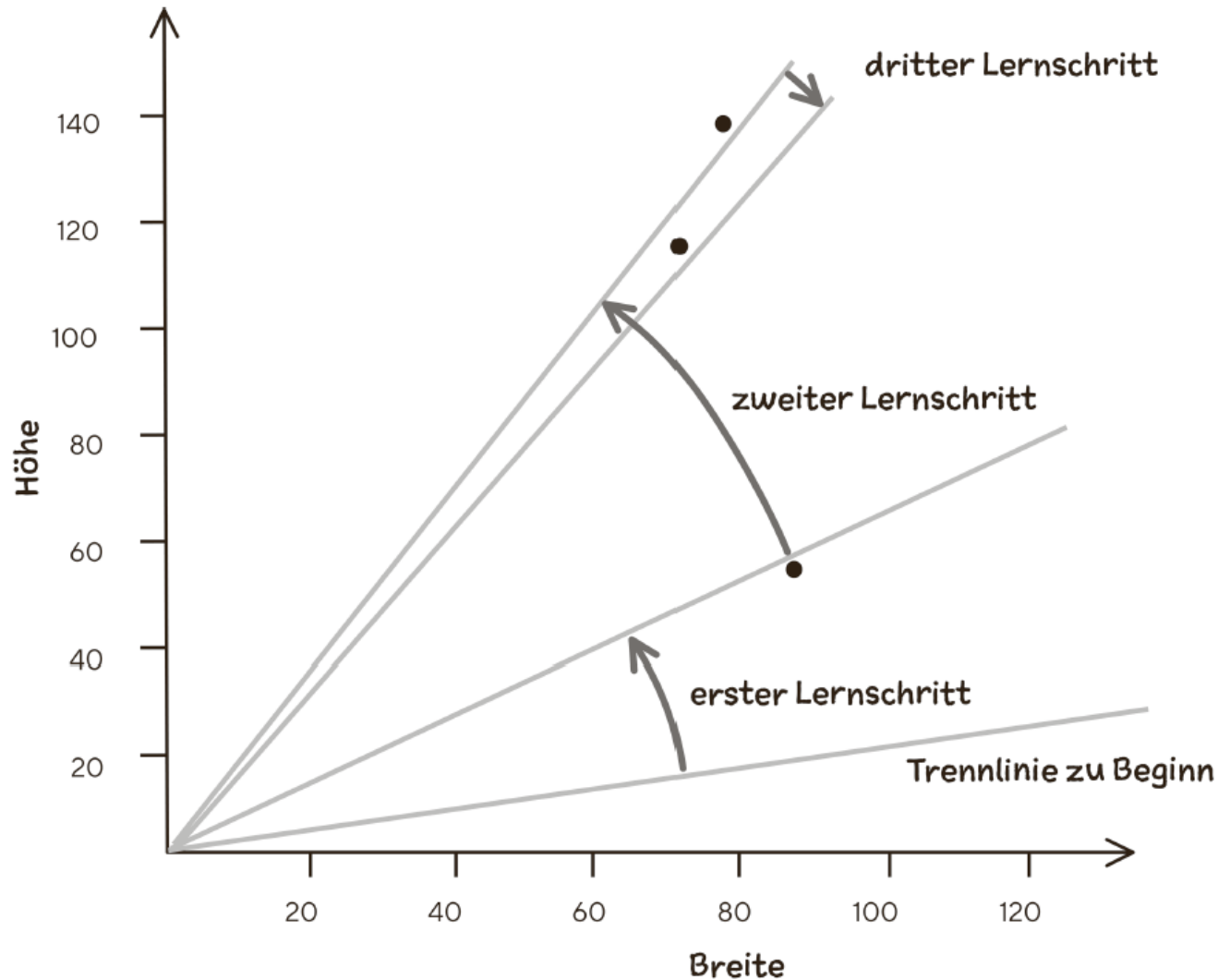
Breite	Höhe	Etikett
88	55	Hund
77	135	Mensch
70	115	Mensch

Lernschritt 3



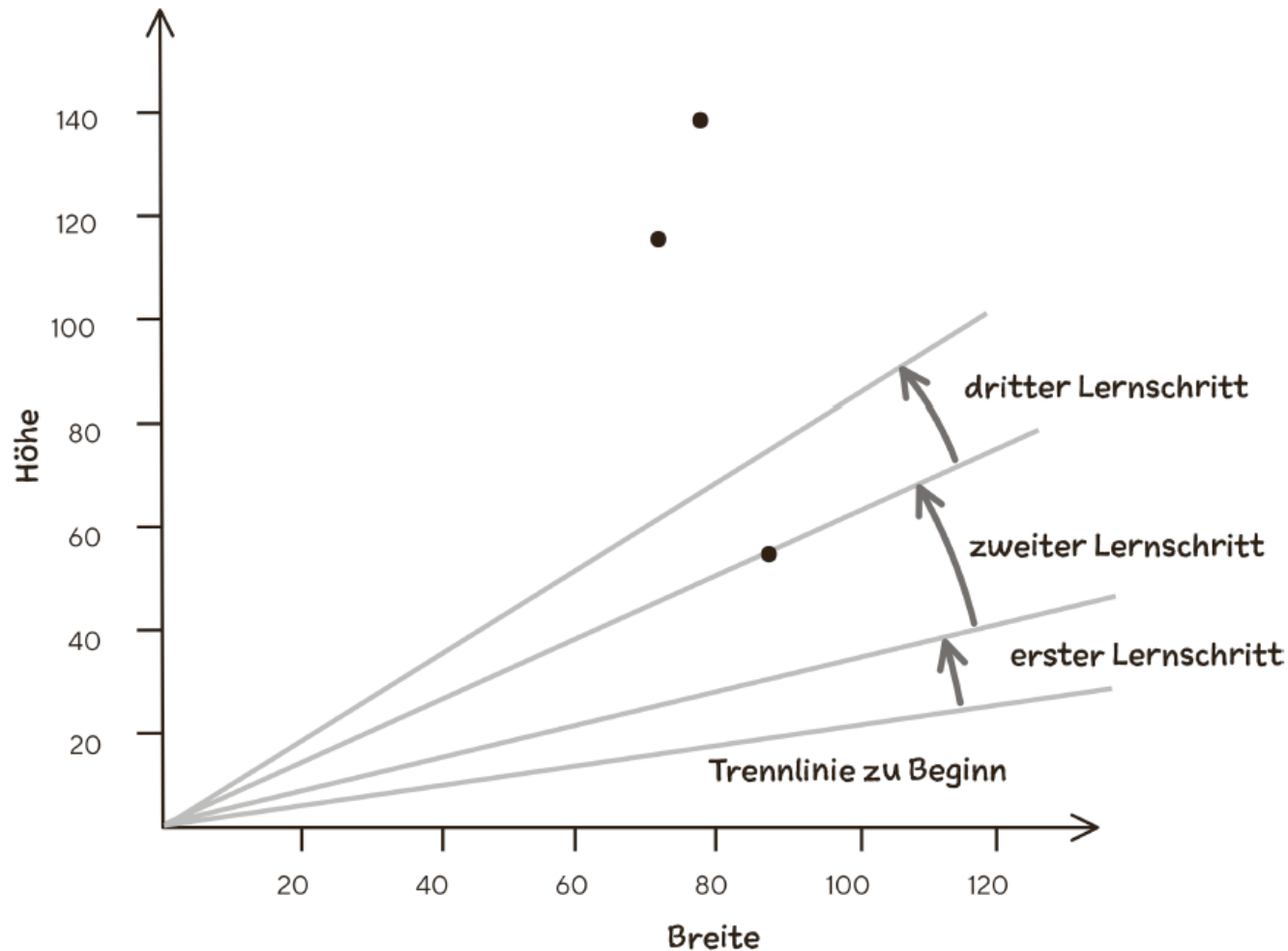
Breite	Höhe	Etikett
88	55	Hund
77	135	Mensch
70	115	Mensch

Die ersten drei Lernschritte



Breite	Höhe	Etikett
88	55	Hund
77	135	Mensch
70	115	Mensch

Moderiertes Lernen



Breite	Höhe	Etikett
88	55	Hund
77	135	Mensch
70	115	Mensch

$$\Delta a = \alpha \cdot \frac{e}{x}$$

Lernrate
(Zahl zwischen 0 und 1)

Moderieren = Abmildern



Ein lernfähiges Vorhersageprogramm

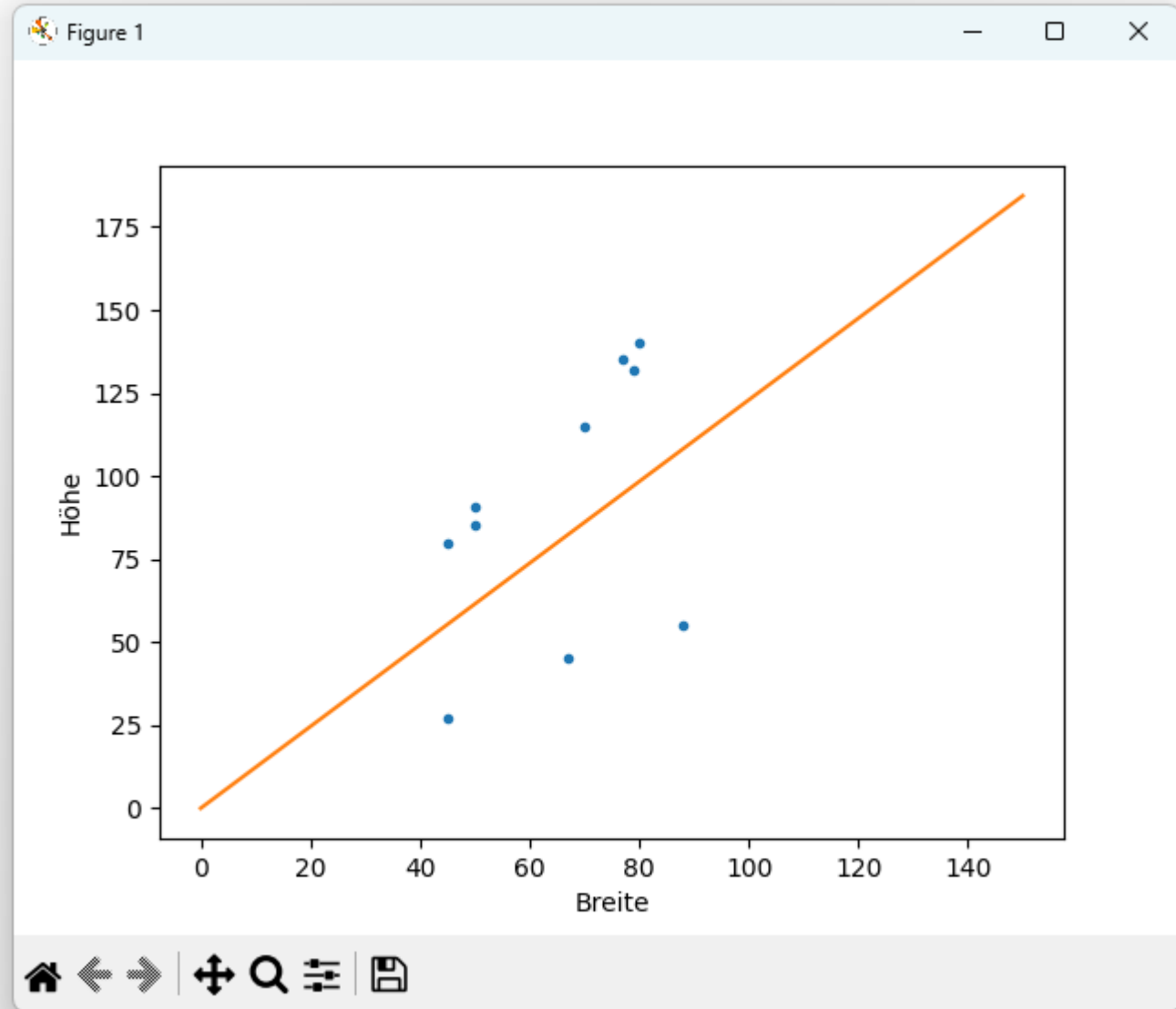
```
# klassifizierer.py
LR = 0.2  # Lernrate

# etikettierte Daten
DATEN = [(77, 135, 'M'), (88, 55, 'H'), (70, 115, 'M'),
          (50, 85, 'M'), (80, 140, 'M'), (45, 27, 'H'),
          (79, 132, 'M'), (50, 91, 'M'), (45, 80, 'M'), (67, 45, 'H')]

# Training
a = 0.2  # Anfangswert der Steigung der Trennlinie
for breite, höhe, label in DATEN:
    if label == 'H':
        t = höhe + 1
    else:
        t = höhe - 1
    e = t - a * breite
    da = LR * e/breite
    a += da
```

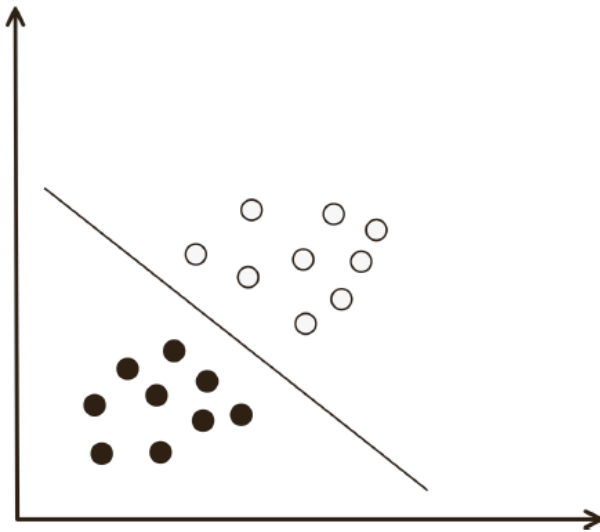
```
# Vorhersagen
eingabe_breite = input('Breite: ')
while eingabe_breite != '':
    eingabe_höhe = input('Höhe: ')
    breite = float(eingabe_breite)
    höhe = float(eingabe_höhe)
    if höhe < a * breite:
        print('Hund')    # unter der Trennlinie
    else:
        print('Mensch')
    eingabe_breite = input('Breite: ')
print('Auf Wiedersehen')
input()    # Warten bis Eingabetaste gedrückt
```

Daten plotten

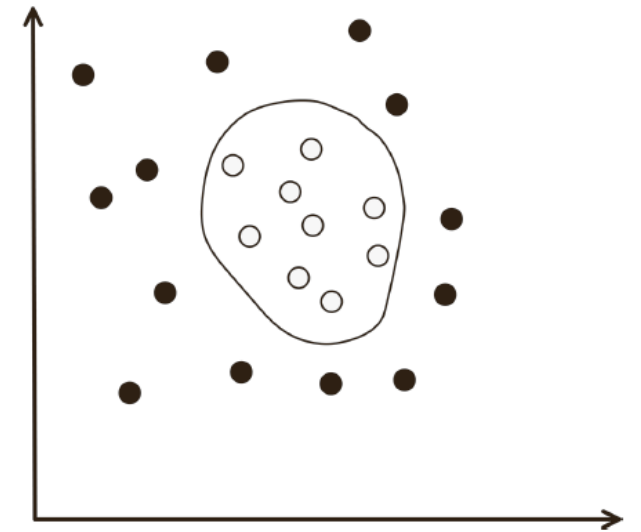
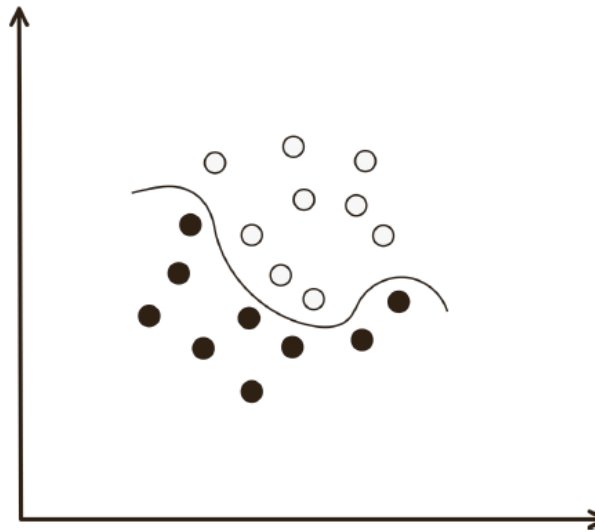


Lineare Separierbarkeit

linear separierbar



nicht linear separierbar



Daten in einer csv-Datei speichern



Daten lesen

Stream ist Folge von
Textzeilen

Datei zum Lesen als
Textdatei öffnen

```
# daten_lesen_split.py
stream = open('trainingsdaten.csv', mode='r')

for zeile in stream:
    datenliste = zeile.split(',')
    print(datenliste)

stream.close()
```


Übung 4 Lernfähiges Vorhersageprogramm

- 1) Experimentieren Sie mit dem Programm `klassifizierer.py`. Ändern Sie z.B. die Lernrate und prüfen Sie ob die Vorhersagen immer noch stimmen.
- 2) Beantworten Sie die folgenden Fragen zum Verständnis des Programms:
 - An welcher Stelle spielt das Etikett eines Datensatzes eine Rolle?
 - An welcher Stelle lernt das Programm?
- 3) Überlegen Sie sich selbst eine Frage zum Verständnis des Programms. Stellen Sie diese Frage Ihrem Nachbarn.
- 4) Überarbeiten Sie das Starterprojekt `klassifizierer.py` und sorgen Sie dafür, dass das Programm die Trainingsdaten aus der Datei `trainingsdaten.csv` holt.

Weitere Aufgabeideen? https://docs.google.com/document/d/140INsIEWA_AwUwtpVMCZqtH7_eOkU8rmvfrgWqE5M3E/edit?usp=sharing

Lösungen 4

Aufgabe 2

```
LR = 0.2  # Lernrate
```

```
# etikettierte Daten
```

```
DATEN = [(77, 135, 'M'), (88, 55, 'H'), (70, 115, 'M'),  
          (50, 85, 'M'), (80, 140, 'M'), (45, 27, 'H'),  
          (79, 132, 'M'), (50, 91, 'M'), (45, 80, 'M'), (67, 45, 'H')]
```

```
# Training
```

```
a = 0.2      # Anfangswert der Steigung der Trennlinie
```

```
for breite, höhe, label in DATEN:
```

```
    if label == 'H':
```

```
        t = höhe + 1
```

```
    else:
```

```
        t = höhe - 1
```

```
    e = t - a * breite
```

```
    da = LR * e/breite
```

```
    a += da
```

In Abhängigkeit vom Etikett wird das Target für die Neuberechnung der Steigung a der Trennlinie gesetzt.

In diesen beiden Zeilen wird die Steigung der Trennlinie neu berechnet. Das ist das Lernen.

Aufgabe 4

```
LR = 0.2  # Lernrate
```

```
# etikettierte Daten lesen
```

```
DATEN = []
```

```
stream = open('trainingsdaten.csv', 'r')
```

```
for zeile in stream:
```

```
    breite, höhe, label = zeile.split(',')
```

```
    datentupel = (int(breite), int(höhe), label[0])
```

```
    DATEN.append(datentupel)
```

```
stream.close()
```

Strings müssen in Zahlen umgewandelt werden

Erster Buchstabe des Strings label

Rückblick

- Manche Datenmengen (aber nicht alle) lassen sich durch eine gerade Trennlinie (Klassifizierer) in zwei Cluster einteilen.
- Beim überwachten Lernen lernt ein Programm mit Hilfe von etikettierten Trainingsdaten, Objekte zu erkennen.
- Etikettierte Trainingsdaten bestehen aus Daten (z.B. Zahlen) und einem Etikett, das den Daten eine Bedeutung zuordnet.
- Die Lernrate ist eine Zahl zwischen 0 und 1. Sie bestimmt, wie stark beim Training eine Einzelbeobachtung in die Neuberechnung des Klassifizierers einfließt.
- Trainingsdaten können als csv-Datei gespeichert werden.
- Mit der String-Methode `split()` kann ein String in eine Liste kleinerer Strings „ausgespalten“ werden.