

Machine Learning (Udemy Kurs)

Theorie

SUPERVISED LEARNING

Regression

Erstellen einer Funktion, die Numerische Werte annähert. Die Funktion wird anhand vorliegender Daten optimiert.

Klassifikation

Anhand der Input Features soll als Output die Klasse ausgegeben werden, die der Daten am ähnlichsten ist.

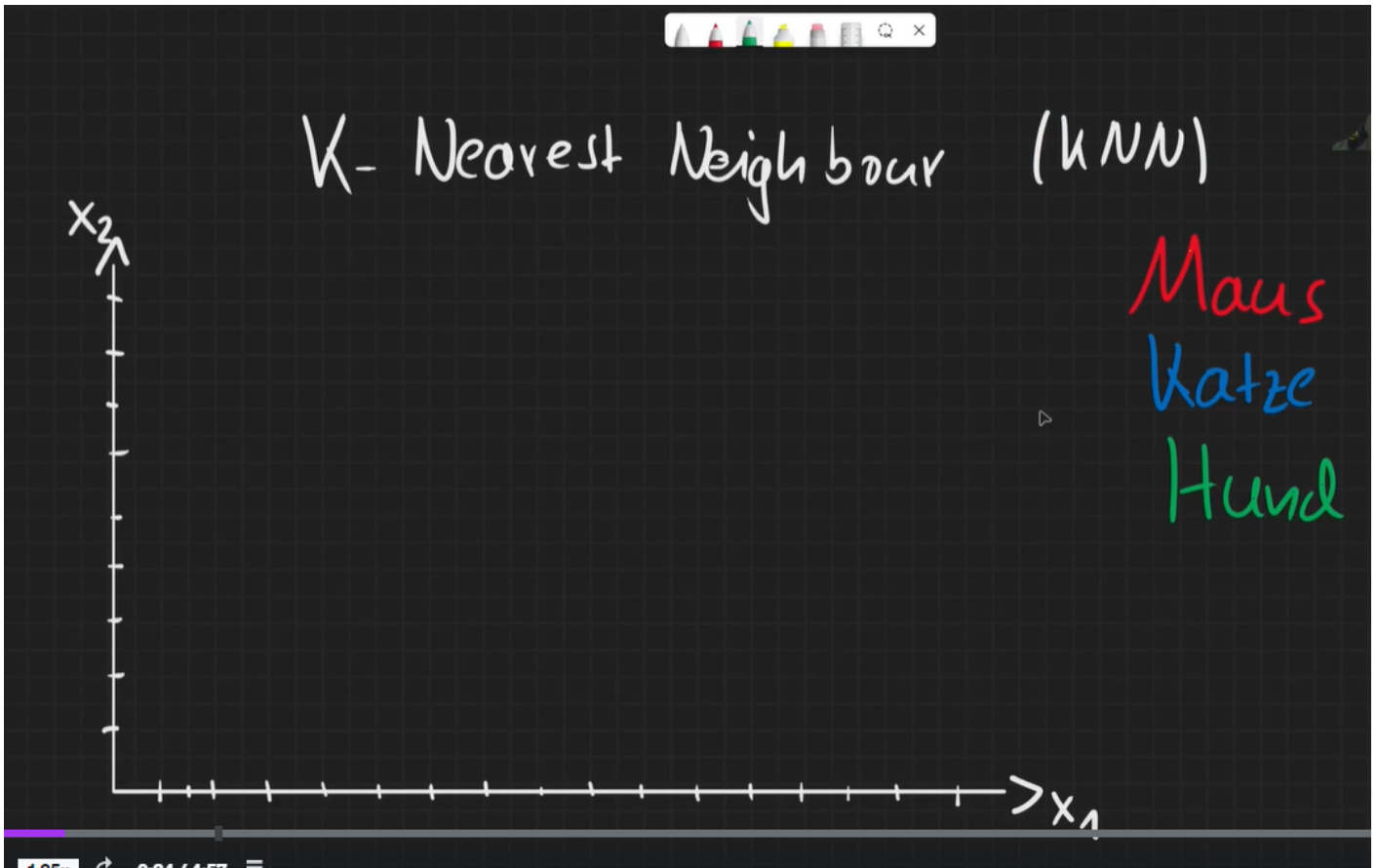
z.B. Das Erkennen von Tierarten auf Bildern

EIGENSCHAFTEN DES MODELLS

- Das Modell soll anhand der Features die richtigen Aussagen treffen
- Wir „Überwachen“ das Programm beim Training
- Beim Training wissen wir welcher Output zu dem jeweiligen Input richtig ist
- Somit sind wir der Lehrer des Modells, der auf Fehler hinweist

DER LERNPROZESS

- Wenn ein Fehler beim Training auftritt, muss erforscht werden warum er auftrat
- Das Programm soll dann nach Anpassung der Parameter ein besseres Ergebnis liefern
- Das Ziel ist dann anhand des Trainings ein allgemeingültiges System zu haben



Praxis

Module

Numpy

```
import numpy as np
```

Ein Array mit Numpy erstellen

```
x = np.array([-2, 1, 3, -10, 22])
```

Matplotlib

```
import matplotlib.pyplot as plt
```

Einen Plot erstellen

```
x = np.array([1,2,3,4,5])
y = np.array([2,-3,5,10,12])

# Scatter Plot # Punkte ohne Verbindung

plt.scatter(x, y, color="red")
plt.legend(['f(x)'])
plt.title('This is a title')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.show()
```

Mit zeilen verbinden

```
# Plot

plt.scatter(x, y, color="red")
plt.plot(x,y,color="blue")
plt.legend(['f(x)'])
plt.title('This is a title')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.show()
```

List slicing und Arrays

```
import numpy as np

l1 = [i for i in range(20)] # Erstellt eine Liste mit Werten von 0 bis 19

l2 = l1[0:20:2] # Erstellt eine Liste aus der ersten Liste und zwar nur jeden 2. Wert

l3 = l1[:10] # Gibt die ersten 10 wieder

# Arrays mit Numpy erstellen
my_array = np.zeros(shape=(2,2)) # erstellt ein array mit 2 x 2 Zellen

my_rershaped_array = np.reshape(my_array, newshape=(4,))
```

```
# Zufallszahlen generieren
my_random_array = np.random.randint(low=0, high=11, size=20) # Ganzzahlen mit Randint

my_random_array2 = np.random.uniform(low=0.0, high=10.0, size=20) # Floatzahlen erstellen
```

Type Annotations

Festlegen welche Datentypen für einen Parameter infrage kommen

```
class Car:
    def __init__(self, name: str, oem: str, hp: int, year: int) -> None: # -> None bedeutet es wird nichts zurück
        gegeben
        self.name = name
        self.oem = oem
        self.hp = hp
        self.year = year

    def get_info(self) -> str: # -> str bedeutet es wird ein String zurück gegeben
        return "Name: " # self.name

def main():
    car1 = Car(10, "Audi", 400, 2022)
    info1 = car1.get_info()
    print(info1)

if __name__ == "__main__":
    main()
```

f-Strings

```
# Mit f-string kann man die Variable direkt in den String schreiben

f"Name: {self.name} OEM: {self.oem} HP: {self.hp}"
```

Revision #6

Created 17 May 2023 08:22:00 by Hermann

Updated 17 May 2023 10:37:21 by Hermann