

# Data Science & Machine Learning

Dieter Greipl

2022-01-22

# Contents

# Willkommen

Dieses Skript entstand (und entsteht) aus meinen Lehrveranstaltungen rund um das Thema **Data Science & Machine Learning**. Die Inhalte richten sich an Studierende, die erste Schritte auf das KI-Spielfeld wagen und das Potential von datengetriebenen Lösungsverfahren verstehen wollen.

Insofern richtet sich die Darstellung an Studierende mit vertieftem Interesse an KI, die einen für Studierende angemessenes Vorwissen im Bereich Mathematik mitbringen. Vorkenntnisse im Bereich der Programmierung sind nicht nötig, aber natürlich hilfreich.

Ich habe mich bemüht, zahlreiche Übungsbeispiele und Youtube-Videos einzubauen. Viele Themen lassen sich so besser darstellen. Sofern es Medien im Netz gibt, die die Sachverhalte gut darstellen, werde ich entsprechenden Links einbauen. Der Autor muss ja nicht der Meinung sein, alles besser zu können. Gleichwohl darf dadurch der rote Faden nicht verloren gehen.

## 0.1 Vorbereitungen

Dieses Skript ist als Unterlage für zahlreichen praktische Übungen mit Python angelegt. Ich werde hierzu **Colab-Notebooks** verwenden. Sie brauchen hierzu ein **Google-Konto**.

Noch einige Hinweise an Studierende meiner Module:

- Die folgende Youtube-Playlist kann zur Vertiefung einzelner Stoffteile nutzen: Youtube Playlist
- Wenige Passagen in diesem Skript sind eventuell in englischer Sprache gehalten.
- Dieses Skript
  - befindet sich in Teilen im Aufbau, leichte Fehler sind also möglich (und wahrscheinlich - für Hinweise bin ich dankbar)
  - geht nach der Prüfung off-line

# Chapter 1

## Python - Quickstart

### 1.1 Colab-Notebooks

### 1.2 Hallo Welt

```
print("Hallo Welt")
```

Ausgabe:

```
#> Hallo Welt
```

### 1.3 Variablen

Variablen sind Platzhalter für Werte, wir sprechen vom *Wert einer Variable*. Im nachfolgendem Beispiel wird der Variable mit dem Namen `x` in Zeile 1 der Wert 1 zugewiesen. Variablen haben immer einen Namen. Einen Wert erhalten sie erst durch eine sog. Zuweisung (wie in Zeile 1). In Zeile 2 drucken wir den Wert aus. Führen Sie also folgende Python-Befehle aus:

```
x = 1  
print(x)
```

Die erstmalige Zuweisung eines Wertes an eine Variable heißt Initialisierung.

### 1.4 Datentypen

#### 1.4.1 Datentyp “Zahlen”

##### 1.4.1.1 Ganze Zahlen und rationale Zahlen

Zahlen sind recht einfach zu verstehen, wir haben ja oben schon einige Beispiele gesehen. Hier nochmal eine Zusammenfassung wichtiger Beispiele mit rationalen Zahlen:

```
a = 2
b = 1/3
c = 1.1
# Funktionen
d = a + b; print(d)
d = a - b; print(d)
d = a * b; print(d)
d = a / b; print(d)
```

## 1.4.2 Datentyp “Strings”

Zeichenketten sind ebenfalls recht einfach zu verstehen. Führen Sie folgendes Beispiel aus:

```
vorname = "Hans"
nachname = 'Huber'
name = vorname + ", " + nachname
print(name)
```

### 1.4.2.1 f-Strings

Statt eine zusammengesetzte Zeichenkette mit dem “+” - Operator aufzubauen, kann ein sogenannter f-String verwendet werden (siehe Zeile 3). f-Strings bringen für uns keine neue Funktion, machen aber die Verknüpfung von Strings einfacher.

```
vorname = "Hans"
nachname = 'Huber'
name = f"{vorname}, {nachname}"
print(name)
```

## 1.4.3 Datentyp “Boolean”

Es gibt in der Theorie unendlich viele Zahlen und Zeichenketten, aber nur zwei Wahrheitswerte: wahr oder falsch. In Python: `True` und `False`

```
z1 = False;
print (z1)
print( type(z1) );

z2 = 1 < 4;
print( z2 );
print( type(z2) );

print ("z1 and z2:", z1 and z2);
print ("z1 or z2:", z1 or z2);
```

## 1.5 Operatoren

Die Bildschirmabzüge dieses Kapitels sind der Webseite [https://www.w3schools.com/python/python\\_operators.asp](https://www.w3schools.com/python/python_operators.asp) entnommen. Erarbeiten Sie sich die Operatoren selbst in kleinen Programmen, so wie wir das zu Zahlen bereits oben gemacht haben.

### 1.5.1 Arithmetische Operatoren

Operator	Name	Example
+	Addition	$x + y$
-	Subtraction	$x - y$
*	Multiplication	$x * y$
/	Division	$x / y$
%	Modulus	$x \% y$
**	Exponentiation	$x ** y$
//	Floor division	$x // y$

Figure 1.1: bild1

### 1.5.2 Vergleichsoperatoren

Operator	Name	Example
==	Equal	$x == y$
!=	Not equal	$x != y$
>	Greater than	$x > y$
<	Less than	$x < y$
>=	Greater than or equal to	$x >= y$
<=	Less than or equal to	$x <= y$

### 1.5.3 Logische Operatoren

Operator	Description	Example
and	Returns True if both statements are true	<code>x &lt; 5 and x &lt; 10</code>
or	Returns True if one of the statements is true	<code>x &lt; 5 or x &lt; 4</code>
not	Reverse the result, returns False if the result is true	<code>not(x &lt; 5 and x &lt; 10)</code>

## 1.6 Datentypen in der Übersicht

Nachfolgendes Programmstück beschreibt exemplarisch die nun bekannten Datentypen. Mit `type()` kann man sich den Datentyp einer Variable ausgeben lassen (oft hilfreich!)

```
x = 1;
print( f"1 : {type(x)}")

x = 1.1;
print( f"1.1 : {type(x)}")

x = 4/2;
print( f"4/2 : {type(x)}")

x = 'String';
print( f"'String' : {type(x)}")

x = [1,2,4];
print( f"[1,2,3] : {type(x)}")

x = (1,2);
print( f"(1,2) : {type(x)}")

x = 1 < -3
print( f"False : {type(x)}")
```

### 1.6.1 Zusammenfassung

Eingebauter Datentyp	Kürzel	Beispiele
Text (Strings)	str	<code>x = "Haw-Landshut"</code>
Zahlen (Numerisch)	int, float	<code>x = 1</code> oder <code>x = 1.1</code>
Listen (Arrays)	list	<code>x = [1,2,3]</code>
Tupel	tuple	<code>x = (1,2)</code>
Wahrheitswerte	bool	<code>x = (1 &lt; 3)</code>

## Chapter 2

# Daten

### 2.1 Der Iris-Datensatz

Der Iris-Datensatz enthält Messungen von jeweils 50 Blüten zu drei verschiedenen Lilien-Arten (*setosa*, *versicolor*, *virginica*)

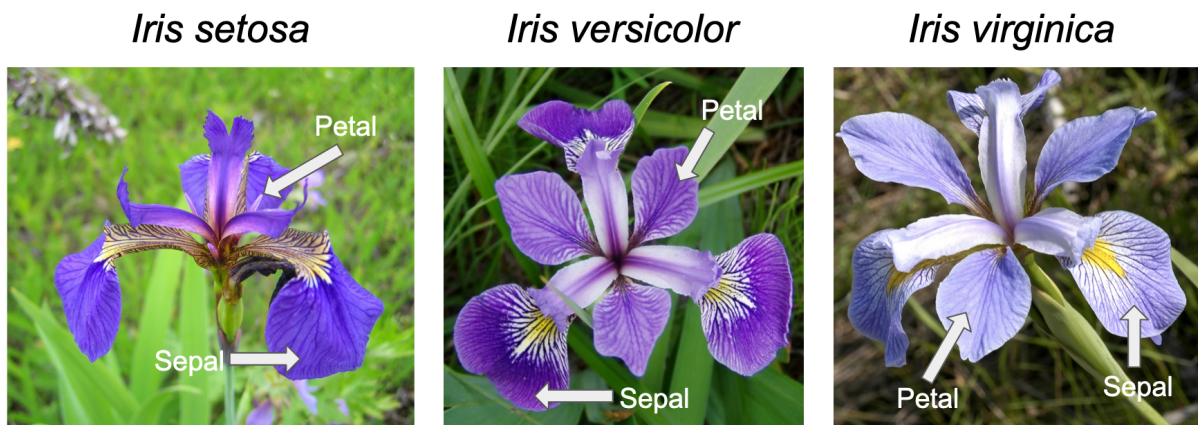


Figure 2.1: Download

Gemessen werden pro Blüte in cm

- die Länge und Breite des Kronblattes (Petalum, petal) sowie
- die Länge und Breite des Kelchblattes (Sepalum, sepal)

#### 2.1.1 Datensatz

Folgender - in der Community wohlbekannter - Datensatz liegt uns vor (Sie finden die Daten hier).



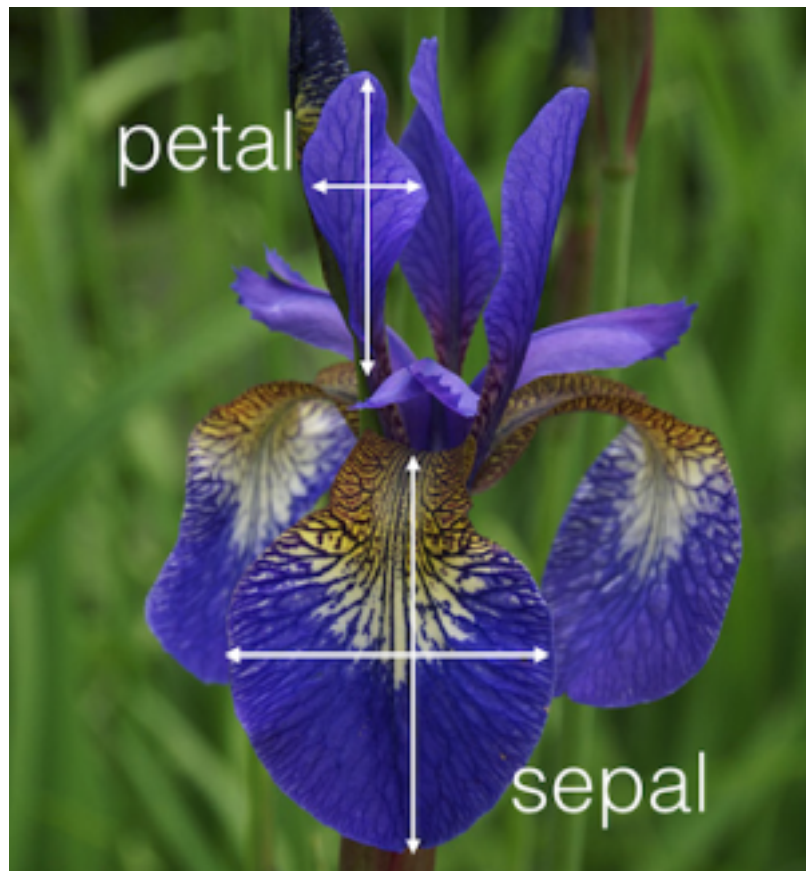


Figure 2.2: image (190)

	A	B	C	D	E	F
1	sepal_len	sepal_wid	petal_len	petal_wid	class	
2	7,2	3,6	6,1	2,5	virginica	
3	5	3,6	1,4	0,2	setosa	
4	4,4	3	1,3	0,2	setosa	
5	4,7	3,2	1,6	0,2	setosa	
6	6,7	3,1	4,7	1,5	versicolor	
7	5,7	3,8	1,7	0,3	setosa	
8	6,9	3,1	5,4	2,1	virginica	
9	6,1	2,8	4,7	1,2	versicolor	
10	6,3	3,3	6	2,5	virginica	
11	6,5	3	5,2	2	virginica	
12	5,9	3	5,1	1,8	virginica	
13	6,3	2,5	5	1,9	virginica	
14	6,9	3,1	4,9	1,5	versicolor	
15	5,4	2,7	1,5	0,2	setosa	

Figure 2.3: Iris-Datensatz

## 2.2 Datentypen

### 2.2.1 Elementare Datentypen

#### 2.2.1.1 Zahlen

#### 2.2.1.2 Strings

#### 2.2.1.3 Logische Werte

#### 2.2.1.4 Elementare Datentypen in Python

### 2.2.2 Komplexe Datentypen

#### 2.2.2.1 Datum

#### 2.2.2.2 Uhrzeit

#### 2.2.2.3 Bilder

#### 2.2.2.4 Komplexe Datentypen in Python

## 2.3 Skalenniveaus


### 2.3.1 Überblick

Scale	Operations	Description	Statistics	Example
Nominal	$=, \neq$	values have no natural order; they describe unordered categories	Mode (Modus)	München, Hamburg, Essen
Ordinal	$<, >$	values have a defined order; difference of values is undefined or has no clear or meaningful definition	Median	Schulnoten, Tabellenplatz in der Bundesliga
Interval	$+, -$	differences of values have the same meaning; adding provides useful results; zero point is not naturally/globally defined	Mean	Temperatur
Ratio	$\cdot, /$	zero point is naturally defined	(Generalized) Mean	Alter

Bemerkungen:

1. Skalenniveaus sind nicht immer klar zuzuordnen.
2. Auf nominalen Datenskalen lassen sich stets *künstliche Ordnungen* (und damit ordinale Datenskalen) definieren.
3. Bilden sie keine Mittelwerte auf Daten mit ordinalen Datenskalen!
4. Nominale und ordinale Datenskalen heißen auch *kategorisch* oder *qualitativ*.
5. Intervall und Ratio-Datenskalen heißen auch *metrisch*.

Ergänzend: Die fünf Skalenniveaus: Einfach und verständlich erklärt ([statistikpsychologie.de](http://statistikpsychologie.de))



	A	B	C	D	E
1	sepal_len	sepal_wid	petal_len	petal_wid	class
2	7,2	3,6	6,1	2,5	virginica
3	5	3,6	1,4	0,2	setosa
4	4,4	3	1,3	0,2	setosa

Figure 2.4: Skalenniveaus bei Iris

## 2.3.2 Skalenniveaus im Iris-Datensatz

**2.3.2.0.1 Von Nominal zu Ordinal** Wir werden später folgende eindeutige Zuordnung treffen:

Nominaler Wert	Ordinaler Wert
setosa	0
versicolor	1
virginica	2

## Chapter 3

# Matplotlib und Seaborn

Mathplotlib (<https://matplotlib.org/>) ist eine Sammlung von Funktionen (Bibliothek) zum Visualisieren von Daten. Wir verwenden matplotlib zusammen mit dem ergänzenden Programmpaket Seaborn.

Wichtig: Sie müssen die folgenden beiden Zeilen stets am Beginn des Programms stehen haben.

```
import matplotlib.pyplot as plt
import seaborn as sns
```