

# Data Science und Machine Learning

Einführung für Studierende der Betriebswirtschaftslehre und  
Wirtschaftsinformatik



Dieter Greipl  
Hochschule Landshut

### **Danke**

Dieses Buch entstand im Sommersemester 2022. Vielen Dank an meine Studierenden für zahlreiche Hinweise.

Copyright

...

# 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

# Chapter 1

## Python - Quickstart

Es wird sie nicht überraschen, dass Data-Science nur mit Unterstützung eines Computers und speziellen Algorithmen zielführend ist. Hierfür verwenden wir die Programmiersprache Python. Ich empfehle dringend ein der folgenden Varianten zur Nutzung von Python zu verwenden. Wir starten in beiden Varianten mit dem typischerweise ersten Programm für Anfänger, dem Hallo-Welt-Programm. Dieses Programm gibt lediglich den digitalen Gruß “Hallo Welt” in dem vorgesehen Ausgabe Bereich aus.

### 1.1 Hallo Welt


Unser kleines Begrüßungsprogramm besteht nur aus einer Zeile

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

Die Ausgabe dieses Programms ist:

```
#> Hallo Welt
```

#### Variante 1: Colab-Notebooks

Unter dem Link <https://colab.research.google.com/> können sie ein sogenanntes Colab-Notebook erstellen. In die Programmzelle können tragen sie die Befehle ein. Ein Klick auf  führt die Programmzeilen aus und schreibt eventuelle Ausgaben unter die Programmzelle.

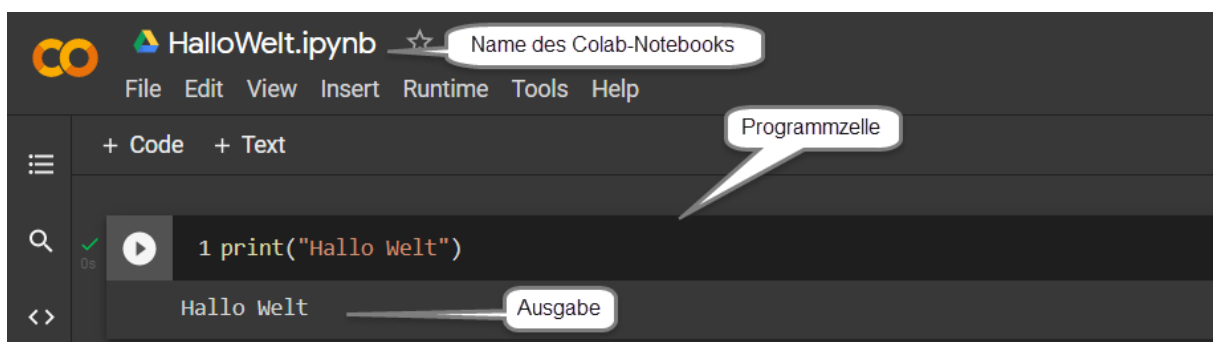


Figure 1.1: Colab-Notebook mit Programmzelle

Versuchen sie es! Mit **File->Save** können sie das Notebook abspeichern. Weitere Hinweise zum Umgang mit Colab-Notebook finden sie unter <https://research.google.com/colaboratory/faq.html>. Besonders smart ist die Funktion von Textzellen. Sie erlauben das Hinzufügen eigener Texte vor oder nach den Programmzellen; so können sie ein eigenes Skript erstellen.

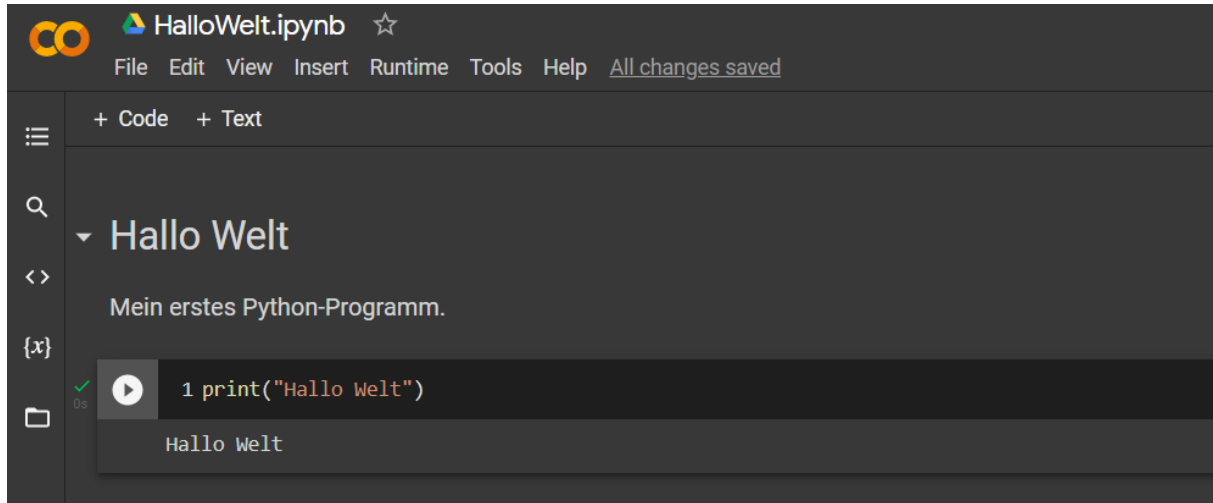


Figure 1.2: Textzelle über der Programmzelle

## Variante 2: Installation von Python

Alternativ können sie Python auch auf ihrem Rechner installieren. Die Installationsanleitung finden sie auf <https://www.python.org/>. Um die Installation zu prüfen verwenden sie den Befehl

```
python --version
```

Als Ausgabe erhalten sie die installierte Version von Python, zum Beispiel **Python 3.9.6**. Sie können nun mit einem einfachen Texteditor, z.B. Notepad++, die Programmbefehle in eine Datei schreiben. Wenn sie die Zeile `print("Hallo Welt")` in die Datei *HalloWelt.py* schreiben, so können sie das Programm mit folgendem Befehl ausführen.

```
python HalloWelt.py
```

Sie sollten nun die Ausgabe **Hallo Welt** sehen.

Verwenden sie die für Python-Programme übliche Dateierweiterung *.py*. Keine Leerzeichen im Dateinamen!

## 1.2 Programme und Fehler

Sie werden eventuell Syntax-Fehler in ihren Programmen haben. Syntax-Fehler entstehen, wenn Python ihr Programm nicht versteht. Oft geht es dabei um "Rechtschreibfehler". Schreiben sie also fälschlicherweise `prin("Hallo Welt")`, so erhalten sie bei der Programmausführung eine Fehlermeldung:

```
line 1, in <module>
    prin("Hallo Welt")
NameError: name 'prin' is not defined
```

Häufig lässt sich diese Meldung leicht verstehen, und sie können den Fehler korrigieren.

Läuft ein Programm ohne Fehlermeldung ab, so kann man daraus natürlich nicht auf die Korrektheit des Programms schließen! Ihr Programm kann also auch noch logisch falsch sein.

## 1.3 Python lernen

Es geht uns nicht in erster Linie darum gute Python-Programmierer zu werden. Unsere Kenntnisse müssen aber für die Lösung oder Analyse unserer Problemstellungen ausreichen. Das legt auch den Umfang an Python Know-How fest. Es gibt hierfür zahlreiche gute Bücher und Internetquellen. Eine ausgewählter Link, auf den ich regelmäßig verweisen werden ist

<https://www.w3schools.com/python/>

Suchen sie doch unter *Python Intro* nach dem Hallo-Welt-Beispiel. Sie finden dort unter *Try it Yourself* eine weitere Möglichkeit Python-Programme auszuführen!

# Chapter 2

## Daten

### 2.1 Elementare Datentypen

Daten<sup>1</sup> sind Ergebnisse von Beobachtungen, Messungen oder Berechnungen, die in einer bestimmten Form notiert, also aufgeschrieben sind. Häufig sprechen wir auch von *Werten*, statt von Daten. Ein Werte kann zum Beispiel die Zahl 27 oder die Note “*sehr gut*” sein. Offensichtlich gehören beide Werte zu verschiedene Typen von Werten, den sogenannten Datentypen. Wir beschäftigen uns mit den Datentypen *Zahlen*, *Text* und *Boolean*.

#### Zahlen

Werte dieses Datentyps sind z.B. 1,  $-1$ , 1.7 oder  $1/3$ . Wie sie wissen, lässt sich die Menge der Zahlen noch weiter einteilen in natürliche Zahlen und reelle Zahlen (und noch ein paar mehr, die aber vorerst nicht gebraucht werden. Wir verwenden die üblichen Symbole  $\mathbb{N}$  für die natürlichen Zahlen und  $\mathbb{R}$  für die reellen Zahlen.

Wir notieren Zahlen wie üblich und verwenden in der Dezimalnotation den Punkt als Trennsymbol.

#### Text

Werte dieses Datentyps sind zum Beispiel “Baum”, “Hans Huber” oder “sehr gut”. Zeichenketten beginnen und enden mit einem Anführungszeichen. In der Regel macht uns diese Notationen keine Probleme - manchmal wird es trotzdem ungemütlich: Kann eine Zeichenkette ein Anführungszeichen enthalten? Gibt es einen Unterschied zwischen der Zahl 123 und der Zeichenkette “123”? Wir vertiefen das hier nicht, sondern gehen die Fragen an, sobald sie uns begegnen.

#### Logischer Datentyp (Boolean)

Dieser Datentyp umfasst nur zwei Werte, die sogenannten Wahrheitswerte. Wir werden in diesem Text die Notation *True* und *False* verwenden.

Wir nennen diese Datentypen “elementar”, weil uns eine weitere Aufteilung nicht sinnvoll erscheint. Der Begriff “elementar” ist nicht ganz korrekt, weil z.B. die Zahl 123 ja eine Folge von Ziffern und die Zeichenkette “Baum” eine Folge von Buchstaben ist. Elementar wäre also eher der Datentyp *Ziffer* oder *Buchstabe*, als der Datentyp *Zahl* oder *Text*. Auch über diese begriffliche Ruppigkeit sehen wir zu Beginn hinweg. In der Programmierung werden sie aber eine Rolle spielen.

Stellen sie sich jeden Datentyp als Menge vor. Die Elemente der Menge sind die zulässigen Werte des Datentyps. Offensichtlich ist besitzt die Menge der Zahlen oder Texte unendlich viele Elemente, während der Logische Datentyp nur zwei Werte kennt.

---

<sup>1</sup>sing. Datum



## 2.2 Der Iris-Datensatz

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

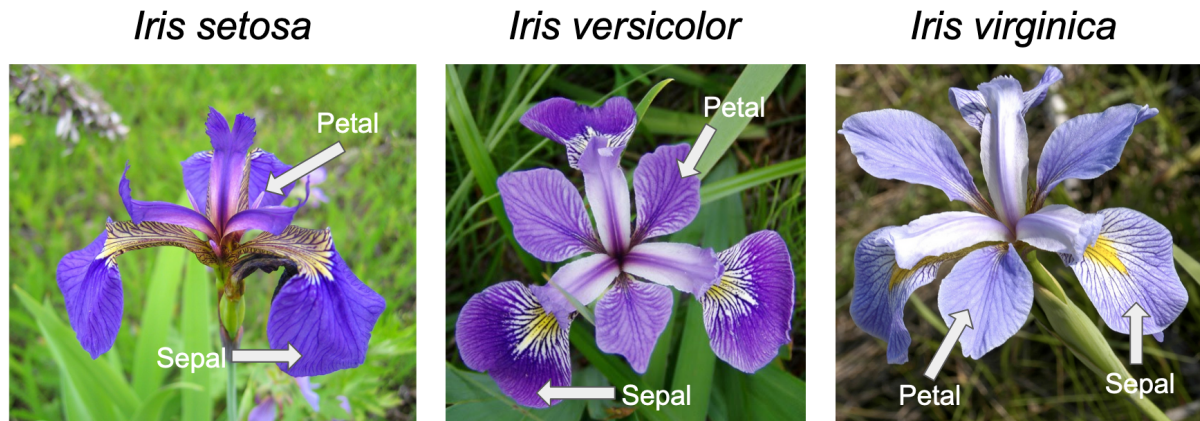


Figure 2.1: Download

Gemessen werden pro Blüte in cm

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

### 2.2.1 Datensatz

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

## 2.3 Datentypen

### 2.3.1 Elementare Datentypen

Zahlen

Strings

Logische Werte

Elementare Datentypen in Python

### 2.3.2 Komplexe Datentypen

Datum

Uhrzeit

Bilder

Komplexe Datentypen in Python

## 2.4 Skalenniveaus

### 2.4.1 Überblick

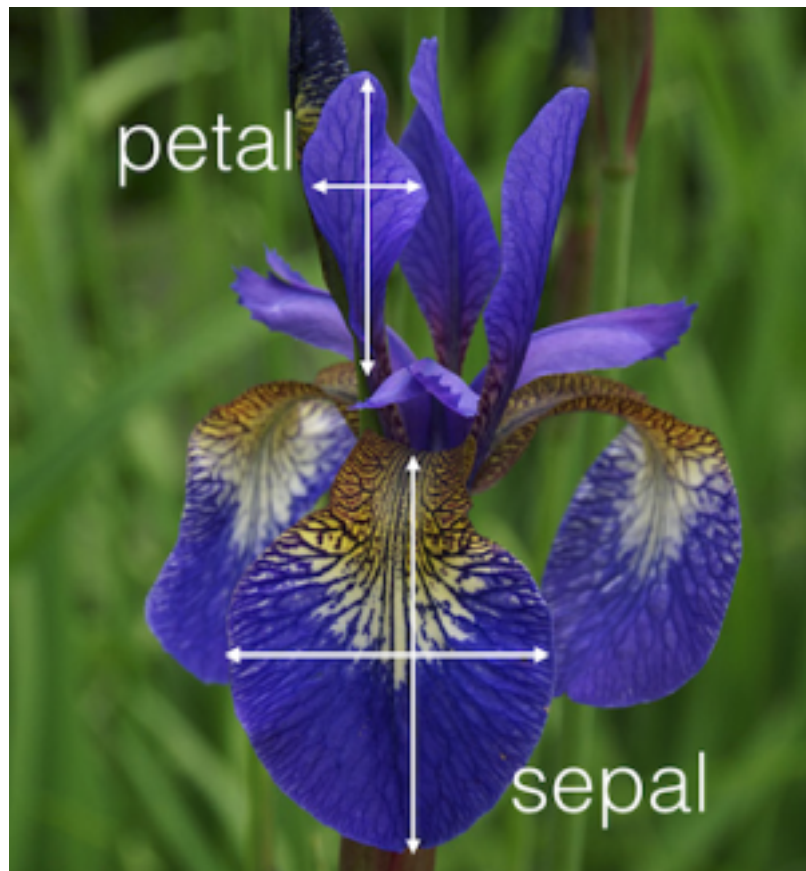


Figure 2.2: image (190)

	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

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))

## 2.4.2 Skalenniveaus im Iris-Datensatz

	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

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

---

Nominaler Wert	Ordinaler Wert
setosa	0
versicolor	1
virginica	2

---

## Chapter 3

# Datenstrukturen mit Python

# Latex Part

# Chapter 4

# Chapter

## 4.1 Latex Section (H2)

Ich habe mich bemüht, zahlreiche Übungsbeispiele und Youtube-Videos einzubauen. Viele Themen lassen sich so besser darstellen. Sofern es Medien im Netz gibt,

Parskip!! 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 verl

### 4.1.1 Latex Subsection (H3)

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 verl

#### Latex SubSubSection (H4)

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 verl

**Latex Paragraph (H5)** 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 verl