

Blatt 7

Vincent Kümmerle und Elvis Gnaglo

3. Dezember 2025

1 Zotero

Machinelles Lernen wird verwendet, um Muster in Datensammlungen herauszufinden. [1]

The screenshot shows a Zotero library entry for a journal article. The entry is titled "Quantum machine learning" and is categorized as a "Zeitschriftenartikel" (Journal Article). The authors listed are Biamonte, Jacob; Wittek, Peter; Pancotti, Nicola; Rebentrost, Patrick; Wiebe, Nathan; and Lloyd, Seth. The article is published in the journal "Nature", volume 549, issue 7671, pages 195-202, dated 2017-09. The language is English (en). The DOI is 10.1038/nature23474 and the ISSN is 1476-4687. The URL is https://www.nature.com/articles/nature23474. The entry was downloaded on 2.12.2025 at 10:26:08. The archive location is www.nature.com.

Quantum machine learning	
Infos	
Eintragsart	Zeitschriftenartikel
Titel	Quantum machine learning
Autor	Biamonte, Jacob
Autor	Wittek, Peter
Autor	Pancotti, Nicola
Autor	Rebentrost, Patrick
Autor	Wiebe, Nathan
Autor	Lloyd, Seth
Publikation	Nature
Band	549
Ausgabe	7671
Seiten	195-202
Datum	2017-09
Reihe	
Titel der Reihe	
Reihe Text	
Zeitschriften-Abkürzung	
Sprache	en
DOI	10.1038/nature23474
ISSN	1476-4687
Kurztitel	
URL	https://www.nature.com/articles/nature23474
Heruntergeladen am	2.12.2025, 10:26:08
Archiv	
Standort im Archiv	
Bibliothekskatalog	www.nature.com
Signatur	

Abbildung 1: Zeitschriftenartikel über Quantum Machine Learning in der Zotero-Bibliothek.

Das Buch “Experimentalphysik 1” von Wolfgang Demtröder erklärt die Grundlagen der Mechanik und Wärmelehre. [2]

Experimentalphysik 1: Mechanik und Wärme	
Info	
Citation key	demtroderExperimentalphysik1Mechanik2008
Item Type	Book
Title	Experimentalphysik 1: Mechanik und Wärme
Author	Demtröder, Wolfgang
Series	
Series Number	
Volume	
# of Volumes	
Edition	
Place	
Publisher	Springer
Date	2008
# of Pages	
Language	
ISBN	
Short Title	
URL	
Accessed	
Archive	
Loc. in Archive	
Library Catalog	
Call Number	
Rights	
Extra	
Date Added	12/2/2025, 10:28:02 AM
Modified	12/2/2025, 10:31:36 AM

Abbildung 2: Buch über die Mechanik und Wärmelehre in der Physik.

2 List Comprehension

```
1 import random
2 import matplotlib.pyplot as plt
3 import math
4
5 def generate_random_point_in_square(length):
```

```

6     """
7     Generate a random point in a square of a given side length
        centered around the origin with the sides aligned with x- and
        y-axis.
8     """
9     return tuple([length*random.uniform(-0.5, 0.5), length*random.
        uniform(-0.5, 0.5)])
10
11 def distance(p1, p2):
12     """
13     Measures the Euclidean distance between the points p1 and p2 in 2
        D.
14     """
15     # Berechnung des euklidischen Abstands: wurzel((x2-x1)^2 + (y2-y1
        )^2)
16     return math.sqrt((p1[0] - p2[0])**2 + (p1[1] - p2[1])**2)
17
18 def point_in_circle(p, center=(0,0), radius=1.0):
19     """
20     Checks if the provided point p lies in a circle of a given radius
        around the provided center.
21     """
22     # Ein Punkt liegt im Kreis, wenn sein Abstand zum Mittelpunkt
        kleiner oder gleich dem Radius ist.
23     return distance(p, center) <= radius
24
25
26 # TODO: generate a list of 10000 random points in a square of side
        length 2.0 using generate_random_point_in_square and list
        comprehension
27 # Wir nutzen _ als Laufvariable, da wir den Index nicht benötigen.
28 full_list_of_points = [generate_random_point_in_square(2.0) for _ in
        range(10000)]
29
30 # TODO: generate a list of those points in full_list_of_points that
        are within a radius of 1.0 of the origin (again using list
        comprehension)
31 # Wir filtern die Liste basierend auf der Bedingung point_in_circle
32 filtered_list_of_points = [p for p in full_list_of_points if
        point_in_circle(p, center=(0,0), radius=1.0)]
33
34 # TODO: print the ratio of the number of points in
        filtered_list_of_points to the total number of generated points
35 ratio = len(filtered_list_of_points) / len(full_list_of_points)
36 print(f"Verhältnis (Punkte im Kreis / Gesamt): {ratio}")

```

```

37 print(f"Annäherung an Pi (Verhältnis * 4): {ratio * 4}")
38 # Erklärung: Fläche Kreis / Fläche Quadrat = (pi*r^2) / (2r)^2 = pi
    /4. Daher ist das Verhältnis ca. pi/4.
39
40 # TODO: Generate separate lists of the x- and y-values of
    full_list_of_points and filtered_list_of_points
41 # x-Werte sind am Index 0, y-Werte am Index 1 jedes Tupels
42 x_values_full = [p[0] for p in full_list_of_points]
43 y_values_full = [p[1] for p in full_list_of_points]
44
45 x_values_filtered = [p[0] for p in filtered_list_of_points]
46 y_values_filtered = [p[1] for p in filtered_list_of_points]
47
48 # Plotten
49 plt.figure(figsize=(6, 6)) # Quadratische Darstellung für korrekte
    Proportionen
50 plt.scatter(x_values_full, y_values_full, s=1, label='Außerhalb',
    color='blue')
51 plt.scatter(x_values_filtered, y_values_filtered, s=1, label='
    Innerhalb', color='red')
52 plt.legend(loc='upper right')
53 plt.title(f'Monte Carlo Simulation (Verhältnis = {ratio})')
54 plt.axis('equal') # Wichtig, damit der Kreis rund und nicht verzerrt
    aussieht
55 plt.show()

```

3 Dateien lesen und Daten plotten

Literaturverzeichnis

- [1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, S. Lloyd, *Nature* **2017**, 549, 195–202.
- [2] W. Demtröder, *Experimentalphysik 1: Mechanik Und Wärme*, Springer, **2008**.