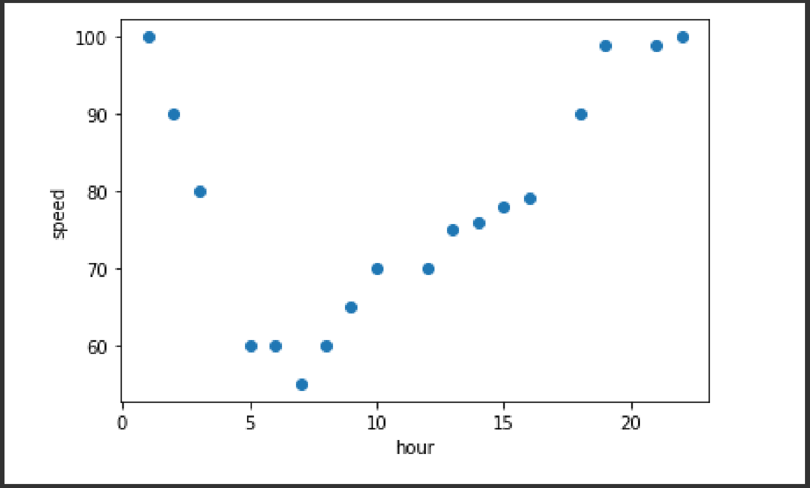
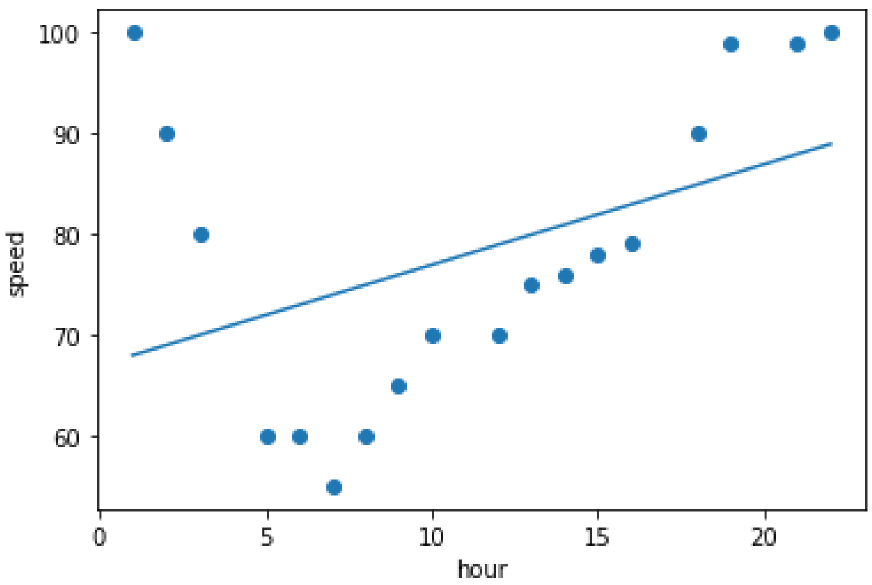
POLYNOMIALE REGRESSION

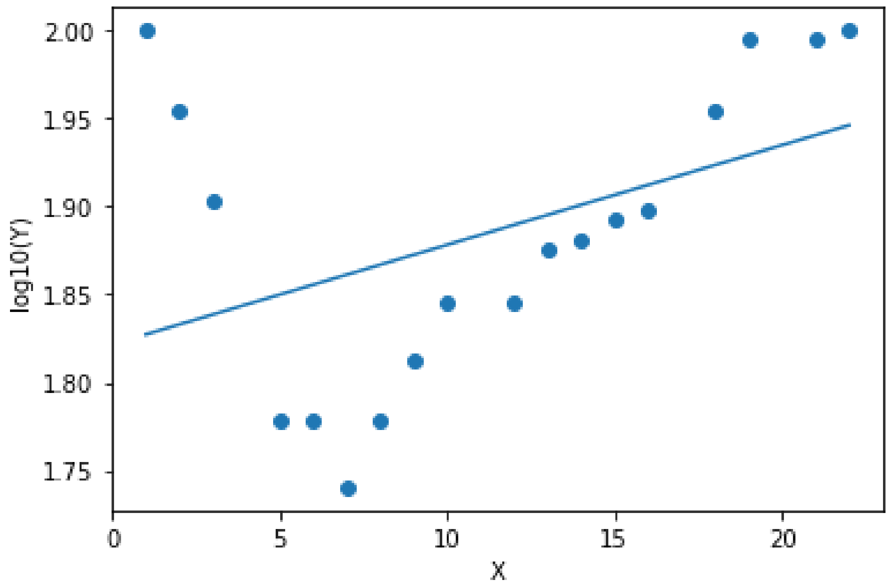
DAS PROBLEM

- Daten beschreiben die nicht lineare Zusammenhänge haben
- Ähnlich zu Thema letzter Sitzung

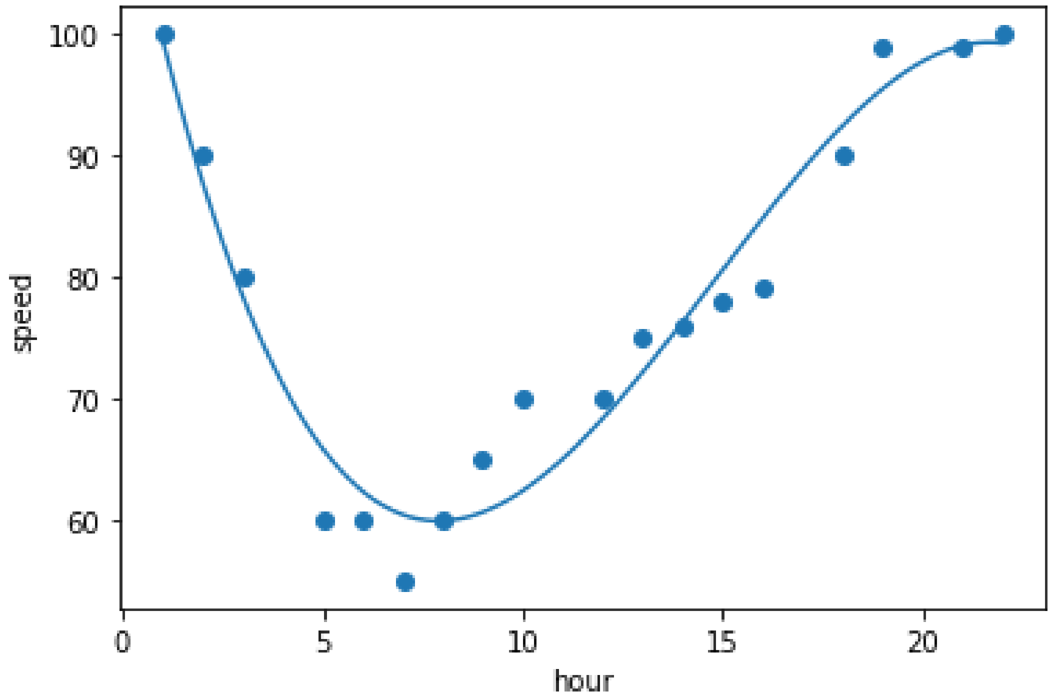




speed
hour
100
90
80
70
60
0
5
10
15
20



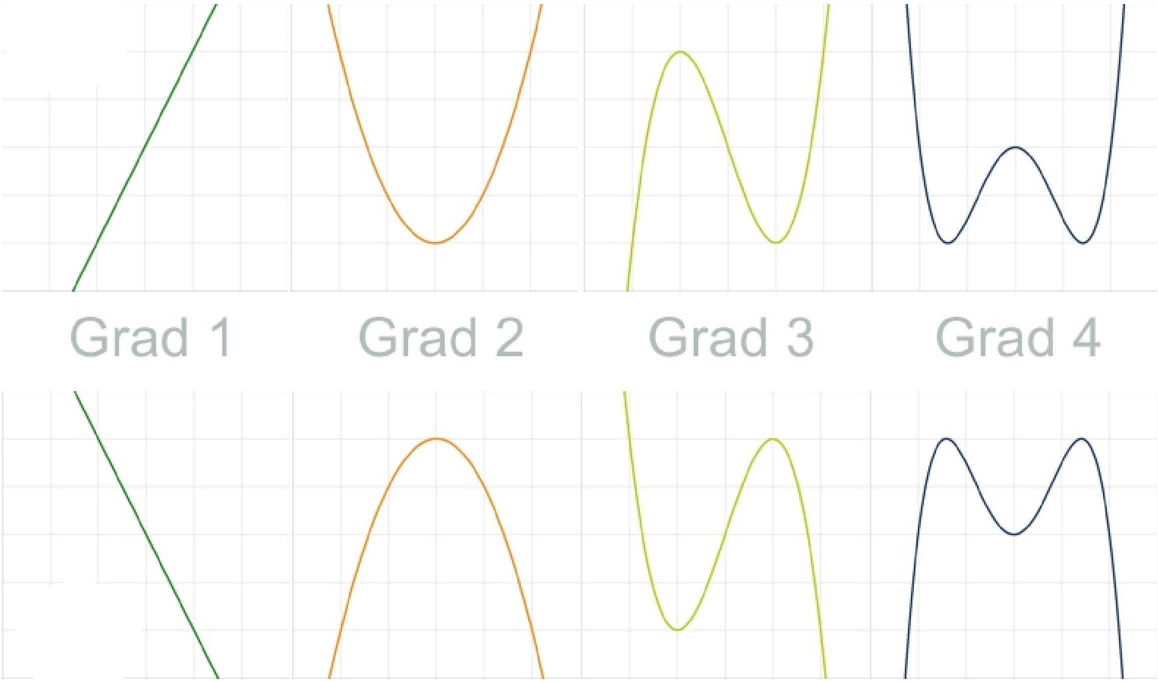
log10(Y)
X
2.00
1.95
1.90
1.85
1.80
1.75
0
5
10
15
20



DIE LÖSUNG

WAS SIND POLYNOME?

$$\hat{y} = a + b_1 x + b_2 x^2 + b_3 x^3 + \cdots + b_n x^n$$



VERSCHIEDENE ARTEN VON POLYNOMEN

Problem – welcher Grad soll die Funktion haben?

PRINZIP

- Annahme von Polynome 2. Ordnung

$$\hat{y} = a + b_1 \cdot x + b_2 \cdot x^2.$$

- Summe der quadrierten Abweichungen minimieren

$$\sum_{i=1}^{n} \left[y_i - (a + b_1 \cdot x_i + b_2 \cdot x_i^2) \right]^2$$

Nach Bortz

PRINZIP

$$
\begin{aligned}
\sum_i y_i &= an & &+ b_1 \sum_i x_i + b_2 \sum_i x_i^2, \\
\sum_i x_i \cdot y_i &= a \sum_i x_i & &+ b_1 \sum_i x_i^2 + b_2 \sum_i x_i^3, \\
\sum_i x_i^2 \cdot y_i &= a \sum_i x_i^2 & &+ b_1 \sum_i x_i^3 + b_2 \sum_i x_i^4.
\end{aligned}
$$

Lineares Gleichungssystem

Verweis auf Substitutionsverfahren
Oder Gauß

Nach Bortz

PRINZIP

Objekt-Nr.	x	y	$x \cdot y$	x^2	x^3	x^4	$x^2 \cdot y$
1	1,1	1,3	1,43	1,21	1,33	1,46	1,57
2	1,3	3,7	4,81	1,69	2,20	2,86	6,25
3	1,5	4,4	6,60	2,25	3,38	5,06	9,90
4	2,2	5,4	11,88	4,84	10,65	23,43	26,14
5	2,5	5,8	14,50	6,25	15,63	39,06	36,25
6	3,3	5,5	18,15	10,89	35,94	118,59	59,90
7	3,4	5,2	17,68	11,56	39,30	133,63	60,11
8	3,7	2,9	10,73	13,69	50,65	187,42	39,70
9	3,8	3,7	14,06	14,44	54,87	208,51	53,43
10	4,1	2,0	8,20	16,81	68,92	282,58	33,62
Summen:	26,9	39,9	108,04	83,63	282,87	1002,60	326,87

Nach Bortz

PRINZIP

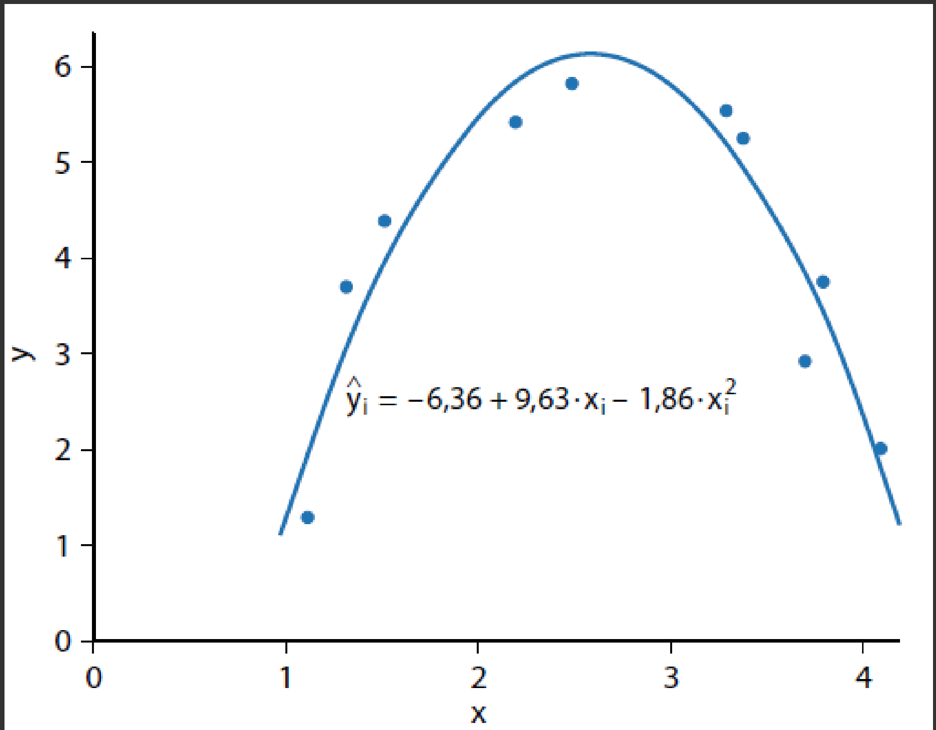
$$\sum_i y_i = an \qquad + b_1 \sum_i x_i + b_2 \sum_i x_i^2,$$
$$\sum_i x_i \cdot y_i = a \sum_i x_i \quad + b_1 \sum_i x_i^2 + b_2 \sum_i x_i^3,$$
$$\sum_i x_i^2 \cdot y_i = a \sum_i x_i^2 \quad + b_1 \sum_i x_i^3 + b_2 \sum_i x_i^4.$$

x	y	$x \cdot y$	x^2	x^3	x^4	$x^2 \cdot y$
26,9	39,9	108,04	83,63	282,87	1002,60	326,87

$$39{,}9 = 10 \cdot a \qquad + 26{,}9 \cdot b_1 \quad + 83{,}63 \cdot b_2,$$
$$108{,}04 = 26{,}9 \cdot a \quad + 83{,}63 \cdot b_1 \quad + 282{,}87 \cdot b_2,$$
$$326{,}87 = 83{,}63 \cdot a + 282{,}87 \cdot b_1 + 1002{,}60 \cdot b_2.$$

Nach Bortz

PRINZIP



https://www.arndt-bruenner.de/mathe/scripts/gleichungssysteme.htm

PRINZIP

$$\hat{y} = a + b_1 x + b_2 x^2 + b_3 x^3$$

$$\hat{y} = a + b_1 x_1 + b_2 x_2 + b_3\, x_3$$

x_1 =x, $x_2 = x^2$, $x_3 = x^3$

UMSETZUNG

- import numpy as np
- import pandas as pd
- import matplotlib.pyplot as plt

UMSETZUNG I - NUMPY

```
import numpy as np
import statsmodels.formula.api as smf

#data
x = np.array([0.0, 1.0, 2.0, 3.0, 4.0, 5.0])
y = np.array([0.0, 0.8, 0.9, 0.1, -0.8, -1.0])

#grad der Funktion
degree = 3

#für die Summary benötigt
df = pd.DataFrame(columns=['y', 'x'])
df['x'] = x
df['y'] = y

weights = np.polyfit(x, y, degree)

model = np.poly1d(weights)
results = smf.ols(formula='y ~ model(x)', data=df).fit()
```

```
#Input as array with 1 dimension
```

`Polyfit`

Returns Polynomial coefficients, highest power first.

UMSETZUNG NUMPY II

```
#Results
print(results.summary())

#Visualisierung
plt.scatter(x, y, color = 'blue')

xModel = np.linspace(min(x), max(x))
yModel = np.polyval(weights, xModel)

plt.plot(xModel, yModel)


plt.title('Polynomial Regression')
plt.xlabel('X')
plt.ylabel('Y')

plt.show()
```

UMSETZUNG II - SKLEARN

```
from sklearn.linear_model import LinearRegression

from sklearn.preprocessing import PolynomialFeatures

import statsmodels.api as sm


#data

x = np.array([0.0, 1.0, 2.0, 3.0, 4.0, 5.0])

y = np.array([0.0, 0.8, 0.9, 0.1, -0.8, -1.0])

x = x.reshape(-1,1)
```

```
#input als 2-d array
```

```
poly = PolynomialFeatures(degree = 3)
X_poly = poly.fit_transform(x)

X_vals = np.linspace(min(x), max(x))
lin2 = LinearRegression()
lin2.fit(X_poly, y)
X_vals_poly = poly.transform(X_vals)
y_vals = lin2.predict(X_vals_poly)
```

UMSETZUNG II – SKLEARN II

```
#results
model = sm.OLS(y, X_poly).fit()
print(model.summary())

#Visualisierung
plt.scatter(x, y)
plt.plot(X_vals, y_vals)
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```

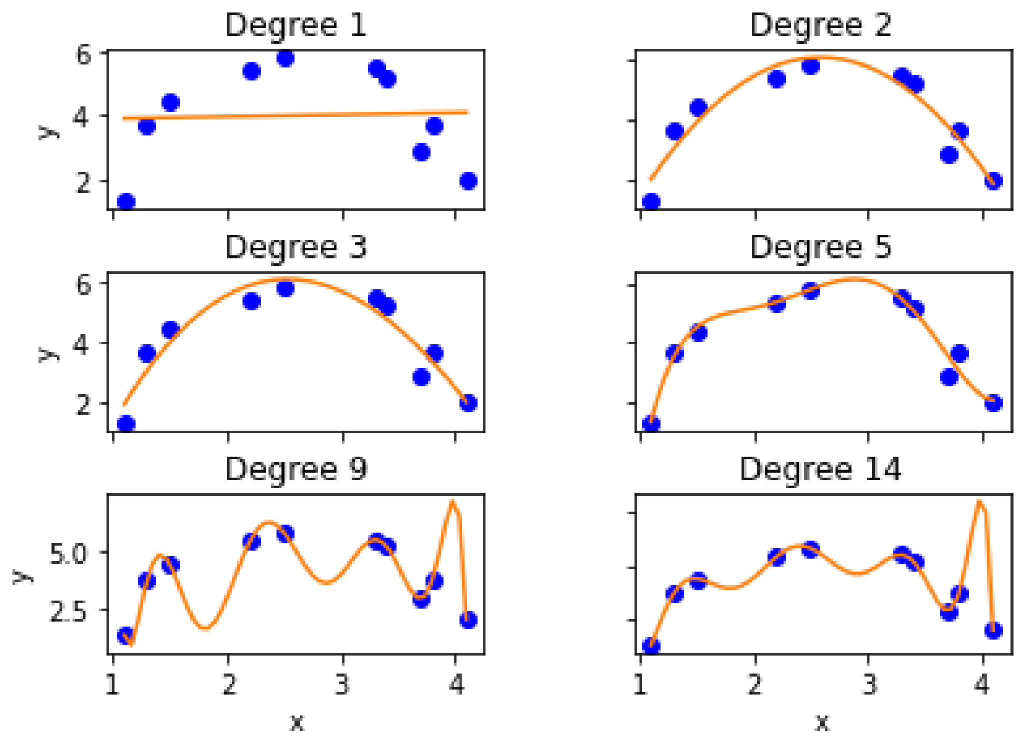
ZUSAMMENFASSUNG UMSETZUNG

	Numpy	**SKlearn**
Input	Ein-dimensionales Array	Zwei-dimensionales Array
Summary	Über smf import statsmodels.formula.api as smf Braucht deswegen df	Über sm import statsmodels.api as sm
Prinzip	Berechnung der weights Dann Training des Models	Umwandlung in Lineare Regression – dann Training dieser
Predictions für einen Wert von 4.5	pred = 4.5 np.polyval(weights, pred)	pred = np.array(4.5) pred = pred.reshape(-1, 1) lin2.predict(poly.fit_transform(pred))

```
df1 =
pd.read_csv(r"https://raw.githubusercontent.com/ck282/statistic0812/main/salaries
.txt", sep="\t")
```

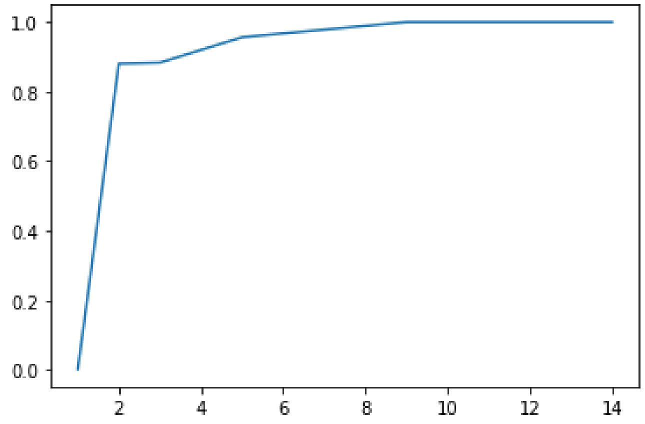
FRAGE NACH DEGREE

- Jackson: Grad größer als 3 oder 4 selten
- Gefahr Overfitting



```
plt.plot(degrees, rsquared)
```

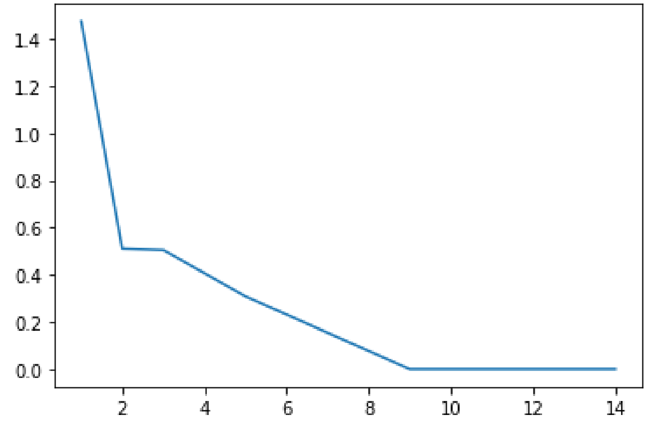
[<matplotlib.lines.Line2D at 0x25282735c90>]




```
degrees = [1,2,3,5,9,14]

rsquared = []


for d in degrees:

    model = np.poly1d(np.polyfit(x, y, d))

    rsquare = smf.ols(formula='y ~ model(x)', data=df).fit().rsquared


    rsquared.append(rsquare)
```

```
plt.plot(degrees, rmses)
```

[<matplotlib.lines.Line2D at 0x25281f1df60>]




```
from statsmodels.tools.eval_measures import rmse

degrees = [1,2,3,5,9,14]
rmses = []

for d in degrees:

    weights = np.polyfit(x, y, d)
    ypred = np.polyval(weights, x)

    c_rmse = rmse(y, ypred)


    rmses.append(c_rmse)
```

HAUSAUFGABE

- Aufgabe 1: https://raw.githubusercontent.com/ck282/statistic0812/main/salaries.txt
- A) Plotte die Daten mit einer polynomialen Regression mit einem Grad von 3. Benutze dabei entweder numpy oder sklearn.
- B) Was ist die Funktionsgleichung?
- C) Treffe eine Vorhersage für einen x-Wert von 6.5
- D) Wähle eine bereits behandelte Modellierung aus und vergleiche, ob diese besser oder schlechter auf die Daten passt.

Aufgabe 2: https://raw.githubusercontent.com/ck282/statistic0812/main/values.txt

- Lade die Daten ein und finde heraus welcher Grad/Degree für die Daten am besten geeignet ist, begründe deine Antwort.

QUELLEN

- Bortz, Jürgen. *Statistik: Für Human- und Sozialwissenschaftler (Springer Lehrbuch)*. 6., Vollst. überarb. u. aktualisierte, Springer, 2022.
- Jackson, Dr. S. *Chapter 7 Polynomial Regression | Machine Learning*. 6. April 2022, bookdown.org/ssjackson300/Machine-Learning-Lecture-Notes/polynomial-regression.html.
- Loong, Joshua. „Fitting Polynomial Regressions in Python“. *Joshua Loong*, 3. Oktober 2018, joshualoong.com/2018/10/03/Fitting-Polynomial-Regressions-in-Python.
- *Polynomial Regression - which python package to use?* https://zerowithdot.com/polynomial-regression-in-python/
- *Python Machine Learning Polynomial Regression*. www.w3schools.com/python/python_ml_polynomial_regression.asp.
- *Vorlesung Intelligente Datenanalyse*. fuzzy.cs.ovgu.de/studium/ida.