Lineare_Regression_Moore_Penrose

April 5, 2021

1 Lineare Regression mit Moore-Penrose-Pseudo-Inversen

Wir können die Koeffizienten einer Linearen Regressionsgleichung auch mit Hilfe der Linearen Algebra ermitteln. Allgmeine benötigt man bei k Koeffizienten (Unbekannten) k linear unabhängige Gleichungen, um eine Lösung zu ermitteln. Ist Y der Ergebnisvektor und X die Koeffizientenmatrix, so kann man schreiben:

$Y = X \cdot \beta$

Gesucht sind die Koeffizienten β. Multipliziert man beide Seiten der Gleichung mit der inversen Matrix von X, also X^-1

$X^-1 \cdot Y = X^-1 \cdot X \cdot \beta$

$X^-1 \cdot X$ ergibt die Einheitsmatrix, sodass sich ergibt:

$\beta = X^-1 \cdot Y$

Hier haben wir allerdings wesentlich mehr Gleichungen (entspricht Stichprobengröße) und erhalten somit ein überbestimmtes Gleichungssystem. Hier hilft uns die Moore-Penrose-Pseudo-Inverse X^+, die wie folgt definiert ist:

$X^+ = (X^T X)^{-1} X^T$

Eingesetzt in die obige Gleichung ergibt sich:

$\beta = X^+ Y \cdot Y = (X^T X)^{-1} X^T \cdot Y$

Diese Gleichung setzen wir jetzt mit Hilfe von *numpy*-Funktionen um. Wir benötigen:

- X^T: Wir müssen also die Matrix X transponieren (aus Zeilen werden Spalten, aus Spalten werden Zeilen. Dies geschieht ganz einfach durch ".T", also zum Beispiel x.T, wenn x die Matrix ist.
- Multiplikation von Matrizen: Dies können wir durch den @-Operator durchführen. Sind a und b Matrizen (2-dimensionale Numpy-Arrays), so können wir mit a@b eine Matrixmultiplikation durchführen.
- Schließlich müssen wir noch die Inverse einer Matrix ermitteln. Wollen wir von der Matrix a die Inverse berechnen, also a^{-1}, so erledigt dies die Funktion *np.linalg.inv*.

Mit diesen Informationen können wir nun die Koeffizienten des überbestimmten Gleichungssystems lösen. Wir wollen wieder den Verbrauch der Autos in Abhängigkeit von Gewicht und Leistung ermitteln.

Laden wir zuerste den Datensatz:

[1]:
```
import pandas as pd
url = "https://raw.githubusercontent.com/troescherw/datasets/master/autos.csv"
autos = pd.read_csv(url)
autos.head()
```

[1]:

	Verbrauch	Leistung	Gewicht
0	11.20	82	1310
1	11.20	82	1437
2	10.32	69	1160
3	10.99	82	1607
4	12.58	130	1720

In der Spalte *Verbrauch* stehen die vorherzusagenden Werte, also der Ergebnisvektor y:

[4]:
```
y = autos.Verbrauch
```

[ ]:
```
Die Spalten *Leistung* und *Gewicht* bilden die Koeffizienten-Matrix X:
```

[7]:
```
x = autos[["Leistung", "Gewicht"]]
```

Allerdings benötigen wir noch einen Koeffizienten für die Konstante β_0. Da es sich eben um eine Konstante handelt, fügen wir in die Koeffizientenmatrix noch eine Spalte mit Einsen ein:

[8]:
```
x.insert(0, "Beta_0", 1)
x.head()
```

[8]:

	Beta_0	Leistung	Gewicht
0	1	82	1310
1	1	82	1437
2	1	69	1160
3	1	82	1607
4	1	130	1720

Nun wenden wir die Formel von oben an:

[10]:
```
import numpy as np
betas = np.linalg.inv(x.T @ x) @ x.T @ y
betas
```

[10]:
```
0    1.493910
1    0.023576
2    0.005399
dtype: float64
```

Auf das gleiche Ergebnis kommt auch die Funktion *ols*:

[14]:
```
import statsmodels.formula.api as sm
model = sm.ols("Verbrauch~Leistung+Gewicht", data=autos).fit()
model.params
```

[14]:
```
Intercept    1.493910
Leistung     0.023576
Gewicht      0.005399
dtype: float64
```