



Übungslektion 13 – Machine Learning II

Informatik II

20. / 21. Mai 2025

Heutiges Programm

- Cross-Validation
- Grid Search für Polynomregression
- Neuronale Netze

Last week: ML I

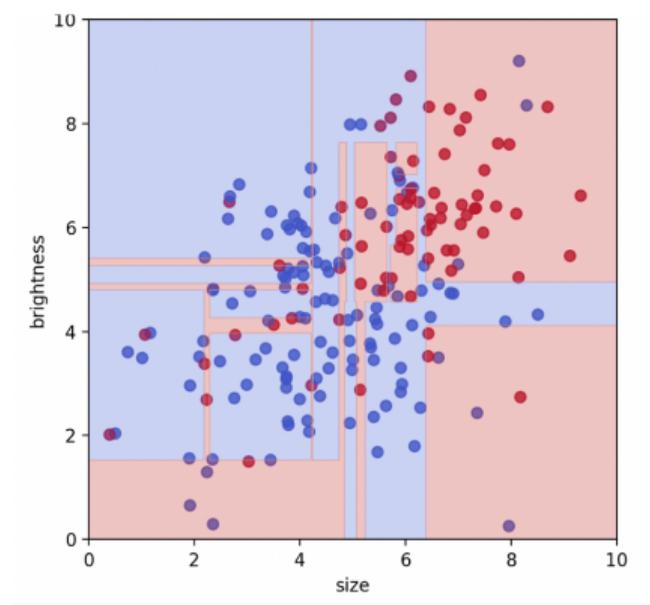
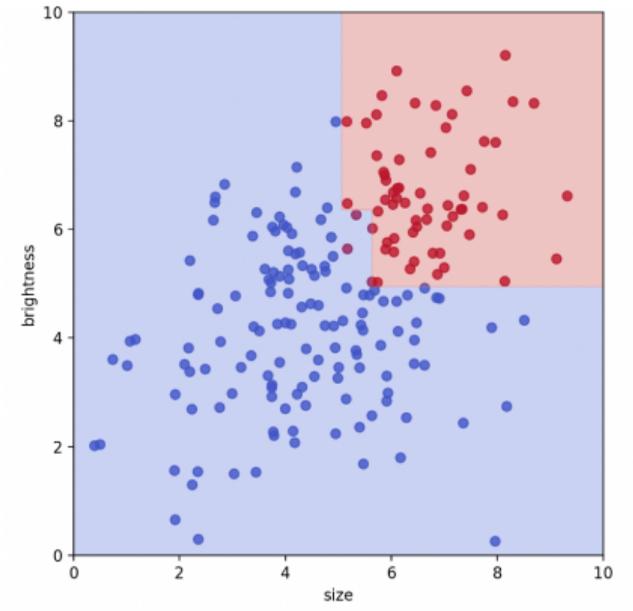
Gibt es Fragen?

- Cancer Detection
- Diabetes Prediction
- Gini Index

1. Cross-Validation

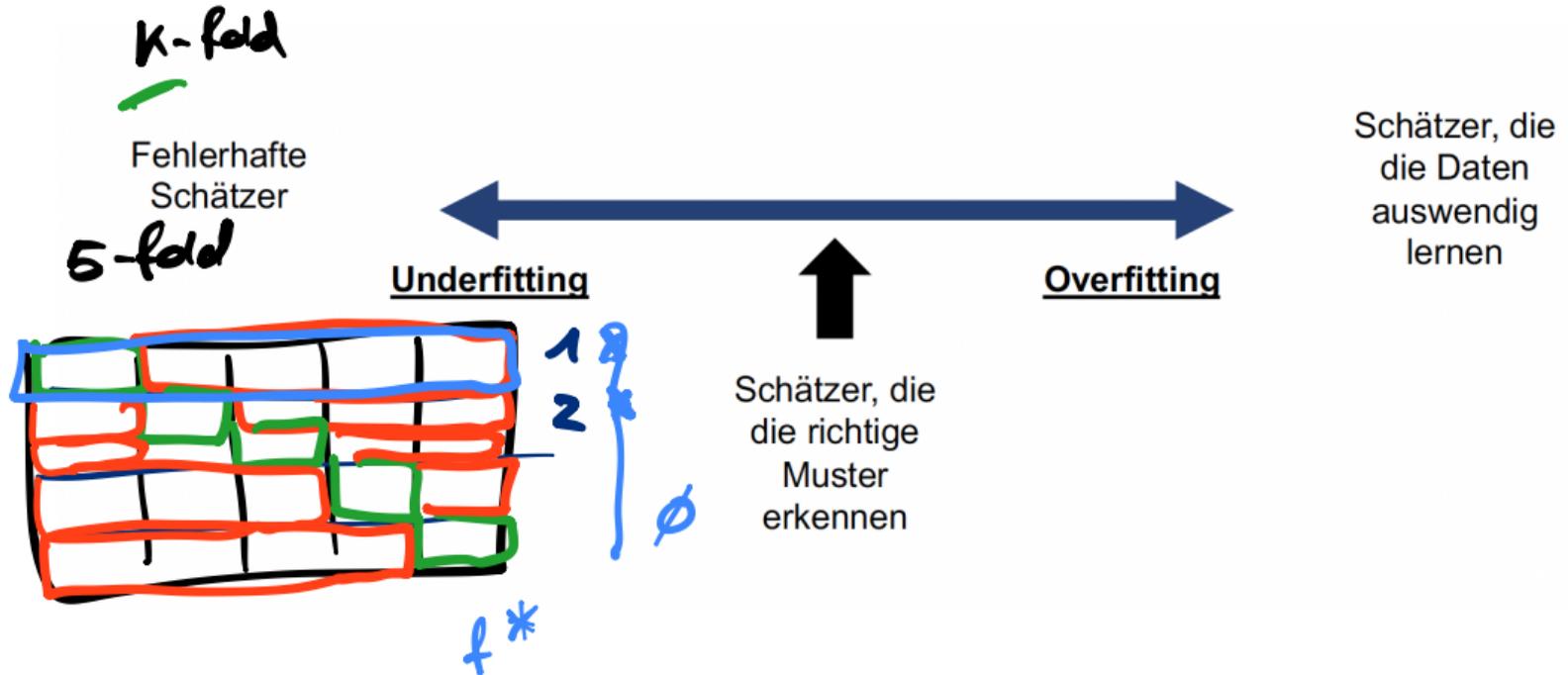
Zusammenfassung

Wenn es Rauschen in dem Datensatz gibt, kann das Training von ML-Modellen falsch laufen.



Zusammenfassung

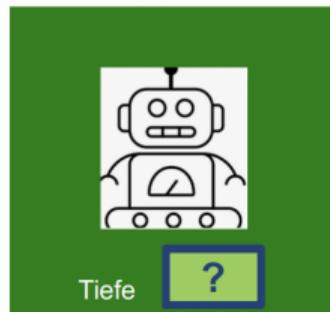
Um dies zu korrigieren, muss man die richtigen Hyperparameter für das Modell vor dem Training auswählen.



Zusammenfassung

Grid-Search mit Cross-Validation ist eine beliebte Methode, um die richtigen Hyperparameter zu finden.

$\left. \begin{array}{l} \text{max_depth} \\ = [2, 20, 200] \end{array} \right\}$



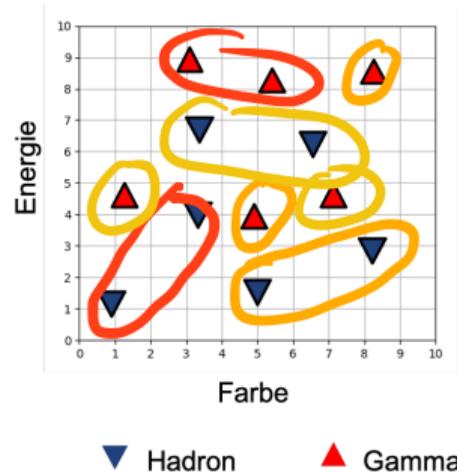
	Tiefe=2	Tiefe=20	Tiefe=200
% mistakes in 	20	5	50
% mistakes in 	15	3	80
% mistakes in 	13	4	70

ETH Zürich

Übung

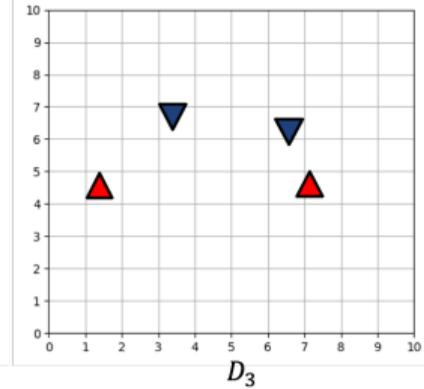
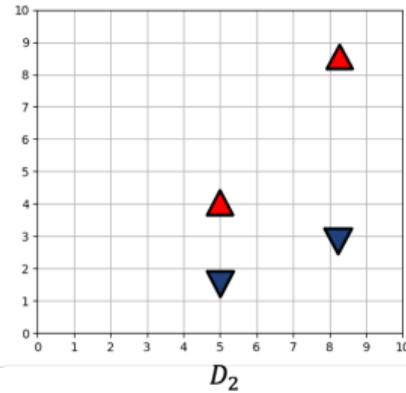
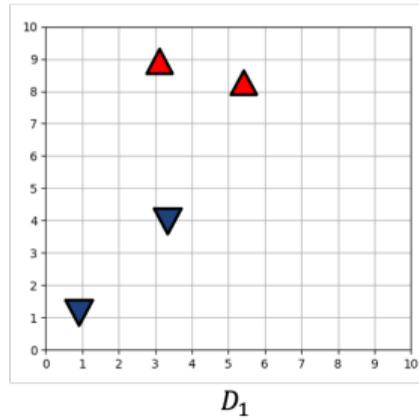
Betrachten Sie den folgenden Datensatz.

3-fold



Übung

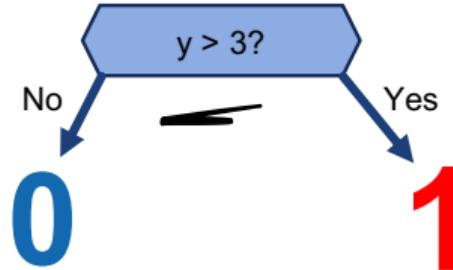
Der Datensatz ist wie folgt in drei Teile unterteilt.



$\gamma > \epsilon$!

Übung

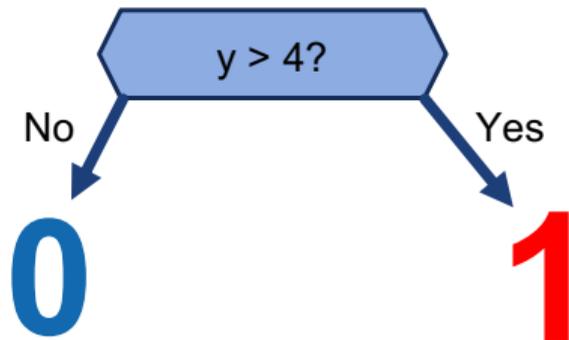
Das Modell besteht aus allen Bäumen, die nur eine Frage über die x - oder y -Koordinate stellen. Ein Schätzer in diesem Modell sieht so aus:



Berechnen Sie für jeden der drei Teile einen Schätzer für diesen Teil und bewerten Sie seine Genauigkeit anhand der anderen beiden Teile.

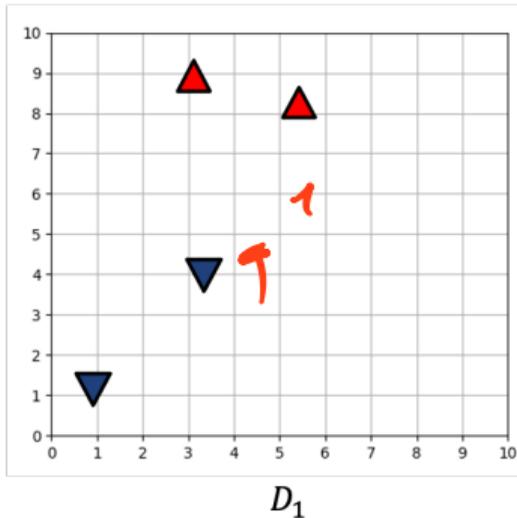
Lösung

- Für D_1 betrachten Sie den folgenden Schätzer. Er prognostiziert 1 gdw $y > 4$.
- In D_2 macht der Schätzer 1 Fehler. In D_3 macht er 2 Fehler.

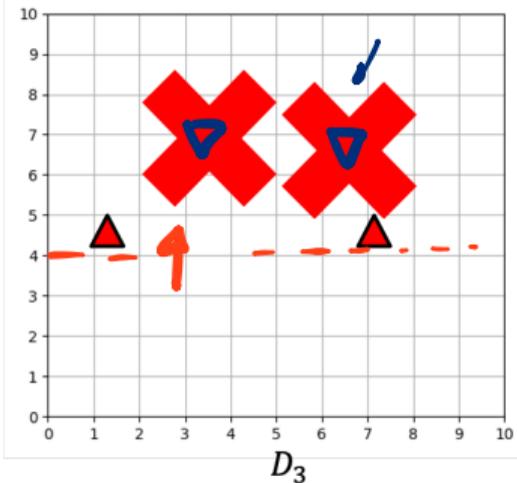
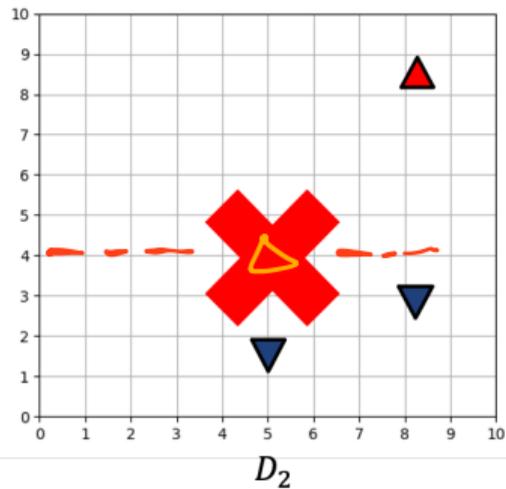


Lösung

γ

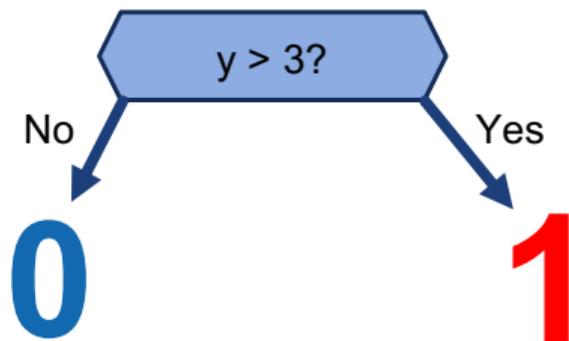


$$\gamma > 4$$

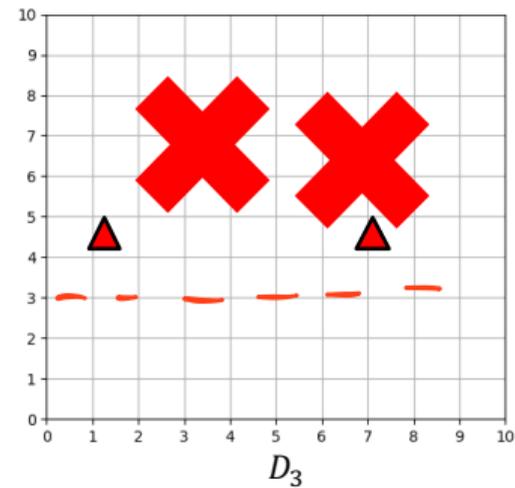
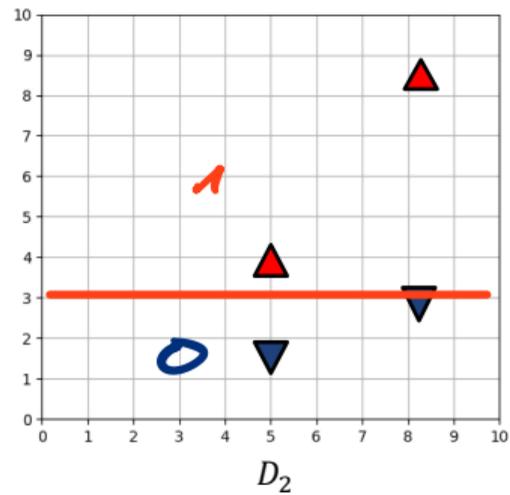
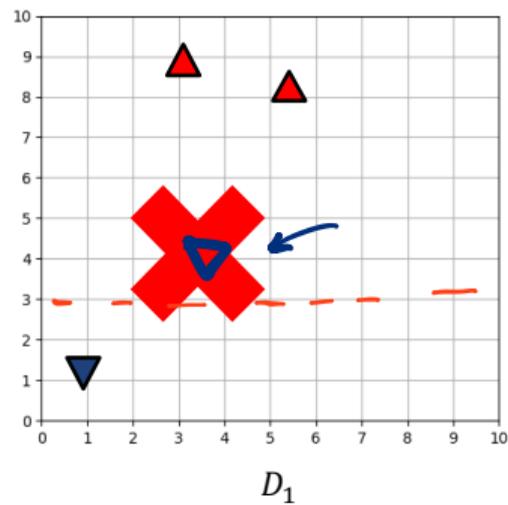


Lösung

- Für D_2 betrachten Sie den folgenden Schätzer. Er prognostiziert 1 gdw $y > 3$.
- In D_1 macht der Schätzer 1 Fehler. In D_3 macht er 2 Fehler.



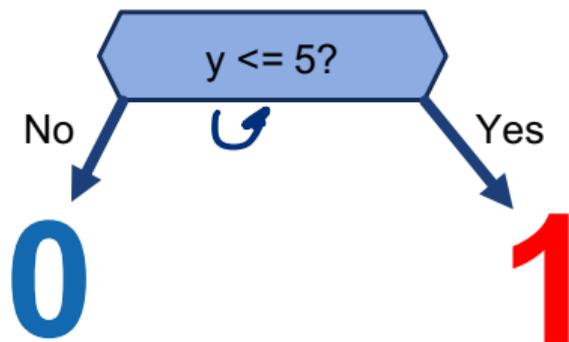
Lösung



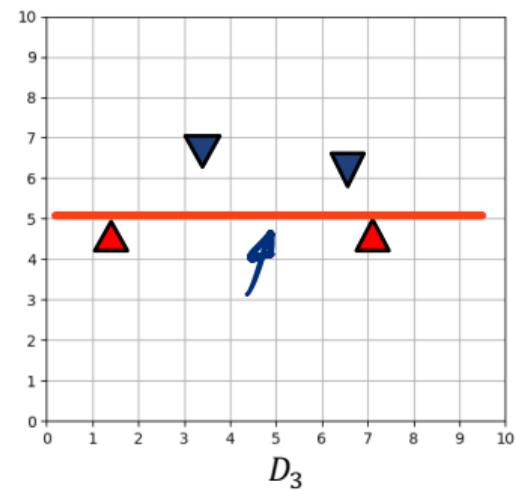
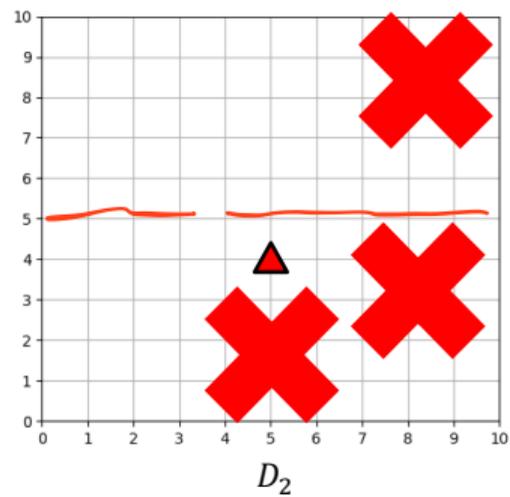
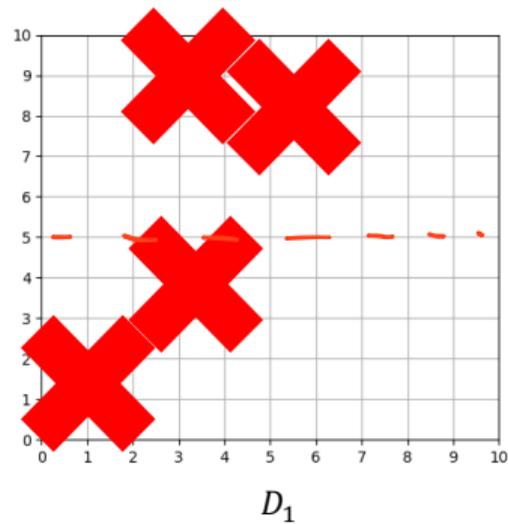
$\gamma > 3$

Lösung

- Für D_3 betrachten Sie den folgenden Schätzer. Er prognostiziert 1 gdw $y \leq 5$.
- In D_1 macht der Schätzer 4 Fehler. In D_2 macht er 3 Fehler.



Lösung

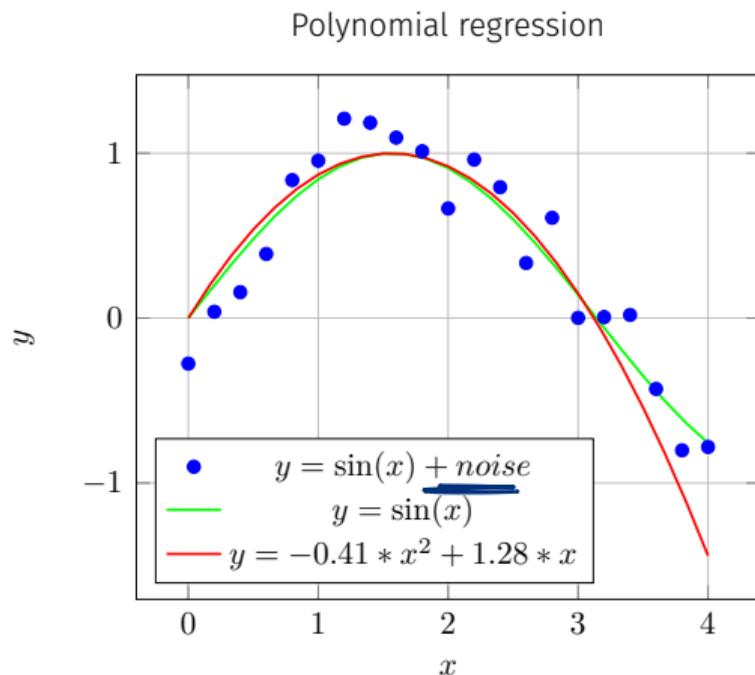


$y=5?$

2. Grid-Search für Polynomregression

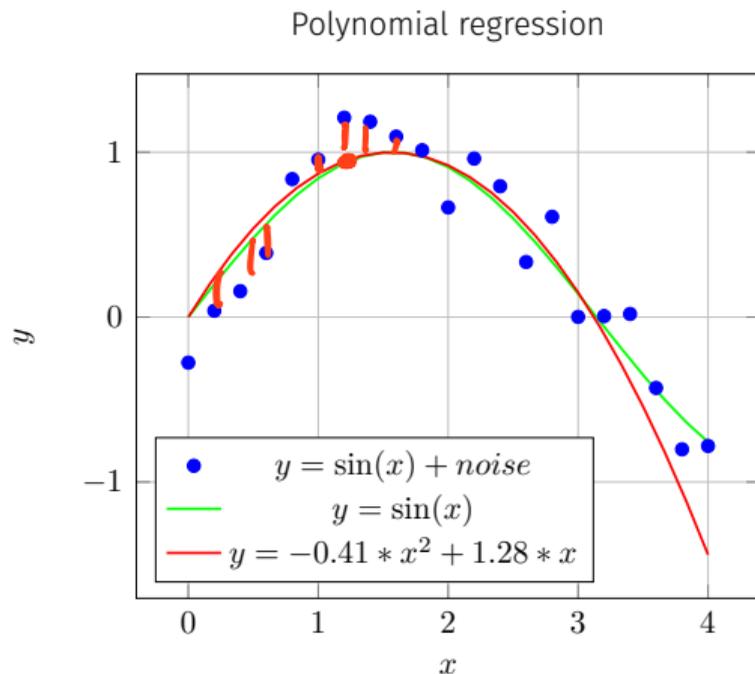
Übung

- Sie erhalten einen Datensatz $D = \{(x_1, y_1), \dots, (x_n, y_n)\} \subseteq \mathbb{R}^2$ mit Punkten in \mathbb{R}^2 .



Übung

- Sie erhalten einen Datensatz $D = \{(x_1, y_1), \dots, (x_n, y_n)\} \subseteq \mathbb{R}^2$ mit Punkten in \mathbb{R}^2 .
- Wir möchten ein Polynom f finden, so dass $f(x_i)$ so nahe wie möglich an y_i liegt, für $i \leq n$.



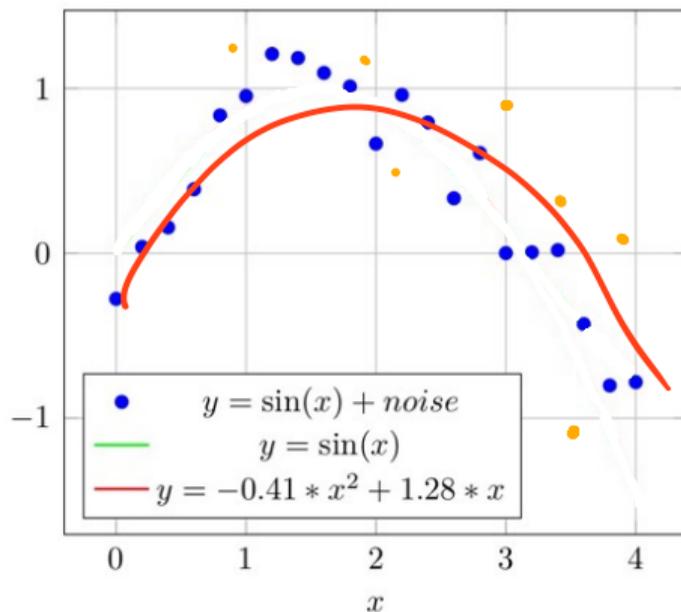
Angenommen, als Modell verwenden wir die Menge aller Polynome und als Verlustfunktion verwenden wir den mittleren quadratischen Fehler (MSE). Wir erinnern daran, dass der MSE, für einen Datensatz D und eine Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$, wie folgt definiert ist:

$$MSE(D, f) = \sum_{i \leq n} \|y_i - f(x_i)\|^2.$$

Übung

Beantworten Sie die folgenden Fragen:

- Wie kann es zu **Overfitting** bei diesem Datensatz kommen?



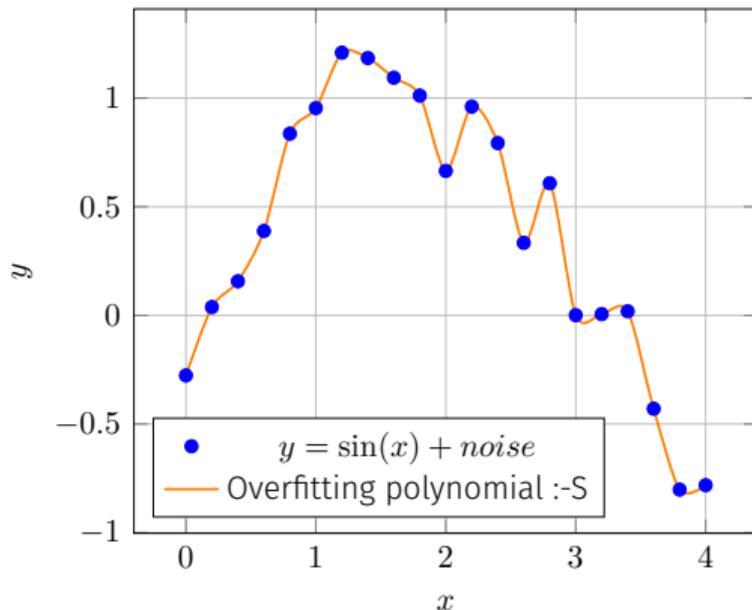
Übung

Beantworten Sie die folgenden Fragen:

- Wie kann es zu Overfitting bei diesem Datensatz kommen? Ein Polynom kann durch alle Punkte hindurch verlaufen, ohne das korrekte zugrundeliegende Polynom zu lernen.

Polynom - 20

Polynomial regression

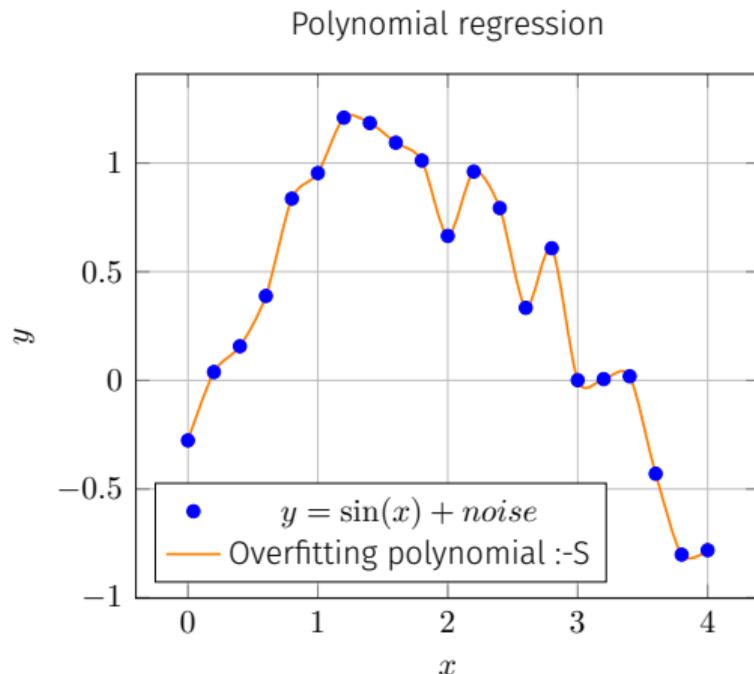


20

Übung

Beantworten Sie die folgenden Fragen:

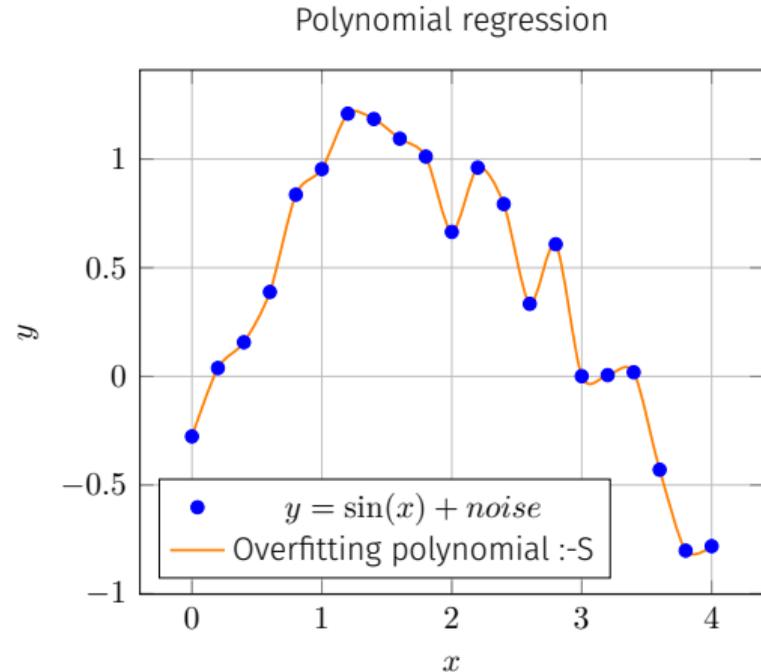
- Wie kann es zu Overfitting bei diesem Datensatz kommen? Ein Polynom kann durch alle Punkte hindurch verlaufen, ohne das korrekte zugrundeliegende Polynom zu lernen.
- Welchen Hyperparameter können wir steuern, um Overfitting zu vermeiden?



Übung

Beantworten Sie die folgenden Fragen:

- Wie kann es zu Overfitting bei diesem Datensatz kommen? Ein Polynom kann durch alle Punkte hindurch verlaufen, ohne das korrekte zugrundeliegende Polynom zu lernen.
- Welchen Hyperparameter können wir steuern, um Overfitting zu vermeiden? Den Grad des Polynoms.



Übung

Beschreiben Sie die Schritte zur Anpassung eines Polynoms an einen Datensatz mit Grid-Search und Cross-Validation.

Übung

Beschreiben Sie die Schritte zur Anpassung eines Polynoms an einen Datensatz mit Grid-Search und Cross-Validation.

- Wählen Sie eine Menge $C = \{d_1, \dots, d_m\} \subseteq \mathbb{N}$ von Kandidaten für die Grade der Polynome.

Übung

Beschreiben Sie die Schritte zur Anpassung eines Polynoms an einen Datensatz mit Grid-Search und Cross-Validation.

- Wählen Sie eine Menge $C = \{d_1, \dots, d_m\} \subseteq \mathbb{N}$ von Kandidaten für die Grade der Polynome.
- Teilen Sie den Datensatz in k Teile D_1, \dots, D_k (Falten) auf

Übung

Beschreiben Sie die Schritte zur Anpassung eines Polynoms an einen Datensatz mit Grid-Search und Cross-Validation.

- Wählen Sie eine Menge $C = \{d_1, \dots, d_m\} \subseteq \mathbb{N}$ von Kandidaten für die Grade der Polynome.
- Teilen Sie den Datensatz in k Teile D_1, \dots, D_k (Falten) auf
 - Für jeden Teil $i = 1, \dots, k$ und jedes $d \in C$, tun Sie folgendes.

Übung

Beschreiben Sie die Schritte zur Anpassung eines Polynoms an einen Datensatz mit Grid-Search und Cross-Validation.

- Wählen Sie eine Menge $C = \{d_1, \dots, d_m\} \subseteq \mathbb{N}$ von Kandidaten für die Grade der Polynome.
- Teilen Sie den Datensatz in k Teile D_1, \dots, D_k (Falten) auf
 - Für jeden Teil $i = 1, \dots, k$ und jedes $d \in C$, tun Sie folgendes.
 - Trainieren Sie ein Polynom $f_{i,d}$ des Grades d auf \bar{D}_i , wobei $\bar{D}_i = D \setminus D_i$.

Übung

Beschreiben Sie die Schritte zur Anpassung eines Polynoms an einen Datensatz mit Grid-Search und Cross-Validation.

- Wählen Sie eine Menge $C = \{d_1, \dots, d_m\} \subseteq \mathbb{N}$ von Kandidaten für die Grade der Polynome.
- Teilen Sie den Datensatz in k Teile D_1, \dots, D_k (Falten) auf
 - Für jeden Teil $i = 1, \dots, k$ und jedes $d \in C$, tun Sie folgendes.
 - Trainieren Sie ein Polynom $f_{i,d}$ des Grades d auf \bar{D}_i , wobei $\bar{D}_i = D \setminus D_i$.
 - Berechnen Sie den MSE $e_{i,d} = \text{MSE}(D_i, f_{i,d})$.

Übung

Beschreiben Sie die Schritte zur Anpassung eines Polynoms an einen Datensatz mit Grid-Search und Cross-Validation.

- Wählen Sie eine Menge $C = \{d_1, \dots, d_m\} \subseteq \mathbb{N}$ von Kandidaten für die Grade der Polynome.
- Teilen Sie den Datensatz in k Teile D_1, \dots, D_k (Falten) auf
 - Für jeden Teil $i = 1, \dots, k$ und jedes $d \in C$, tun Sie folgendes.
 - Trainieren Sie ein Polynom $f_{i,d}$ des Grades d auf \bar{D}_i , wobei $\bar{D}_i = D \setminus D_i$.
 - Berechnen Sie den MSE $e_{i,d} = \text{MSE}(D_i, f_{i,d})$.
- Geben Sie das Polynom f^* aus, das den kleinsten MSE über alle k Teile erreicht hat. Das heißt,

$$f^* = f_{d^*} \quad \text{wobei} \quad d^* = \arg \min_d \left(\sum_{i=1}^k e_{i,d} \right)$$

Implementieren Sie nun in Code Expert ein Python-Programm, das Grid-Search und Cross-Validation verwendet, um Polynome anzupassen.

3. Neuronale Netze

Wiederholung

Wiederholung

- Neuronale Netze gehören zu den leistungsfähigsten Maschinenlernmodellen.

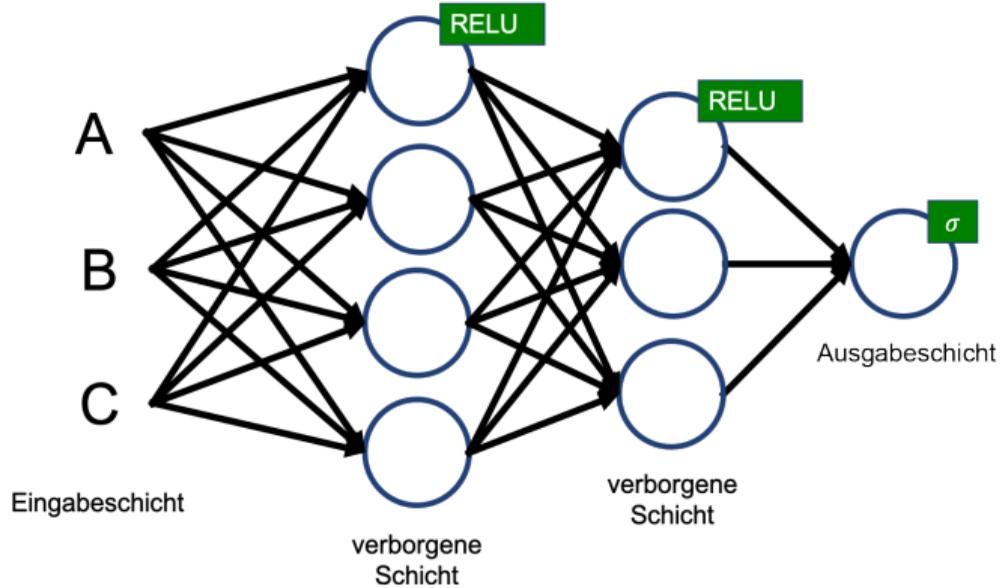
Wiederholung

- Neuronale Netze gehören zu den leistungsfähigsten Maschinenlernmodellen.
- Ihre Leistungsfähigkeit resultiert aus der Fähigkeit, automatisch Merkmale zu berechnen, die zur Lösung der Aufgabe relevant sind.

Wiederholung

- Neuronale Netze gehören zu den leistungsfähigsten Maschinenlernmodellen.
- Ihre Leistungsfähigkeit resultiert aus der Fähigkeit, automatisch Merkmale zu berechnen, die zur Lösung der Aufgabe relevant sind.
- Ein neuronales Netz ist das Ergebnis von mehreren Schichten von Arrays logistischer Regressoren.

Wiederholung



Zufällige Initialisierung

Die zufällige Initialisierung von Gewichten ist ein wesentlicher Schritt beim Training neuronaler Netze.

- Wenn alle Neuronen mit dem gleichen Anfangsgewicht beginnen, lernen sie die gleichen Merkmale und das Netzwerk kann keine komplexen Muster lernen.
- In Code Expert fixieren wir die Startwerte des Zufallszahlengenerators mit einem *random seed*, um stets die dieselben Ergebnisse zu erhalten (Reproduzierbarkeit).

Beantworten Sie die folgenden Fragen:

Beantworten Sie die folgenden Fragen:

- Warum übertreffen neuronale Netze andere Maschinenlernmodelle wie Entscheidungsbäume oder logistische Regression?

Beantworten Sie die folgenden Fragen:

- Warum übertreffen neuronale Netze andere Maschinenlernmodelle wie Entscheidungsbäume oder logistische Regression? Neuronale Netze können komplexere Flächen im euklidischen Raum beschreiben.

Beantworten Sie die folgenden Fragen:

- Warum übertreffen neuronale Netze andere Maschinenlernmodelle wie Entscheidungsbäume oder logistische Regression? Neuronale Netze können komplexere Flächen im euklidischen Raum beschreiben.
- Warum übertreffen tiefe neuronale Netze (Netze mit mehreren Schichten) flache neuronale Netze?

Beantworten Sie die folgenden Fragen:

- Warum übertreffen neuronale Netze andere Maschinenlernmodelle wie Entscheidungsbäume oder logistische Regression? Neuronale Netze können komplexere Flächen im euklidischen Raum beschreiben.
- Warum übertreffen tiefe neuronale Netze (Netze mit mehreren Schichten) flache neuronale Netze? Jede Schicht in einem neuronalen Netz lernt, Muster zu erkennen, die auf den von vorherigen Schichten erkannten Mustern basieren. Daher lernen tiefere Schichten, kompliziertere Muster zu erkennen als die ersten Schichten.

Übung

Beantworten Sie die folgenden Fragen:

Übung

Beantworten Sie die folgenden Fragen:

- Was ist der Unterschied zwischen ReLU und Sigmoid?

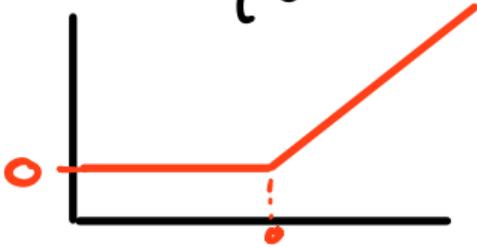
Übung

Beantworten Sie die folgenden Fragen:

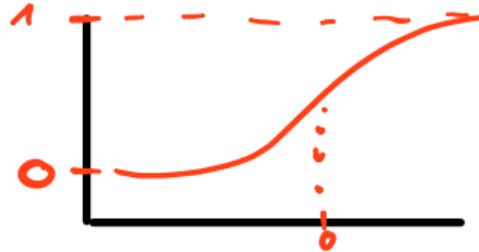


- Was ist der Unterschied zwischen ReLU und Sigmoid? Eine Sigmoidfunktion gibt einen Wert zwischen 0 und 1 aus und wird für Klassifizierungszwecke verwendet; das heißt, wenn wir zwischen zwei Klassen von Objekten unterscheiden wollen. Sigmoidfunktionen werden in der Regel für die letzte Schicht verwendet. ReLU gibt einen Wert zwischen 0 und ∞ aus und wird normalerweise für alle anderen Schichten verwendet.

$$\text{Relu} \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases}$$



$$\text{Sigmoid } \sigma(x) = \frac{1}{1+e^{-x}}$$



Handwritten notes in red ink:

- Limit as $x \rightarrow \infty$: $\frac{1}{1+e^{-x}} \rightarrow \frac{1}{1+0} = 1$
- Limit as $x \rightarrow -\infty$: $\frac{1}{1+e^{-x}} \rightarrow \frac{1}{1+\infty} = 0$

Beantworten Sie die folgenden Fragen:

- Was ist der Unterschied zwischen ReLU und Sigmoid? Eine Sigmoidfunktion gibt einen Wert zwischen 0 und 1 aus und wird für Klassifizierungszwecke verwendet; das heißt, wenn wir zwischen zwei Klassen von Objekten unterscheiden wollen. Sigmoidfunktionen werden in der Regel für die letzte Schicht verwendet. ReLU gibt einen Wert zwischen 0 und ∞ aus und wird normalerweise für alle anderen Schichten verwendet.
- Welche Python-Klasse repräsentiert Modelle für neuronale Netze in Scikit-learn?

Übung

Beantworten Sie die folgenden Fragen:

- Was ist der Unterschied zwischen ReLU und Sigmoid? Eine Sigmoidfunktion gibt einen Wert zwischen 0 und 1 aus und wird für Klassifizierungszwecke verwendet; das heißt, wenn wir zwischen zwei Klassen von Objekten unterscheiden wollen. Sigmoidfunktionen werden in der Regel für die letzte Schicht verwendet. ReLU gibt einen Wert zwischen 0 und ∞ aus und wird normalerweise für alle anderen Schichten verwendet.
- Welche Python-Klasse repräsentiert Modelle für neuronale Netze in Scikit-learn? `sklearn.neural_network.MLPClassifier`

Programmieraufgabe: Implementieren Sie ein Python-Skript, das ein neuronales Netz trainiert. Es muss lernen, gelbe von lila Punkten zu unterscheiden, wie in der Abbildung gezeigt.

4. Hausaufgaben

Exercise 11: Intro ML II

Auf <https://expert.ethz.ch/enrolled/SS25/mavt2/exercises>

- Spirals
- Diabetes with regression trees and grid search
- Confusion Matrix
- Sine Regression

Abgabedatum: Montag 26.05.2025, 20:00 MEZ

NO HARDCODING

Fragen oder Anregungen?