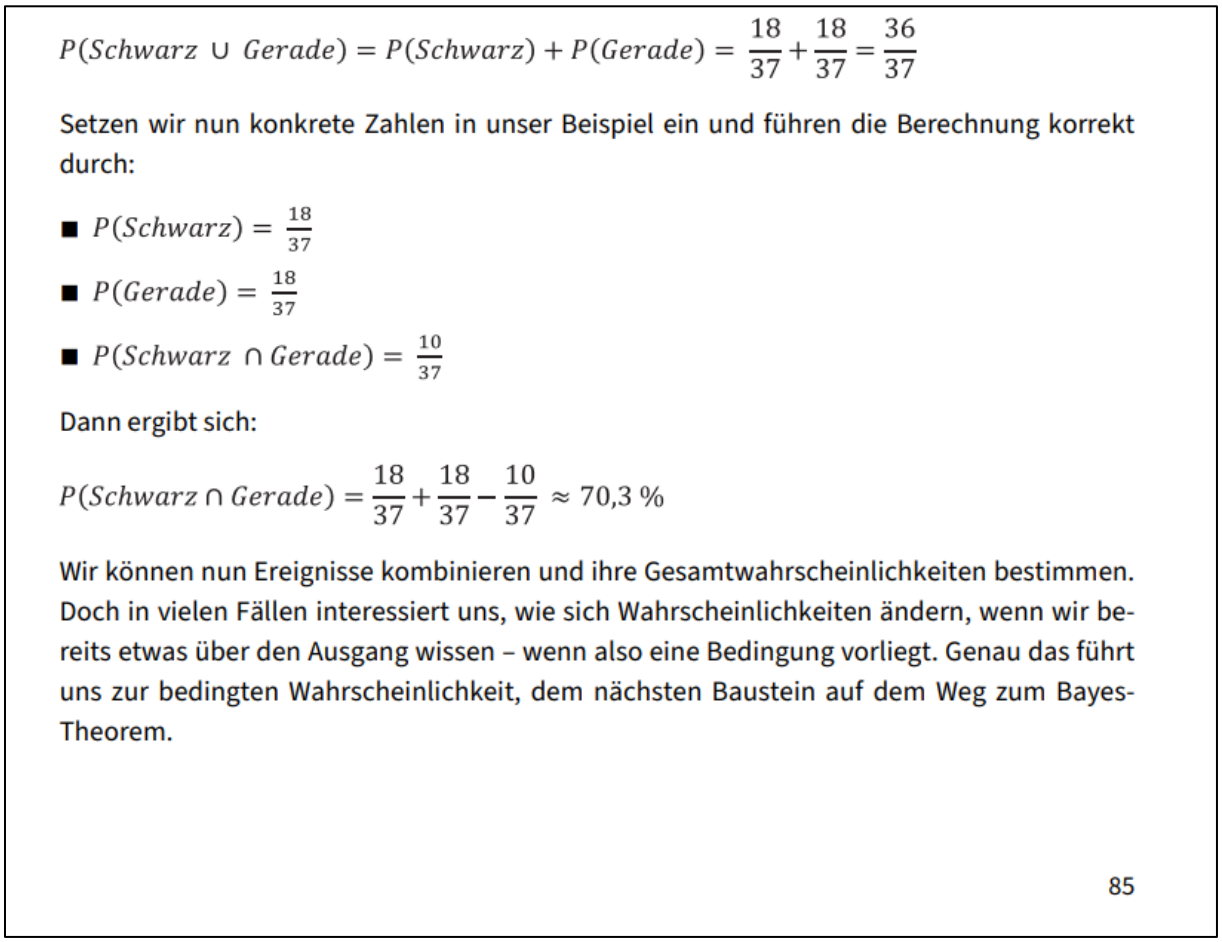
Errata

Im Folgenden finden Sie die Errata aus dem Buch „KI-Algorithmen in Python – Die Klassiker von k-Nearest Neighbours bis zum Reinforcement Learning“. Einem Screenshot von der betroffenen Seite folgt immer die Fehlerbeschreibung.

Seite 85



$$P(Schwarz \cup Gerade) = P(Schwarz) + P(Gerade) = \frac{18}{37} + \frac{18}{37} = \frac{36}{37}$$

Setzen wir nun konkrete Zahlen in unser Beispiel ein und führen die Berechnung korrekt durch:

- $P(Schwarz) = \frac{18}{37}$
- $P(Gerade) = \frac{18}{37}$
- $P(Schwarz \cap Gerade) = \frac{10}{37}$

Dann ergibt sich:

$$P(Schwarz \cap Gerade) = \frac{18}{37} + \frac{18}{37} - \frac{10}{37} \approx 70{,}3\ \%$$

Wir können nun Ereignisse kombinieren und ihre Gesamtwahrscheinlichkeiten bestimmen. Doch in vielen Fällen interessiert uns, wie sich Wahrscheinlichkeiten ändern, wenn wir bereits etwas über den Ausgang wissen – wenn also eine Bedingung vorliegt. Genau das führt uns zur bedingten Wahrscheinlichkeit, dem nächsten Baustein auf dem Weg zum Bayes-Theorem.

85

Fehler

In der letzten Formel über dem Fließtext befindet sich das falsche Formelzeichen. Es muss die Vereinigung sein:

$$P(Schwarz \cup Gerade) = \frac{18}{37} + \frac{18}{37} - \frac{10}{37} \approx 70{,}3\%$$

Seite 149

Ein R^2 von 1 entspricht einer perfekten Anpassung. Ein R^2 von 0 zeigt, dass das Modell nicht besser ist als der einfache Mittelwert. Negative Werte bedeuten, dass das Modell sogar schlechter abschneidet als dieses einfache Referenzmodell. Abbildung 5.5 veranschaulicht diesen Zusammenhang: Die blaue Linie zeigt die Regressionsgerade des Modells, die horizontale Linie den einfachen Mittelwert. Je näher die Regressionsgerade an den Datenpunkten liegt, desto höher das R^2. R^2 ergänzt die Fehlermaße um eine relative Perspektive, ersetzt sie jedoch nicht.

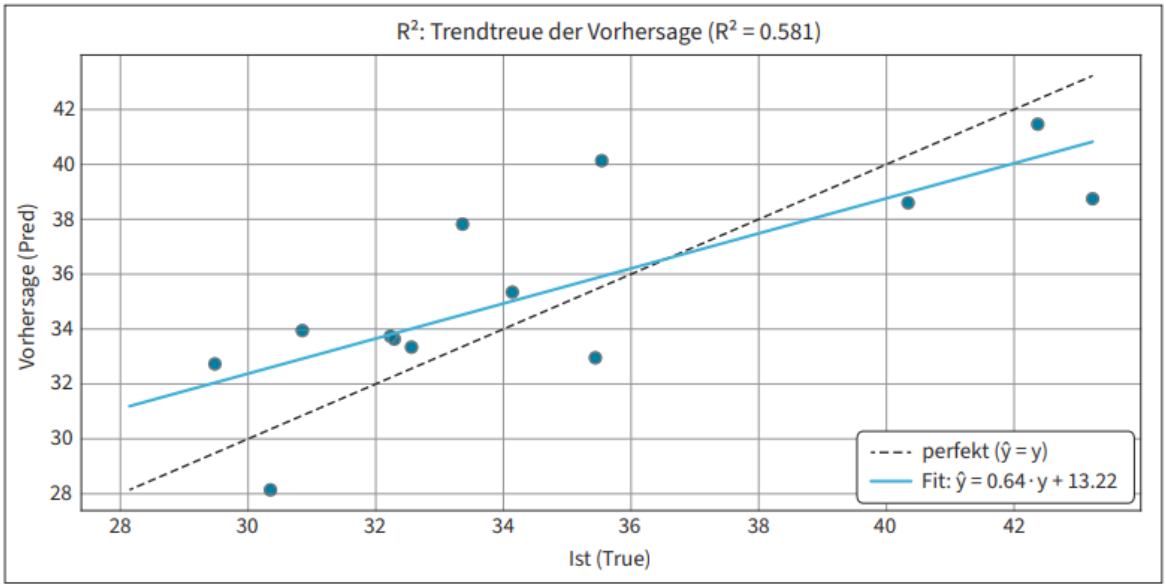


Abbildung 5.5: Darstellung des R^2-Werts

Fehler

Hier muss eine Textanpassung vorgenommen werden:

„*Abbildung 5.5* veranschaulicht diesen Zusammenhang: Die gestrichelte Linie markiert die perfekte Vorhersage (ŷ = y), die blaue Linie zeigt eine an die Punkte angepasste Gerade. Je näher die Datenpunkte an der gestrichelten Diagonalen liegen, desto höher ist R^2.“

Seite 249

8.4 Algorithmusbeschreibung

Der Ablauf der linearen Regression lässt sich in wenigen einfachen Schritten beschreiben. Egal, ob die Parameter später analytisch oder iterativ bestimmt werden, das Verfahren folgt stets demselben Grundprinzip.

1. **Modell festlegen**: Zunächst wird das Modell festgelegt. Bei der einfachen linearen Regression ist dies eine Gerade. Sie beschreibt, wie aus einem Eingabewert x eine Vorhersage $\hat{y}$ für den Zielwert berechnet wird.

   $\hat{y} = m \bullet x + b$

2. **Fehler bestimmen**: Für jeden Datenpunkt wird die Abweichung zwischen dem tatsächlichen Wert y_i und der Modellvorhersage $\hat{y}_i$ berechnet:

   $e_i = y_i \hat{y}_i$

3. **Fehler zusammenfassen und minimieren**: Um die Gesamtgüte des Modells zu bewerten, werden die quadrierten Residuen summiert:

   $$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

   Das Quadrieren verhindert ein Aufheben positiver und negativer Abweichungen und gewichtet große Fehler stärker. Diese Summe kennen Sie bereits aus Kapitel 5, »Evaluationsmetriken«, als Sum of Squared Errors (SSE). Ziel der linearen Regression ist es, die Parameter m und b so zu wählen, dass diese Summe minimal wird.

4. **Parameter bestimmen und Modell nutzen**: Durch Minimierung der Fehlerfunktion erhält man die optimalen Parameter für Steigung und Achsenabschnitt. Ist das Modell einmal

249

Fehler

Im zweiten Aufzählungspunkt „Fehler bestimmen:“ fehlt in der Formel ein Minus. Richtig ist:

$e_i = y_i - \hat{y}_i$

Seite 437

Hier sind zwei Ergebnisse des neuronalen Netzes, einmal in der Konfiguration mit drei Neuronen im ersten Hidden Layer [2, 3, 2, 3]:

```
Testpunkte:
0.90, 0.85 -> erwartet: bestanden        | Netz: nicht bestanden  |
probs=['0.27', '0.22', '0.51']
0.70, 0.50 -> erwartet: Nachprüfung      | Netz: nicht bestanden  |
probs=['0.27', '0.22', '0.51']
0.45, 0.45 -> erwartet: nicht bestanden  | Netz: nicht bestanden  |
probs=['0.21', '0.22', '0.58']
```

Und einmal in der Konfiguration mit sechs Neuronen im ersten Hidden Layer [2, 6, 3, 2]:

```
Testpunkte:
0.90, 0.85 -> erwartet: bestanden        | Netz: bestanden        |
probs=['0.96', '0.04', '0.00']
0.70, 0.50 -> erwartet: Nachprüfung      | Netz: Nachprüfung      |
probs=['0.00', '0.64', '0.36']
0.45, 0.45 -> erwartet: nicht bestanden  | Netz: nicht bestanden  |
probs=['0.00', '0.00', '1.00']
```

13.4.5 Mechanistic Interpretability

Die Frage, was ein neuronales Netz tatsächlich »gelernt« hat, ist Gegenstand eines aktiven Forschungsgebiets: *Mechanistic Interpretability*. Bei kleinen Netzen wie dem hier vorgestellten kann man die Gewichte noch manuell untersuchen und vorsichtige Schlüsse ziehen. Bei

Fehler

Hier hat sich in der Mitte der Seite ein Zahlendreher eingeschlichen:

„Und einmal in der Konfiguration mit sechs Neuronen im ersten Hidden Layer [2, 6, 2, 3]“

Seite 456

Daraus ergibt sich:

$Q(Schreibtisch,Arbeiten) = 5 + \gamma \bullet V(Schreibtisch)$

Wählt der Agent statt der Arbeit am Schreibtisch das Meeting, ergibt sich diese Q-Funktion:

$Q(Schreibtisch,Meeting) = 1 + \gamma \bullet V(Schreibtisch)$

Der Wert des Meeting-Zustands ist insgesamt niedriger als der des Schreibtischs, daher wird auch der Aktionswert entsprechend kleiner. Halten wir fest: Eine Entscheidung ist gut, wenn sie in einen guten Zustand führt.

Der Agent lernt aus seiner Erfahrung. Das wird dadurch realisiert, dass der Wert eines Zustands vom Wert seiner Folgezustände abhängt. Die Bellman-Idee ist also rekursiv. Um den Wert eines Zustands zu bestimmen, nutzen wir die folgende formale Beschreibung für Zustände:

Fehler

Hier wurde ein Begriff in Klammern vertauscht. Richtig ist für die zweite Formel auf der Seite:

„Wählt der Agent statt der Arbeit am Schreibtisch das Meeting, ergibt sich diese Q-Funktion:
$Q(Schreibtisch,Meeting) = 1 + \gamma \cdot V(Meeting)$“

Seiten 472 bis 474

Abschnitt 14.8.4, *Der Agent* – Listing 14.1, *Der Tic-Tac-Toe-Agent*

Was ist passiert?

Im abgedruckten Quelltext der Klasse *Agent* passt ein Vorzeichen nicht. Der Agent lernt dadurch nicht richtig und spielt auch nach dem Training nicht besser, als es der Zufall erwarten lässt. Außerdem stimmte das Listing im Buch nicht mit dem Quelltext im Download-Bereich überein.

So gehen Sie vor

Der Quelltext im Download-Bereich ist korrigiert. Laden Sie bitte die aktuelle Version herunter und verwenden Sie diese anstelle des abgedruckten Listings. Wenn Sie das Listing bereits abgetippt oder eine ältere Version heruntergeladen haben, ersetzen Sie die Datei durch die neue. Der Text des Kapitels bleibt davon unberührt, das Verfahren und die Erklärungen gelten weiter.