

Woche 07: Wie war das nochmal? – Überblick und Ausblick

Skript

Erarbeitet von
Dr. Maike Mayer

Lernziele	1
Inhalt	1
Einstieg	1
Die Nachbarschaft ist entscheidend	2
Wann ist es zu viel?	3
Wichtig: Wissen, was die KI tut.....	3
Auf geht's!	4
Quellen	4
Disclaimer	4

Lernziele

- Erinnern der Ziele der Regressionen
- Erinnern der Definition von Accuracy
- Nachvollziehen der Zusammenhänge zwischen den Inhalten

Inhalt

Einstieg

In der letzten Woche hast du die logistische und die lineare Regression als Beispiele für statistische Lernmethoden im Bereich des Maschinellen Lernens kennengelernt. Mit Hilfe einer Regression kann man Zusammenhänge zwischen Zielmerkmalen und Features bestimmen und dann bei neuen Beobachtungen Vorhersagen für das Zielmerkmal treffen.

Das hat nichts mit Wahrsagerei oder mit Kristallkugeln zu tun, sondern mit Statistik! Und wir bleiben auch in dieser Woche bei statistischen Lernmethoden. Ich hatte ja schon angekündigt, dass dir einige begegnen werden. Diese Woche geht es um den sogenannten k-nearest neighbours Algorithmus oder auch k-nächste Nachbarn Algorithmus. Mit diesem Algorithmus lassen sich Beobachtungen klassifizieren und dafür kommt es auf die Nachbarn dieser Beobachtung in einer Datenmenge an ...

Einblendung Illustrationen

Die Nachbarschaft ist entscheidend

Der k-nearest neighbours Algorithmus wird häufig für Klassifikationen verwendet. Eine neue Beobachtung wird anhand ihrer Features oder Eigenschaften einer Kategorie – auch Klasse genannt – zugeordnet. Es geht hier also wieder um kategoriale Daten, die in der vergangenen Woche bereits bei der logistischen Regression (mit der S-Kurve) relevant waren. Kurze Wiederholung: Kategoriale Daten sind beispielsweise Studienfächer, Wohnorte oder Vereinsmitgliedschaften.

Einblendung Schlagwort/Illustrationen

Ok, es geht also um die Zuordnung von neuen Beobachtungen zu einer bestimmten Klasse. Aber wie funktioniert das Ganze? Im Prinzip ist es eine Frage der Nachbarschaft. Der KNN-Algorithmus geht davon aus, dass ähnliche Dinge oder Personen auch ähnliche Eigenschaften haben. Um diese Ähnlichkeit zu bestimmen, wird die Nähe zwischen den Datenpunkten herangezogen. Der Algorithmus schaut sich eine bestimmte Anzahl an Nachbarn (die wir mit „k“ vorgeben) an, die in der Datenmenge am nächsten an unserer neuen Beobachtung liegen, und ordnet die neue Beobachtung dann der Klasse zu, zu der die meisten der einbezogenen Nachbarn gehören. Dieser Ansatz wird auch als „lazy“ oder träge bezeichnet. Neue Beobachtungen werden einfach mit den Trainingsdaten verglichen und dann eingeordnet. Es ist also kein Training erforderlich.

Einblendung Illustrationen/Schlagwörter

Auch wenn kein Training erforderlich ist, programmieren muss man den Algorithmus trotzdem. Daher zeigen wir dir in Python am Beispiel von Pinguinen direkt auch, wie du einen KNN-Algorithmus umsetzen kannst. Dafür brauchst du nochmal das Modul Scikit-learn aus der letzten Woche und auch Numpy Arrays werden wieder relevant. Und dir begegnet auch ein Qualitätsmaß aus Woche 5 wieder: die Genauigkeit oder auch Accuracy. Mit Hilfe dieses Maßes kannst du überprüfen, wie gut dein KNN-Algorithmus arbeitet. Was Accuracy nochmal genau war? Die Accuracy oder Genauigkeit beschreibt das Verhältnis der korrekten Klassifikationen zu den insgesamt vorgenommenen Klassifikationen. Es geht darum, wie viele der gemachten Klassifikationen korrekt waren. Wenn also beispielsweise 19 von 20 Hunde- und Katzenbildern korrekt als Hund bzw. Katze identifiziert wurden, liegt die

Accuracy des Systems bei 95 %. Und diesen Accuracy-Wert kannst du dir auch in Python ausgeben lassen.

Aber nochmal zurück zu unseren statistischen Lernmethoden. Sowohl Regressionen als auch KNN-Algorithmen benötigen Features oder Eigenschaften von Beobachtungen. Das klingt jetzt erstmal so, als wäre es immer gut, möglichst viele Features und damit auch möglichst viele Informationen zur Verfügung zu haben ... Aber gibt es auch ein „zu viel“ bei Features?

Wann ist es zu viel?

In der letzten Woche hast du bereits gelernt, was passiert, wenn man die Komplexität des Zusammenhangs zwischen Zielmerkmal und Feature über- bzw. unterschätzt. Aber auch zu viele Features können ein Problem beim Training unserer Modelle werden. Denn je nachdem, mit welcher Funktion ein Modell arbeitet, werden alle Features als gleich wichtig betrachtet. Haben wir aber eigentlich unwichtige Features dabei, wird die Vorhersage oder die Klassifikation des Modells dadurch schlechter.

Ein weiteres Problem: Mit jedem Feature werden auch mehr Daten benötigt, um das Modell zu trainieren. Das kann sehr schnell ziemliche Ausmaße annehmen und wird auch gerne als Fluch der Dimensionalität bezeichnet ... Und auch das wollen wir uns in dieser Woche näher anschauen, denn darauf sollte man ebenfalls bei der Wahl seiner Trainingsdaten achten.

Einblendung Illustrationen/Schlagwörter

Über Trainingsdaten hast du ja schon viel gehört und du lernst in dieser Woche bereits die zweite statistische Lernmethode des Maschinellen Lernens kennen. Das ist schon einiges und du weißt auch bereits, dass es ethisch einiges zu beachten gibt. Um KI-Systeme einschätzen zu können, ist es also offensichtlich nicht verkehrt, zu wissen, was KI eigentlich tut ...

Wichtig: Wissen, was die KI tut

Zu wissen bzw. nachvollziehen zu können, was ein KI-basiertes System tut, ist aus vielen Gründen wichtig. Es hilft beispielsweise dabei, Akteurinnen und Akteure für negative Konsequenzen der KI-Anwendung verantwortlich zu machen (Stichwort: Rechenschaftspflicht), weil nachvollzogen werden kann, woran ein bestimmtes Fehlverhalten liegt. Es trägt aber auch zu der Akzeptanz von KI-basierten Systemen bei, wenn den Menschen klar ist, wie diese Systeme funktionieren bzw. arbeiten.

Daher ist es vielleicht nicht überraschend, dass Transparenz einer der Grundsätze ist, der in ethischen Leitlinien zum Thema Künstliche Intelligenz oft genannt wird. Transparenz bezieht sich auf das Ausmaß, in dem die innere Funktionsweise eines KI-Systems für den Menschen offen und erklärbar – also nachvollziehbar – ist. Kann man die Funktionsweise eines Systems nicht nachvollziehen, spricht man übrigens von sogenannten Blackbox-KI-Systemen.

Quelle [1, 2]

Auf geht's!

In dieser Woche lernst du also den k-nearest neighbours Algorithmus kennen und wir zeigen dir auch direkt, wie du ihn in Python umsetzen kannst. Wir werden außerdem über Transparenz bei Systemen sprechen, die auf Künstlicher Intelligenz basieren. Zusätzlich wird es um den Fluch der Dimensionalität gehen und du erfährst, was man dagegen tun kann. Und abschließend haben wir dann noch ein paar Anwendungsbeispiele für den KNN-Algorithmus für dich zusammengestellt.

Einblendung der Videotitel

Und wer weiß, vielleicht siehst du deine Nachbarschaft demnächst mit ganz anderen Augen. Aber jetzt erstmal viel Spaß in dieser Woche!

Quellen

- Quelle [1] Jobin, A., Ienca, M., & Vayena, E. (2019). The global landscape of AI ethics guidelines. *Nature Machine Intelligence*, 1(9), 389–399. <https://doi.org/10.1038/s42256-019-0088-2>
- Quelle [2] Diakopoulos, N. (2020). Transparency. In *The Oxford handbook of ethics of AI* (pp. 197–213). Oxford University Press.

Disclaimer

Transkript zu dem Video „Woche 07: Wie war das nochmal? – Überblick und Ausblick“, Dr. Maïke Mayer.

Dieses Transkript wurde im Rahmen des Projekts ai4all des Heine Center for Artificial Intelligence and Data Science (HeiCAD) an der Heinrich-Heine-Universität Düsseldorf unter der Creative Commons Lizenz [CC-BY 4.0](https://creativecommons.org/licenses/by/4.0/) veröffentlicht. Ausgenommen von der Lizenz sind die verwendeten Logos, alle in den Quellen ausgewiesenen Fremdmaterialien sowie alle als Quellen gekennzeichneten Elemente.