**Training Fiche Vorlage**

|  |  |  |
| --- | --- | --- |
| **Titel** | Einführung in maschinelles Lernen | |
| **Schlüsselwörter (meta tag)** | Maschinelles Lernen, supervised learning , Klassifizierung, Regression, AI, künstliche Intelligenz, Naive Bayes, Entscheidungsbäume, Random Forest, neuronale Netze, Deep Learning | |
| **Sprache** | Deutsch | |
| **Zielsetzungen / Ziele / Lernergebnisse** | * **Erfahre mehr über KI, maschinelles Lernen, Deep Learning und wie sie sich auf Data Science beziehen** * **Lerne verschiedene Algorithmen kennen, die beim maschinellen Lernen verwendet werden, z. B:** * **Naive Bays** * **Entscheidungsbäume** * **Random Forests** * **Neuronale Netze** * **Kurzer Überblick darüber, wie die Performance von Algorithmen des maschinellen Lernens bewertet werden kann** | |
| **Lehrgang:** | | |
| **Datenwissenschaftliche Kompetenz** | |  |
| **Modul Datenvisualisierung und visuelle Analyse** | |  |
| **Einführung in die Datenwissenschaft für Human- und Sozialwissenschaften** | |  |
| **Software** | |  |
| **Maschinelles Lernen** | | **X** |
| **Datenwissenschaft für den guten Zweck** | |  |
| **Datenjournalismus und Geschichtenerzählen** | |  |
| **Beschreibung** | Dieses Skript enthält Definitionen der grundlegenden Konzepte des maschinellen Lernens sowie Beschreibungen der wichtigsten verwendeten Methoden, einschließlich einiger spezifischer Beispiele und Anwendungen. Du kannst das Skript entweder oberflächlich lesen, um ein grundlegendes Verständnis des Fachgebiets zu erlangen, oder die ausführlicheren Beschreibungen lesen, insbesondere den Abschnitt über die Methoden, um ein Verständnis des maschinellen Lernens auf mittlerem Niveau zu erlangen.  Statistik und maschinelles Lernen sind die wichtigsten Werkzeuge für deine Arbeit als Datenwissenschaftler:in. Wenn du die verschiedenen Methoden des maschinellen Lernens verstehst - wie sie funktionieren, was ihre Hauptvorteile sind und wie ihre Performance bei einer bestimmten Aufgabe zu bewerten ist – kannst du bessere Entscheidungen darüber treffen, wann du sie einsetzen solltest, und werde zu einer:einem vielseitigeren Data-Science-Expert:in. | |
| **Inhalt in 3 Ebenen gegliedert** | 1. Einführung in maschinelles Lernen   Data Science ist eine empirische Disziplin, die Daten mit verschiedenen Methoden, hauptsächlich aus der Statistik und dem maschinellen Lernen, kombiniert, um Probleme zu lösen und fundierte Entscheidungen zu ermöglichen. Statistik wurde in einem separaten Kurs behandelt, daher werden wir uns hier auf den Bereich des maschinellen Lernens (ML) konzentrieren.  1.1 Begriffsbestimmungen [BASIC]  Es gibt viele Schlagworte, die mit ML in Verbindung gebracht werden - die beiden bekanntesten sind Künstliche Intelligenz (KI) und Deep Learning (DL). KI ist das Studiengebiet, das sich mit Algorithmen befasst, die Aufgaben ausführen können, die normalerweise mit menschlicher "Intelligenz" verbunden sind. Dazu gehören z. B. Algorithmen, die Bilder erkennen oder Texte zu "verstehen" scheinen (ja, wie ChatGPT), die sich selbstständig fortbewegen können (Roboter oder selbstfahrende Autos) oder komplexe Entscheidungen treffen (z. B. wem man einen Kredit gibt oder welche Bewerber man einstellt).  Wenn die Methode zur Bewältigung dieser Aufgaben darin besteht, der Maschine Schritt-für-Schritt-Anweisungen zu geben, dann wird dies oft als "symbolische KI" oder "heuristische KI" bezeichnet. Tatsächlich gibt es KI schon seit den 1950er Jahren. Bis die Computertechnologie leistungsfähiger und die Datenmenge größer wurden (vor etwa 15-20 Jahren), waren die meisten KI-Systeme tatsächlich symbolische KI. Die Zunahme der verfügbaren Daten und der Rechenleistung hat dazu geführt, dass ein zweiter Zweig der KI immer beliebter und leistungsfähiger geworden ist: ML - "Lernen" nach Beispielen. ML ist im Grunde ein Feld, das Algorithmen untersucht und entwickelt, welche zur Erkennung von Mustern in Daten verwendet werden können. Bei ML erhält die Maschine die Anweisungen, "wie ein Muster zu finden ist", gemeinsam mit vielen Beispielen; aus diesen Beispielen erkennt der Algorithmus ein Muster und verwendet dieses Muster, um "neue" Probleme zu lösen.  Deep Learning ist ein Teilbereich von ML. DL ist eine Sammlung von Methoden, die auf neuronalen Netzen basieren, auf die wir später noch näher eingehen werden.  1.2 Arten des maschinellen Lernens  Algorithmen für Maschinelles Lernen können in drei Klassen von Algorithmen unterteilt werden: supervised Learning, unsupervised Learning und reinforcement Learning.  In der folgenden Abbildung werden die verschiedenen Arten des maschinellen Lernens dargestellt und einige Beispiele für Anwendungsszenarien oder Anwendungsfälle für jede Art von Maschinellem Lernen gegeben.    Abbildung 1: Arten von ML-Algorithmen  **Supervised Learning**  Alle Algorithmen, die die Methode supervised Learning verwenden, benötigen gelabelte Daten für Training, Validierung und Test.  Gelabelte Datensätze sind Datensätze, die Merkmalsvariablen (auch als unabhängige Variablen oder Attribute bezeichnet) und eine Zielvariable (auch als abhängige Variable bezeichnet) enthalten. In einem Algorithmus zur Erkennung von Kreditrisiken könnte ein markierter Datensatz beispielsweise Elemente wie Alter, Geschlecht, Kontostand, Kreditwürdigkeit und beantragter Kreditbetrag als Attribute und eine Zielvariable enthalten, z. B. ob diese Person ihren Kredit zurückgezahlt hat oder nicht. Andere Beispiele wären ein Datensatz mit Bildern von Haustieren, welche mit Etiketten versehen sind, die das Tier bezeichnen, oder ein Datensatz mit Merkmalen wie dem täglichen Aktienwert eines Unternehmens in den letzten sechs Monaten, dem Jahresdurchschnitt der letzten fünf Jahre und der Anzahl der Mitarbeiter:innen, wobei die Zielvariable der Aktienwert des Unternehmens am nächsten Tag wäre.  Je nach Art der Zielvariablen kann der supervised Learning-Algorithmus als Klassifizierung oder Regression bezeichnet werden. Wenn die Zielvariable aus einer endlichen Anzahl von Kategorien besteht, wird der Algorithmus im Allgemeinen als Klassifizierungsalgorithmus bezeichnet. Handelt es sich bei der Zielvariable hingegen um eine quantitative (oder numerische) Variable, so gehört der Algorithmus zur Klasse der Regressionsalgorithmen.  **Unsupervised Learning**  Unsupervised Learning wird verwendet, um Muster in ungelabelten Daten zu erkennen. Einige der beliebtesten Arten des unsupervised Learnings sind:   * Clustering: Identifizierung ähnlicher Gruppen in den Daten, ohne im Voraus zu wissen, nach welchen Gruppen zu suchen ist; * Erkennung von Anomalien: Bestimmen, welche Instanzen sich "sehr stark" von den übrigen Beispielen im Datensatz unterscheiden; * Dimensionsreduktion: Verringerung der Dimension des Merkmalsraums - dazu gehören Methoden wie die Hauptkomponentenanalyse oder LDA.   **Reinforcement Learing (RL)**  Reinforcement Learing wird eingesetzt, um eine optimale Strategie in Situationen abzuleiten, in denen der algorithmische Agent mit einer gegebenen Umgebung interagieren und eine Reihe von Entscheidungen treffen muss, bevor das Endergebnis bekannt ist (d. h. das Feedback ist nicht unmittelbar: Erfolg oder Misserfolg, Sieg oder Niederlage). RL-Methoden werden am häufigsten bei Spielen, beim autonomen Fahren und bei der Robotermobilität eingesetzt.  Manchmal wird eine vierte Klasse von Algorithmen in Betracht gezogen: das half-supervised Learning. Dabei handelt es sich um eine Mischung aus supervised und unsupervised Learning, die sich aufgrund der Kosten für die Beschaffung etikettierter Daten zunehmender Beliebtheit erfreut.  Oft hilft dir die Art des vorliegenden Problems und die Art der verfügbaren Daten bei der Entscheidung, welche Art von Algorithmen für maschinelles Lernen du verwenden kannst. Versuchst du Gruppen von Datenpunkten mit einer gewissen Ähnlichkeit zu identifizieren, ohne eine klare Vorstellung davon zu haben, wie diese Gruppen aussehen sollten? Dann benötigst du unsupervised Learning. Geht es bei deinem Problem um die Entwicklung einer optimalen Strategie in einer Situation, in der die Rückmeldung (Erfolg/Misserfolg) nicht unmittelbar erfolgt? Dann bist du auf der Suche nach einer Lösung mit reinforcement Learning. Oder hast dueinen festen Satz von Kategorien und möchtest neue Datenpunkte automatisch diesen vorgegebenen Klassen zuordnen? Dann ist das supervised Learining.  Die genaue Festlegung, welche Methode des supervised/unsupervised/reinforcement Learnings zu wählen ist, ist jedoch eine viel schwierigere Angelegenheit. ML ist eine empirische Wissenschaft, und du musst in der Regel mehrere verschiedene Algorithmen ausprobieren und deren Performance vergleichen, um "die beste Option" zu ermitteln.  Aus diesem Grund werden im nächsten Abschnitt verschiedene ML-Techniken mit ihren Schwächen und Stärken beschrieben, und im letzten Abschnitt wird untersucht, wie ihre Performance zu bewerten ist.   1. Überblick über ML-Algorithmen   Dieser Abschnitt gibt einen Überblick über verschiedene Algorithmen, die im Bereich ML verwendet werden. Die Algorithmen sind unterschiedlich komplex und reichen von einfachen Algorithmen wie Entscheidungsbäumen bis hin zu komplexeren Algorithmen wie Random Forests.  Dieser Abschnitt ist keineswegs erschöpfend, sondern soll dir einen Eindruck von der Tiefe und Vielfalt der im Bereich des maschinellen Lernens verfügbaren Techniken vermitteln.   * 1. Grundlagen der Statistik [BASIC]   Die lineare Regression ist ein Algorithmus, der für Regressionsprobleme des überwachten Lernens verwendet wird. Die logistische Regression basiert auf den Konzepten der linearen Regression, wird aber trotz des Wortes "Regression" im Namen eigentlich für Klassifizierungsprobleme verwendet.  Wenn du dir einige Konzepte und Algorithmen der ML genauer ansiehst, wirst du feststellen, dass sie oft auf Varianten der linearen oder logistischen Regression hinauslaufen. Zum Beispiel war ein Neuron in einem neuronalen Netz oft eine einfache logistische Regression (oder etwas noch Einfacheres, wie eine piecewise line!)  Obwohl sie auch Teil des ML-Toolkits sind, wurden die lineare und die logistische Regression ausführlich in den Modulen zu Statistik erklärt und werden hier nicht weiter beschrieben. Siehe das STATS-Skriptum.   * 1. Naïve Bayes-Klassifikator [BASIC]   Naïve Bayes ist ein einfacher Klassifizierungsalgorithmus, der häufig als Baseline (zum Vergleich mit anderen, komplexeren Algorithmen) z. B. bei der Verarbeitung natürlicher Sprache verwendet wird.  Naïve Bayes verwendet das Bayes-Theorem, um das Problem der Bestimmung der Wahrscheinlichkeit, dass eine Instanz angesichts ihrer Attribute X = [x1 , ..., xN ] zur Klasse Y gehört, in das einfachere Problem der Bewertung der Häufigkeit des Attributs xi , wenn die Instanz zur Klasse Y gehört, umzuwandeln.  Das Bayes-Theorem ist eine einfache mathematische Formel zur Berechnung bedingter Wahrscheinlichkeiten. Das Theorem besagt, dass:  P ( Y|X) = , wobei  P (Y) ist die Wahrscheinlichkeit, dass das Ereignis Y eintritt,  P) ist die Wahrscheinlichkeit, dass beide Ereignisse eintreten,  P (Y|X) ist die Wahrscheinlichkeit, dass Y eintritt, wenn X eintritt (die bedingte Wahrscheinlichkeit von Y bei X).  Das Bayes-Theorem kann auch folgendermaßen formuliert werden  P) = P (X|Y)x P (Y) = P (Y|X)x P (X), und so kann das Problem der Bestimmung von P (Y|X) in das Problem der Bestimmung von P (X|Y) umgewandelt werden.  Warum ist dies nützlich? Weil die relativen Häufigkeiten von X und Y in den Trainingsdaten verwendet werden können, um P (X|Y) zu bestimmen.  Sie kann gute Ergebnisse liefern, wenn   * alle Attribute sind mehr oder weniger gleich wichtig für die Bestimmung der Zielklasse; * bei einer festen Zielklasse sind die Attribute voneinander unabhängig (kannst du dir vorstellen, warum diese Annahme wichtig ist?)   Naïve Bayes gibt es in verschiedenen Varianten:   * Gaußscher NB: wird verwendet, wenn die Attributvariablen numerisch sind und angenommen werden kann, dass sie einer Gaußschen Verteilung folgen * Einfacher NB: wird verwendet, wenn die Attributvariablen kategorisch sind * Multinomialer NB: wird am häufigsten im Zusammenhang mit der Verarbeitung natürlicher Sprache verwendet, wobei die Attribute Wörter in einem Dokument sind.   1. Entscheidungsbäume [INTERMEDIATE]   Ein Entscheidungsbaum ist ein supervised Learning-Algorithmus, der zur Klassifizierung und Regressionsmodellierung verwendet werden kann. Entscheidungsbäume sind sowohl eine Methode zur Darstellung von Informationen als auch ein Algorithmus zur Erkennung von Mustern in Daten. Ein Entscheidungsbaum-Algorithmus gibt nämlich die Informationen, die er aus den Trainingsdaten "gelernt" hat, in Form eines Entscheidungsbaums aus.  Wie sieht also ein Entscheidungsbaum aus?   * Entscheidungsbäume bestehen aus Knoten (nodes) und Zweigen, wobei ein Knoten ganz oben steht * Jeder Knoten "stellt eine Frage" in Bezug auf die Datenattribute und hat Verzweigungen in Abhängigkeit von den möglichen Antworten. Wenn ein Attribut zum Beispiel "Collegejahr" lautet und die möglichen Attributwerte sind (Freshman, Sophomore, Junior, Senior), dann könnte der Knoten, der der Frage "Welches Jahr im College?" entspricht, 4 Zweige haben. In einem binären Entscheidungsbaum hätte ein Knoten immer genau zwei Verzweigungen - so könnte sich der Knoten "Collegejahr = Junior?" zunächst in "Ja" und "Nein" verzweigen, und der "Nein"-Zweig könnte dann einen weiteren Knoten "Collegejahr = Freshman?" haben, der sich in "Ja" und "Nein" verzweigt, usw. * Entscheidungsbäume werden vom obersten Knoten abwärts durchlaufen: An jedem Knoten muss eine Entscheidung darüber getroffen werden, welchem Zweig als Nächstes gefolgt werden soll, und zwar auf der Grundlage des Werts/der Werte eines bestimmten Attributs/mehrerer Attribute * Dies geht weiter, bis die Endknoten (oder "Blattknoten") erreicht sind. Diese Knoten haben keine weiteren Verzweigungen und stellen die Schlussfolgerung bzw. Vorhersage dar.     Abbildung 2: Klassifikationsbäume  Ein Baum, dessen Blätter Klassen oder Kategorien sind, wird als Klassifikationsbaum bezeichnet. Handelt es sich bei den Blättern um Funktionen (meist numerische Konstanten oder auch Linien), so handelt es sich um einen Regressionsbaum.  Entscheidungsbaum-Algorithmen werden mit Methoden aus der Informationstheorie konstruiert und versuchen, bei jedem Schritt einen Baum nach dem Prinzip des "größten Informationsgewinns" zu konstruieren. Die Anzahl der Verzweigungen und die Tiefe des Baums sind in der Regel Entscheidungen, die der Datenwissenschaftler treffen muss - oft ist ein wenig Experimentieren mit verschiedenen Werten erforderlich.  Es ist auch gut zu bedenken, dass Bäume mit einer größeren Anzahl von Zweigen und einer größeren Tiefe mehr Flexibilität bieten, aber dies muss sorgfältig gegen die erhöhte Wahrscheinlichkeit einer Überanpassung und die Tatsache abgewogen werden, dass Bäume mit weniger Zweigen und einer geringeren Tiefe deutlich verständlicher sind.   * 1. Random Forests [INTERMEDIATE]   Ein Random Forest ist eine Sammlung von vielen Entscheidungsbäumen, die als [Ensemble](https://en.wikipedia.org/wiki/Ensemble_learning) arbeiten. Random Forests sind eine besondere Art des "Ensemble-Lernens" - eine Klasse von Methoden, welche (in der Regel einfache) Modelle kombinieren, um die Vorhersagegenauigkeit durch Vielfalt zu verbessern.  Random Forests bestehen aus mehreren zufällig ausgewählten Entscheidungsbäumen und kombinieren deren Vorhersagen. Sie variieren in der Anzahl der Bäume, die sie enthalten, und in der Tiefe der einzelnen Bäume.  Random Forests werden oft als eine Kombination aus der Erklärbarkeit von Entscheidungsbäumen und der Leistungssfähigkeit und höheren Genauigkeit komplexerer Methoden angesehen. Random Forests und andere baumbasierte Ensemble-Methoden wie Gradient Boosting sind nach wie vor recht beliebt und können hochmoderne Ergebnisse erzielen (ja, es muss nicht immer ein neuronales Netz sein).   * 1. Hierarchisches Clustering [BASIC]   Clustering ist eine breite Palette von Techniken des supervised Learnings. Das Ziel besteht darin, Strukturen und Ähnlichkeiten in den Daten zu erkennen: eine Gruppierung der Beispiele im Datensatz zu finden, so dass die Beispiele in einer Gruppe einander irgendwie ähnlich sind und sich von den Beispielen in anderen Gruppen unterscheiden. Eine beliebte Anwendung wäre die Erstellung von Profilen über Verbraucher:innen – d.h. die Identifizierung von "Verbrauchertypen" – damit Werbung gezielter eingesetzt werden kann.  Hierarchisches Clustering und K-means Clustering sind zwei der bekanntesten Clustering-Techniken. Hierarchisches Clustering erzeugt eine baumartige Struktur (in diesem Fall üblicherweise als Dendrogramm bezeichnet), die an einem obersten Knoten beginnt, der den gesamten Datensatz enthält, und sich an jedem Knoten rekursiv in kleinere Dendrogramme verzweigt, wobei "ähnliche" Elemente in denselben Zweig gelangen. Diese Art des Clusterns bietet verschiedene Granularitätsebenen: An der Spitze des Dendrogramms haben wir ein breiteres Konzept von "ähnlich", und je weiter wir nach unten gehen, desto feiner sind die Unterschiede zwischen den Zweigen.   * 1. K-Means-Clustering [BASIC]   Während beim hierarchischen Clustering keine Informationen über die Anzahl der Gruppen oder Cluster erforderlich sind, in die die Daten aufgeteilt werden sollen, ist dies beim K-Means-Clustering der Fall. Beim K-Means-Clustering wird der Datensatz tatsächlich in K verschiedene Gruppen aufgeteilt.  Es ist oft nicht von vornherein klar, in wie viele Gruppen ein Datensatz unterteilt werden muss. Aus diesem Grund besteht ein Teil deiner Aufgabe als Datenwissenschaftler:in darin, mit einigen verschiedenen Werten von K zu experimentieren, um den "besten" zu finden.  Der K-Means-Algorithmus geht davon aus, dass jede Instanz im Datensatz ein Punkt in einem Vektorraum mit einer bestimmten Abstandsfunktion (normalerweise euklidisch) ist. Zunächst wird jede Instanz im Datensatz nach dem Zufallsprinzip genau einem von K Clustern zugewiesen, und dann wird für jedes Cluster ein Schwerpunkt oder Mittelwert berechnet. Anschließend wird jeder Punkt dem Cluster zugeordnet, dessen Schwerpunkt am nächsten liegt; die Mittelwerte der Cluster werden erneut berechnet und die Punkte erneut zugeordnet. Dieser Prozess wird so lange fortgesetzt, bis die Neuzuordnung die Clusterzugehörigkeit der Punkte im Datensatz nicht mehr ändert.  Ein Wort der Vorsicht: Die Cluster sind nicht robust, und insbesondere die anfänglichen zufälligen Zuordnungen von Punkten zu Clustern haben einen starken Einfluss auf die Ergebnisse. Wir sollten daher den K-means-Algorithmus mehrmals ausführen und dann die beste Clusterung auswählen.  Und wie kann man feststellen, welches das beste ist? Wenn wir bereits eine Vorstellung vom Abstand haben, können wir für jedes Cluster berechnen, wie groß die Unterschiede zwischen den Punkten in diesem Cluster sind. Nehmen wir die Summe über alle K Cluster: Wenn die Gruppen einen Sinn ergeben und jedes Cluster Punkte enthält, die einander ähnlich sind, dann erwarten wir, dass die Summe klein ist - wir wählen also das Cluster mit der geringsten Summe.   * 1. Neuronale Netze   Ein neuronales Netz besteht aus einer Reihe miteinander verbundener Einheiten (so genannter "Neuronen"), wie sie in der folgenden Abbildung dargestellt sind.  Jedes Neuron nimmt mehrere Eingaben auf, weist jeder Eingabe ein Gewicht zu, kombiniert sie dann und lässt sie durch eine Aktivierungsfunktion laufen, um eine Ausgabe zu erzeugen. Die Sigmoidfunktion wird häufig als Aktivierungsfunktion verwendet - das bedeutet, dass das Neuron wie eine logistische Regression funktioniert! Die derzeit am häufigsten verwendete Aktivierungsfunktion ist jedoch noch einfacher - sie wird als gleichgerichtete lineare Einheit (ReLU) bezeichnet und nimmt den Wert f(x) = x an, wenn die Eingabe x positiv ist, und f(x) = 0, wenn x negativ ist.    Ein neuronales Netz wird gebildet, indem diese so genannten Neuronen in Schichten (layers) organisiert werden.  Beim Training eines neuronalen Netzes wird versucht, die Werte für die Netzgewichte zu ermitteln, die den Vorhersagefehler bei den Trainingsdaten minimieren (gemessen an einer bestimmten Verlustfunktion).    Wie du sehen kannst, sind die Bausteine eines neuronalen Netzes recht einfach. Was sie so komplex macht, ist die schiere Anzahl der "Neuronen", über die sie verfügen, die Anzahl der Schichten und die verschiedenen Möglichkeiten, wie die Neuronen miteinander verbunden werden können.   1. Bewertung der Performance    1. Genauigkeit und Co.   Es gibt viele Metriken, die zur Messung der Performance eines trainierten Modells verwendet werden können. Welche Metrik verwendet wird, hängt von der Art des Modells (supervised, unsupervised, reinforcement Learning; Klassifizierung vs. Regression) und vom Anwendungskontext ab. Wir werden uns auf supervised Learning konzentrieren.  Im Rahmen des supervised Learnings müssen Datensätze in Trainings-, Validierungs- und Testdatensätze aufgeteilt werden. Die Testdatensätze sollten weder beim Training noch bei der Validierung zu sehen sein: Sie sollten "weggesperrt" werden und erst ganz zum Schluss herausgeholt werden, um zu testen, wie das Modell auf völlig neuen Daten funktioniert. Nur dann, wenn die Testdaten repräsentativ für den beabsichtigten Verwendungskontext des Modells sind, kann die Performance des Modells auf den Testdaten als Indikator für seine Performance im "echten" Leben angesehen werden. Das bedeutet auch, dass unterschiedliche Verwendungskontexte unterschiedliche Testsätze erfordern!  Validierungsdaten werden verwendet, um das "beste" Modell zu wählen. Angenommen, du hast einen Entscheidungsbaum-Klassifikator, bei dem du versuchst zu entscheiden, was die beste "Tiefe" ist, und du willst ihn auch mit einem Naïve Bayes-Klassifikator vergleichen: Verwende die Performance auf dem Validierungsdatensatz, um den Vergleich durchzuführen. Es sei noch einmal darauf hingewiesen, dass ein Datensatz, der zur Validierung verwendet wurde, ***nicht*** als Testdatensatz verwendet werden ***kann***. Unter Berücksichtigung dieses Grundsatzes kannst du jedoch die Validierungsdaten für mehr als eine Validierung oder einen Modellvergleich verwenden.  Der Trainingsdatensatz schließlich ist der Datensatz, der zum Trainieren des Modells verwendet wird. Idealerweise sollten auch die Validierungsdaten vollständig von den Trainingsdaten getrennt sein. In Fällen, in denen nur wenige Daten zur Verfügung stehen, ist es jedoch möglich, Bootstrapping oder Kreuzvalidierung (siehe unten) zu verwenden, um den Trainingsdatensatz sowohl für das Modelltraining als auch für die Modellvalidierung zu nutzen.  Sobald ein Test- oder Validierungsdatensatz erstellt ist, müssen wir auch wissen, wie wir die Performance des Modells messen können. Denke daran, dass bei einem supervised Learning-Algorithmus die Beispiele im Datensatz alle den "richtigen" Zielwert haben, der mit dem vom Modell vorhergesagten Wert verglichen werden kann.   * Die am häufigsten verwendete Performance-Kennzahl für Regressionsmodelle ist MSE (mean squared error). Berechnet wird die mittlere quadratische Abweichung zwischen dem tatsächlichen Zielwert und der Modellvorhersage. Dies sollte in deinem Statistikkurs behandelt worden sein und wird hier nicht weiter ausgeführt. * Die am häufigsten verwendete Performancekennzahl für die Klassifizierung ist die Genauigkeit: Dies ist einfach die Gesamtzahl der richtigen Klassifizierungen im Verhältnis zur Gesamtzahl der Instanzen im Datensatz.   Diese Messgrößen sind jedoch nicht immer die "besten", wie die folgenden Beispiele zeigen.  Binäre Klassifikatoren sind Klassifizierungssysteme, bei denen es nur zwei mögliche Zielklassen gibt: nennen wir sie POSITIV und NEGATIV.  Wir werden verschiedene Performancemetriken für diese untersuchen, und warum sie unter bestimmten Umständen der Genauigkeit vorzuziehen sind.  Beginnen wir mit einem häufig verwendeten Hilfsmittel, um die Performance eines binären Klassifikators zu verstehen: die Konfusionsmatrix.    Unter Verwendung der Terminologie der Konfusionsmatrix können wir eine Formel für die Genauigkeit aufstellen:  Genauigkeit = (TP + TN)/(TP + TN + FP + FN)  Wann würden wir eine andere Messgröße als die Genauigkeit verwenden wollen?   * Wenn die Zielklassen in unserem Test-Datensatz stark unausgewogen sind: Wenn beispielsweise 95% POSITIV und nur 5% NEGATIV sind, dann hätte ein Klassifikator, der einfach alles als POSITIV klassifiziert, eine erstaunliche Genauigkeit von 95 %. Aber wäre das sinnvoll? * Ist es wichtiger, alle POSITIVEN Elemente korrekt zu identifizieren (z. B. bei einer medizinischen Diagnose wollen wir sicherstellen, dass wir eine Krankheit erkennen, damit wir mit der Behandlung beginnen können)? Oder ist es wichtiger, falsche POSITIVE zu vermeiden?   Eine erweiterte Version der Konfusionsmatrix (siehe unten) kann bei der Wahl der Metrik helfen:    Wenn wir also alle POSITIVEN Elemente identifizieren müssen, sollte unser Modell eine hohe *Sensitivität* bzw. *True Positive Rate (TPR)* haben. Wenn wir stattdessen falsche POSITIVE vermeiden wollen, sollte unser Modell die *Falsch-Positiv-Rate (FPR)* minimieren - was bei Betrachtung der Konfusionsmatrix gleichbedeutend ist mit der Maximierung der *Spezifität* bzw. der *Rate der echten Negative.*  Selbst wenn klar ist, dass wir die richtige Kennzahl (oder Kennzahlen - wir können versuchen, mehr als eine zu optimieren oder ein Gleichgewicht zwischen mehreren zu finden) haben, wann ist der Punkt erreicht, an dem wir sagen: "Das ist gut genug", und beschließen, das Modell zu verwenden? Auf diese Frage gibt es keine Lehrbuchantwort - sie hängt vom jeweiligen Kontext ab.  Betrachten wir als Beispiel eine "reale" Anwendung: automatische Erkennung von Hassreden in sozialen Medien.  *Nach den im Rahmen des "Barometro dell'Odio"-Projekts von Amnesty International Italien gewonnenen Daten (siehe die data4good-Folien) machen Hassreden etwa 1 % der politischen Online-Inhalte aus.* Da die Zielklasse so unausgewogen ist, ist die Genauigkeit nicht die beste Wahl der Metrik. Angenommen, wir haben ein Modell für Hassreden entwickelt, das für eine hohe TPR und eine niedrige FPR optimiert wurde: Es erreicht 99 % TPR und 1 % FPR.  ***Wie viele von 100 Kommentaren, die das Modell als Hassrede einstuft, sind dann voraussichtlich tatsächlich neutrale Kommentare? Versuche, das selbst herauszufinden, bevor du die Excel-Tabelle unten liest!***    Du kannst die obige Tabelle verwenden, um mit verschiedenen TPRs, FPRs und Prävalenzen herumzuspielen. Dies sollte dir ein Gefühl für die Bedeutung nicht nur von Metriken auf der Grundlage unseres Test-Datensatzes vermitteln, sondern auch für den Versuch, die Auswirkungen des Modells in seinem Anwendungskontext zu verstehen.  Zum Beispiel: Wenn du wüsstest, wie viel Prozent der neutralen Kommentare als Hassrede eingestuft würden, würdest du dann empfehlen, das Modell zur Klassifizierung und *automatischen Zensierung von* Hassrede-Kommentaren zu verwenden?   * 1. Bootstrapping   Bootstrapping basiert auf einer Zufallsstichprobe mit Ersetzung aus den Trainingsdaten – d. h. man nimmt die oft zitierte Vase mit den farbigen Kugeln, die in Wahrscheinlichkeitstexten so häufig vorkommt, und zieht zufällig eine Kugel heraus, notiert die Farbe und wirft die Kugel wieder in die Vase wirft. Das bedeutet, dass ein und dieselbe Beobachtung mehrmals gezogen werden kann, während andere Beobachtungen möglicherweise gar nicht gezogen werden.  Diese statistische Tatsache wird ausgenutzt: Aus den Trainingsdaten werden so oft Stichproben gezogen, bis ein neuer Trainingsdatensatz mit demselben Umfang entsteht. Die Beobachtungen, die bei diesem Verfahren nie gezogen wurden, werden in den Validierungsdatensatz aufgenommen. Die Validierungsergebnisse werden zum Vergleich der verschiedenen Algorithmen herangezogen.   * 1. Kreuzvalidierung   Es gibt verschiedene Möglichkeiten, eine Kreuzvalidierung durchzuführen, aber wir konzentrieren uns auf die n-fache Kreuzvalidierung und setzen der Einfachheit halber n = 5.  Der Trainingsdatensatz wird nach dem Zufallsprinzip in 5 etwa gleich große Untergruppen aufgeteilt.   * Im ersten Durchgang nehmen wir die Datengruppe 1 als Validierungsdaten und trainieren mit den übrigen Daten (Gruppen 2, 3, 4, 5). * Im zweiten Durchgang wird die zweite Datengruppe zur Validierung beiseite gelegt, und der Algorithmus trainiert auf den anderen Datengruppen (1,3,4,5). * Wir fahren auf diese Weise fort, bis alle 5 Datengruppen genau einmal als Validierungsdaten gedient haben. * Wir haben dann 5 Validierungsergebnisse (z.B. Fehlerrate für Klassifizierung, MSE für Regression), die zum Vergleich der verschiedenen Algorithmen verwendet werden können. * Sobald die Validierung abgeschlossen ist und ein "bestes" Modell ausgewählt wurde, kann es auf dem gesamten Datensatz neu trainiert werden.   1. Andere Überlegungen   Es gibt Situationen, in denen diese Maßstäbe für die Performance nicht ausreichen. Betrachten wir das folgende Beispiel, in dem ein Bildklassifikator ein Muster erkannt hat und Bilder als "Hund" und "Wolf" klassifizieren kann.  C:\Users\Rania\Pictures\Picture1.png  Wie wird der Algorithmus wohl die nächsten beiden Bilder einordnen?  **"Hund" "Wolf"**    Das Bild auf der linken Seite wurde als "Hund" eingestuft. Das rechte Bild als "Wolf".  Warum? Weil das Modell eigentlich nicht Hund vs. Wolf, sondern Schnee vs. keinen Schnee erkannt hat.    Dieses Beispiel ist inspiriert von dem Artikel "Warum sollte ich Ihnen vertrauen?" [1]. Solange das Modell zu komplex ist, als dass wir verstehen könnten, welche Muster es gelernt hat und warum eine bestimmte Vorhersage getroffen wurde, ist es für uns schwierig, Fehler zu erkennen. Es gibt Situationen, in denen es viel wichtiger sein kann, zu verstehen, welche Muster das Modell gelernt hat, als ein paar zusätzliche Prozentpunkte an Genauigkeit zu gewinnen.  Neben der Erklärbarkeit könnten weitere Anforderungen an das Modell die Sicherheit (z. B. gegen Hacker oder Datenvergifter), der Datenschutz (wenn der Algorithmus sensible Daten verarbeiten muss) oder die Nichtdiskriminierung sein (siehe die Folien zu data4good). Es gibt viele Kriterien, die zusammen das "beste" Modell ergeben - die Genauigkeit ist nur ein Kriterium unter vielen.   * 1. Weitere Lektüre   Mit diesem Skript hast du gerade erst deine ML-Reise begonnen. Wenn du neugierig bist, mehr erfahren und einige Probleme ausprobieren möchtest, empfehlen wir dir das Lehrbuch "An Introduction to Statistical Learning" [2]. | |
| **Selbstbeurteilung (Multiple-Choice-Fragen und Antworten)** | 1. Welche der folgenden Aussagen beschreibt am ehesten den Ansatz des maschinellen Lernens zur Bewältigung einer Aufgabe?   A) Schritt-für-Schritt-Anweisungen befolgen  **B) Mustererkennung anhand historischer Daten oder früher Versuchen und Anwendung des Musters**  C) Weitere Versuche nach dem Zufallsprinzip bis zum Erfolg   1. Welcher Algorithmus geht davon aus, dass die Eingangsmerkmale voneinander unabhängig sind?   A) Neuronale Netze  B) Entscheidungsbäume  **C) Naive Bayes**   1. In wie viele Untergruppen wird dein Trainingsdatensatz bei einer 5-fachen Kreuzvalidierung unterteilt? Und wie oft wird der Algorithmus insgesamt trainiert?   **A) 5 Untergruppen; 5 mal trainiert**  B) 4 Untergruppen; 5 mal trainiert  C) 5 Untergruppen; 6 mal trainiert | |
| **Ressourcen (Videos, Verweislink)** | * <https://medium.com/@lyon-nlp/labeling-tools-for-nlp-36a8179f15d8> * <https://towardsdatascience.com/introduction-to-machine-learning-with-graphs-f3e73c38d4f8>  James, G. et al, *An Introduction to Statistical Learning,* 2nd ed., 2021. Verfügbar unter [https://www.statlearning.com/.](https://www.statlearning.com/) | |
| **Verwandtes Material** | * Foto von [Colin Davis](https://unsplash.com/@cd163601?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText) auf [Unsplash](https://unsplash.com/de/fotos/ZTD_WOL4cXA?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText): red setter * Foto von [Ashlee Marie](https://unsplash.com/pt-br/@ashleew?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText) auf [Unsplash](https://unsplash.com/de/fotos/JUnjhuBSB2M?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText): Hund auf der Wiese * Foto von [Oscar Sutton](https://unsplash.com/@o5ky?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText) auf [Unsplash](https://unsplash.com/de/fotos/2WofMaFL37g?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText): Labrador * Foto von ractapopoulos auf [Pixabay](https://pixabay.com/images/id-2288533/): Wolfsgeheul * Foto von StormmillaGirl auf [Pixabay](file:///C:/Users/Rania/AppData/Roaming/Microsoft/Word/Bild%20von%20%3Ca%20href=%22https:/pixabay.com/de/users/stormmillagirl-749682/): Wolf in verschneiten Wäldern * Foto von Leila LaRochelle auf [pexels](file:///C:/Users/Rania/AppData/Roaming/Microsoft/Word/Foto%20von%20Leila%20Larochelle:%20https:/www.pexels.com/de-de/foto/schnee-winter-tier-wolf-10709572/): Wolf im Schnee von oben * Foto von Steve auf [pexels](https://www.pexels.com/de-de/foto/brauner-wolf-682375/): brauner Wolf, kein Schnee * Foto von Maurizio Izzo auf [Pixabay](https://pixabay.com/images/id-4970700/): Deutscher Schäferhund im Schnee | |
| **Verwandte PPT** | 1. Statistik 2. Data4goodd | |
| **Literaturverzeichnis** | [1] Ribeiro, M. et al, "Why Should I Trust You?": Erklärung der Vorhersagen eines beliebigen Klassifikators. Verfügbar auf arxiv: <https://arxiv.org/abs/1602.04938>  [2] James, G. et al, *An Introduction to Statistical Learning,* 2nd ed., 2021. Verfügbar unter [https://www.statlearning.com/.](https://www.statlearning.com/) | |
| **Zur Verfügung gestellt von** | [Women in AI Austria] | |