# Machine Learning

## Einführung in Machine Learning

Der Bereich des maschinellen Lernens (Machine Learning) basiert auf mathematischen Methoden wie Linearer Algebra, Wahrscheinlichkeitstheorie und Statistik. Ziel ist es, Abhängigkeiten zwischen den Daten zu erkennen und Muster zu identifizieren, um auf Basis dieser Muster und Abhängigkeiten Prognosen zu erstellen.

## Lernmethoden und -ansätze

Das maschinelle Lernen lässt sich, wie in der folgenden Abbildung zu sehen ist, in drei Hauptkategorien unterteilen:

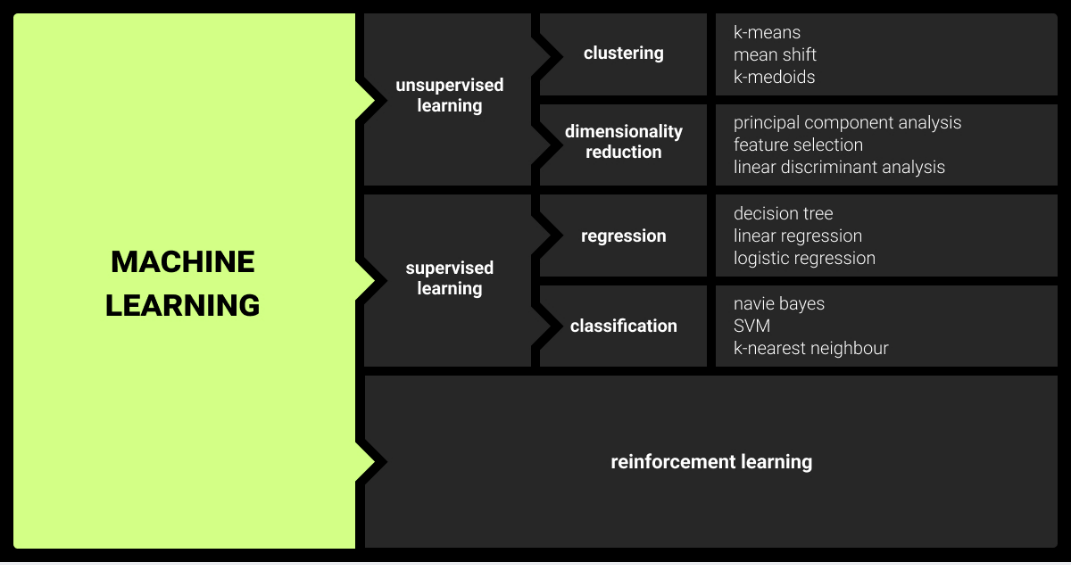


Abbildung 1: Maschinelles Lernen - Arten - Zweck - Algorithmen[[1]](#footnote-1)

* **Supervised Learning (überwachtes Lernen):** In dieser Kategorie des maschinellen Lernens wird der Algorithmus mit gelabelten Daten trainiert. Das bedeutet, dass der Algorithmus sowohl die Merkmale (Features) als auch die zugehörigen Zielwerte (Targets) erhält. Das Modell lernt, wie bestimmte Kombinationen von Merkmalen zu bestimmten Ergebnissen führen. Ziel ist es, anhand der erkannten Muster Vorhersagen für neue, unbekannte Daten zu treffen. Überwachtes Lernen hat das Ziel, eine Funktion zu finden, die bestimmte Eingabedaten mit bestimmten Ausgabedaten verknüpft[[2]](#footnote-2).

Supervised Learning wird für zwei Arten von Problemen eingesetzt:

* + **Regression:** Diese Methode ermöglicht eine Vorhersage von kontinuierlichen numerischen Werten, z.B. Preis, Kosten, Noten, Temperaturen, etc.

Wichtige Regressionsmodelle sind:

* + - Decision Tree
    - Linear Regression
    - Logistic Regression
  + **Klassifikation:** Diese Methode ermöglicht eine Zuordnung von Daten zu bestimmten Kategorien, z.B. Betrug/nicht Betrug, krank/gesund, Spam/nicht Spam, etc.

Wichtige Klassifikationsmodelle sind:

* + - Support Vector Machine
    - Naive Bayes
    - K-Nearest Neighbour
* **Unsupervised Learning (unüberwachtes Lernen):** In dieser Kategorie werden die Modelle ohne gelabelte Daten trainiert, d.h. es werden keine Zielwerte (Targets) dem Modell übergeben. Stattdessen versucht das Modell, aus nur den reinen Merkmalen (Features) zu lernen und Muster zu erkennen. Ziel des unüberwachten Lernens ist es, eine Funktion zu entwickeln, die Eingabedaten clustert oder gruppiert[[3]](#footnote-3).

Unsupervised Learning wird beispielweise für folgende Aufgaben verwendet:

* + **Clusterbildung:** Ähnlicher Datenpunkte werden so gruppiert, dass die Datenpunkte innerhalb einer Gruppe sehr ähnlich sind, während sie sich von Daten anderer Gruppen deutlich unterscheiden, z.B. Erstellung von Kundensegmenten.

Häufig verwendete Algorithmen zur Clusterbildung:

* + - k-means
    - Mean Shift
    - k-medoids
  + **Dimensionsreduktion:** Bei Daten mit einer hohen Dimensionsanzahl wird versucht, diese auf die wichtigsten Dimensionen zu reduzieren

Häufig verwendete Algorithmen zur Dimensionsreduktion:

* + - Principal Component Analysis
    - Feature Selection
    - Linear Discriminant Analysis
* **Reinforcement Learning (bestärkendes Lernen):** Der Lernagent lernt, wie in Abbildung X dargestellt, hier ähnlich wie ein Mensch. Für das vom System gelieferte Ergebnis erhält er von seiner Umgebung ein Feedback. Eine Belohnung, wenn das Ergebnis korrekt ist, und eine Bestrafung, wenn das Ergebnis falsch ist. Dadurch optimiert sich das ML-System anhand dieser Feedbacks. Um das gewünschte Ergebnis zu erreichen, benötigt der Lernagent, anderes als beim Supervised Learning, keine vorab bereitgestellten Daten (Features und Targets), aus denen er lernt. Stattdessen lernt er explorativ anhand der Trial-and-Error-Methode, um die Belohnung zu maximieren und Bestrafungen zu vermeiden[[4]](#footnote-4).

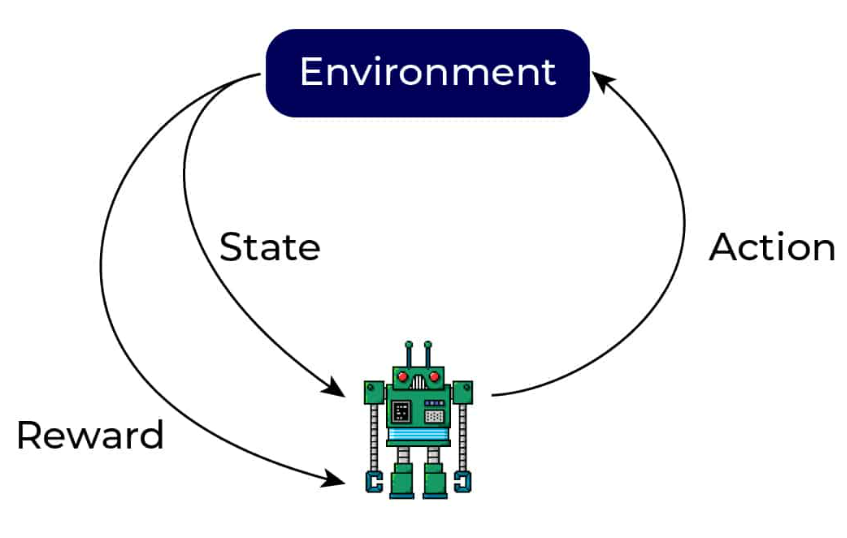
****

Abbildung 2: Reinforcement Learning[[5]](#footnote-5)

## Der Lebenszyklus eines ML-Modells

Um die gewünschten Ergebnisse eines KI-Systems zu erreichen, durchläuft dessen Entwicklung mehrere wichtige Phasen.

### Definition des Problems

In dieser Phase wird ein bestehendes Problem genau analysiert und eine geeignete Lösung formuliert.

### Datengewinnung und -beschaffung

Die für das KI-Modell relevanten Daten werden gesammelt, da sie die Grundlage für das Training des Modells bilden.

### Auswahl geeigneter Modelle

Jedes KI-Modell ist für bestimmte Problemstellungen geeignet. Zusätzlich spielen verfügbare Daten sowie weitere Faktoren, wie Verarbeitungsgeschwindigkeit, Abhängigkeiten zwischen den Daten und technische Anforderungen eine zentrale Rolle bei der Modellauswahl.

Bevor das Modell ausgewählt wird, sollten folgende Fragen berücksichtigt werden:[[6]](#footnote-6)

* Was ist mein Problem, und wie lässt es sich lösen?
* Habe ich alle notwendigen Daten zur Verfügung?
* Sind die Daten, die ich habe, für das Modell geeignet und nutzbar?
* Gibt es Abhängigkeiten oder Zusammenhänge zwischen den Daten?
* Ist mir ein schnelleres Ergebnis wichtiger oder ein präziseres?

Nachfolgend sind wichtige Punkte aufgeführt, die die Auswahl eines passenden Machine-Learning-Modells erleichtern:

* **Lineare Regression:** Mithilfe von Linear Regression werden die Beziehungen zwischen abhängigen und unabhängigen Variablen durch eine Linie (Beste Regression-Linie) dargestellt.[[7]](#footnote-7) Dieser Algorithmus wird verwendet, um reale Werte bzw. Targets (z. B. Preis, Kosten, Gewinn, etc.) anhand mehrerer anderen realen Werte bzw. Features zu schätzen. Ein Beispiel aus der realen Welt ist die Vorhersage des Gewichts anhand der Körpergröße und des Körperbaus[[8]](#footnote-8).
* **Logistic Regression:** Es handelt sich hierbei um eine n Klassifizierungsalgorithmus. Er sagt die Wahrscheinlichkeit (zwischen 0 % und 100 %) für das Auftreten eines diskreten Werts bzw. Target (z. B. True oder False, Fake oder Echt) anhand anderer unabhängiger Werte bzw. Features voraus, indem er eine logistische Funktion anpasst.[[9]](#footnote-9)
* **Decision Tree:**  Dieser Algorithmus wird hauptsächlich verwendet, um Klassifizierungsprobleme zu lösen, kann jedoch auch für Regressionsprobleme eingesetzt werden. Er funktioniert sowohl mit numerischen als auch mit kategorischen Werten bzw. Features. Der Algorithmus teilt die Daten basierend auf wichtigen Attributen und Techniken wie Gini, Informationsgewinn, Chi-Quadrat und Entropie in mehrere Homogene Gruppen auf, um ein Ergebnis (Target) vorherzusagen.[[10]](#footnote-10)
* **Support Vector Machine (SVM):** Es handelt sich hierbei um eine Klassifizierungsmethode, bei der jedes Feature in einer eigenen Dimension dargestellt wird. Ziel ist es, Gruppen so zu trennen, dass sie durch eine Linie (im Fall von zwei Dimensionen bzw. Features) oder eine Hyperplane (in höheren Dimensionen bzw. Features) optimal getrennt zu werden. Der Algorithmus maximiert den Abstand (Margin) zwischen der Hyperplane und den nächstgelegenen Datenpunkten (den sogenannten Support Vectors) jeder Gruppe. Eine größere Margin gewährleistet eine robuste Trennung, sodass neue Datenpunkte zuverlässig den entsprechenden Gruppen zugeordnet werden können.[[11]](#footnote-11)
* **Naive Bayes:** Naive Bayes ist ein Klassifizierungsalgorithmus, der auf dem Bayes-Theorem basiert und gut mit sehr guten Datenmengen umgehen kann. Er berechnet die Wahrscheinlichkeiten für das Auftreten eines Ergebnisses (Target) basierend auf gegebenen Merkmalen (Features). Dieser Algorithmus geht dabei von der Annahme aus, dass die Features untereinander unabhängig sind.[[12]](#footnote-12)
* **k-Nearest Neighbors (kNN):** kNN ist einfache und intuitive Methode des maschinellen Lernens, die sowohl für Klassifikations- als auch für Regressionsprobleme eingesetzt wird. Sie wird jedoch häufiger zur Lösung von Klassifikationsaufgaben verwendet. Diese Methode speichert alle Trainingsdaten, d. h., es findet keine explizite Modellbildung statt. Bei neuen Daten berechnet kNN die Distanz (beispielsweise die euklidische, Manhattan- oder Minkowski-Distanz) zu den k nächsten Nachbarn (wobei k die Anzahl der betrachteten Nachbarn ist) und ordnet den Punkt einer Gruppe zu. Die Wahl von k hat dabei einen großen Einfluss auf die Aussagekraft des Algorithmus. Ein weiterer Nachteil ist die Empfindlichkeit des Algorithmus gegenüber Ausreißern, da diese die Ergebnisse stark beeinflussen können. Zudem erfordert kNN eine Skalierung der Daten, da unterschiedliche Wertebereiche zu einer Verzerrung der Ergebnisse führen können.[[13]](#footnote-13)
* **K-Means:** K-Means ist eine unüberwachte Methode des maschinellen Lernens, die zur Clusterbildung verwendet wird. Dabei steht k für die Anzahl der Cluster, die gebildet werden sollen. Ziel des Algorithmus ist es, Daten innerhalb eines Clusters möglichst ähnlich zu machen, während die Unterschiede zwischen den Clustern maximiert werden. In jedem Cluster gibt es einen Mittelpunkt, den sogenannten Centroid. Der Algorithmus misst die Distanz zwischen jedem Punkt und dem Centroid. Die Summe der quadrierten Abstände aller Punkte eines Clusters zum Centroid wird als Sum of Squares (SS) für diesen Cluster bezeichnet. Wenn man diese Summe für alle Cluster berechnet, ergibt sich der Gesamt-SS-Wert der Lösung. Mit einer größeren Anzahl von Clustern wird der Gesamt-SS-Wert kleiner, da die Punkte im Durchschnitt näher an den jeweiligen Centroids liegen.
* **Dimensionality Reduction Algorithm:** Da Behörden, Unternehmen und andere Organisationen heutzutage viele Daten über Ihre Kunden erfassen,[[14]](#footnote-14)steigt die Anzahl der vorhandenen Dimensionen stetig an. Um ein robustes Machine-Learning-Modell zu entwickeln, müssen relevante Dimensionen (Features) ausgewählt werden. Dies ist bei einer enormen Anzahl von Dimensionen sehr zeitaufwändig und mit erheblichem Aufwand verbunden. Die Methode der Dimensionsreduktion (Dimensionality Reduction) ist ein unüberwachter Machine-Learning-Algorithmus, der dabei hilft, die Dimensionen auf die für das Training des Modells wesentlichen und wichtigsten Merkmale zu reduzieren. Der Algorithmus verwendet Methoden wie Entscheidungsbäume, Random Forest oder PCA und berücksichtigt dabei Korrelationsmatrizen sowie das Verhältnis fehlender Werte[[15]](#footnote-15).
* **Random Forest:** Dieser Random-Forest-Algorithmus basiert auf Entscheidungsbäumen und nutzt mehrere Bäume (Ensemble-Learning), um präzisere Ergebnisse zu liefern. Die Trainingsdaten werden mithilfe des Bootstrapping-Verfahrens in mehrere zufällige Teilmengen aufgeteilt. Jede Teilmenge dient als Grundlage, um einen einzelnen Entscheidungsbaum zu trainieren. Jeder Baum liefert anschließend eine eigene Vorhersage.

Da Entscheidungsbäume sowohl für Klassifikations- als auch für Regressionsaufgaben geeignet sind, kann Random Forest ebenfalls für beide Problemtypen eingesetzt werden, jedoch am Häufigsten für Klassifikationsaufgaben.

* + Bei Klassifikation wird das häufigste Ergebnis (Modus) der Bäume als endgültige Vorhersage übernommen.
  + Bei Regression wird der Durchschnitt der Ergebnisse aller Bäume berechnet und als Vorhersage genutzt.

Im Vergleich zu einzelnen Entscheidungsbäumen, die einen niedrigen Bias (gut) und eine hohe Varianz (schlecht) aufweisen, kombiniert Random Forest die Vorteile beider Aspekte. Er erzielt sowohl einen niedrigen Bias als auch eine niedrige Varianz, wodurch das Modell robuster und genauer wird.

Bias und Varianz sind wichtige Qualitätsmerkmale eines Modells. Ein Modell mit geringem Bias und niedriger Varianz liefert zuverlässige und präzise Vorhersagen. Bei Random Forest laufen die Modelle unabhängig parallel (Bagging).

* **Gradient Boosting Algorithm:** Dies ist eine Ensemble-Learning-Methode. Beim Gradient Boosting werden mehrere Modelle sequentiell trainiert (Boosting), wobei jedes Modell darauf abzielt, die Fehler des vorherigen Modells zu korrigieren.[[16]](#footnote-16)

In der folgenden Abbildung werden die Modellstrukturen von Bagging und Boosting graphisch veranschlicht:

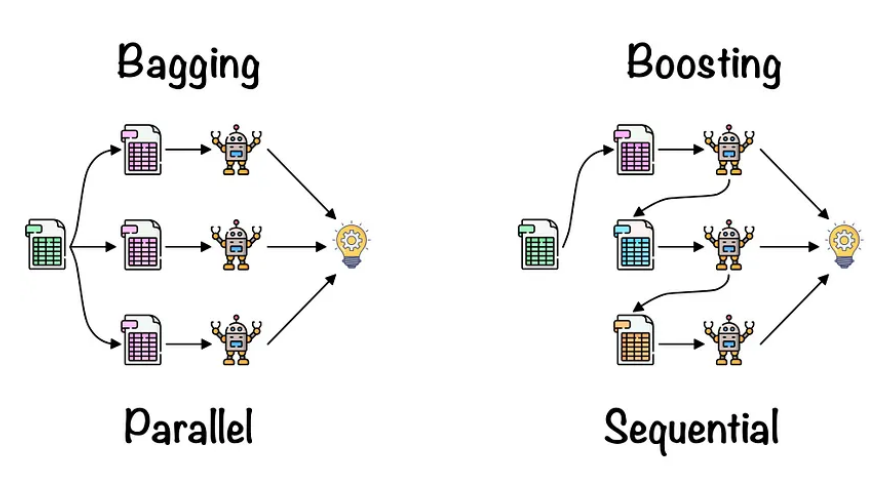
****

Abbildung 3: Modellstrukturen von Bagging und Boosting[[17]](#footnote-17)

Die heutzutage von Machine-Learning-Ingenieuren am Häufigsten verwendeten Algorithmen sind Entscheidungsbäume in Form von Random Forest und Gradient Boosting,[[18]](#footnote-18) da sie aufgrund ihrer Robustheit und Genauigkeit (geringe Varianz und geringer Bias) bei Vorhersagen überzeugen.

### Datenaufbereitung und -bereinigung

Die gesammelten Daten liegen nicht in einem Format vor, das der KI-Algorithmus direkt nutzen kann. Deshalb werden sie so aufbereitet, dass sie für das Modell geeignet und verarbeitbar sind.

Je nach Art der Daten, z.B. Zahlen, Text, Fotos, Videos, Audios, etc., existieren verschiedene Datenverarbeitungsschritte.

Der Ablauf der Preprocessing-Prozess ist bei Zahlendaten ist wie folgt:

**Umgang mit fehlenden Werten:** Zunächst werden die fehlenden Werte (Nullwerte) identifiziert. Anschließend können entweder die Datensätze, die diese Werte enthalten, abhängig davon, wie viele Datensätze Nullwerte aufweisen, gelöscht oder die fehlenden Werte durch geeignete Ersatzwerte aufgefüllt werden.

Bei einer großen Anzahl an Nullwerten innerhalb der Spalte ist es sinnvoll, die fehlenden Werte durch andere geeignete Werte zu ersetzen, um die Daten optimal für das Training des Modells zu nutzen.

Falls der fehlende Wert ein String ist, kann er durch den häufigsten Wert (Mode) der Spalte ersetzt werden. Bei numerischen Werten stehen mehrere Möglichkeiten zur Verfügung, wie die Verwendung des Medians oder des Mittelwerts. Um den passenden Ersatzwert für numerische Werte zu bestimmen, wird eine explorative Analyse der betroffenen Spalte durchgeführt. Hier bietet scikit-learn mithilfe SimpleImputer die Möglichkeit, fehlende Daten mit dem Median oder Mittelwert zu ersetzen.

Sind die Daten normalverteilt (Normal Distribution), wird der Mittelwert verwendet, um die fehlenden Werte zu ersetzen. Liegt jedoch eine Schiefe (Skewed Distribution) in den Daten vor (d.h. die Verteilung der Daten hat eine asymmetrische Form, bei der die Daten mehr nach rechts (Left Skewed Distribution bzw. Negative Skewness) oder links (Right Skewed Distribution bzw. Positive Skewness) als zur Mitte hin konzentriert sind), wird stattdessen der Median verwendet.

Scikit-Learn bietet darüber hinaus die Möglichkeit, fehlende Werte mithilfe von Machine-Learning-Algorithmen, wie dem KNNImputer, dem IterativeImputer etc., zu ersetzen. Dabei werden die fehlenden Werte auf Grundlage der Korrelation innerhalb des Datensatzes sowie mithilfe ähnlicher Datensätze geschätzt und ausgefüllt.

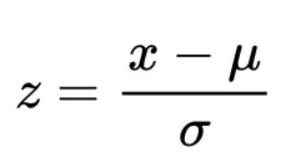
**Enkodierung kategorialer Daten:** Da Machine-Learning-Algorithmen nicht direkt mit String-Spalten, z.B. Stadt, Familienstand, Produktkategorie, Herkunft, Schulabschluss etc., arbeiten können, müssen diese Daten zuerst enkodiert werden.   
  
Scikit-Learn bietet verschiedene Möglichkeiten, um kategoriale Daten zu enkodieren:

* **OneHotEncoder**: Jeder Wert in der kategorialen Spalte (Dimension) wird in eine eigene Spalte umgewandelt, d.h. die Anzahl der erzeugten Spalten entspricht der Anzahl der eindeutigen Werte (Unique Values) in der ursprünglichen Spalte. Die erzeugten Spalten erden mit binären Werten (0 oder 1) gefüllt. Ein Wert von 1 bedeutet, dass der aktuelle Datensatz den entsprechenden Wert in der ursprünglichen Spalte hat, während 0 bedeutet, dass der aktuelle Datensatz einen anderen Wert hat.
* **LabelEncoder**: Jeder Ausprägung wird eine fortlaufende zahl zugewiesen. Daten mit der gleichen Ausprägung erhalten die gleiche Zahl, z.B. 0 für „Deutschland“, 1 für die „USA“, 2 für „Schweden“ und so weiter. Das Ergebnis ist eine einzige Spalte.
* **OrdinalEncoder**: Dies funktioniert ähnlich wie der **LabelEncoder**, mit dem Unterschied, dass die Reihenfolge der Ausprägungen beim **OrdinalEncoder** eine wichtige Rolle spielt. Ein Beispiel wäre, 0 für „Bachelor“, 1 für „Master“, 2 für „Doktor“.

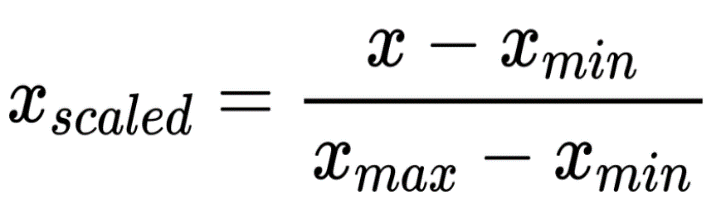
**Skalierung numerischer Features:** Numerische Features mit unterschiedlichen Wertenbereichen müssen vor der Anwendung von Machine-Learning-Modellen skaliert werden. Das bedeutet, dass alle Features auf einen ähnlichen Wertebereich transformiert werden sollten. Dies ist wichtig, da viele Machine-Learning-Algorithmen empfindlich auf große Wertebereiche reagieren und Features mit hohen Werten dadurch ein höheres Gewicht bei der Vorhersage erhalten als Features mit kleinen Werten. Dies kann die Genauigkeit der Prognose negativ beeinflussen.  
  
Auch die Zielspalte (numerische Target-Spalten) kann skaliert werden, muss jedoch nach der Vorhersage zurückskaliert werden, um die tatsächlichen Werte zu erhalten.

Zum Skalieren von numerischen Features oder Targets existieren Methoden wie:[[19]](#footnote-19)

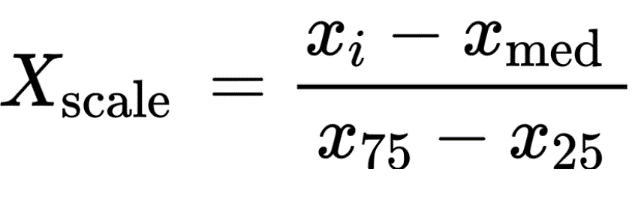
* **StandardScaler:** Dies ist eine einfache und schnelle Methode, die verwendet wird, wenn die Daten einer Normalverteilung folgen. Numerische Daten werden so skaliert, dass der Mittelwert und die Standardabweichung eines numerischen Features nach der Skalierung 0 bzw. 1 betragen. Diese Methode ist für normal große, normalverteilte Daten ohne Ausreißer geeignet. Das Bedeutet, dass ein Ausreißer das Ergebnis erheblich beeinflussen kann und in diesem Fall eine andere Skalierungsmethode erforderlich ist. Diese Methode verwendet den Z-Score und skaliert die numerischen Daten anhand des Mittelwerts und der Standardabweichung wie folgt:



* **MinMaxScaler:** Die numerischen Daten werden auf einen bestimmten Wertebereich skaliert, in der Regel zwischen 0 und 1. Dieses Intervall kann jedoch angepasst werden. Diese Methode hat den Vorteil, dass die relative Beziehung zwischen den Daten erhalten bleibt. Wie der StandardScaler ist auch diese Methode bei Ausreißern empfindlich. Sie wird als Alternative zum StandardScaler verwendet, wenn man einen Mittelwert von 0 und eine Standardabweichung von 1 vermeiden möchte. Diese Methode ist einfach zu implementieren und arbeitet schnell. Sie skaliert die numerischen Daten anhand des minimalen und maximalen Werts wie folgt:



* **RobustScaler:** Diese Skalierungsmethode kann gut mit Ausreißern umgehen und behält die relative Beziehung zwischen den Daten bei. Der Skalierungsbereich bewegt sich zwischen dem ersten und dritten Quartil und ist bei bedarf anpassbar. Diese Methode skaliert die numerischen Daten anhand des Medians sowie des ersten und dritten Quartils wie folgt:



In der folgenden Abbildung wird Datenverteilung vor der und nach der Skalierung mit den oben genannten drei Methoden visualisiert.

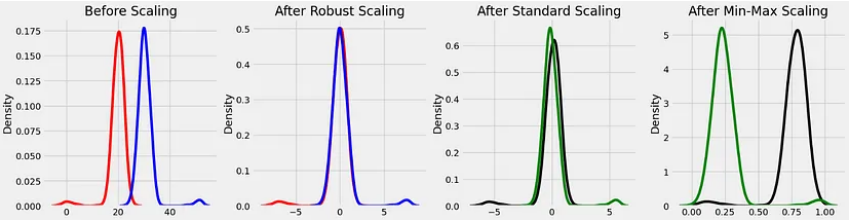


Abbildung 4: Datenverteilung vor und nach der Skalierung[[20]](#footnote-20)

### Aufteilung der Datenbestände

Die verarbeiteten Daten werden in Trainings- und Testdaten unterteilt, meist im Verhältnis 80% Trainingsdaten zu 20% Testdaten. Diese Aufteilung ermöglicht eine fundierte Evaluierung des Modells in einer späteren Phase.

### Training des Modells

Das gewählte Modell wird mithilfe der Trainingsdaten trainiert, um Muster zu erkennen und Zusammenhänge zu lernen.

### Bewertung und Validierung des Modells

Das trainierte Modell wird anschließend mithilfe der Testdaten auf Richtigkeit und Zuverlässigkeit geprüft. Die Ergebnisse der Testdaten werden mit den durch das trainierte Modell prognostizierten Werten verglichen.

### Implementierung und Anwendung

Das trainierte und getestete Modell wird schließlich eingesetzt, um Vorhersagen auf Basis neuer Daten zu treffen.

Literaturverzeichnis:

|  |  |
| --- | --- |
| [Cosg/2018] | Hasan Hüseyin Coşgun: Which data scaling technique should I use?, 2018, online im Internet: URL: <https://medium.com/@hhuseyincosgun/which-data-scaling-technique-should-i-use-a1615292061e> [Stand 01.12.2024] |
| [Dey/2024] | Roshmita Dey: Bagging v/s Boosting, 2024, online im Internet: URL: <https://medium.com/@roshmitadey/bagging-v-s-boosting-be765c970fd1> [Stand 11.12.2024] |
| [Nalc/2022] | Sarp Nalcin: StandardScaler vs. MinMaxScaler vs. RobustScaler: Which one to use for your next ML project?, 2022, online im Internet: URL: [https://medium.com/@onersarpnalcin/standardscaler-vs-minmaxscaler-vs-robustscaler-which-one-to-use-for-your-next-ml-project-ae5b44f571b9](https://medium.com/@onersarpnalcin/standardscaler-vs-minmaxscaler-vs-robustscaler-which-one-to-use-for-your-next-ml-project-ae5b44f571b9%20) [Stand 14.12.2024] |
| [Nami/2024] | Karyna Naminas: Machine Learning Algorithm: How to Choose for ML Workflows in 2024, 2024, online im Internet: URL: <https://labelyourdata.com/articles/how-to-choose-a-machine-learning-algorithm> [Stand 10.12.2024] |
| [Ray/2024] | Sunil Ray: Top 10 Machine Learning Algorithms You Must Know, 2024, online im Internet: URL: <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/> [Stand 12.12.2024] |
| [Wutt/2023] | Laurenz Wuttke: Reinforcement Learning: Wenn KI auf Belohnungen reagiert, 2023, online im Internet: URL: <https://datasolut.com/reinforcement-learning/> [Stand 08.12.2024] |

1. [Nami/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-1)
2. Vgl. [Ray/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-2)
3. Vgl. [Ray/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-3)
4. Vgl. [Wutt/2023] (in englischer Sprache), o. S. [↑](#footnote-ref-4)
5. [Wutt/2023] (in englischer Sprache), o. S. [↑](#footnote-ref-5)
6. Vgl. [Nami/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-6)
7. Vgl. [Nami/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-7)
8. Vgl. [Ray/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-8)
9. Vgl. [Ray/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-9)
10. Vgl. [Ray/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-10)
11. Vgl. [Ray/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-11)
12. Vgl. [Ray/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-12)
13. Vgl. [Ray/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-13)
14. Vgl. [Ray/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-14)
15. Vgl. [Ray/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-15)
16. Vgl. [Dey/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-16)
17. Vgl. [Dey/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-17)
18. Vgl. [Nami/2024] (in englischer Sprache), o. S. [↑](#footnote-ref-18)
19. Vgl. [Nalc/2022] (in englischer Sprache), o. S. [↑](#footnote-ref-19)
20. [Cosg/2018] (in englischer Sprache), o. S. [↑](#footnote-ref-20)