HOCHSCHULE RUHR WEST
UNIVERSITY OF APPLIED SCIENCES

Erklärbare KI im Gesundheitswesen - Implementierung und Evaluation eines Lösungsansatzes mit Hilfe des Kaggle Datensatzes „Symptoms and COVID Presence“

Geschrieben von: Jan Cieslik 10006950
Fachbereich 1
Angewandte Informatik

Erstgutachter
Michael Schellenbach
Wissenschaftlicher Mitarbeiter
Institut Informatik, Hochschule Ruhr West

Zweitgutachter
Dr.-Ing. Ahmad Rabie
Lehrkraft für besondere Aufgaben
Institut Informatik, Hochschule Ruhr West

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

RKI	Robert Koch Institut
FI	Feature Importance
PFI	Permutated Feature Importance
AI	Artificial Intelligence
KI	Künstliche Intelligenz
XAI	Explainable Artificial Intelligence
PDP	Partial Dependence Plot
LIME	Local Interpretable Model-agnostic Explanations
SHAP	Shapley Additive Explanations
DARPA	Defense Advanced Research Project Agency
SARS	Schweres Akutes Respiratorisches Syndrom

1 Einleitung

1.1 Motivation

Die Corona-Pandemie veränderte auch nach der zweijährigen Dauer viele Aspekte des alltäglichen Lebens. Hand in Hand mit gesellschaftlichen Einschränkungen, die in der Allgemeinheit für Unmut sorgte und die Wirtschaft erschütterte, stellt sie auch einen Meilenstein in der Sammlung und Auswertung medizinischer Daten weltweit dar. Die umfassende Dokumentation und Bekanntgabe täglicher Daten wie Infektionszahlen, Hospitalisierungs- und Sterberate durch weltbekannte Organisationen wie das Robert-Koch-Institut oder die World Health Organisation bieten eine noch nie gebotene Forschungsgrundlage. So riefen diverse Organisationen und Regierungen dazu auf, diese Datenmenge zu nutzen, um Zusammenhänge zu erschließen, die dabei helfen können, diese Pandemie und auch zukünftige Notstände schneller und effizienter bewältigen zu können.

Zu diesem Zweck bietet sich eine Untersuchung und Auswertung mit Hilfe von maschinellem Lernen an. So können Klassifikationsmodelle verwendetet werden, um zuverlässige Diagnosen zu stellen und Clustering Algorithmen, um neue Zusammenhänge aus den Daten zu erschließen. Allerdings bleibt ein Problem bei der Nutzung von maschinellem Lernen: Die Modelle müssen nicht nur funktionieren und akkurate Vorhersagen treffen, sondern allen voran, verständlich und erklärbar sein. Denn gerade im Gesundheitswesen ist es von höchster Wichtigkeit, die Ergebnisse und Diagnosen auch belegen zu können, denn dies schafft Vertrauen in die Technik.

1.2 Aufbau und Zielsetzung

Diese wissenschaftliche Arbeit widmet sich dem Forschungsgebiet der Erklärbaren Künstlichen Intelligenz. Das Ziel ist es, ein grundlegendes Verständnis für den Oberbegriff und seiner Facetten wie Einsatzgebieten und unterschiedlichen Algorithmen zu vermitteln, Kernbegriffe und Modelle der Erklärbaren Künstlichen Intelligenz zu erläutern und Methoden, die die Erklärbarkeit und Interpretierbarkeit von Modellen verbessern, am implementierten Beispiel zu präsentieren.

Die Arbeit teilt sich in zwei Teile: Der erste Teil vermittelt theoretische Grundlagen der Künstlichen Intelligenz, dem Maschinellen Lernen, dem Forschungsgebiet der Erklärbaren Künstlichen Intelligenz und der Rolle der KI im Gesundheitswesen. Der zweite Teil behandelt die Implementierung fünf unterschiedlicher Modelle des maschinellen Lernens. Zu Präsentationszwecken wird exemplarisch die Bearbeitung und die Funktionsweise der verwendeten Algorithmen beschrieben, die Grundlagen der Methoden der XAI erläutert und die Ergebnisse dieser inspiziert und interpretiert bzw. erklärt. Abschließend folgen die Evaluation und der Vergleich der verschiedenen Modelle und der Methoden und ein Ausblick auf potenzielle Fortsetzung gegeben.

2 Theoretische Grundlagen

Für das Verständnis des praktischen Abschnittes dieser Arbeit werden grundlegende Begrifflichkeiten der Künstlichen Intelligenz, des Maschinellen Lernens und der Erklärbarkeit und Interpretierbarkeit von Algorithmen und Methoden des Maschinellen Lernens vorausgesetzt. Infolgedessen werden diese zunächst erläutert und anschließend in Kontext des Gesundheitswesens, mit Fokus auf die COVID-19 Pandemie, gesetzt.

2.1 Künstliche Intelligenz

Eine einheitliche, allseits anerkannte Definition des Begriffes der künstlichen Intelligenz existiert derweil nicht, da der Mensch ebenso daran scheitert, menschliche Intelligenz als solche zu definieren.[1] Die eine „klassische“ Intelligenz, die die gesamte menschliche Kognition beschreibt, ist in der heutigen Wissenschaft nicht mehr prävalent. Vielmehr wird von unterschiedlichen Arten von Intelligenz gesprochen, die unter Umständen unterschiedliche Kompetenzen beschreiben. Diese sind: emotionale, körperliche, klassische, naturalistische, sowie die künstlerische Intelligenz. Vorrangig sind die ersten drei Intelligenzen bekannt. Die emotionale Intelligenz beschreibt einerseits die Fähigkeit des Menschen sich selbst, seine Emotionen und seine Gedanken differenziert wahrzunehmen, andererseits diese Punkte bei anderen Individuen wahrzunehmen. Die klassische Intelligenz umfasst die logischen, sprachlichen und problemlösenden Kompetenzen. Abschließend werden die motorischen Fähigkeiten und der Umgang mit dem eigenen Körper als körperliche Intelligenz zusammengefasst.

Allein diese drei Punkte durch eine Maschine nachahmen zu lassen, ohne, dass diese nur einer Abfolge von Verzweigenden Abfragen, die mit „Ja“ oder „Nein“ beantwortet werden können, folgt gestaltet sich als schwierig.

Zu eben diesem Zweck hat Alan Turing im Jahre 1950 den sogenannten Turing-Test entwickelt. Nach Durchführung des Tests soll die Aussage getätigt werden können, ob ein Computer oder eine Maschine eine zum Menschen gleichwertige Kognition besitzt. [2] Hierfür kommuniziert ein Proband mit einer weiteren Person und einem Computer ohne, dass bekannt ist, wer der Computer und wer die zweite Person ist. Kann der Proband zum Schluss nicht differenzieren, ob es sich bei dem Kommunikationspartner um einen Menschen oder eine Maschine handelt, gilt der Test als bestanden. Dies ist der erste Meilenstein in der Geschichte der künstlichen Intelligenz, auch wenn die Bedeutung dieses Begriffes sich in den letzten 72 Jahren deutlich verändert hat. Der Forschungsbereich erhielt nur sechs Jahre später in der Konferenz in Dartmouth, USA 1956 seinen Namen „Artificial Intelligence“ zu Deutsch, „Künstliche Intelligenz“. [3]

Von da an beschleunigte sich die Entwicklung und ein Meilenstein folgt auf dem nächsten. 1966 entstand der erste funktionstüchtige Chatbot „ELIZA“, 1988 die Gründung des deutschen Forschungszentrums für künstliche Intelligenz, die seit jeher deutlichem Zuwachs an Forschern aus der ganzen Welt erfahren haben.[4] Weiter geht es mit dem Meilenstein der künstlichen Intelligenz und Robotik, als der erste „RoboCup“ 1997 stattfand, eine Veranstaltung in der zum ersten Mal humanoide Roboter gegeneinander im Fußball antreten. Weiterhin schaffte es die vom IBM entwickelte KI den Weltmeister im Schach zu bezwingen. [5]

Durch diese Leistung beflügelt und durch die darauffolgende Kommerzialisierung angetrieben, entstanden die unterschiedlichen Teilgebiete der Künstlichen Intelligenz.

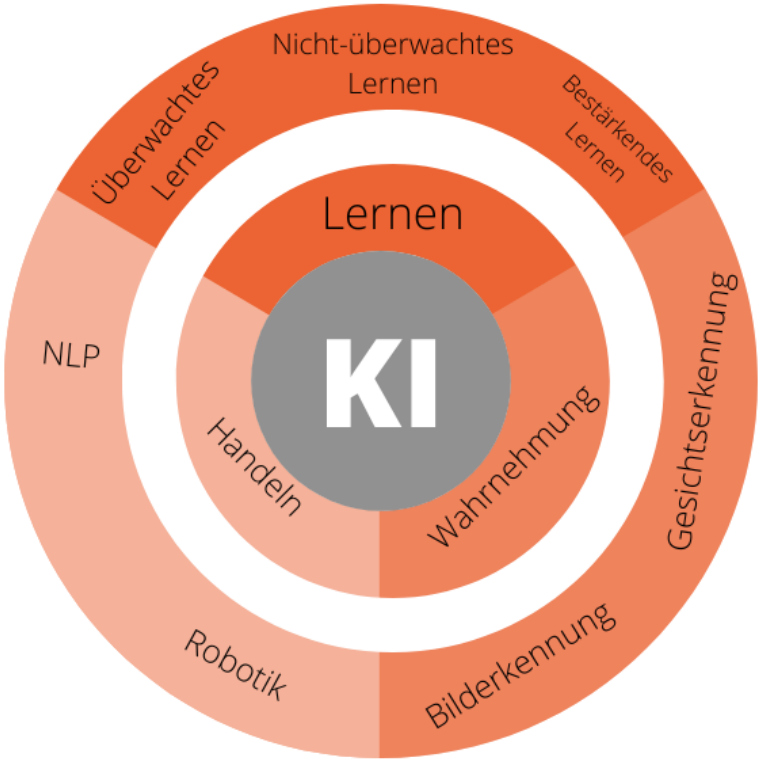


Abbildung 1: Auswahl von Themengebieten der Künstlichen Intelligenz

Um die verschiedenen Fähigkeiten des Menschen aufzugreifen und zu optimieren, gibt es unterschiedliche Ansätze in der künstlichen Intelligenz. So können intelligente Systeme die Wahrnehmung, das Handeln und Lernprozesse des Menschen versuchen nachzuahmen.

Unter die Disziplin des Handelns fallen zum Beispiel die Gebiete des *Natural Language Processing* (NLP) und der Robotik. NLP verschreibt sich der Erkennung, Verarbeitung und Generierung der menschlichen Sprachen. Dementsprechend gilt es in der Fachrichtung gesprochene Texte in geschriebenen Text umzuwandeln, die Syntax beziehungsweise Rechtschreibung eines vorliegenden Textes zu erkennen, bewerten und zu korrigieren und im Idealfall die Semantik, sprich die Bedeutung der Worte, korrekt erfassen zu können. [6] Die *Robotik* implementiert in erster Linie keine KI-Systeme, da viele der Tätigkeiten eines Roboters, wie Roboterarme die in Fertigungsstraßen, simple und repetitive Aufgaben durchführen, die normalerweise keinerlei Lernprozesse oder Intelligenz benötigen. [7] Ein Einsatzgebiet von intelligenten Robotern sind die heimischen Räumlichkeiten. So halten Putz - und Staubsaugroboter, die mittels maschinellen Lernens unteranderem Routen optimieren, in immer mehr Wohnungen und Häusern Einzug.

Die *Bild -und Gesichtserkennung* gehören zu den Hauptgebieten des wahrnehmenden Teilgebietes der Emulation der menschlichen Intelligenz. Ziel der Bilderkennung, auch *Computer Vision genannt*, ist es, Informationen aus Bilddateien, seien es Bilderfolgen oder einzelne Bilder, zu extrahieren, Zusammenhänge oder Muster zu erkennen und diese zu interpretieren bzw. zu klassifizieren. [8] Gerade die Mustererkennung erweist sich als vorteilhaft für das Gesundheitswesen. Vermehrt werden maschinelle Lernalgorithmen zur Mustererkennung in Röntgen -und Computertomographieaufnahmen eingesetzt, um

Anomalien wie Geschwüre oder Krebszellen hervorzuheben, die sonst vom menschlichen Auge übersehen werden könnten. [9]

Künstliche Intelligenz ist, wie bereits vorangestellt, nicht einheitlich definierbar. Das schiere Ausmaß an unterschiedlichen Ausprägungen mit unterschiedlichen Fokussen der diversen Fähigkeiten, Sinneseindrücken und Ausprägungen an Intelligenzen, die es zu emulieren gilt, führt dazu, dass die künstliche Intelligenz sich schlichtweg als Teil -und Forschungsgebiet der Informatik etabliert hat.

Der Kern der meisten Anwendungen der künstlichen Intelligenz, und das unabhängig von dem Ziel der Anwendung, ist das maschinelle Lernen. Dementsprechend muss dieser Aspekt gesondert dargestellt werden.

Um das Verhalten eines Modells oder Systems nach dem eigentlichen Training einfach zu charakterisieren und gleichzeitig eine Bewertung zu geben, wurden die drei Begriffe *Underfitting*, *Overfitting* und *Generalisierung* geprägt.

Underfitting bedeutet, dass das System, meistens aufgrund von zu weniger Daten, weder auf Testdaten noch Trainingsdaten gut operiert.

Overfitting bezeichnet ein System, welches die Trainingsdaten „auswendig“ gelernt hat und somit auf diese perfekt reagiert, allerdings auf Testdaten miserable Ergebnisse liefert.

Das Ziel des Trainings eines Systems ist die *Generalisierung*: Das System liefert sowohl auf Trainingsdaten als auch auf Testdaten zufriedenstellende Ergebnisse und kann dementsprechend auch auf zuvor noch nicht bekannte Daten angemessen reagieren.

2.2 Maschinelles Lernen

Als Teilgebiet der künstlichen Intelligenz teilt das maschinelle Lernen einen Großteil der Geschichte der KI. Die größten Meilensteine traten jedoch wesentlich später ein. Die Jahre 2016 und 2017 beweisen sich als besondere Jahre in der Entwicklung vom maschinellen Lernen. Im Jahr 2016 besiegte die KI „AlphaGo“ einen Profispieler in Go. [10] Go ist ein im asiatischen Raum beliebtes Brettspiel, welches Ähnlichkeiten zu Schach aufweist, dementsprechend also ein hohes Maß an Kreativität und strategischem Denken erfordert. Nur ein Jahr später wird der Nachfolger von AlphaGo „AlphaGo Zero“ entwickelt. Der radikale Unterschied zu seinem Vorgänger ist, dass AlphaGo Zero lediglich mit den Grundregeln des Brettspiels ausgestattet ist und sich sämtliche Taktiken und Spielzüge selbst durch Modellbildung und Lernalgorithmen beibringt. Bereits nach drei Tagen schaffte dieser es, und das ohne menschliches Zutun, seinen Vorgänger in einem Spiel zu schlagen. [11]

Maschinelles Lernen umfasst also eine Auswahl an Modellen und Algorithmen, um menschliches Lernen nachzuahmen und das in Form eines Computerprogrammes festzuhalten. Hier stellt sich demnach erst einmal die Herausforderung, wie sich die Lösung dieser Aufgabe vom konventionellen Programmieren abhebt. Die Herangehensweise des klassischen Programmierens ist einerseits festzustellen, welche konkrete Aufgabe durch das Programm erfüllt werden soll und anschließend eine Folge von Regeln und Arbeitsschritten festzuhalten und in einer der zahlreichen Programmiersprachen zu implementieren. [12] Ein einfaches Beispiel hierfür wäre die Implementierung der Fakultätsfunktion. Diese ist wie folgt definiert:

$$n! = \prod_{k=1}^{n} k \ = 1 * 2 * 3 * \ldots * n$$

Dies lässt sich einfach als Programm niederschreiben. Der Nutzer wird aufgefordert eine Zahl n einzugeben, daraufhin wird eine Schleife beginnend bei 1 bis zu der eingegebenen Zahl n durchlaufen und der Wert einer Iterationsvariable bei jeder Wiederholung multipliziert.

Dies wäre als Pseudocode wie folgt ausgedrückt:

```
1.  #factorial calculation
2.
3.  Read number
4.  factorial = 1
5.  i = 1
6.  while i <= number
7.    Factorial = factorial * i
8.    i = i + 1
9.  end while
10. print factorial
```

Für dieses Problem ist also ein iteratives und konventionelles Programm leicht erstellbar. Soll ein solches Programm jedoch ein Bild auf seinen Inhalt untersuchen, sprich was darauf abgebildet ist, ist die Definition solcher Regeln und Anweisungsabfolgen praktisch unmöglich. Im maschinellen Lernen geht es also genau genommen nicht darum, wie etwas gemacht werden soll, sondern um die Lösung der Aufgabe an sich. Der Computer erhält Eingabe und Ausgabe und „erstellt“ das Programm, um die Aufgabe zu lösen selbst.

Um dieses Ziel zu erreichen, gibt es drei Hauptherangehensweisen im maschinellen Lernen die für die unterschiedlichen Zwecke prädestiniert sind. Diese sind das überwachte Lernen (supervised Learning), das nicht-überwachte Lernen (unsupervised Learning) und das bestärkende Lernen (reinforcement Learning).

Schlussfolgernd lässt sich also sagen, dass die Stärke von Algorithmen des maschinellen Lernens genau darin liegen Informationen aus Datensätzen zu extrahieren und zu lernen. Diese Datensätze definieren sich als umfangreiche Datenquelle, die die nötigen Informationen enthalten.

2.2.1 Überwachtes Lernen

Eines der drei gängigsten Verfahren des maschinellen Lernens ist das sogenannte überwachte Lernen. Das Kernmerkmal dieser Methode ist das die Menge an Informationen/Daten der Eingabe „gelabelt“ ist. Ein Label kann als eine Kategorie betrachtet werden. Ein simples Beispiel hierfür wäre die Auswertung von Bildern mit dem Ziel der richtigen Unterscheidung zwischen Tier und Mensch. So wird jedem Bild das jeweils passende Label zugewiesen, wodurch im Nachhinein korrekt evaluiert werden kann, ob das Modell die Daten richtig erkannt und kategorisiert hat.

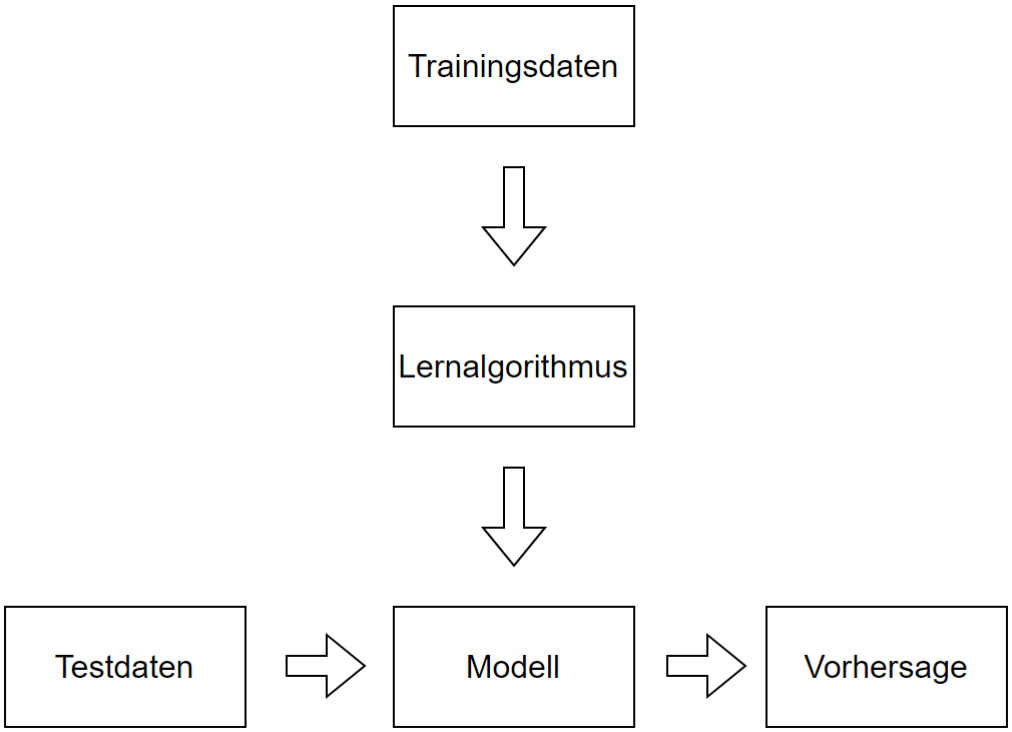


Abbildung 2: Vereinfachter Ablauf überwachtes Lernen

Im Allgemeinen ist der Prozess des Lernens in vier Schritte unterteilt. Zuallererst wird der zu untersuchende Datensatz in Trainings -und Testdaten unterteilt (für die Unterteilung gibt es mehrere Möglichkeiten, die hier der Einfachheit vernachlässigt werden). Anschließend werden die Trainingsdaten in den ausgewählten Lernalgorithmus eingespeist und somit ein Modell gebildet. Im nächsten Schritt werden nun die Testdaten, die dem Modell bis dato unbekannt sind, durch das Modell ausgewertet. Die daraus resultierende Vorhersage oder Klassifikation kann nun evaluiert werden und das Modell gegebenenfalls angepasst oder durch weitere Lernzyklen verbessert werden.

Überwachte Lernalgorithmen werden zu genau zwei Zwecken eingesetzt: *Klassifikation* und *Regression*.

Klassifikation

Die Klassifikation beschreibt die zuverlässige und korrekte Kategorisierung der bearbeiteten Daten. Das übergeordnete Ziel dieser Methode ist die Entwicklung eines Modells durch den Algorithmus. Dieses Modell soll auf eine Eingabemenge an Informationen so reagieren, dass als Ausgabe eine Information ausgegeben wird, die Rückschlüsse auf die Kennzeichnung (also genau genommen der Klassifikation) der Eingabe zulässt.[13, S. 15-16] Ausschlaggebend ist hierbei, dass die gewünschten Ergebnisse im Vorhinein bekannt sind und das Modell so direkt bewertet werden kann. Gängige Algorithmen der Klassifikationsprobleme des überwachten Lernens sind K-Nächste Nachbarn (KNN), Entscheidungsbäume, Neuronale Netze und Support Vector Machines (SVM).

Im Folgenden wird exemplarisch das Vorgehen eines Algorithmus von Entscheidungsbäumen erläutert. Hier soll jedoch gesagt sein, dass es unterschiedliche Algorithmen zur Lösung von Problemen mit Entscheidungsbäumen gibt. [14]

Grundsätzlich ist ein Baum in der Informatik als Datenstruktur anzusehen.[15] Diese Datenstruktur, und das in jeglicher Form, zeichnet sich durch die folgenden Merkmale aus:

1. Der Baum besitzt eine Wurzel.
2. Der Baum besitzt *n* Knoten.
3. Alle Knoten, abgesehen von der Wurzel, sind genau mit einem weiteren Knoten verbunden, welcher Elternteil genannt wird.
4. Zu jedem Knoten verläuft ein einzigartiger Pfad zu jedem Knoten,
5. Die letzten Knoten im Baum, die keine Kinder besitzen, werden Blätter genannt.
6. Jeder Elternknoten mit seinen Kinderknoten kann als eigenständiger Teilbaum gesehen werden.

Diese Grundlagen treffen genauso auf die Entscheidungsbäume zu, jedoch liegt der große Unterschied in der Bedeutung der Knoten und Blätter.

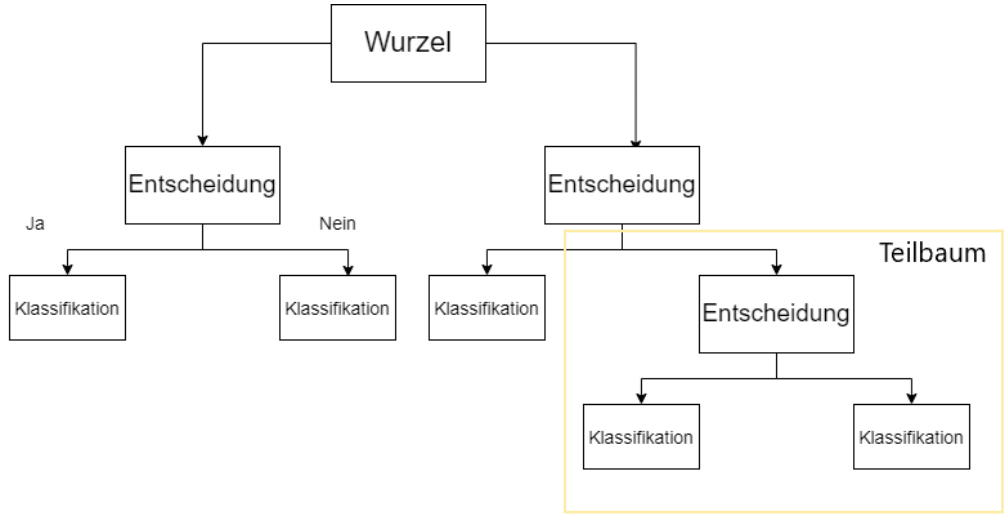


Abbildung 3: Entscheidungsbaum

Die Knoten formulieren in diesem Fall eine Bedingung bzw. eine Entscheidung, die die Eingabemenge teilt. Dies kann sich beliebig oft wiederholen, um jeden einzelnen Datenpunkt zuzuweisen. Die Blätter symbolisieren die Entscheidung, also die Einordnung und Klassifikation des betrachteten Datenpunktes. [16]

Regression

Im Gegensatz zu Klassifikationsproblemen werden Ansätze der Regression nicht zu dem Zweck verwendet klare Aussagen wie „bekommt einen/keinen Kredit“ zu tätigen, sondern um bei diesem Beispiel zu bleiben, die konkrete Summe vorherzusagen, die der Interessent potenziell bekommen könnte. Ein weiteres Einsatzgebiet ist die Vorhersage bzw. Forecasting. Hier wird mittels statistischer Methoden und maschinellem Lernen z.B. versucht Preise auf dem Aktienmarkt aufgrund historischer Datensätze vorherzusagen oder den Marktwert eines Hauses basierend auf Faktoren wie Inflation, ursprünglicher Verkaufspreis oder ähnlichem zu bestimmen.

Die lineare Regression zielt darauf ab, den Abstand von Datenpunkten zu einer gewählten linearen Funktion, also einer Geraden, möglichst zu minimieren, ohne diese dabei exakt zu treffen bzw. zu schneiden.

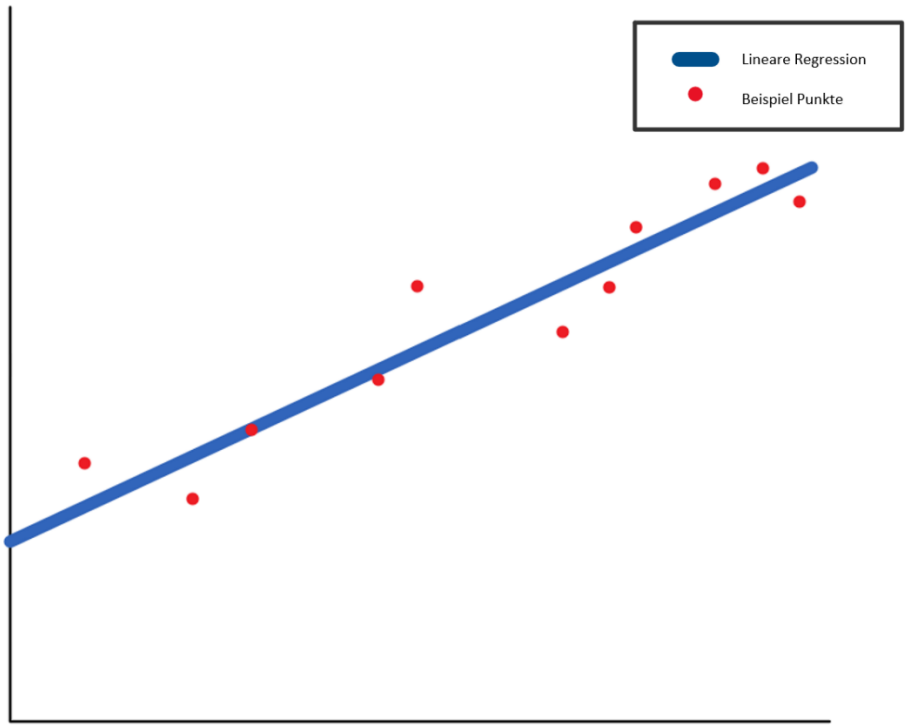


Abbildung 4: Beispiel lineare Regression

Zunächst muss also eine zu optimierende Funktion festgelegt werden. Diese lautet:

$$f_{w,b}(x) = wx + b$$

Hier definiert w einen N-dimensionalen Vektor, wobei N die Anzahl der Merkmale des Datensatzes beschreibt und $b \in \mathbb{R}$, wodurch auch $y = f_{w,b} \in \mathbb{R}$ gilt. Um nun den Abstand von zu den Datenpunkten zu verringern, muss der sogenannte Zielwert der sogenannten Kostenfunktion verringert werden. Dazu werden die Parameter w und b verändert bis der Ausdruck minimal wird.

$$\frac{1}{N}\sum_{i=1}^{N}(f_{w,b}(x_i) - y_i)^2$$

In diesem Beispiel wird als Verlustfunktion der quadratische Fehler verwendet, jedoch soll hier gesagt sein, dass genau wie in den meisten anderen Bereichen des maschinellen Lernens, eine stattliche Anzahl an weiteren mathematischen Funktionen verwendet werden kann. [17]

An dieser Stelle muss auch der wohl populärste Ansatz des maschinellen Lernens erwähnt sein: die *Neuronalen Netze*.

2.2.1.1 Neuronale Netze und das Perzeptron

Die Nachbildung menschlicher Gehirnstrukturen und deren Funktionsweise beschäftigt die Forschungswelt seit langem. Die Idee hinter den *neuronalen Netzen* lässt sich stark von den Neuronen des menschlichen Gehirns und deren Verbindung und Aktivierung inspirieren. Ein solches Netzwerk besteht aus einer Eingabe-Schicht, einer Ausgabe-Schicht und einer unbestimmten Anzahl an Zwischenschichten, die als „versteckte“ Schichten (engl. *hidden*) betitelt werden. Die Eingabe-Schicht enthält die Variablen, die das Netz als Eingabe nutzen und verarbeiten soll. Die Ausgabe-Schicht gibt die relative Wahrscheinlichkeit aller möglicher Klassen aus, wobei die Klassifikation mit der höchsten Wahrscheinlichkeit als richtige interpretiert wird. Jede Schicht ist mit der nächstfolgenden Schicht verbunden. Die Verbindungen erhalten die Gewichte, die in der Berechnung der Aktivierungsfunktion der Knoten der nächsten Schicht verwendet werden. Die Aktivierungsfunktion dient zur Bestimmung, ob der nächste verbundene Knoten für die Weiterberechnung verwendet wird. Ist das Ergebnis dieser Aktivierungsfunktion oberhalb eines Schwellwertes, wird der verbundene Knoten genutzt. Die Gewichte sind dem Modell zu Beginn unbekannt. Das Training zielt dementsprechend auf die Findung der optimalen Gewichte für das gegebene Problem ab. Hier ist es nötig das neuronale Netz in zahlreichen Iterationen wiederholt zu trainieren. Dieser hohe Aufwand resultiert in einem der genausten Vorhersagegenauigkeiten. [18, S. 253-263]

Der Ursprung der *neuronalen Netze* liegt in dem Konzept des *Perzeptron*.

Das *Perzeptron* bildet die Grundlage neuronalen Netze. Dieses Konzept geht auf Frank Rosenblatt zurück und versucht die menschlichen Gehirnstrukturen der Neuronen nachzubilden. [19] Dass das *Perzeptron* die Grundlage für *neuronale Netze bildet*, bedeutet, dass es die kleinste Einheit ist. Ein einfaches, einzelnes *Perzeptron* wird als „einlagig“ betitelt, die Anhäufung bzw. Schichtung von mehreren *Perzeptronen* wird mehrlagiges oder mehrschichtiges (engl. *multi-layer Perceptron) Perzeptron* genannt.

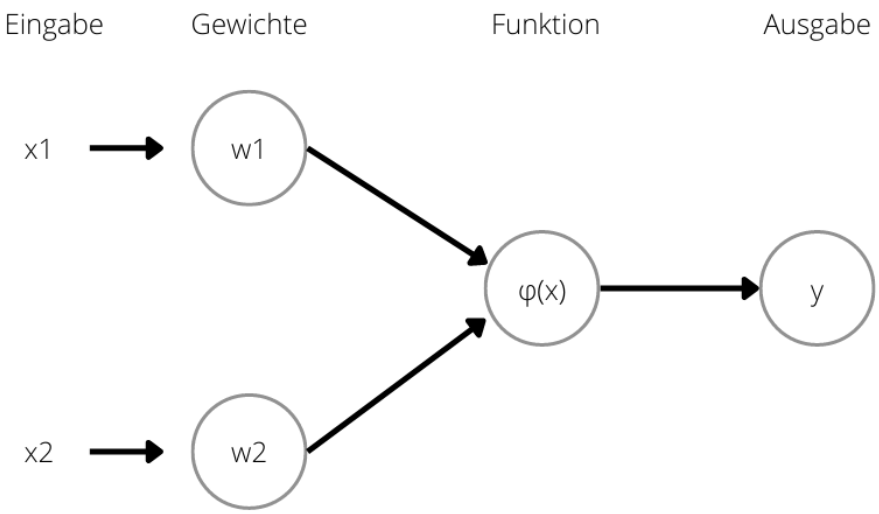


Abbildung 5: Einlagiges Perzeptron

Das *Perzeptron* in seiner einfachsten Form berechnet einen Ausgabewert, indem zunächst die Eingabewerte mit ihren dazugehörigen Gewichten multipliziert werden. In der Praxis wird hier noch ein sogenannter „Bias“ hinzugefügt. Dieser Bias definiert einen systematischen Fehler. Nach der Multiplikation werden die errechneten Werte aufsummiert. Die resultierende Summe liegt in der Regel zwischen 0 und 1. Anschließend wird im einfachsten Fall die Heaviside Funktion verwendet. Diese betrachtet nur, ob der Wert, in diesem Fall die Summe, über oder unter einem Schwellwert liegt. Ist der Wert über dem Schwellwert ist das Ergebnis 1, liegt der Wert unter dem Schwellwert so ist das Ergebnis 0.

2.2.2 Nicht-überwachtes Lernen

Das nicht-überwachte Lernen, im Englischen „unsupervised learning“ unterscheidet sich stark von dem überwachten Lernen. Der wohl größte Unterschied ist allerdings, dass die Daten nicht im Vorhinein gelabelt und kategorisiert wurden. Dies deutet auf die Problematik hin, dass die Ergebnisse nicht effizient bewertet werden können und die Nutzung von nicht-überwachten Lernalgorithmen für viele Anwendungen schlicht und ergreifend unbrauchbar ist. Ganz im Gegenteil ermöglichen diese Methoden jedoch vollkommen neue Zusammenhänge und Aussagen zu unbekannten und/oder unübersichtlichen Daten herauszustellen. Die zwei Hauptaufgaben dieser Methoden sind einerseits das *Clustering* und andererseits die *Dimensionalitätsreduktion*. [20]

Clustering

Clustering bedeutet aus dem Englischen übersetzt „Zusammenlagerung“ oder einfacher „Gruppierung“ [21]. Mit Hilfe von Clustering versucht man Zusammenhänge und Muster in Datensätzen aufgrund von geometrischer Mathematik, wie z.B. durch Berechnen der Euklidischen Distanz, abzuleiten. Die Euklidische Distanz ist im zweidimensionalen Raum wie folgt definiert: $d(a,b) = (b_1 - a_1)^2 + (b_2 - a_2)^2 = \sqrt{a^2 + b^2}$. Also im zweidimensionalen Spezialfall auch bekannt als Satz des Pythagoras. In diesem Sinne werden die Daten anhand ihrer Ausprägungen (Gewicht, Alter, Größe, Abschluss, o.ä.) in einem n-dimensionalen Koordinatensystem aufgetragen, wobei *n* der Anzahl der Ausprägungen entspricht. Anschließend werden, wenn auch in einfachem Falle visuell ersichtlich, die Distanzen aller Punkte zueinander berechnet und dann bei möglichst geringer Nähe gruppiert. So entstehen die sogenannten Cluster die Punkte, die eine möglichst geringe Distanz zueinander aufweisen gruppieren und andererseits eine möglichst große Distanz zu den anderen Clustern besitzen. [22]

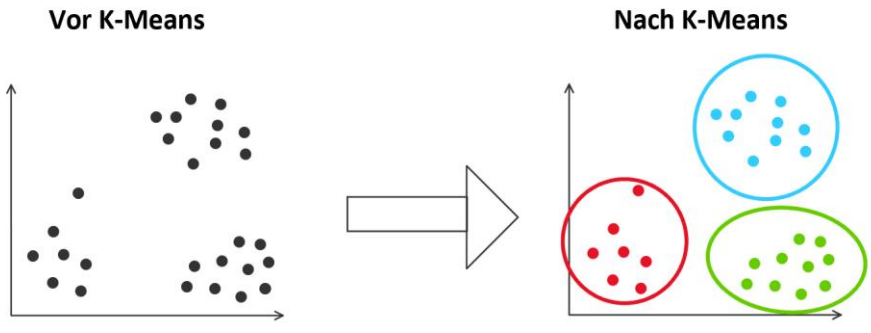


Abbildung 6: K-Means

Stellvertretend wird hier der *K-Means Clustering* Algorithmus erläutert.

Der Algorithmus beginnt mit der Auswahl von *K*. *K* bezeichnet die Anzahl an Clustern, welche sich einerseits zufällig identifizieren lässt oder durch Berechnung der Summe des quadrierten Fehlers der Datenpunkte innerhalb eines Clusters. Anschließend werden *K* zufällige Punkte ausgewählt, die zu jeweils einem Cluster gehören. Als nächstes wird der Abstand von jedem der übrigen Punkte zu den *K* Clustern berechnet und der gerade betrachtete Punkt zu dem Cluster zugeordnet der ihm am nächsten ist. Ist nun jeder Punkt einem Cluster zugeordnet, folgt die Berechnung des Durchschnittswertes (engl. *mean*) jedes Clusters. Nun wiederholt sich die Berechnung der Abstände mit dem Unterschied, dass nun der Abstand von allen Punkten zu den Clustermittelwerten relevant ist. Eine Iteration ist beendet, sobald die Punkte nicht mehr anders zuzuordnen sind. Da die Bestimmung der Gruppierungen von einer zufälligen Auswahl im ersten Schritt abhängig ist, führen mehrere Durchführungen dieser Methode auf demselben Datensatz zu unterschiedlichen Ergebnissen. [13, S. 154-163]

Dimensionalitätsreduktion

Mittels der *Dimensionalitätsreduktion* wird versucht, einen hochdimensionalen Datensatz auf einen Raum geringerer Dimension zu übertragen. Da bei diesem Vorgehen keine relevanten Informationen verloren gehen dürfen, die die grundlegende Aussage des später verwendeten Modells verändern, gilt es zunächst Attribute (Features) mit geringer Korrelation, also mit geringer Aussagekraft, zu entfernen. Ein weiterer Ansatz ist es nahe/ähnliche Features und Datenpunkte zusammenzufassen, um somit keinerlei Information zu verlieren. Das ist nicht nur hilfreich, um eine Visualisierung der Daten zu vereinfachen, sondern verringert auch die Trainingszeit und die Erklärbarkeit. Ein Ansatz der Dimensionalitätsreduktion ist die Hauptkomponentenanalyse („Principle Component Analysis“, kurz PCA). [18]

Die Hauptkomponentenanalyse basiert auf den mathematischen Konzepten der Eigenvektoren und Singulärwerten. Um die Hauptkomponenten zu bestimmen, bedarf es nur weniger Schritte. Zunächst werden für jede der betrachteten Variablen der Durchschnitt berechnet. Dieser ist nötig, um die Daten zu normalisieren. Die Normalisierung drückt sich graphisch so aus, dass der Mittelpunkt aller Datenpunkte im Ursprung des Koordinatensystems liegt.

Als nächstes wird ähnlich der Regression ein Gerade gesucht, die zwei Bedingungen erfüllt:

1. Die Gerade muss durch den Ursprung verlaufen.
2. Die Summe der Quadrate der Abstände aller Datenpunkte zu dieser Geraden müssen minimal sein.

Die Gerade, die diese Bedingungen erfüllt beschreibt die erste Hauptkomponente. Die Steigung dieser Geraden gibt Aufschluss darüber in welchem Verhältnis die in der Hauptkomponenten vereinten Variablen zueinanderstehen, genau genommen also welche der betrachteten Komponenten einflussreicher in das Ergebnis einfließt. Bereits im dreidimensionalen Raum lässt sich so leicht bestimmen, welche Variablen vernachlässigt werden können, ohne, dass das Ergebnis maßgeblich verändert wird. [23]

Der Vollständigkeit halber soll hier noch das teilüberwachte Lernen erwähnt sein. Das teilüberwachte Lernen wird durch eine Kombination des überwachten und nicht-überwachten Lernens charakterisiert. Anwendung findet es dann, wenn der Datensatz nur teilweise kategorisiert, also gelabelt, ist. Über Methoden des nicht-überwachten Lernens lassen sich Rückschlüsse von gelabelten Daten auf nicht gelabelten Daten ziehen, sodass daraus

Zusammenhänge deduziert werden können. Im Rahmen dieser Arbeit wird im Folgenden jedoch nicht weiter auf dieses Teilgebiet eingegangen.

2.2.3 Bestärkendes Lernen

Das bestärkende Lernen ist stark an dem menschlichen Lernen im Kindesalter oder dem Training von zum Beispiel Hundewelpen angelehnt. Betrachtet man das Beispiel eines Hundes. Der Hund soll Lernen sich auf Befehl hin hinzusetzen, was er natürlich zu Beginn nicht versteht, da er verständlicherweise nicht der menschlichen Sprache mächtig ist. Der Welpe reagiert auf welche Art auch immer, nur nicht, wie er soll und wird dementsprechend nicht belohnt oder sogar bestraft. Der Hund beobachtet das und merkt sich, dass er so auf die Aufforderung nicht reagieren sollte und probiert beim nächsten Mal etwas anderes zu tun. Setzt er sich nun hin wie gewünscht erhält er eine Belohnung und merkt sich das. Je öfter er für die richtige Reaktion belohnt wird umso gefestigter wird das gelernte Verhalten. In der Theorie des bestärkenden maschinellen Lernens wird der Hund, bzw. das jeweilige Programm, als Agent betitelt und das Herrchen als Umwelt. [24] Die Belohnung und/oder Bestrafung wird meist in Form von einem Punktestand realisiert. Soll ein Roboter z.B. durch ein Labyrinth finden erhält er für jeden Schritt, den der Roboter erst einmal zufällig wählt, einen Punkt und wird so mit jeder Iteration versuchen den Punktestand zu minimieren, um den effizientesten und schnellsten Weg durch das Labyrinth zu finden.

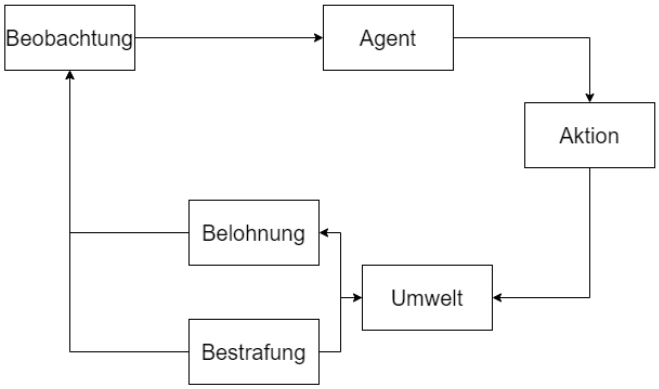


Abbildung 7: modellhafte Darstellung bestärkendes Lernen

Nachdem nun die verschiedenen Typen des maschinellen Lernens inklusive zugehöriger Algorithmen vorgestellt wurden, bleibt noch eine Frag in diesem Zusammenhang offen: Wieso verhält sich das Modell so, wie es das tut? Oftmals reicht die alleinige Qualität eines Systems nicht mehr aus, um die Nutzung zu rechtfertigen. Dies spielt oftmals dann eine Rolle, wenn sensible Daten involviert sind. Um dieser Frage nachzugehen, widmet sich der relative junge Forschungszweig der *Erklärbaren Künstlichen Intelligenz*.

2.3 Erklärbare Künstliche Intelligenz

Der Wunsch danach die Entscheidungen eines Systems der künstlichen Intelligenz, genauer des maschinellen Lernens, nachvollziehen zu können, ist kein neuer. Bereits in den siebziger Jahren wurde ein Prototyp zur Diagnostizierung bakterieller Infektionen im Blut entwickelt. Dieser Prototyp war fähig anzugeben, welcher seiner einprogrammierten Regeln einen wie starken Einfluss auf das Ergebnis hatten. [25]

Mit zunehmendem Interesse nach dem „KI-Winter“ zu Beginn der neunziger Jahre, gelang es schnelle Fortschritte im Gebiet der künstlichen Intelligenz zu erzielen. Die starke Zunahme an verwendbaren Daten spielte dabei eine ausschlaggebende Rolle. Das Aufkommen des Internet of Things, Wearables wie Smart Watches und des Ambient Assisted Livings, führen zu einem deutlichen Anstieg an (Sensor-)Daten. Allerdings ging der Schritt zunächst dahin performantere und genauere Modelle, wie die (tiefen) neuronalen Netze, zu kreieren. Die neuronalen Netze werden im Allgemeinen auch als Black Box Modelle betitelt. Diese sind charakterisiert durch eine hohe Präzision in ihren Ergebnissen, geben allerdings keinerlei Auskunft über die beeinflussenden Faktoren, die zu ihren Entscheidungen geführt haben. Weder die Modelle an sich sind erklärbar noch ihre Entscheidungen, noch können sie bis dato von dem jeweiligen Entwickler durchschaut und erklärt werden. [26]

Die immer stärkere Einbindung von künstlicher Intelligenz in das alltägliche Leben fordert im Gegenzug allerdings auch das Verständnis hinter den Entscheidungen. Gerade in Bereichen wie im Finanzwesen oder dem Gesundheitswesen können Entscheidungen nicht einfach hingenommen werden. Es muss ein gewisses Vertrauen geschaffen werden. Dies ist nicht erreichbar, wenn grundlegende Beweggründe unbekannt bleiben. [27]

Anlässlich dieses mangelnden Einblicks entstand DARPAs ‚Explainable Artificial Intelligence Program‘. [28] Dieses Programm hat das Ziel dem Trend der KI zu immer performanteren Blackbox Modellen entgegenzuwirken und den nötigen Einblick in KI-Systeme voranzutreiben. Hierzu wurde ein zwei Phasen Prozess geplant und durchgeführt. Die erste Phase umfasst die Erfassung und Evaluation der bestehenden erklärbaren Systeme. Phase 2 widmet sich der Weiterentwicklung und Integration bestehender erklärbaren Modelle in nicht erklärbare Systeme. [29]

Im Fokus des Forschungsgebiets liegt also der Kompromiss zwischen Transparenz und Performanz der Systeme. So fällt auch bei genauer Betrachtung auf, dass Algorithmen wie die bereits erklärten Entscheidungsbäume von „Haus aus“ erklärbar und transparent sind. Folgt man einem Pfad von Wurzel bis zu einem Blatt werden alle Entscheidungen bis hin zu der Klassifizierung eindeutig definiert. Ganz im Gegensatz zu der gewünschten und gegebenen Transparenz gehört es zu den langsamsten Systemen.

2.3.1 Erklärbarkeit VS Interpretierbarkeit

Aber was bedeutet Erklärbarkeit und Interpretierbarkeit genau? Wo liegt der Unterschied zwischen den beiden und welche Methoden existieren, um beide Faktoren im Fachbereich des maschinellen Lernens miteinzubeziehen?

Ein grundlegendes Verständnis darüber was eine Erklärung im ursprünglichen Sinne umfasst und wie diese Definition auf Probleme des maschinellen Lernens übertragen werden können, bilden den Grundpfeiler für das Forschungsgebiet der Erklärbaren Künstlichen Intelligenz.

Laut Duden bedeutet *erklären* so etwas wie „etwas deutlich machen“ oder „erläutern, sodass Zusammenhänge deutlich werden“. [30] Dies lässt sich also darauf runterbrechen, dass Sachverhalte so umschrieben werden müssen, dass jegliche kausale Hintergründe besprochen und, in der Regel, die Frage „Warum?“, in all ihren Ausprägungen, beantwortet ist. Dieser Vorgang gestaltet sich bereits in der Kommunikation von Mensch zu Mensch, sei es schriftlich oder verbal, als eine schwierige Aufgabe. Dementsprechend müssen die Rahmenbedingungen in diesem Sachverhalt in Hoffnung auf eine befriedigende Erklärung durch ein Programm bzw. nicht menschliches System deutlich abgesteckt werden. [31]

Genauso wichtig wie die Erklärbarkeit ist der Begriff der Interpretierbarkeit. Etwas zu *interpretieren* verhält sich Synonym zu „etwas auszulegen“, also die Bedeutung und die Beweggründe hinter Handlungen, Aussagen, Verhaltensweisen, Ergebnissen herauszustellen. Ein allseits bekanntes Beispiel ist es in der schulischen Ausbildung die Bedeutung von Musikstücken oder Gedichten zu finden. [32] Die Schwierigkeit in diesem Kontext liegt allerdings darin, dass die Interpretation, also das Herauslesen der Bedeutung individuell variiert. Je nach persönlicher Erfahrung aber auch Wissensstand oder kulturellen Hintergründen lässt sich vieles ambivalent deuten. Zusätzlich Bedarf es ein gewisses Hintergrundwissen zum jeweiligen Bereich um Inhalte, Anspielungen, oder ähnliches nachvollziehen und richtig einordnen zu können. Hierfür muss sich ein jeder nur die Interpretation eines Gedichtes vor Augen führen. Die Ergebnisse würden deutlich von Person zu Person variieren, wobei bei keinem der Ergebnisse der Interpretation gewiss sein kann, dass diese auch wirklich dem entspricht, was der jeweilige Autor beabsichtig hat.

Was bedeutet dies nun für die Erklärbarkeit und Interpretierbarkeit der künstlichen Intelligenz?

Interpretierbarkeit charakterisiert das Verständnis der kausalen Zusammenhänge zwischen Ursache und Wirkung – in dem Feld des maschinellen Lernens also der Zusammenhänge zwischen Eingabe und Ausgabe. Nun ist es das Ziel, dass auch Nutzer, die kein Domänenwissen haben, diese Schlüsse ziehen können. So sollten Fragen wie: Warum ist der Einfluss eines Features oder einer Variablen so viel stärker als das einer anderen? Was sind die Grenzen des Modells bezogen auf die Ein -und Ausgabe? Ein Modell, welches allein Aufschluss über diese Aussagen trifft, ist demnach vertrauenswürdiger als eines, welches keine Rückschlüsse zulässt. Zusätzlich ermöglicht diese Einsicht auch dem entwickelnden Ingenieur oder Entwickler deutliche Vorteile. So kann er ein Modell und die zugehörige Software besser entwickeln, trainieren, warten und vor allem debuggen. [33] Erklärbarkeit umfasst in diesem Kontext alles was Interpretierbarkeit in sich vereint, mit dem einzigen Unterschied, dass es in Sachen Transparenz der Abläufe innerhalb des Modells noch tiefer geht und genauer definiert und Menschen verständlich wiedergibt.

Es werden drei Arten von Transparenzen unterschieden:

1. **Modell Transparenz**: Bietet die Möglichkeit einzusehen, wie das Modell trainiert wurde und wie die entsprechenden Koeffizienten z.B. einer Kostenfunktion ermittelt wurden.
2. **Design Transparenz**: Die Erklärung was die Auswahl von Architekturen, Algorithmen und Parameter beeinflusst hat.
3. **Algorithmische Transparenz**: Stellt die Faktoren, Variablen, Gewichte heraus, die die Entscheidung des verwendeten Algorithmus beeinflusst haben.

Eine weitere Darstellung der Transparenz von Systemen der künstlichen Intelligenz wird beschrieben mittels drei unterschiedlicher Level von Transparenz: [34]

1. **Implementation**: Im ersten Level ist die Beziehung von Eingabe zu Ausgabe bekannt, sowie alle Parameter.
2. **Spezifikation**: Das zweite Level umfasst alle Informationen, die zur endgültigen Umsetzung und Training geführt haben. Hierzu gehören unter anderem: der verwendete Datensatz, Hyperparameter, Kostenfunktion, Verlustfunktion.
3. **Interpretierbarkeit**: Die dritte und letzte Stufe soll Auskunft über die Mechanismen an sich liefern, die z.B. Prinzipien umfassen die Hinweise auf die Ausgabeschicht geben. Dieser Aspekt ist allerdings bis dato noch nicht umgesetzt

Die Einhaltung oder auch die Einführung der Transparenzen hat deutliche Vorteile in diversen Bereichen. So lassen sich einerseits bessere Entscheidungen treffen und andererseits Algorithmen besser optimieren. Zu wissen, wie das Modell entscheidet unterscheidet sich schließlich stark davon, ob der Entwickler die Hintergründe zu diesen Entscheidungen kennt. Gerade dieses Hintergrundwissen führt andererseits auch zu einem gestärkten Vertrauen in diese Systeme. [33, S. 18-21]

2.3.2 Kategorien von KI-Systemen

Für das folgende Verständnis sind auch die drei Kategorien Whitebox, Blackbox und Glassbox notwendig, die die unterschiedlichen Modelle und Systeme in der künstlichen Intelligenz charakterisieren.

White Box?	Model Class	Properties that Increase Interpretability					Task		Performance Rank	
		Expl.	Linear	Monotone	Non-Interactive	Regul.	Regr.	Classif.	Regr.	Classif.
✓	Linear Regression	●	●	●	●	●	✓	✗	6	
✓	Regularized Regression	●	●	●	●	●	✓	✓	7	8
✓	Logistic Regression	●	●	●	●	●	✗	✓		5
✓	Gaussian Naïve Bayes	●	●	●	●	●	✗	✓		7
✓	Polynomial Regression	●	●	●	●	●	✓	✓	2	
✓	RuleFit	●	●	●	●	●	✓	✓	8	
✓	Decision Tree	●	●	●	●	●	✓	✓	5	3
✓	k-Nearest Neighbors	●	●	●	●	●	✓	✓	9	6
✗	Random Forest	●	●	●	●	●	✓	✓	3	4
✗	Gradient Boosted Trees	●	●	●	●	●	✓	✓		2
✗	Multi-layer Perceptron	●	●	●	●	●	✓	✓	1	1

Abbildung 8: Eigenschaften und Klassifizierung von ML Modellen [33, S. 124]

Whitebox

Die sogenannten White-Box Modelle sind „von Haus aus“ verständlich, erklärbar und interpretierbar. Das bedeutet, dass das Verhalten der Modelle klar ist, wie es zu den Ergebnissen kommt, also wie Eingabe auf Ausgabe abgebildet wird und welche Variablen den stärksten Einfluss haben. Sie entsprechen der ersten Stufe, der im vorherigen Abschnitt erläuterten, drei Stufen der Transparenz. Gängige Algorithmen dieses Typs sind lineare und logistische Regression, Entscheidungsbäume oder k-Nächste Nachbarn. Obwohl diese Vorgehensweisen das menschliche Verständnis im Fokus haben, muss erwähnt sein, dass diese weniger performant sind als Vertreter der Blackbox Kategorie.

Blackbox

Genau das Gegenteil zu den Whitebox Modellen stellen die Vertreter der Blackbox Modelle dar. Obwohl die Präzision und die Performanz in diesem Ansatz deutlich über der von Whitebox Modellen liegen, ist es, zumindest von Hause aus, unmöglich Einblick in die Algorithmen selbst zu bekommen. Es lässt sich (so gut wie) keine Aussage über die internen Strukturen, Verhaltensweise und Grundlage für die Vorhersagen treffen. Die Modelle, die diese Charakteristiken geprägt haben, sind unter anderem die künstlichen Neuronalen Netze. Diese bestehen aus einer Inputschicht, einer Vielzahl an versteckten Schichten und einer Ausgabeschicht (vgl. Abbildung 9). Die verstecken Schichten werden häufig als schwarzer Kasten (engl. *black box*) dargestellt, da keine qualitative Aussage darüber getroffen werden kann, was genau in ihnen geschieht. [35]

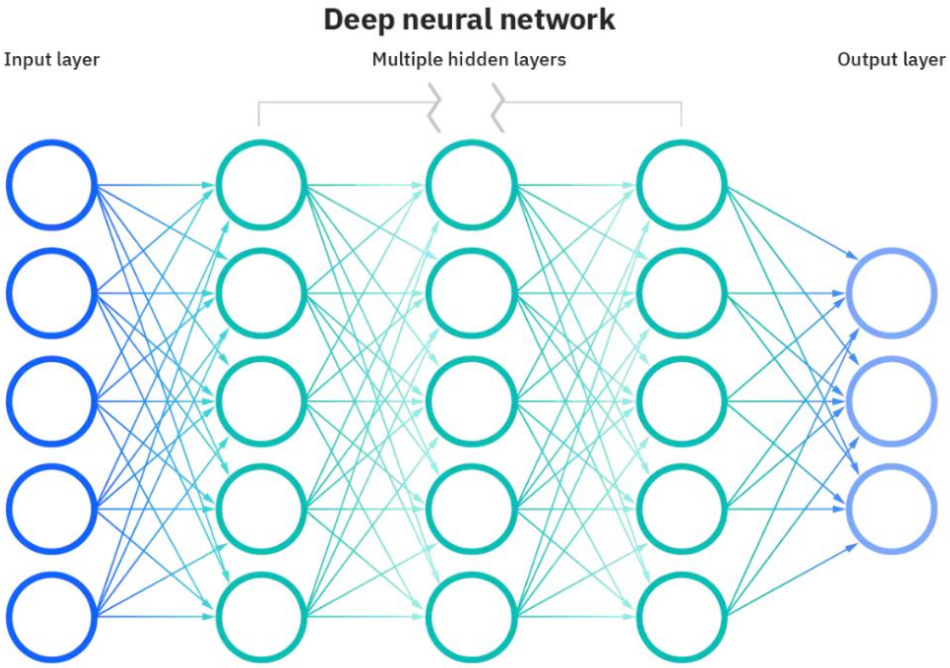


Abbildung 9: Künstliches Neuronales Netz [36]

Glassbox**:**

Glassbox, lose übersetzt gläserner Kasten, wird einerseits Synonym zu Whitebox verwendet. Andererseits wird damit das Ergebnis von Blackbox Modellen, die durch zusätzliche Algorithmen und Funktion erklärbar gemacht wurden, betitelt. Um dieses Ziel zu erreichen kann

- die Anzahl der Ausprägungen der Eingabeschicht reduziert
- die Beziehung zwischen Ausprägungen und Vorhersagen als monoton beschränkt
- die relevanten Ausprägungen des Eingabestromes einer Instanz eines Modells hervorgehoben
- das Ergebnis des Blackbox Modells mittels eines Whitebox Modells wie Entscheidungsbäumen approximiert und interpretiert

werden. [37]

2.4 Künstliche Intelligenz im Gesundheitswesen

Die künstliche Intelligenz bzw. das maschinelle Lernen hat in den letzten Jahren ein enormes Wachstum erfahren. So wurden Applikationen, die das maschinelle Lernen inkorporieren für Bereiche des E-Commerce, des Rechts, der Verwaltung, Digitaler Assistenzen, sowie des Verkehrs und der Mobilität etabliert. [38]

Auch wenn alle diese Bereiche ihre eigenen (rechtlichen) Einschränkungen und Prämissen haben, ist der Einsatz „neuartiger“ Technologie gerade im Gesundheitswesen besonders reglementiert.

2.4.1 Einsatzgebiete im Gesundheitswesen

Speziell Anwendungsfälle des Gesundheitswesens können von der künstlichen Intelligenz profitieren. So lassen sich die Daten von Wearables wie Smart Watches genutzt werden, um Indikationen über den Status einer Erkrankung oder Hinweise zum Aufsuchen eines Arztes zu geben. Weiterhin kann die enorme Menge an Daten dafür genutzt werden individuelle Medikation für z.B. an Diabetes erkrankten Personen zu optimieren. Abgesehen von direkten diagnostischen oder therapeutischen Verfahren können Modelle auch dafür verwendet werden, medizinische Geräte zu optimieren und Wartungen vorherzusagen oder zu vereinfachen. Selbst bei dieser geringen Anzahl an Beispielen wird schnell deutlich, welche Vorteile der Einsatz von maschinellem Lernen im Gesundheitswesen haben kann. Zusammengefasst führt der Einsatz zu besserer Diagnose und Therapie von Patienten als auch zu einer Verbesserung für die Gesundheitswirtschaft, die so ihre Wertschöpfungskette optimieren und damit wiederum ihren Kunden, den Patienten, effizienter helfen können. [39]

Der Fokus im Gesundheitswesen liegt auf der Erhaltung von Menschenleben und der Versorgung und Verbesserung von gesundheitsfördernden Maßnahmen für Menschen. Daher ist die Schaffung von Vertrauen zu einer maschinellen Unterstützung, abseits von der ohnehin schon intensiven Überprüfung dieser, von großer Wichtigkeit. Dieses Vertrauen kann nur geschaffen werden, wenn die verwendete Technologie durch und durch verstanden wurde und das im Idealfall nicht nur von Experten mit dem entsprechenden Domänenwissen.

Abseits von den bereits erwähnten Implikationen des Bedarfs nach Vertrauen und regulatorischer Einflüsse spielen auch ethische Aspekte und die bereits erwähnten Transparenzen eine besondere Rolle.

2.4.2 Das Beispiel COVID-19 Pandemie

Da sich die folgende Ausarbeitung auf Daten der Covid-19 Pandemie stützt, werden zunächst Kernbegriffe der Pandemie erläutert.

Der Unterschied zwischen einer Pandemie und einer Epidemie ist die räumliche Variable. Sowohl Epidemie also auch Pandemie berücksichtigen das zeitlich begrenzte aber stark angehäufte Auftreten einer Infektionskrankheit. Allerdings ist diese bei einer Pandemie nicht örtlich beschränkt, sondern die ganze Welt betreffend. [40]

Die Namensgebung des im allgemeinen Sprachgebrauch verwendeten Corona Virus bezieht sich auf die Form des Virus unter dem Mikroskop. Dieser ähnelt einer Krone (engl. *corona)*. Die vollständige Bezeichnung des Corona Virus lautet SARS-CoV-2 und ist ein Akronym der

Bezeichnung „severe acute respiratory syndrome coronavirus type 2“. Eine ähnliche Variante trat bereits 2003 das erste Mal auf. Prädominante Symptomatiken dieser Erkrankung ähneln der einer Grippe, werden allerdings durch Atemnot und Husten ergänzt. Ungefähr 20% der Erkrankten benötigten zusätzliche Sauerstoffzufuhr. [41] Der signifikante Differenz zwischen SARS und SARS-CoV-2 liegt darin, dass die Symptome einer SARS Infektion vor der potenziellen Übertragung auf Dritte auftreten. Dadurch konnten Infektionsketten leichter durchbrochen werden. Bei SARS-CoV-2 ist dies allerdings nicht der Fall. Die Übertragung der SARS-CoV-2 Viren findet über Aerosole statt. Aerosole bezeichnen die Mischung aus flüssigen Teilchen (z.B. Speichel) in einem Gasgemisch (z.B. Sauerstoff). Der Corona Virus wird also über im Speichel befindlichen Erregern beim Ausatmen, Husten oder Niesen freigesetzt. [42]

Obwohl die Infektion mit dem Covid-19 Virus in seiner Schwere stark von Individuum zu Individuum variieren kann, erkranken Männer statistisch gesehen häufig schwerer an SARS-CoV-2. Auch die Letalität ist bei männlichen Patienten deutlich höher als bei Frauen. [43]

Unabhängig vom Geschlecht treten jedoch folgende Symptome am häufigsten auf:

- Husten
- Fieber
- Schnupfen
- Störung des Geruchs -und Geschmackssinns

Weitere weniger häufig auftretende Symptome sind Halsschmerzen, Kopf -und Gliederschmerzen, seltener Atemnot/Atembeschwerden und Befall des Magen -und Verdauungstraktes. Schwere pneumologische, also die Lunge und Atmung betreffende, Folgen bis hin zu Lungenversagen treten in Deutschland weniger stark ausgeprägt auf. Das Ableben von nur 1,8% der an Covid-19 erkrankten Personen konnte tatsächlich auf die Infektion zurückgeführt werden. [44]

Durch die weltweite Ausbreitung der Infektionskrankheit SARS-CoV-2 und die strategische Eindämmung und Datenerhebung bezüglich dieser, konnte eine riesige Menge an Daten erfasst werden. Diese Datenmenge kann nun in der Retrospektive einen guten Einblick in die Sachlage, den Verlauf und die Effektivität der Maßnahmen zur Bekämpfung der Ausbreitung liefern. Dieser Grundbaustein könnte ausschlaggebend sein, um in einem zukünftigen Notstand effektivere und effizientere Entscheidung treffen zu können.

3 Methoden und Implementierung der Erklärbaren Künstlichen Intelligenz

Dieses Kapitel thematisiert nun die Implementierung fünf verschiedener Modelle des maschinellen Lernens und erläutert daran das Vorgehen der Modellerstellung und Implementierung. Zusätzlich wird eine Auswahl an Methoden vorgestellt, welche das Ziel haben, die Erklärbarkeit und die Interpretierbarkeit der Modelle zu verbessern.

Die Modelle, die hier verwendet werden, sind:

- (Multi-) Lineare Regression, da sie eine etablierte und zuverlässige Methode des Maschinellen Lernens bietet.
- Logistische Regression, beweist sich als präziser Whitebox Klassifikator
- Entscheidungsbäume (Klassifikation), da Entscheidungsbäume die von Hause aus erklärbarsten Modellen darstellen.
- Random Forest (Regression), bauen auf den Entscheidungsbäumen auf und beheben einige Mankos der Entscheidungsbäumen. Random Forest Algorithmen bilden hier den Kontrast zu den Entscheidungsbäumen.
- Support Vector Machines als Klassifikator und als Regressor, werden in dieser Arbeit als Beispiel für Blackbox Modelle verwendet.

In den vier dafür erstellten „*Jupyter Notebooks*“ befindet sich die Implementierung sowie die Dokumentation der Modelle und deren Erstellung.

Der zur Auswertung und Veranschaulichung verwendete Datensatz trägt den Namen „Symptoms and COVID Presence (May 2020 data)[1]“. Die gesammelten Daten entstammen der „World Health Organization“ und sollen hauptsächlich zur binären Klassifikation genutzt werden. Der Datensatz enthält anonymisierte Daten einer Vielzahl an Personen. Diese geben Aufschluss über verschieden Symptome und Begebenheiten im Zusammenhang mit SARS-CoV-2 und einem Indikator, der angibt, ob die jeweilige Person nachweislich an Covid-19 erkrankt ist.

Zur Erstellung der Implementierung und der Modelle wird *Python*[2], *Anaconda*[3] und *Jupyter Notebooks*[4] verwendet. Die wichtigsten mit Anaconda verwalteten Bibliotheken sind:

- *NumPy*[5]: Eine Sammlung an mathematischen Funktionen und Möglichkeiten zur numerischen Berechnung von Vektoren und Matrizen
- *Pandas*[6]: Ein Tool zur Analyse und Manipulation von Datensätzen
- *Scikit-Learn*[7]: Umfassende Bibliothek des maschinellen Lernens, die unter anderem Klassifikation, Regression und Clustering Algorithmen bereitstellt.
- *Matplotlib*[8]: Bibliothek zur Visualisierung von mathematischen Daten.

[1] https://www.kaggle.com/datasets/hemanthhari/symptoms-and-covid-presence
[2] https://www.python.org/
[3] https://www.anaconda.com/
[4] https://jupyter.org/
[5] https://numpy.org/
[6] https://pandas.pydata.org/
[7] https://scikit-learn.org/stable/
[8] https://matplotlib.org/

3.1 Aufbau Modellentwicklung und Ausprägungen des Datensatzes

Im folgenden Abschnitt wird der grundsätzliche Aufbau der Implementierungen und der Modellentwicklung beschrieben. Anschließend wird der verwendete Datensatz mitsamt seinen Ausprägungen erläutert.

3.1.1 Modellaufbau

Die vier verschiedenen Modelle werden folgendermaßen aufgebaut:

0. **Einbinden der notwendigen Bibliotheken**
   Der erste Schritt in der Implementierung beinhaltet das Laden und Einbinden der für das aktuelle Modell benötigten Bibliotheken. Im maschinellen Lernen wird meist zunächst *NumPy*, *Pandas*, *Scikit-Learn* und *Matplotlib* eingebunden. Diese sind nötig, um mit den vorliegenden Daten effizient arbeiten zu können und die Ergebnisse anschließend zu visualisieren.

1. **Laden des Datensatzes**

   Nachdem die nötigen Bibliotheken eingebunden sind, wird der Datensatz zur Verwendung in das Programm geladen. Für die Verarbeitung diverser Dateitypen bietet *Pandas* zahlreiche Funktionen an, wie auch für die verwendete Datei im CSV-Format.

2. **Untersuchen des Datensatzes**
   Bevor Verfahren des maschinellen Lernens auf den Datensatz angewendet werden, bietet sich zunächst die „Exploratory Data Analysis“ an. [45] Diese beinhaltet in der einfachsten Form das Begutachten der Daten, deren Aufbau und Vollständigkeit. Zusätzlich besteht die Möglichkeit statistische Methoden anzuwenden, um z.B. Ausreißer in den Daten entdecken zu können. Ein weiterer Ansatz ist es, Teile der Daten zu visualisieren und graphisch aufzubereiten, um weiteren Aufschluss über die Datenlage zu erhalten.

3. **Transformation**
   Bevor die Daten verarbeitet werden können, müssen diese erst einmal für die Maschine verständlich gemacht werden. Enthält der Datensatz z.B. Worte wie „Ja“ und „Nein“ müssen diese in binäre Werte transformiert werden, damit diese verwendet werden können. Weiterhin sollten numerische Werte oder Daten im Vorhinein vereinheitlicht werden, um eine konsistente Verarbeitung zu ermöglichen und unrealistisch große oder kleine Werte eliminiert werden.

4. **Modellierung**
   Anschließend müssen die Ziel -und Feature Variablen getrennt, der Datensetz in Trainings -und Testdatensatz aufgeteilt und das Modell auf die Daten angewandt werden.

5. **Evaluation**
   Die Ergebnisse, also genau genommen die Vorhersagen bzw. Klassifikation oder Regression, können nun mittels verschiedener statistischer und mathematischer Verfahren evaluiert werden und so auch die Performanz des gesamten Modells bewertet werden.

6. **Visualisierung**
   Abschließend können die verschiedenen Ergebnisse (je nach Modell) oder in diesem Fall zusätzliche Methoden, die die Erklärbarkeit erhöhen, graphisch dargestellt werden.

3.1.2 Ausprägungen des Datensatzes

Der Datensatz „Symptoms and COVID Presence“ enthält 21 Spalten. Die ersten 20 Spalten beschreiben jeweils eine Ausprägung, in diesem Fall ein Symptom oder eine Tätigkeit, die für oder gegen eine potenzielle Erkrankung sprechen. Die letzte Spalte erfasst die diagnostizierte Infektion mit SARS-CoV-2. Die Ausprägungen sind:

- Breathing Problem: Treten Atembeschwerden auf?
- Fever: Hat der Proband Fieber?
- Dry Cough: Hat der Proband Husten ohne Auswurf?
- Sore throat: Verspürt der Proband Schmerzen beim Schlucken?
- Running Nose: Läuft die Nase des Probanden?
- Asthma: Liegt eine Asthmaerkrankung vor?
- Chronic Lung Disease: Liegt eine chronische Erkrankung der Atemwege vor?
- Headache: Treten Kopfschmerzen auf?
- Heart Disease: Leidet der Proband an einer Erkrankung des Herzens?
- Diabetes: Leidet der Patient an Typ 1 oder Typ 2 Diabetes?
- Hyper Tension: Hat der Patient Bluthochdruck?
- Fatigue: Verspürt der Proband verstärkte Müdigkeit?
- Gastrointestinal: Treten Probleme im Magen -und Verdauungstrakt auf?
- Abroad travel: Ist der Proband kürzlich gereist?
- Contact with COVID Patient: Hatte der Proband Kontakt zu einer mit Sars-CoV-2 infizierten Person?
- Attended Large Gathering: Hat der Proband an einer Versammlung teilgenommen?
- Visited Public Exposed Places: Hat er öffentliche Orte besucht?
- Family working in Public Exposed Places: Arbeiten Familienangehörige an öffentlichen Orten oder Einrichtungen?
- Wearing Masks: Wurde eine Maske getragen?
- Sanitization from Market: Wurde (Hand-) Desinfektionsmittel genutzt?
- COVID-19: Wurde Sars-CoV-2 diagnostiziert?

Alle Fragen zu den Symptomen und Verhaltensweisen können mit „Ja“ und „Nein“ beantwortet werden. Dementsprechend ist der Datensatz prädestiniert dafür durch Klassifikationsmodelle evaluiert zu werden. Zur weiteren Veranschaulichung einiger Methoden zur Verbesserung der Erklärbarkeit werden dennoch auch Regressionsmodelle auf den Datensatz angewandt.

3.2 (Multi-) Lineare Regression

Der zuvor beschriebene Aufbau der Implementierungen wird nun zunächst am Beispiel der Linearen Regression in seiner Gesamtheit dargestellt und wesentliche Bestandteile erläutert. Anschließend wird ein erstes Verfahren der erklärbaren künstlichen Intelligenz eingeführt und am Modell erklärt.

Vorab sei jedoch erwähnt, dass die „multiple“ lineare Regression sich nur in der Anzahl der unabhängigen Variablen, also der Ausprägungen von der konventionellen linearen Regression unterscheidet.

3.2.1 Explorative Datenanalyse

Bevor ein Algorithmus auf einen Datensatz angewendet werden kann, sollten die vorliegenden Daten untersucht und begutachtet werden. Hier stellen sich neben dem allgemeinen Aufbau und den tatsächlichen Werten auch weitere Fragen wie nach z.B. der Vollständigkeit der Daten, der Verteilung und möglicher Ausreißer. Zusätzlich lassen sich so bereits im Vorhinein Rückschlüsse auf Beziehungen und Auswirkungen ziehen.

Hierfür bietet das Tool *Pandas* einige Möglichkeiten zur Darstellung, Analyse und Manipulation der Daten. Nach Einlesen des Datensatzes liefert der Aufruf der Pandas Funktion „.head()“ die ersten fünf Zeilen des Datensatzes als überschaubare Tabelle.

	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	Headache	Heart Disease	Diabetes	...
0	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes	...
1	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	No	No	...
2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	...
3	Yes	Yes	Yes	No	No	Yes	No	No	Yes	Yes	...
4	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	...

Abbildung 10: Auszug aus Datensatz nach Aufruf von .head()

Bereits aus den ersten fünf der insgesamt 5434 Zeilen und insgesamt 21 Spalten lassen sich Informationen ziehen, die in erster Linie Aufschluss über die Weiterverarbeitung geben. So stehen in jeder Spalte, die die Features repräsentieren, String Literale.

Hier kommt das „One-Hot Encoding“ zum Tragen. Das „One-Hot Encoding“ wird dann verwendet, wenn eine numerische Datengrundlage benötigt wird. Unterschieden wird hierbei zwischen ordinaler und nominaler Kodierung. [46] Eine nominale Kodierung erzeugt für jede Antwortmöglichkeit eine weitere Variable, also in der obigen Tabelle eine weitere Spalte. Ein Resultat wäre für die erste Spalte „Breathing Problem“: „Breathing Problem-Yes“ und „Breathing Problem-No“. Das würde den vorliegenden Datensatz künstlich aufblähen und unübersichtlich machen. Die ordinale Kodierung, die für jede Antwortmöglichkeit eine aufsteigende Zahl vergibt, ist hier angebrachter. So wird aus „Yes“ eine 1 und aus „No“ eine 0.

	Breathing Problem	Fever	Dry Cough	Sore throat	Running Nose	Asthma	Chronic Lung Disease	Headache	Heart Disease	Diabetes	...
0	1	1	1	1	1	0	0	0	0	1	...
1	1	1	1	1	0	1	1	1	0	0	...
2	1	1	1	1	1	1	1	1	0	1	...
3	1	1	1	0	0	1	0	0	1	1	...
4	1	1	1	1	1	0	1	1	1	1	...

Abbildung 11: Datensatz nach Encoding

Nach diesem Schritt ist der Datensatz für die Maschine bzw. in diesem Fall das Modell verständlich. Überprüft werden sollte allerdings, ob fehlende Einträge oder gar Leerzeilen bzw. Leerspalten in den Daten existieren. Dies ließe sich bei der verwendeten Datengrundlage auch händisch überprüfen, aber auch hierfür gibt es eine nützliche Funktion in *Pandas*: „isNull(). isNull() überprüft, ob in einer der Kategorien (also Spalten) leere Zellen vorhanden sind. Der Datensatz ist vollständig gefühlt, sodass für jede Kategorie „false“ zurückgegeben wird. Wäre dies nicht der Fall, wäre eine Herangehensweise beispielsweise bei mehr fehlenden Einträgen in einer Kategorie die Zeile komplett zu löschen. Bei weniger fehlenden Zellen könnten die fehlenden Einträge durch den Mittelwert der Spalte ersetzt oder durch den Einsatz eines zusätzlichen Modells oder Clustering-Algorithmen angenähert werden.

Der Aufruf der *Pandas* Funktion „.describe()“ erzeugt einen Bericht mit verschiedenen statistischen Auswertungen.

	Breathing Problem	Fever	Dry Cough	Sore throat
count	5434.000000	5434.000000	5434.000000	5434.000000
mean	0.666176	0.786345	0.792602	0.727457
std	0.471621	0.409924	0.405480	0.445309
min	0.000000	0.000000	0.000000	0.000000
25%	0.000000	1.000000	1.000000	0.000000
50%	1.000000	1.000000	1.000000	1.000000
75%	1.000000	1.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000

Abbildung 12: Ausschnitt des Berichts nach Aufruf von .describe()

Abbildung 12 gibt Auskunft über die folgenden Kennwerte der einzelnen Spalten des Datensatzes:

- Count: Anzahl an validen Werten in der Kategorie
- Mean: Durchschnitt der Werte
- Std: Standardabweichung der Werte
- Min: Kleinster auftretender Wert
- Max: größter auftretender Wert
- 25% - 75%: relativer Anteil der **unterschiedlichen** Werte <= 25/50/75%

3.2.2 Modellierung

Nach der Explorativen Datenanalyse und etwaiger potenzieller Anpassungen und Manipulationen des Datensatzes folgt die Modellierung.

Hierfür werden zu Beginn die Eingangsvariablen von der Zielvariablen getrennt. Dies resultiert in einer Matrix *X* und einem Vektor *y*. *X* enthält nun alle Features des Datensatzes abgesehen der Ausprägung „COVID-19“. Diese Ausprägung wird separat im Vektor *y* gespeichert und dient somit gleichzeitig als gewünschtes Ergebnis als auch als Label.

Nach der Aufteilung wird der Datensatz in Trainings -und Testdaten aufgeteilt. Das Lernen und Testen auf ein und derselben Datenlage würde das *Overfitting* nur begünstigen und zu keinem wirklichen Lernerfolg führen. Hierfür wird im *Jupyter Notebook* die Funktion „train_test_split()“ verwendet. Die Funktion erhält als Übergabeparameter die Matrix *X*, den Vektor *y* und den prozentualen Anteil, den der Testdatensatz final haben soll. Der prozentuale Anteil wurde bei 20% angelegt. Dadurch werden 80% des Datensatzes zu Trainingszwecken verwendet und die restlichen 20% zum Testen des Modells reserviert. Auch für das Aufteilen des Datensatzes gibt es unterschiedliche Möglichkeiten. Die am häufigsten verwendete ist die *Kreuzvalidierung*. [47] Der Ablauf ist analog zu der beschriebenen Methode, allerdings durchläuft das Modell mehrere Iterationen des Trainings und der Validierung. Bei jeder Iteration werden unterschiedliche Teile des Datensatzes zu Trainings -und Testzwecken verwendet.

Anschließend wird ein lineares Modell mit Hilfe der *Scikit-Learn* Bibliothek instanziiert und mit den Trainingspartitionen des Datensatzes trainiert. Die lineare Regression in der *Scikit-Learn* Bibliothek versucht standardmäßig eine lineare Funktion zu finden, zu welcher die Summe der Quadrate der Abstände der Datenpunkte zu der linearen Funktion minimal ist. Mathematisch ist dies wie folgt ausgedrückt:

$$\min_{w} \|xw - y\|_2^2$$

Hier gilt:

- $\min_{w}$: kleinster Wert für Gewicht w
- xw: Summe des Datenpunktes und dem Gewicht
- y: Vorhersage

3.2.3 (Multi-) Lineare Regression im Detail

Die zuvor beschrieben lineare Regression ist auf ersten Blick nicht von einer üblichen linearen Funktion zu unterscheiden. Daher wird in der Statistik und im Bereich des maschinellen Lernens folgende mathematische Definition verwendet:

$$y = \beta_0 + \beta_1 X + e$$

Wird jedoch eine Datengrundlage verwendet, die nicht eine einzelne Variable auf eine weitere abbildet, sondern eine Vielzahl an Features oder unabhängige Variablen so wird der Term um diese erweitert. [48]

$$y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n + e$$

Wobei gilt:

- y: Der vorhergesagte Wert
- β_0: y-Achsenabschnitt
- $\beta_1 X_1$: Multiplikation aus dem zugehörigen Gewicht und der unabhängigen Variablen
- e: Modellfehler

Ziel der beiden Herangehensweisen ist es, die Gewichte so anzupassen, dass der Abstand zwischen den vorhergesagten Werten y und den tatsächlichen Werten minimal wird. Hierfür wird repräsentativ die vorgestellte Methode der kleinsten Quadrate verwendet.

Aufgrund der Tatsache, dass die Zielvariable nur die zwei Ausdrücke „Ist an Covid-19 erkrankt“ und „Ist nicht an Covid-19 erkrankt“ abbildet, führt der Einsatz einer linearen Regression nicht zu zufriedenstellenden Ergebnissen.

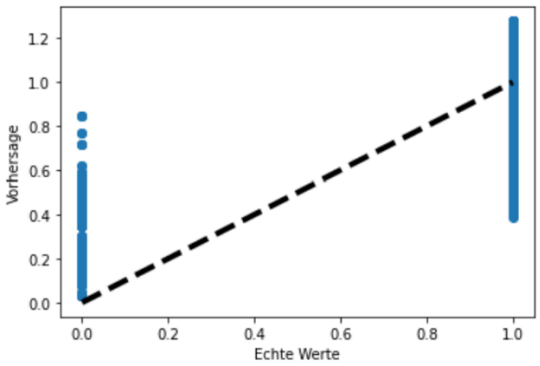


Abbildung 13: Visualisierung der linearen Regression

Der Graph wird durch den kleinsten und größten Wert, also 1 und 0, als Start -und Endpunkt approximiert.

Das Ergebnis der linearen Regression für den verwendeten Datensatz lässt sich leicht interpretieren. Jedoch fällt auf, dass ein anderer Algorithmus verwendetet werden muss, um ein sinnvolles Modell zu bilden. Obwohl die Zielvariable mit „1“ und „0“ gelabelt ist, kann die Vorhersage zwischen 0,0 und ca. 1,25 liegen. An dieser Stelle wäre eine Optimierungsmöglichkeit, die Vorhersage bei Werten kleiner 0,5 auf 0 zu setzen und bei Werten über 0,5 auf 1 aufzurunden. Für das Ergebnis wäre dann eine Darstellung einer linearen Funktion allerdings nicht mehr sinnvoll.

3.2.4 Evaluation

Für die Evaluation bieten sich an dieser Stelle der „*Mean Absolute Error*“, der „*Mean Squared Error*“ [49] und die „*Accuracy*“. [33, S. 81]

$$MAE = \frac{1}{n}\sum_{j=1}^{n} |y_j - \hat{y}\,|$$

$$MSE = = \frac{1}{n}\sum_{j=1}^{n} ({y_j}^2 - \hat{y}^2)$$

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n}$$

MAE berechnet den durchschnittlichen Fehler aus der Differenz zwischen der Vorhersage $\hat{y}$ und dem echten Wert y_j. *MSE* findet besonderen Anklang in Problemstellung, in denen gerade große Fehler besonders unerwünscht sind, da diese durch Quadrierung der Vorhersage und des echten Wertes besonders stark ins Gewicht fallen. Der sogenannte *Accuracy*-Score besticht als eine der simpelsten Möglichkeiten der Evaluation von Modellen, da lediglich der prozentuelle Anteil der korrekten Vorhersagen, Positive sowie negative, gegenüber aller Vorhersagen kalkuliert wird.

Tabelle 1: Ergebnisse der Evaluations Metriken für die (multi-) lineare Regression

Mean Absolute Error	Mean Squared Error	Accuracy
0,20	0,07	57,60%

Die Ergebnisse der Metriken bestätigen erneut, dass die lineare Regression nicht die optimale Wahl für das vorliegende binäre Problem ist. Die Betrachtung der Vorhersage -und wahren Werte, die beide immer zwischen 0 und 1 liegen, legitimiert den geringen *MSE* von 7%. Ein *Accuracy*-Score, zu Deutsch „Genauigkeit“, von nicht mal 60% sagt aber aus, dass das Modell in ungefähr 40% der Fälle die falsche Prognose stellt. Dies macht das Modell bei Status Quo unbrauchbar.

3.2.5 Feature Importance

Eine häufig gestellte Frage bei Betrachtung eines solchen Modells ist: „Was hat zu dem Ergebnis geführt und welche(s) der Features war/waren ausschlaggebend?“.

Um diese Frage zu beantworten, kann als erstes Mittel die „*Feature Importance*“ eingesetzt werden. Die *Feature Importance* gibt den relativen Einfluss jedes einzelnen Features auf das Resultat aus. Da es sich hierbei um einen modellspezifischen Werkzeugkasten handelt, muss die *Feature Importance* für jede Art von Algorithmus unterschiedlich berechnet werden. Eine konkrete Erläuterung dieser folgt in den jeweiligen Abschnitten der hier verwendeten Modelle, sollte diese bestimmt werden.

Im Fall linearer Modelle lässt sich die *Feature Importance* aus den Koeffizienten der Funktion annähern. Betrachtet man erneut die Formel der multiplen linearen Regression $y = \beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n + e$, so können die Koeffizienten β_i als „Wichtigkeit“ oder Gewicht interpretiert werden.

Unter Betrachtung der Abb. 14 fällt lässt sich feststellen, welche der Features einen besonders starken Einfluss auf das Resultat haben. Genauso lässt sich durch reines Betrachten identifizieren, welche Features keine Bedeutung oder gar einen negativen Einfluss haben. Ein negativer Wert der *Feature Importance*, oder in diesem Fall Koeffizienten, repräsentiert eine Erhöhung des Fehlers. Dementsprechend wäre es sinnvoll diese Features aus der Eingabematrix zu entfernen.

In Bezug auf das vorliegende Modell kann also erkannt werden, dass die ausschlaggebendsten Variablen „Breathing Problem“, „Fever“, „Dry Cough“, „Gastrointestinal“, „Contact with Covid-19 Patient“ und „Abroad Travel“ sind. Unter Berücksichtigung der Risikobewertung des Robert-Koch-Instituts scheinen diese Rückschlüsse durchaus plausibel zu sein. [50]

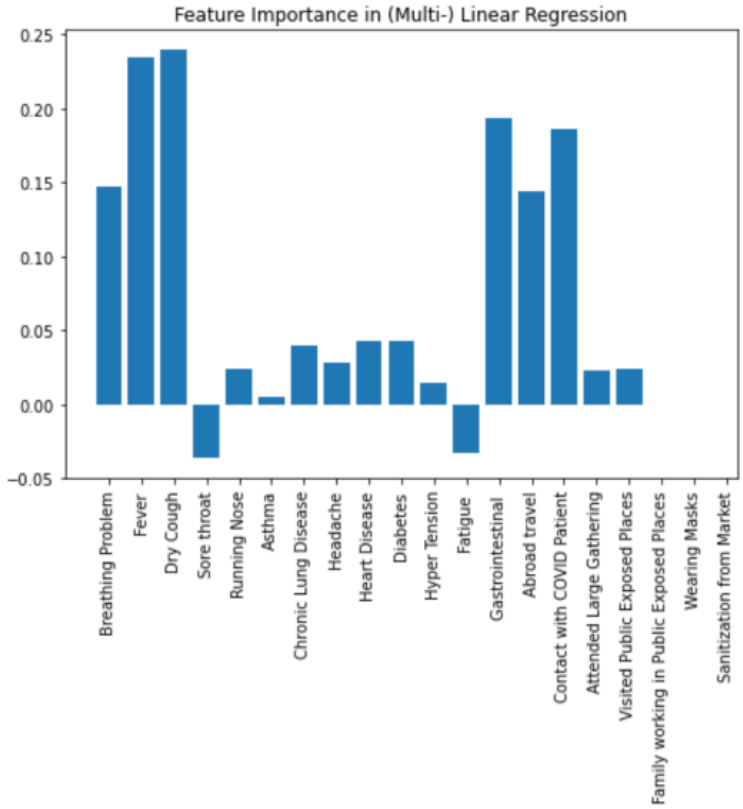


Abbildung 14: Feature Importance der (Multi-) Linearen Regression

Zusätzlich lässt sich feststellen, dass die Features „Sore Throat“ und „Fatigue“ einen negativen Einfluss auf das Ergebnis haben. Dies erscheint erstmal weniger legitim, kann aber auf den Datensatz zurückgeführt werden, der schlicht und ergreifend nicht für eine Regression ausgelegt ist. Die restlichen Features haben nur wenig bis keinen Einfluss auf das Ergebnis. Im Zuge des *Feature Engineerings*, also der Optimierung der Eingabemenge, könnten diese aus den Trainings -und Testdaten entfernt werden, ohne dass die Vorhersagequalität großartig verändert werden würde. Dieses Vorgehen würde im Gegenteil zu einer Verbesserung der Performanz, einer Dimensionalitätsreduktion und somit auch zur Interpretierbarkeit führen. Bezüglich der Interpretation und Auswertung der Feature Importance ist es wichtig, dass diese immer in Bezug auf das verwendete Modell getroffen werden muss, da es sich hierbei um ein modellspezifisches Verfahren handelt.

3.3 Logistische Regression

Alle vorbereitenden Schritte des Modellaufbaus bis zum Training des Modells können ab hier, sollte es nicht anders ausgewiesen sein, als analog betrachtet werden.

3.3.1 Logistische Regression im Detail

Die logistische Regression wird, obwohl der Name etwas anderes vermuten lässt, bei Klassifikationsproblemen verwendet. So verhält sie sich wie eine typische Regression und berechnet eine Wahrscheinlichkeit, um die Zugehörigkeit zu einer Klasse zu bestimmen, resultiert aber in einer Klassifikation. Bei einer Wahrscheinlichkeit oberhalb 50% wird das Ergebnis zu der Klasse 1 und unterhalb 50% der Klasse 0 zugeordnet. Dementsprechend scheint die logistische Regression die bessere Wahl für das vorliegende Klassifikationsproblem zu sein. [51]

Das Verhalten der logischen Regression lässt sich darauf zurückführen, dass die Grundlage auf der *Sigmoid*-Funktion liegt:

$$\sigma(t) = \frac{1}{1 + e^{-t}}, mit\ t = \log\left(\frac{p}{(1-p)}\right)$$

Die Variable p beschreibt die Wahrscheinlichkeit für die Zuordnung zu der Klasse 1. Der Nenner erfasst analog die Wahrscheinlichkeit der Zugehörigkeit zu der Klasse 0. Das Resultat ist die Aufteilung [18, S. 145-146]:

$$\hat{y} = \begin{cases} 1, wenn\ p \geq 0{,}5 \\ 0 \qquad\quad sonst \end{cases}$$

3.3.2 Evaluation

Um das Modell zu bewerten, existieren bis auf die bereits genannten Vertreter *MAE*, *MSE* und *Accuracy* noch weitere Evaluations Metriken.

Hier bieten sich zusätzlich „*Precision*“, „*Recall*“ und die „*F1*“-Metrik an. [33, S. 81-83]

$$Precision = \frac{T_p}{T_p + F_p}$$

$$Recall = \frac{T_p}{T_p + F_n}$$

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

In Worten ausgedrückt, beschreibt *Precision* den relativen Anteil aller wahren positiven Vorhersagen zu allen wahren Vorhersagen also auch eingeschlossen der falsch als positiv klassifizierten Aussagen. Diese Metrik ist besonders relevant, wenn der Anteil an falschen positiven Vorhersagen besonders schwerwiegend ist. *Recall* verhält sich ähnlich, zieht jedoch die falsch negativ vorhergesagten Werte ein. [52] Der Recall-Wert setzt den Fokus auf die falsch negativ prognostizierten Werte. Diese Eigenschaft ist beispielsweise in der Medizin von Vorteil. Wenn ein Tumor nicht als solcher klassifiziert werden kann, ist der genutzte Algorithmus nicht die korrekte Wahl. Ist jedoch keine der beiden Varianten vorzuziehen, kann

der *F1*-Score genutzt werden, der den harmonischen Durchschnitt über *Precision* und *Recall* bildet.

Tabelle 2: Ergebnisse der Evaluations Metriken für die logistische Regression

Mean Absolute Error	Mean Squared Error	Accuracy	Precision	Recall	F1
0,05	0,05	94,76%	97,76%	95,77%	96,71%

Die Ergebnisse der vorgestellten Evaluations Metriken sprechen dafür, dass die logistische Regression eine angebrachtere Wahl als die lineare Regression für den Datensatz ist. Dies lässt sich daran festmachen, dass der *Recall*-Wert bei knapp 96% liegt. Der hohe *Recall*-Wert gibt in erster Linie Aufschluss darüber, wie gering die Anzahl der eigentlich gesunden Probanden mit Covid-19 diagnostiziert werden. Dieser geringfügige Fehler mag auf den ersten Blick ungewollt sein, jedoch rät es sich einen Fehler solcher Art in Kauf zu nehmen. Ansonsten könnte es passieren, dass kranke Personen als gesund diagnostiziert werden, was nicht nur im Falle einer Pandemie ungewünscht ist.

Um einen weiteren Aufschluss über die Verteilungen der Vorhersagen zu erhalten, bietet sich die *Konfusionsmatrix* an. Die *Konfusionsmatrix* stellt die wahren Labels und die Vorhersagen in einer 2x2-Matrix gegenüber.

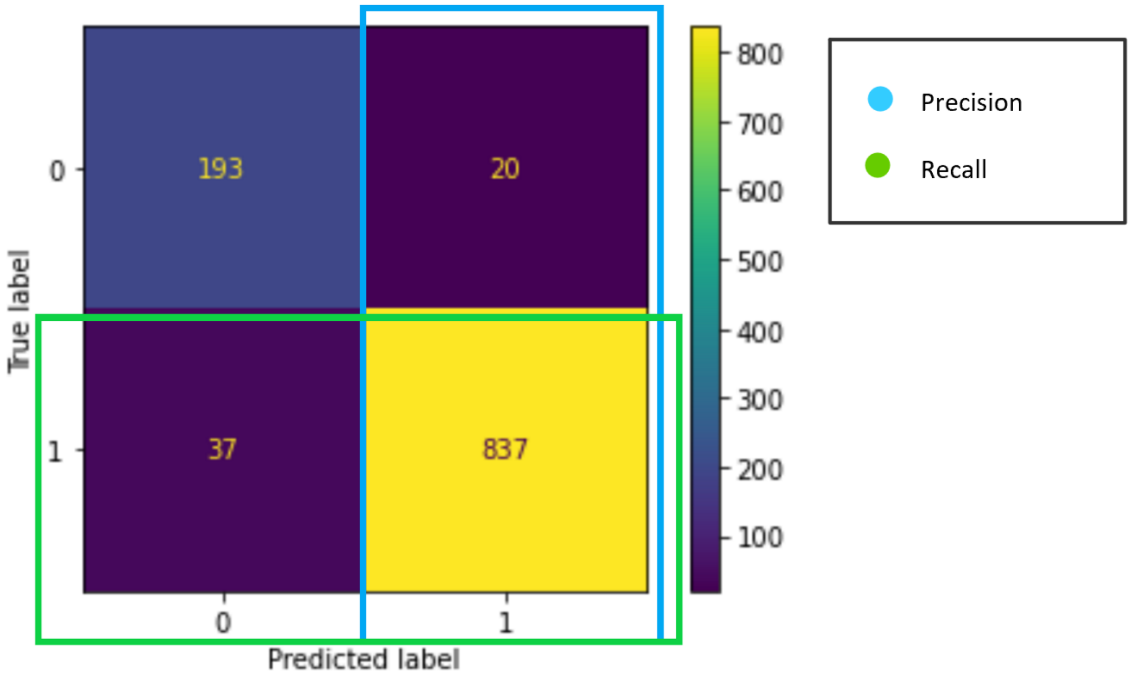


Abbildung 15: Konfusionsmatrix mit Precision und Recall

Auf der Diagonalen der Konfusionsmatrix lassen sich die richtig klassifizierten Vorhersagen ablesen. Klasse 0 (nicht an Covid-19 erkrankt) – 193 und Klasse 1 (an Covid-19 erkrankt) – 837. Unten Links kann die Anzahl an fälschlicherweise negativ und oben rechts die fälschlicherweise positiv klassifizierten Vorhersagen abgelesen werden. Idealerweise sollte die falsch negative Anzahl bei null liegen, dies ist in der Realität nur schwer zu erreichen. Durch

hinzuziehen der Konfusionsmatrix lassen sich Aussagen darüber treffen, wie präzise Aussagen getroffen werden und auch welche Kategorien richtig und falsch klassifiziert wurden.

3.3.3 Feature Importance und Permutation Feature Importance

Analog zur linearen Regression wird die *Feature Importance* aus den Koeffizienten entnommen.

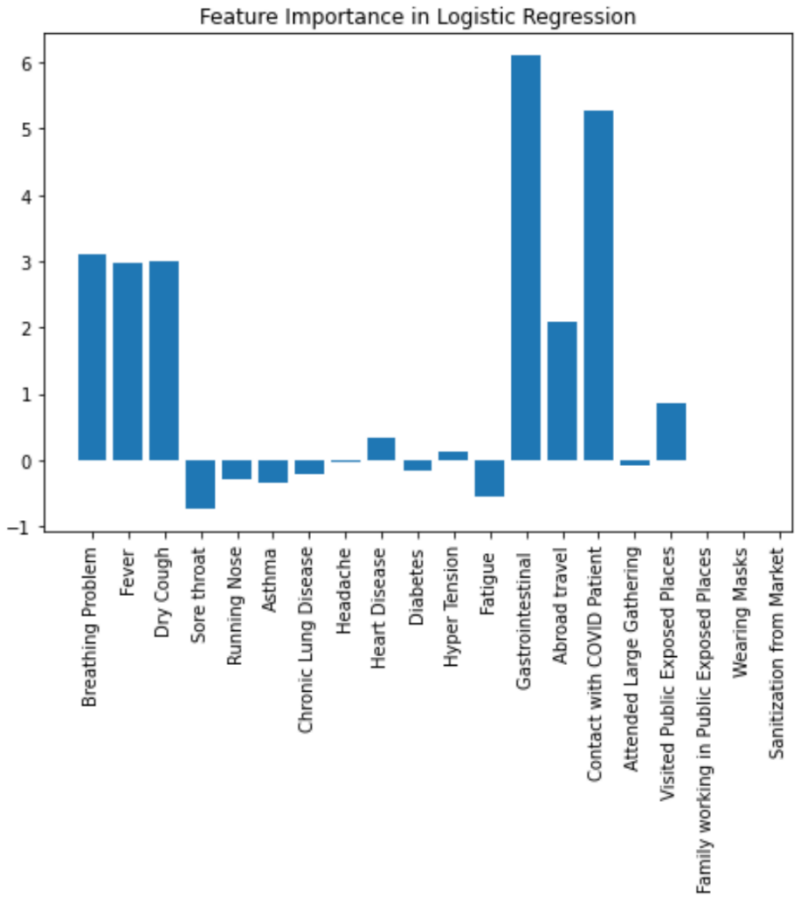


Abbildung 16: Feature Importance der logistischen Regression

Anhand Abbildung 16 lässt sich bestätigen, dass die *Feature Importance* tatsächlich modellspezifisch ist, also immer von dem verwendeten Modell abhängt. Währen bei der linearen Regression noch „Breathing Problem“, „Fever“, „Dry Cough“ die drei größten Koeffizienten waren, sind es in der logistischen Regression die Features „ Gastrointestinal“ und „Contact with COVID Patient“. Abgesehen von letzterem kann man sagen, dass die ausschlaggebenden Punkte weniger plausibel sind als die der linearen Regression. Dieser Punkt wirft die Frage auf, inwiefern man einem Modell vertrauen kann, welches scheinbar die richtigen Schlüsse aus den falschen Beweggründen zieht.

Um die Wirkung verschiedener Ausprägungen auf die Vorhersage zu untermauern, zu widerlegen oder weiter zu untersuchen, kann die „*Permutated Feature Importance*“ zu Rate gezogen werden. Der große Vorteil der *PFI* gegenüber der „normalen“ *Feature Importance* ist, dass die *PFI* modellagnostisch ist. Modellagnostisch bedeutet, dass die Ergebnisse dieses Algorithmus unabhängig von dem genutzten Modell sind und dementsprechend bei Verwendung desselben Datensatzes bei jedem Modell dasselbe Ergebnis liefern sollte. [53]

Das Vorgehen der *Permutated Feature Importance* ist simpel. In jeder Spalte des tabellenförmigen Datensatzes wird nach und nach der Inhalt zufällig durcheinander gewürfelt und somit neu angeordnet. So wird die Verbindung z.B. des zutreffenden Symptoms zu der Diagnose eines Probanden aufgehoben. Dies resultiert in einem künstlichen erzeugten Eingang neuer, noch nicht vom Modell gesehener Daten. Die Folge ergibt sich in dem sich verändernden Fehler. Wird der Fehler für eines der Features besonders groß nach dem Permutieren, so muss dieses Feature besonderen Einfluss auf das Ergebnis haben.

	Feature	Importances Mean
12	Abroad travel	0.075437
14	Attended Large Gathering	0.066697
13	Contact with COVID Patient	0.035189
0	Fever	0.034269
1	Dry Cough	0.033119

Abbildung 17: Auszug aus Ergebnissen der Permutated Feature Importance der logistischen Regression

Vergleicht man nun die die Ergebnisse der *PFI* mit den Koeffizienten der logistischen Regression, so ist das Feature „Gastrointestinal“ nicht mal mehr unter den fünf wichtigsten Features, obwohl es vorher so gewertet wurde. Es stellt sich sogar raus, dass „Abroad Travel“ nun den stärksten Einfluss hat. Diese Ergebnisse scheinen legitimer als die ursprünglichen Ergebnisse des Modells, vor Allem unter Miteinbeziehung der Risikobewertung des RKIs.

3.4 Entscheidungsbäume und Random Forest

Der nun folgende Abschnitt befasst sich mit den Umsetzungen der Entscheidungsbäume und des Random Forest. Weiterhin werden diese Modelle genutzt, um die beiden Methoden *Partial Dependence Plots* und *Shapley Values* beispielhaft zu illustrieren und zu erläutern.

3.4.1 Entscheidungsbaum als Klassifikator

Der grobe Aufbau von binären Entscheidungsbäumen wurde in dem theoretischen Abschnitt bereits beleuchtet. Jedoch stellt sich noch die Frage, wie genau die Reihenfolge der Features im Baum und die Schwellwerte der Knoten zustande kommt.

Der Ablauf der Baumerstellung gestaltet sich wie folgt:

1. Die erste Spalte, also das erste Feature, wird als Wurzelknoten gewählt
2. Die zweite Ebene des Baumes bildet die Zielvariable
3. In beiden Kinderknoten bzw. in den Blättern wird aufsummiert, wie oft über den gesamten Datensatz der jeweilige Pfad in einer positiven Klassifizierung resultiert (1) und wie oft in einer negativen (0)
4. Der Vorgang wird mit allen Features wiederholt, bis die Kinder „rein" sind, also nur in positiven oder negativen Klassifikationen resultieren oder diesem am nächsten kommt.
5. Der „reinste" Teilbaum wird als Wurzel verwendet. Wenn die Blätter unrein sind, wird der Vorgang mit allen übriggebliebenen Features wiederholt, um die weiteren Verzweigungen und Teilbäume aufzuspannen.
6. Die Blätter, die die Klassifikation zeigen, sind rein.

Der beschriebene Vorgang lässt sich auch mathematisch ausdrücken. Auch hier gibt es unterschiedliche Möglichkeiten das Maß an „Unreinheit" zu bestimmen. In den meisten Fällen wird jedoch die „*GINI-Impurity*" verwendet. [54]

Die *GINI-Impurity* gibt für die einzelnen Blätter eines Entscheidungsbaums den relativen Anteil an positiven und negativen Klassifikationen an. Für einzelne Knoten mit zwei Klassen lässt das Unreinheitsmaß sich mit folgender Formel berechnen:

$$G = 1 - (p_0^2 + p_1^2), mit\ p = \frac{N_i}{N}$$

Hier gilt:

- $p_{0/1}$: Wahrscheinlichkeit Klasse 0 oder 1
- N : Gesamt Anzahl Stichproben im Blatt
- N_i: Anzahl an Stichproben mit Zugehörigkeit zur Klasse i

Um einen Baum zu erstellen, resultiert daraus die Berechnung der *GINI-Impurity* für jede Abbildung eines Features auf die Zielvariable. Die Abbildung, die die geringste *GINI-Impurity* aufweist, bildet die Wurzel des Baumes.

Bei der vorliegenden binären Klassifikation ist der Schwellwert simpel zu bestimmen. Da die Features nur die Zustände 1 und 0 annehmen können liegt dieser bei 0,5.

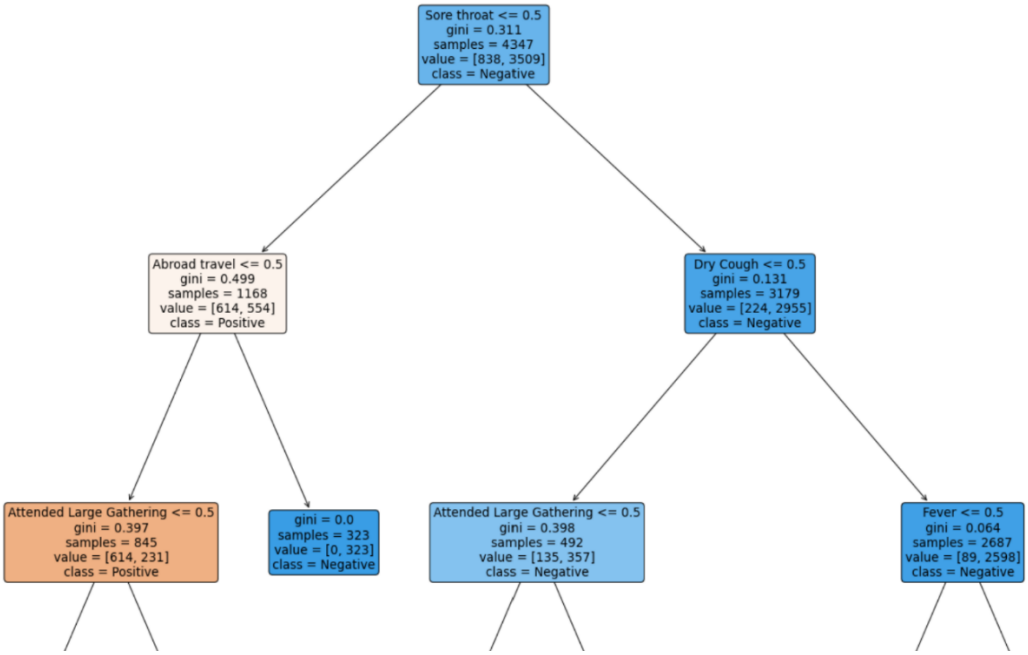


Abbildung 18: Entscheidungsbaum (Tiefe 2)

In Abbildung 18 ist der Entscheidungsbaum aus der Implementierung dargestellt. Aufgrund der nicht darstellbaren Größe bei 20 Features wurde die Tiefe 2 gewählt, damit die Inhalte der einzelnen Knoten und Blätter ersichtlich bleiben. Jedem Knoten kann die zugrundliegende Bedingung entnommen werden. Zusätzlich wird die vorgestellte *GINI-Impurity*, die Anzahl der Samples, die unter diese Bedingung fallen und die entsprechende Aufteilung der Samples in „trifft zu“ und „trifft nicht zu“ dargestellt. Die „class“ Variable enthält die bis zu dem jeweiligen Knoten wahrscheinlichste Klassifizierung der Samples. Den Blättern ist die endgültige Klasse des betrachteten Samples zu entnehmen. In den Informationen des Blattes im linken Teilbaum der Abbildung ist auch ersichtlich, dass die Blätter rein sind, da eine *GINI-Impurity* von 0,0 ausgewiesen wird und alle Samples der negativen Klasse zugeordnet werden.

3.4.2 Random Forest

Neben den Entscheidungsbäumen widmet sich die Implementierung dem Random Forest Algorithmus. Ein großer Nachteil der herkömmlichen Entscheidungsbäume liegt in der Veranlagung zum *Overfitting*. Obgleich Entscheidungsbäume eines, wenn nicht das verständlichste Modell bieten, macht es diese Eigenschaften in vielen Einsatzgebieten schlichtweg unbrauchbar. Gerade der Einsatz sogenannter Ensemble Algorithmen bieten die Möglichkeit das Risiko auf *Overfitting* zu reduzieren und die *Generalisierung* zu optimieren. [55]

Der Verbund mehrerer Entscheidungsbäume resultiert in einem Random Forest. Abgesehen von der verbesserten Generalisierungsfähigkeit, ist auch ein Anstieg in der Performanz gegenüber den klassischen Entscheidungsbäumen bemerkbar. Im Kontrast dazu steht allerdings der Verlust der Erklärbarkeit und Interpretierbarkeit des Modells.

Random Forests nutzen sogenannte „bagging“ Algorithmen [56], um den Datensatz in Teildatensätze aufzuteilen und auf verschiedene Teilbäume zu verteilen, das sogenannte „Bootstraping“. Oberflächlich betrachtet werden Teile der Testdaten stichprobenartig separiert. Auf Grundlage dieser Stichproben werden einzelne Entscheidungsbäume trainiert. Diese Art

des „Sampling“ eignet sich vor Allem für große Datensätze, da so in den Teildatensätzen bis zu ca. 62% einzigartiger Stichproben aus dem Hauptdatensatz enthalten sind. Bei Features mit einer hohen Korrelation zueinander können diese verstärkt in mehreren Entscheidungsbäumen präsent sein.

Anschließend werden die einzelnen Entscheidungsbäume wie bereits vorgestellt, ausgewertet und eine Vorhersage berechnet. Unabhängig davon, ob der Random Forest als Klassifikator oder Regressor verwendet wird, berechnet sich die Gesamtvorhersage folgendermaßen:

$$\hat{y} = \frac{1}{B}\sum_{n=1}^{B}(f_n(x))$$

Hierbei gilt:

- $\hat{y}$: Vorhersage
- B: Anzahl Entscheidungsbäume
- n: Index des Entscheidungsbaums
- f_n: Ergebnis des Entscheidungsbaums n
- x: Sample, für das die Vorhersage berechnet werden soll

Die finale Vorhersage wird also durch den Durchschnitt der Vorhersagen der konstruierten Entscheidungsbäume bestimmt. [57]

3.4.3 Partial Dependence Plots

Zusätzlich zu den bereits vorgestellten Techniken *Feature Importance* und *Permutated Feature Importance* ist es möglich, die sogenannten Partial Dependence Plots zu verwenden, um den Effekt und die Komplexität einer unabhängigen auf eine abhängige Variable festzustellen und zu visualisieren. Komplexität umfasst die Dimension der Beziehung, also linearer, quadratischer oder komplexer Art.

Ein großer Vorteil gegenüber den zuvor vorgestellten Techniken liegt in der Beobachtung einer einzelnen unabhängigen Variablen. So lässt sich hervorheben, wie sich ein Feature in seiner vollen Bandbreite über Datensatz auf das Ergebnis auswirkt. [58]

Um die *Partial Dependence* eines Features zu bestimmen, wird zunächst die komplette Spalte ausgewählt und das entsprechende Feature in allen Zeilen auf den kleinsten Wert, hier 0, gesetzt. Dies erzeugt einen künstlichen Datensatz. Die restlichen Spalten bleiben unberührt. Anschließend wird für jede Zeile die Vorhersage bestimmt. Der Durchschnitt dieser manipulierten Vorhersage entspricht dann der *Partial Dependence* für den kleinsten Wert. Der Vorgang wird der Anzahl an Antwortmöglichkeiten entsprechend wiederholt. Die Ergebnisse lassen sich nun in einem Graphen auftragen und interpretieren.

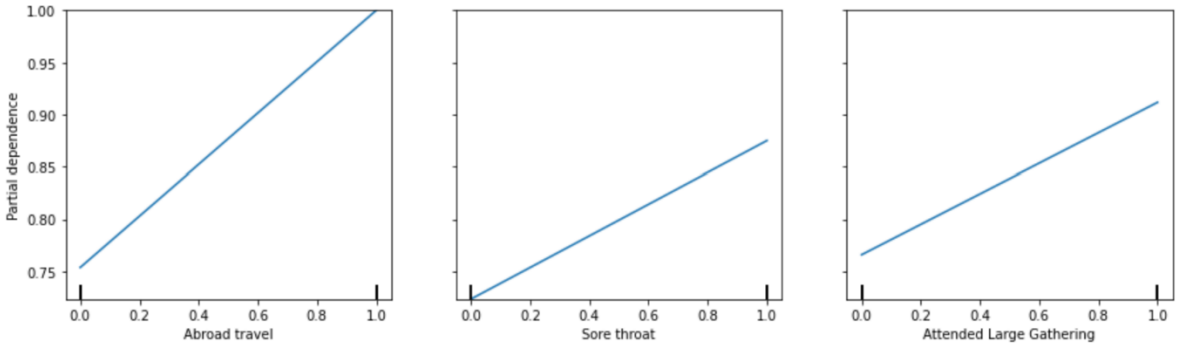


Abbildung 19: Partial Dependence Plots der drei wichtigsten Features

In Abbildung 19 sind die *Partial Dependence Plots* der drei wichtigsten Features des Modells dargestellt. Die Features wurden anhand der (*Permutated*) *Feature Importance* ausgewählt. Auf der x-Achse sind die manipulierten Werte, die das Feature annehmen kann, und auf der y-Achse die Vorhersage, die daraus generiert wurde.

Bei allen drei Features liegt die geringste Wahrscheinlichkeit eine Infektion mit Covid-19 vorherzusagen bei ungefähr 75%. Ist eine Reise ins Ausland angetreten worden ist eine Infektion unvermeidbar, ist dies nicht der Fall gewesen, besteht immer noch eine Wahrscheinlichkeit von ca. 75% die Diagnose zu erhalten. Ähnlich verhält es sich mit dem Beiwohnen von Menschenansammlungen, wobei die Wahrscheinlichkeit auf 90% steigt. Ein gängiges Erkältungssymptom wie Halsschmerzen wird vom Modell ebenso als aussagekräftiger Indikator für die Infektion gehalten.

Partial Dependence Plots erweisen sich als nützliches Hilfsmittel für die Interpretation, jedoch hat auch diese ihre Grenzen. Bei Verwendung des Algorithmus wird stets davon ausgegangen, dass alle Ausprägungen keine Korrelationen aufweisen, also komplett unabhängig voneinander sind. Diese Unabhängigkeit ist in der Realität nur schwer zu gewährleisten, was aber das Verwerfen dieses Werkzeugs nicht legitimiert.

3.4.4 Shapley Values

Oberflächlich betrachtet berechnen die „*Shapley Values*“ die Beteiligung einzelner Faktoren an einem Ergebnis. Das System, das die Grundlage für die *Shapley Values* etabliert hat, ist die Spiele Theorie. Auch hier geht es darum, die Beteiligung unterschiedlicher Spieler an einem Ergebnis zu messen. Die Art des Einflusses, sprich positiv oder negativ, spielt dabei erst einmal nur eine marginale Rolle. Von Interesse ist nur der Deckungsbeitrag, also die Differenz zwischen Verlust und Gewinn. Die *Shapley Values* werden über den Durchschnitt über alle Deckungsbeiträge gebildet. [59]

Dieses Konzept lässt sich auch auf das maschinelle Lernen übertragen. Hier spielt die Auswahl an Features und auch deren Reihenfolge eine besondere Rolle für das Ergebnis der Vorhersage. Der Aspekt der Abhängigkeit der Variablen untereinander wurde von den bereits vorgestellten Verfahren noch nicht berücksichtigt. Die Reihenfolge führt aber zu einem Nachteil der *Shapley Values* – die Rechenzeit. Da in der Theorie bei jeder möglichen Permutation der Features festgestellt werden müsste, wie sich die Ausgabe des Modells verhält, wenn unterschiedliche Features hinzugefügt oder weggelassen werden, sollten die *Shapley Values* approximiert werden. Um die nötige enorme Rechenleistung zu Umgehen bietet sich beispielsweise die Monte Carlo Simulation an. Die Monte Carlo Simulation wird durch das Stichprobenartige

herausziehen diverser Untermengen aus der Gesamtmenge der Features definiert. Dieses Verfahren hat zum Ergebnis, dass deutlich weniger Permutationen überprüft und berechnet werden müssen. Gerade die Rechenleistung ist ein Argument für die Verwendung von Approximationen in diversen Bibliotheken, die *Shapley Values* nutzen oder die Berechnung beinhalten. [60]

Die in der Implementierung verwendete Bibliothek zur Berechnung und Visualisierung der *Shapley Values* heißt *SHAP*[9]. *SHAP* umfasst eine Vielzahl an bereits implementierten Methoden, welche die Interpretation und Erklärung von Modellen mit Tabellen -und Baumstrukturen, aber auch auf Text -und Bildern basierend vereinfachen.

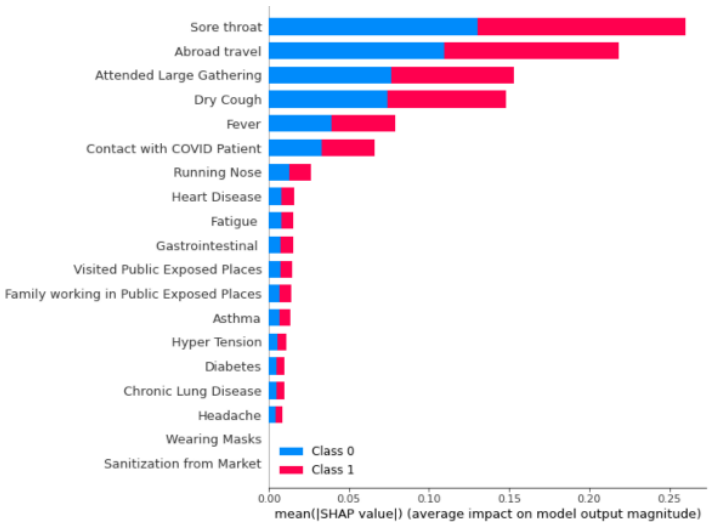


Abbildung 20: SHAP Plot für Entscheidungsbaum

Die x-Achse des in Abbildung 20 dargestellten *SHAP* Plots zeigt den Mittelwert *des Shapley Value*s pro Feature. Auf der y-Achse sind die einzelnen Features gelistet. Dem aufmerksamen Betrachter fällt schnell auf, dass der Plot für Klassifikationen eine hohe Ähnlichkeit zu den Graphen der *Feature Importance* aufweist. Der signifikante Unterschied liegt in der farblichen Trennung des Balkendiagramms. Die von bedeutendsten bis zum unbedeutendsten sortierten Feature sind genau zu 50% in rot und blau unterteilt. Das liegt an der Aussage der *Shapley Values*. Nämlich der Auswirkung der einzelnen Variablen auf die Vorhersage, und zwar in dem gesamten Spektrum, die sie annehmen können. Da der Datensatz lediglich die Aussagen „Ja“ und „Nein“ für die einzelnen Symptome ausdrückt, entsteht die besagte Trennung in genau der Hälfte.

Auch wenn der Datensatz nicht für eine Regression ausgelegt ist, wird der Nutzen von *Shapley Values* gerade bei diesen deutlich, denn probabilistische Vorhersagen bilden einen größeren Raum an potenziellen Varianten der Eingabe ab.

[9] https://shap.readthedocs.io/en/latest/

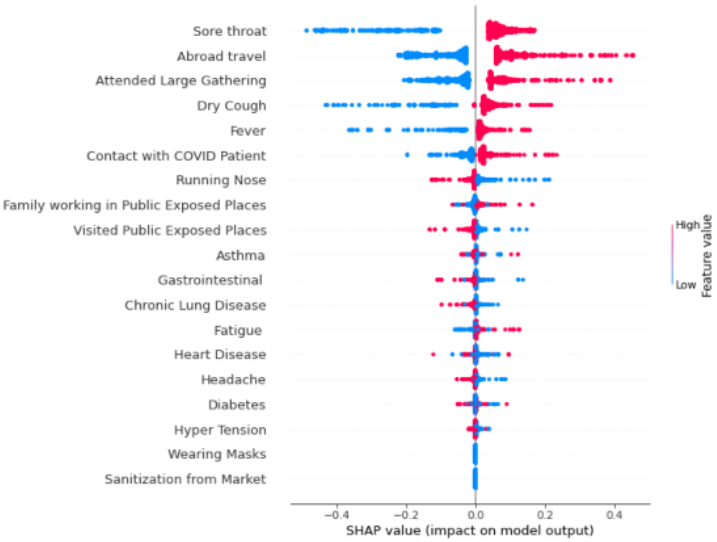


Abbildung 21: SHAP Plot für Random Forest

Der Aufbau des Graphen ist im Grunde derselbe wie der des zuvor vorgestellten Entscheidungsbaumes. Der große Unterschied liegt darin, dass auf der x-Achse nicht mehr der Mittelwert der *Shapley Values* abgebildet ist, sondern der *Shapley Value* passend zu dem jeweiligen Feature. Dieser drückt aus, in welche Richtung also positiv oder negativ, und wie stark ein bestimmter Wert in einem Feature sich auf das Endergebnis, also der Vorhersage, auswirkt.

Untersucht man nun die Auswirkung des Features „Sore Throat“ stellt sich erst einmal heraus, dass es dasselbe Feature ist, welches auch in den Varianten der *Feature Importance* und den *Partial Dependence Plots* zumindest in den Top 3 Features auftrifft. Der Graph lässt sich nun an der Stelle wie folgt interpretieren: Beinahe linear auf der x-Achse lässt sich das Gewicht des Features auf die Vorhersage ablesen. Je kleiner der Wert wird (hier in blau dargestellt), umso eher tendiert die Vorhersage Richtung 0. Ein Wert nahe 0 würde einer Klassifikation von „nicht infiziert“ entsprechen. Analog wird bei einem größeren Wert (im Graphen an der roten Färbung zu erkennen) die Wahrscheinlichkeit für eine Entscheidung, die entsprechende Instanz als „infiziert“ zu klassifizieren, deutlich größer. Interessanterweise spielen unterschiedliche Größen in den Features „Wearing Masks“ und „Sanitization from Market“ keine Rolle in Bezug auf das Ergebnis. Was erneut dafür spricht, dass die beiden Features ohne Verlust an Aussagefähigkeit aus dem Datensatz entfernt werden könnten.

3.5 Support Vector Machines

Als Kontrast zu den bisher vorgestellten Whitebox Modellen soll an dieser Stelle das Blackbox Modell Support Vector Machines als Klassifikator und Regressor vorgestellt werden. Abschließend wird ein letztes Verfahren eingeführt und angewandt, um etwas Licht ins Dunkle des Modells zu bringen. Hierfür muss erst die Grundlage geschaffen werden.

3.5.1 Support Vector Machines im Detail

Die Grundidee von Support Vector Machines gestaltet sich im ein -bis dreidimensionalen simpel. Gesucht ist eine (lineare) Funktion oder Ebene, die die Datenpunkte in zwei Gruppen teilt. Zusätzlich werden parallellaufende „Grenzen“ gezogen, die sogenannten „margins“. Ziel ist es, die Margins, also den Abstand der Datenpunkte zur eigentlichen trennenden Ebene oder

Funktion maximal zu halten. Eine weitere Eigenschaft der SVM ist es, Fehler zuzulassen und in Kauf zu nehmen, um diese maximale Trennung zu erreichen.

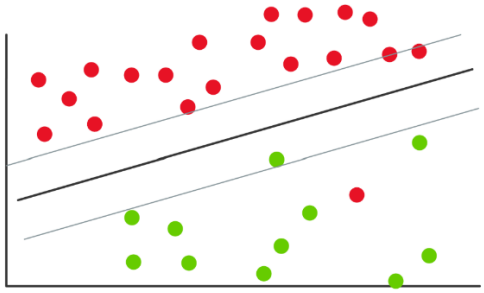


Abbildung 22: Beispielhafte Darstellung einer SVM

Sind die Daten so klar trennbar, ist die Trennung denkbar schnell vollbracht. Sind die Punkte durchmischt oder nicht so klar in zwei Gruppen einteilbar, kommt die eigentliche Stärke der Support Vector Machines ins Spiel.

Wenn in der vorliegenden Dimensionalität keine eindeutige Trennung durch einen *Support Vector Classifier* möglich ist, werden die Daten künstlich in die nächsthöhere Dimension gehoben. Ein simples Beispiel: Ein Datensatz, der die unabhängige Variable Medikamentendosis auf „Ist genesen“ und „Ist nicht genesen“ abbildet, kann graphisch auf einer Dimension abgebildet werden. Jedoch ist keine klare Trennung möglich. Eine mögliche „Kernel“ Funktion der SVM, der die Polynom-Funktionen nutzt, „hebt“ die Werte in die zweite Dimension, indem die Medikamentendosen quadriert werden. Somit entsteht eine zweidimensionale Form, nämlich eine Kurve. Sollten die Daten dennoch nicht trennbar sein, folgt die Berechnung der dritten Dimension. Dieser Vorgang kann beliebig oft wiederholt werden, um einen idealen Trennvektor mit seinen Grenzen für den Datensatz zu finden. [61]

An dieser Stelle soll noch kurz und knapp der sogenannte „Kernel Trick“ erwähnt sein. [62] Die Datenpunkte werden nicht tatsächlich in einen höheren Raum transformiert. Die Beziehung wird anhand der ursprünglichen Dimension beschrieben, jedoch wird das Verhalten in höheren Dimensionen durch Berechnung des Skalarprodukts in einem höheren Raum emuliert. Abgesehen davon, dass alle Räume über der dritten Dimension die menschliche Vorstellungskraft übersteigen und zusätzlich nicht mehr darstellbar sind, führt dieses Vorgehen zu einer erhöhten Rechenleistung.

Die Erstellung eines Modells auf Grundlage der Support Vector Machines klassifiziert dieses als Blackbox Modell. Die Ergebnisse mögen zwar zutreffend sein, jedoch ist nicht mehr nachvollziehbar, was zu den Vorhersagen geführt hat.

3.5.2 Local Interpretable Model-agnostic Explanations

Die bisher vorgestellten Methoden der XAI ermöglichen Aussagen auf globaler Ebene. Jedoch kann es aufschlussreich sein, Anhand einzelner Datenpunkte die Entscheidung nachzuvollziehen. Um diese Interpretationsmöglichkeit bieten zu können, müssen lokale „*Surrogate*“, also Ersatzmodelle eingesetzt werden. Diese approximieren eine einzelne Vorhersage eines Blackbox Modells. Die Vorhersagen werden angenähert, in dem ein künstlicher Datensatz aus Stichproben der Originaldaten erzeugt wird und in ein interpretierbares Modell, z.B. lineare Regression oder Entscheidungsbaum, eingespeist und dieses somit trainiert wird.

Der Vorgang kann für eine beliebige Anzahl an Vorhersagen wiederholt werden und erstreckt sich über Tabellen-,Text -und Bilddaten. Die Labels des neuen Modells bilden die entsprechenden Vorhersagen des Blackbox Modells, die aus einem künstlichen Datensatz resultieren. Die Features ergeben sich aus einer Manipulation des gerade betrachteten Datenpunktes. Dafür werden in der unmittelbaren Umgebung der, im Fall des genutzten Datensatzes, Zeile, neue fiktionale Datenpunkte erzeugt, indem einzelnen Features des zu untersuchenden Punktes verändert werden. Die Veränderung kann durch Addition oder Subtraktion des Features mit der Standardabweichung oder dem Durchschnitt geschehen. Der Algorithmus verwendet in der Durchführung unterschiedliche Kombinationen an Eingabefeatures und findet so, ähnlich der *Feature Importance*, die Features, die das Modell am besten beschreiben und am meisten Gewicht haben. Zusätzlich zielt *LIME* darauf ab, die Anzahl an erforderlichen Features zu minimieren. Abschließend wird für den so neu entstehenden Datensatz ein lineares Modell entsprechend trainiert, um die Differenz zwischen Blackbox und Surrogat Vorhersage zu minimieren. Dementsprechend wird bei Klassifikationsproblemen versucht, die Genauigkeit zu maximieren und bei Regressionsmodellen der Verlust zu minimieren. [63]

Um dieses Vorgehen darzustellen und die Ergebnisse zu interpretieren, wurden zwei Support Vector Machine Modelle trainiert. Eins der Modelle dient als Klassifikator, das andere als Regressor. Die Bibliothek *LIME*[10] bietet alle Möglichkeiten die beschriebene Methode auf ein beliebiges Blackbox Modell anzuwenden.

Um einzelne Instanzen eines Modells auszuwerten, enthält die Bibliothek sogenannte „*Explainer*“ (zu Deutsch „Erklärer“). Um nun eine Instanz, also eine Zeile in tabellenförmigen Daten, zu erhalten, wird nach Instanziierung des *Explainer*-Objekts die Funktion .explain_instance aufgerufen. Die Funktion erhält den Index der zu untersuchenden Instanz, die berechneten Wahrscheinlichkeiten des Modells und eine gewünschte Anzahl an maximal zu nutzenden Features. Die Ausgabe für zwei zufällige Instanzen des Datensatzes sind in Abbildung 23 und 24 zu sehen.

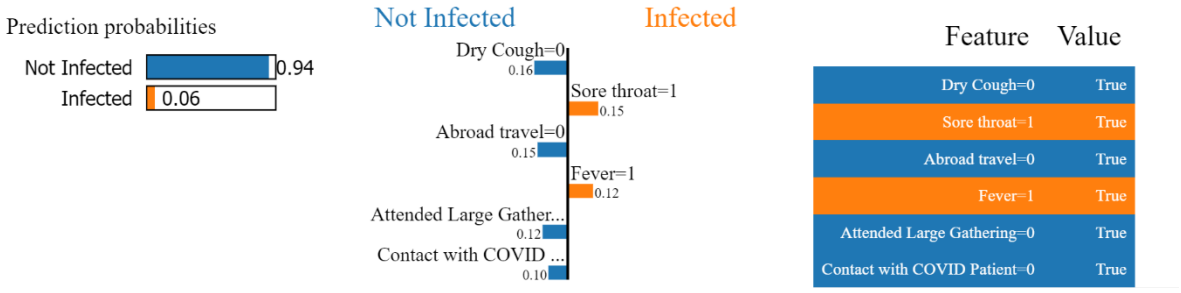


Abbildung 23: Lokale Erklärung einer Instanz (SVM Klassifikation)

Im Falle eines Klassifikationsmodells ist auf der linken Seite die relative Wahrscheinlichkeit der Klassifikation für alle möglichen Klassen zu abgebildet. Die Tabelle rechts gibt die Features samt ihren Werten an und auch, welches Feature zu welcher Klasse zugeordnet wird. Das Diagramm in der Mitte der Abbildung schlüsselt nun die Features, die entsprechenden Werte der Instanz und die Auswirkung auf die Klassifikation auf. Im Falle dieser Instanz wäre die Erklärung für die Diagnose „nicht mit Covid-19 infiziert“, dass der Proband weder Husten noch Kontakt mit einer mit Covid infizierten Person hatte noch eine Reise ins Ausland unternommen

[10] https://lime-ml.readthedocs.io/en/latest/#

hat. Die Argumente für eine Infektion sind lediglich Halsschmerzen und Fieber. Ob die Erklärung Sinn ergibt, liegt aber final im Auge des Betrachters.

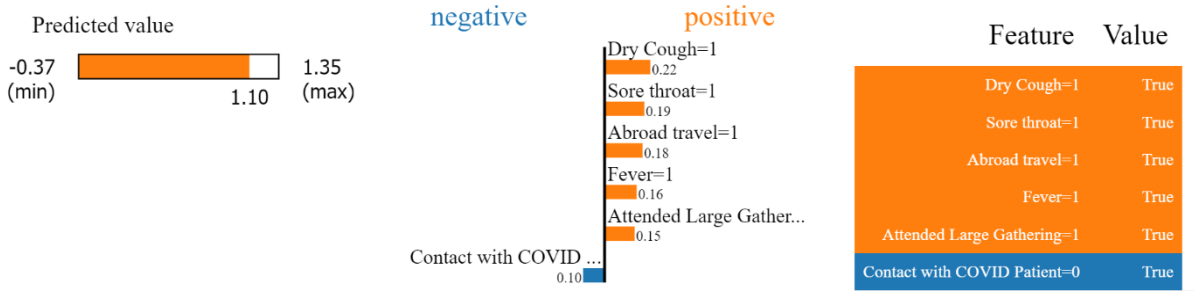


Abbildung 24: Lokale Erklärung einer Instanz (SVM Regression)

Nun das Gegenbeispiel einer Instanz, die die Diagnose „mit Covid infiziert“ erhalten würde. Dieses Beispiel verwendet im Kern eine Support Vector Machine, die der Regression dient. Ziel ist es also, diesmal die Wahrscheinlichkeit für eine Infektion zu berechnen.

Der Aufbau des Plots ist analog zu der Abbildung 23. Der Unterschied liegt darin, dass im linken Abschnitt die relative Wahrscheinlichkeit mit Werten von -0,37 bis 1,35 angegeben werden. Der Proband würde also laut Modell die Diagnose „mit Covid-19 infiziert“ erhalten. Dies macht der Algorithmus daran fest, dass die Instanz fünf der sechs wichtigsten Symptome und Charakteristika, die für eine Infektion sprechen, erfüllt.

Die Verwendung von *LIME* ermöglicht demnach auch Laien, ohne Vorkenntnisse im maschinellen Lernen Vorhersagen zu verstehen, diese zu erläutern, zu interpretieren und gegebenenfalls abzuwiegen, ob diese logisch erscheinen. Experten in diesem Gebiet können zusätzlich die Ergebnisse dazu verwenden, das erstellte Modell optimieren zu können. So kann die Auswahl der Features angepasst werden oder besonders wichtige Aussagen, die auf Domänenwissen beruhen, verstärkt werden.

4 Diskussion

In der vorliegenden wissenschaftlichen Arbeit wurden unterschiedliche Algorithmen und Modelle der Künstlichen Intelligenz vorgestellt, implementiert und ausgewertet. Im nun folgenden Abschnitt werden die Ergebnisse der Modelle hinsichtlich ihrer Performanz und Genauigkeit gegenübergestellt. Anschließend werden die Methoden der Erklärbaren Künstlichen Intelligenz diskutiert und in Zusammenhang mit dem Gesundheitswesen gesetzt.

4.1 Auswertung der Modelle

Angesichts des Fokus des Datensatzes, der die Bewertung durch Klassifikationsmodelle erfordert, werden lediglich die implementierten Modelle in dieser Diskussion berücksichtigt, die dieser Gruppe zugehörig sind. Um in der Realität ein optimales Modell für einen bestimmten Anwendungsfall zu finden, gilt (noch) nicht die Erklärbarkeit als höchste Priorität. Zunächst muss ein Modell entwickelt werden, welches eine möglichst hohe Genauigkeit aufweist und präzise Vorhersagen treffen kann. Sind diese Faktoren zuverlässig erfüllt, gilt es, die Hintergründe und die Plausibilität für die Entscheidungen zu hinterfragen und zu erörtern.

Tabelle 3: Gegenüberstellung der Klassifikationsmodelle Anhand von Evaluations Metriken

Klassifikation	MAE	MSE	Accuracy	Precision	Recall	F1
Logistische Regression	0,05	0,05	94,7%	97,6%	95,7%	96,7%
Entscheidungsbaum	0,018	0,018	98,2%	99,1%	98,6%	98,9%
SVM	0,018	0,018	98,2%	99,1%	98,6%	98,9%

Die Ergebnisse der unterschiedlichen Evaluations Metriken sind der Tabelle 3 zu entnehmen und umfassen *Mean Average Error*, *Mean Squared Error*, *Accuracy*-Score, *Precision*-Score, *Recall* und *F1*-Score der drei Klassifikationsmodelle Logistische Regression, Entscheidungsbaum und Support Vector Machine. Alle drei Modelle haben ähnlich gute Vorhersagen getroffen. Sie verfügen über geringe Fehlerquoten, und auch die anderen vorgestellten Metriken weisen zuverlässige Ergebnisse auf. Bemerkenswert sind jedoch die Nähe der *Precision* und *Recall* Ergebnisse als auch die Identität des Entscheidungsbaumes und der Support Vector Machine. Ersteres ist auf die Implementierung der beiden Funktionen zurückzuführen. Letzteres kann auf den Datensatz an sich deduziert werden. Entscheidungsbäume sind abseits von ihrer langen Rechendauer und ihrem Hang zum *Overfitting*, eine oft genutzte Wahl bei Klassifikationsproblemen. Ein weiterer Aspekt, der zu diesem Ergebnis führt, ist die Instanziierung der Support Vector Machine an sich. Um optimale Ergebnisse des Modells zu erhalten, müssen die Parameter wie allen voran die vorgestellte Kernelfunktion auf das vorliegende Problem angepasst werden. Dafür gibt es bis dato keine hinreichende Vorgabe an Schritten, was ultimativ darin resultiert, dass die Optimierung von Support Vector Machines eher einem „Ausprobieren“ gleicht. [64]

Die Evaluationen durch die vorliegenden Metriken reichen dennoch nicht aus, um die Güte eines Modells vollends zu bestimmen. So ist es beispielsweise möglich oder gar nötig, die Metriken auf Trainings -und Testdaten gleichermaßen anzuwenden. Das Gegenüberstellen der Ergebnisse gibt dann Aufschluss über die Generalisierungsfähigkeit des Modells. Dies resultiert in einer Tendenz zum *Overfitting* oder *Underfitting*, sollte das Modell eine geringe Generalisierungsfähigkeit aufweisen. Ein weiterer möglicher Schritt sind die vorgestellten *Konfusionsmatrizen*. Diese haben für die Modelle, auf die sie angewandt wurden, ähnliche gute

Ergebnisse geliefert mit nur wenigen falsch negativ klassifizierten Werten und einer akzeptablen Menge an falsch positiven Ergebnissen. Im Rahmen dieser Arbeit lässt sich also nur sagen, dass für die Wahl des Klassifikationsmodells aus der Reihe der drei verwendeten keine Rolle spielt.

4.2 Ergebnisse der Methoden der Erklärbaren Künstlichen Intelligenz

Im Hauptteil dieser Arbeit wurden diverse Methoden vorgestellt, die Aufschluss auf globaler und lokaler Ebene der Modelle geben sollen.

Die implementierten Modelle zielen auf eine Unterstützung der Diagnose einer Infektion mit dem Sars-CoV-2 Virus ab. Nachdem die Vorhersagegenauigkeit der einzelnen Modelle gewährleistet und validiert wurde, galt es, die Entscheidungen zu untersuchen und zu legitimieren. Der Einsatz und die Untersuchung der *(Permutated) Feature Importance* in Kombination mit den *Shapley Values* erwiesen sich im Kontext des Gesundheitswesens als nützlich, wenn die Frage gestellt wird, welche Symptome oder Biomarker statistisch gesehen den größten Einfluss auf eine Diagnose haben. [65] In der Auswertung sollte allerdings nicht nur berücksichtig werden, welche die wichtigsten, sondern auch die laut der Erklärung unwichtigsten Features sind. So gaben die Auswertung beispielsweise eine Auswirkung von 0 für die beiden Features, Tragen einer Maske und Verwenden von Desinfektionsmittel an. Daraus ließe sich schließen, dass die Maskenempfehlung während der Pandemie überflüssig gewesen sei. Zieht man im Gegenteil den Datensatz hinzu, wird schnell ersichtlich, dass das Ergebnis lediglich darauf beruht, dass alle 5434 Instanzen keine Maske getragen und kein Desinfektionsmittel genutzt haben. An dieser Stelle erweist es sich also als ratsam, die Grundlage der Erklärung der eingesetzten Methoden mit Quellen oder Domänenwissen zusätzlich zu legitimieren, um so nicht nur das Modell zu optimieren, sondern auch die Erklärung zu belegen oder falsifizieren zu können.

In anderen Instanzen des Gesundheitswesens kann nicht nur die Wirkung der Biomarker, Symptome oder Kennwerte von hoher Bedeutung sein, sondern auch die Schwellwerte, bei denen sich die Auswirkung auf den Organismus verändert. Weiterhin kann auch die Kombination von beispielsweise zweier dieser Merkmale untersucht und dargestellt werden. Hier beweisen sich zusätzlich die *Partial Dependence Plots*. Für den zugrundeliegenden Datensatz, der der Klassifikation vorbehalten ist, hingegen, dient diese Methodik alleinig Präsentationszwecken. Dennoch mindert der Fakt nicht die Einsatzmöglichkeiten. In dieser Arbeit wurden die *Partial Dependence Plots* dazu verwendet die Beteiligung einzelner Features in der gesamten Bandbreite ihrer Ausprägungen auf die Vorhersage einer Infektion zu illustrieren. Die Verwendung ist jedoch nicht auf zwei Dimensionen beschränkt. Verwendet man hierfür drei Dimension, kann die Auswirkung und Abhängigkeit zweier Features auf die Vorhersage dargestellt werden. Ein Anwendungsgebiet für die Verwendung der Methodik im Gesundheitswesen ist die Bestimmung der Wahrscheinlichkeit von Herzerkrankungen, beschränkt auf die Faktoren Alter und Cholesterin. [66]

Die vorherigen Methoden können also einen guten Eindruck für die Relevanz einzelner Features geben und eine Untersuchungsvorlage bieten. Ebenso wichtig ist es gleichwohl, einzelne Vorhersagen zu analysieren. In diesem Kontext zeigte sich das vorgestellte *LIME* Framework als bedeutendes Tool der Bewertung bezüglich der medizinischen Relevanz als auch als Legitimation der Entscheidungen. [67] Von besonderer Aussagekraft ist die Evaluation mehrerer Instanzen und Instanzen aller Klassen, aufgrund dessen die Vollständigkeit der

Erklärungen berücksichtigt werden können. Letztendlich zeigte sich in dem Paper von Kumarakulasinghe et al [67],dass die Entscheidungen, die das *LIME* Modell getroffen hat, eine potente Schnittmenge mit den Aussagen verschiedener praktizierender Ärzte hat, was die medizinische Relevanz des implementierten Systems schlussendlich bestätigt.

5 Zusammenfassung und Ausblick

Das erste Ziel dieser wissenschaftlichen Arbeit war es, relevante Grundlagen der Künstlichen Intelligenz, des Maschinellen Lernens und dem Forschungsgebiet der Erklärbaren Künstlichen Intelligenz und die Bedeutung der Thematik für das Gesundheitswesen zu beleuchten.

Der zweite Hauptaspekt belief auf die Implementierung von beispielhaften Modellen, deren Dokumentation und Erläuterung. Anschließend wurden einige Methoden, welche Einblick in die Entscheidung der Modelle bieten, durchgeführt, erläutert und die Ergebnisse interpretiert. So ließ sich Anhand der linearen Regression der Gesamtaufbau eines solchen Data Science Projekts von Anfang bis Ende und das Gewicht unterschiedlicher Features auf die Vorhersage darstellen. Dieses Gewicht wurde durch die Verwendung der logistischen Regression unter Hinzunahme der *Permutated Feature Importance* bis auf eine leicht veränderte Reihenfolge bestätigt. Der Entscheidungsbaum und Random Forest lieferten die besten Vorhersage Ergebnisse. Diese Ergebnisse ließen sich durch den Einsatz der *Partial Dependence Plots* und der Berechnung der *Shapley Values* spezifizieren und in all ihren Ausprägungen legitimieren. Zum Schluss der Implementierung konnte durch die Support Vector Machines als Blackbox Modell der Kontrast zu den vorherigen Modellen bezüglich der Erklärbarkeit und Interpretierbarkeit aufgezeigt werden. Gleichermaßen boten die *Local Interpretable Model-agnostic Explanations* die interessanteste Methode für das „Öffnen“ der Black Box Modelle. Das Approximieren eines Datenpunktes und die Übertragung auf ein analoges Whitebox Ersatzmodell bot die Möglichkeit, die Vorhersage und die entsprechenden Einflüsse einer Instanz zu inspizieren. Der Einsatz von *LIME* erwies sich somit als erstaunliches Mittel, die Relevanz des Modells im Kontext des Gesundheitswesens zu bestätigen.

Obwohl die Ergebnisse der Untersuchungen und der Modelle für die Kernaussage dieser Arbeit nur eine geringe Rolle spielten, war es möglich, unter Miteinbeziehung von z.B. den Reporten des Robert-Koch-Instituts die Aussagen der Methoden zu bestätigen. Für alle Modelle durchweg schienen die aussagekräftigsten Indikatoren für eine potenzielle Infektion Halsschmerzen, Fieber, trockener Husten, Kontakt zu einer infizierten Person, Reisen und das Beiwohnen von großen Menschenansammlungen in unterschiedlicher Reihenfolge zu sein. Diese Aussagen decken sich mit eigenen Erfahrungen und wissenschaftlichen Berichten. Auch die befremdliche Aussage der Methoden, dass die Faktoren Tragen einer Maske und Verwendung von Desinfektionsmittel keine Rolle spielen, konnte durch Prüfen des Datensatzes begründet werden. Die Verwendbarkeit der Methoden der Erklärbaren Künstlichen Intelligenz sind also im Rahmen dieser wissenschaftlichen Arbeit bestätigt.

Diese Arbeit bietet einen Einstieg und eine Grundlage für die Thematik der Erklärbaren Künstlichen Intelligenz. Allerdings beschränkt sich die Auswertung auf die Verarbeitung von Datensätzen in Tabellenform. Dementsprechend wäre eine Ergänzung in Bezug auf Bilddateien und Sprache im Sinne des Natural Language Processing und eine Erweiterung um die Modellierung mit *neuronalen Netzen* denkbar. Im Rahmen dieser Arbeit war es bedauernswerterweise nicht möglich, die Methoden der XAI in ihrer Gesamtheit zu erfassen, sodass sich ein Nachtrag um die fehlenden Gebiete anbieten würde.

Zusammenfassend lässt sich also bekräftigen, dass Weiterentwicklung, Forschungsdrang und Inkorporation des maschinellen Lernens in das alltägliche Leben unweigerlich mit der Methodik der Erklärbarkeit verbunden sind. Dies ist besonders in Gebieten von höchster Wichtigkeit, in denen ein hohes Maß an Vertrauen in die Modelle nötig ist. Infolgedessen muss

der angesprochen Werkzeugkasten durch und durch exerziert werden, um vollständige, vertrauenswürdige, performante und vor Allem erklärbare Systeme bilden zu können.

6 Anhang

6.1 Elektronische Ressourcen

Datensatz: https://www.kaggle.com/datasets/hemanthhari/symptoms-and-covid-presence

Gitlab Repository: https://gitlab.hs-ruhrwest.de/xpjacies/BA_XAI

Literaturverzeichnis

[1] „Menschliche Intelligenz(en)“ in *Verhaltensorientierte Führung*, S. Franken, Hg., Wiesbaden: GABLER, 2007, S. 26–35, doi: 10.1007/978-3-8349-9539-1_3.

[2] A. M. TURING, „I.—COMPUTING MACHINERY AND INTELLIGENCE“, *Mind*, LIX, Nr. 236, S. 433–460, 1950, doi: 10.1093/mind/LIX.236.433.

[3] John McCarthy, Marvin L. Minsky, Nathaniel Rochester und Claude E. Shannon, „A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955“ (en), *AIMag*, Jg. 27, Nr. 4, S. 12, 2006, doi: 10.1609/aimag.v27i4.1904.

[4] I. Teich, „Meilensteine der Entwicklung Künstlicher Intelligenz“, *Informatik Spektrum*, Jg. 43, Nr. 4, S. 276–284, 2020, doi: 10.1007/s00287-020-01280-5.

[5] *A Brief History of RoboCup*. [Online]. Verfügbar unter: https://www.robocup.org/a_brief_history_of_robocup (Zugriff am: 27. April 2022).

[6] K. R. Chowdhary, „Natural Language Processing“ in *Fundamentals of Artificial Intelligence*, Meherishi und Chowdhary, Hg., New Delhi: Springer India, 2020, S. 603–649, doi: 10.1007/978-81-322-3972-7_19.

[7] R. Murphy, *Introduction to AI robotics*. Cambridge, Massachusetts: The MIT Press, 2019.

[8] A. Rosenfeld, „Computer vision: basic principles“, *Proc. IEEE*, Jg. 76, Nr. 8, S. 863–868, 1988, doi: 10.1109/5.5961.

[9] J. Gao, Y. Yang, P. Lin und D. S. Park, „Computer Vision in Healthcare Applications“ (eng), *Journal of healthcare engineering*, Jg. 2018, S. 5157020, 2018, doi: 10.1155/2018/5157020.

[10] *AlphaGo*. [Online]. Verfügbar unter: https://www.deepmind.com/research/highlighted-research/alphago (Zugriff am: 5. Mai 2022).

[11] D. Silver *et al.*, „Mastering the game of Go without human knowledge“ (eng), *Nature*, Jg. 550, Nr. 7676, S. 354–359, 2017, doi: 10.1038/nature24270.

[12] G. Rebala, A. Ravi und S. Churiwala, „Machine Learning Definition and Basics“ in *An Introduction to Machine Learning*, G. Rebala, S. Churiwala und A. Ravi, Hg., Cham: Springer International Publishing, 2019, S. 1–17, doi: 10.1007/978-3-030-15729-6_1.

[13] A. Burkov, *Machine Learning kompakt: Alles, was Sie wissen müssen*, 1. Aufl. Frechen: MITP Verlags GmbH & Co. KG, 2019. [Online]. Verfügbar unter: https://www.content-select.com/index.php?id=bib_view&ean=9783958459960

[14] V. Nasteski, „An overview of the supervised machine learning methods“, *HORIZONS*, Jg. 4, S. 51–62, 2017, doi: 10.20544/HORIZONS.B.04.1.17.P05.

[15] „Teil I: Grundlegendes zur Entwicklung von Algorithmen“ in *Algorithmen und Datenstrukturen*, N. Blum, Hg., Oldenbourg Wissenschaftsverlag, 2013, S. 1–84, doi: 10.1524/9783486719666.1.

[16] B. Charbuty und A. Abdulazeez, „Classification Based on Decision Tree Algorithm for Machine Learning“, *JASTT*, Jg. 2, Nr. 01, S. 20–28, 2021, doi: 10.38094/jastt20165.

[17] G. Rebala, S. Churiwala und A. Ravi, „Regression“ in *An Introduction to Machine Learning*, G. Rebala, S. Churiwala und A. Ravi, Hg., Cham: Springer International Publishing, 2019, S. 25–40.

[18] A. Geron, *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. Beijing: O'Reilly, 2019.

[Online]. Verfügbar unter: https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=5892320
[19] F. ROSENBLATT, „The perceptron: a probabilistic model for information storage and organization in the brain“ (eng), *Psychological review*, Jg. 65, Nr. 6, S. 386–408, 1958, doi: 10.1037/h0042519.
[20] G. Rebala, A. Ravi und S. Churiwala, „Learning Models“ in *An Introduction to Machine Learning*, Springer, Cham, 2019, S. 19–23, doi: 10.1007/978-3-030-15729-6_2.
[21] *clustering - LEO: Übersetzung im Englisch ⇔ Deutsch Wörterbuch*. [Online]. Verfügbar unter: https://dict.leo.org/englisch-deutsch/clustering (Zugriff am: 9. Mai 2022).
[22] G. Rebala, A. Ravi und S. Churiwala, „Clustering“ in *An Introduction to Machine Learning*, G. Rebala, S. Churiwala und A. Ravi, Hg., Cham: Springer International Publishing, 2019, S. 67–76, doi: 10.1007/978-3-030-15729-6_6.
[23] H. Abdi und L. J. Williams, „Principal component analysis“, *WIREs Comp Stat*, Jg. 2, Nr. 4, S. 433–459, 2010, doi: 10.1002/wics.101.
[24] R. S. Sutton und A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, Massachusetts: The MIT Press, 2018.
[25] *Computer-based medical decision making: from MYCIN to VM*, 1980. [Online]. Verfügbar unter: https://stacks.stanford.edu/file/druid:xt779dh5744/xt779dh5744.pdf
[26] P. P. Angelov, E. A. Soares, R. Jiang, N. I. Arnold und P. M. Atkinson, „Explainable artificial intelligence: an analytical review“, *WIREs Data Mining Knowl Discov*, Jg. 11, Nr. 5, 2021, doi: 10.1002/widm.1424.
[27] C. C. Yang, „Explainable Artificial Intelligence for Predictive Modeling in Healthcare“ (eng), *Journal of healthcare informatics research*, Jg. 6, Nr. 2, S. 228–239, 2022, doi: 10.1007/s41666-022-00114-1.
[28] *Explainable Artificial Intelligence*. [Online]. Verfügbar unter: https://www.darpa.mil/program/explainable-artificial-intelligence (Zugriff am: 10. Mai 2022).
[29] D. Gunning und D. Aha, „DARPA’s Explainable Artificial Intelligence (XAI) Program“ (en), *AIMag*, Jg. 40, Nr. 2, S. 44–58, 2019, doi: 10.1609/aimag.v40i2.2850.
[30] Duden, *erklären*. [Online]. Verfügbar unter: https://www.duden.de/rechtschreibung/erklaeren (Zugriff am: 10. Mai 2022).
[31] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter und L. Kagal, „Explaining Explanations: An Overview of Interpretability of Machine Learning“, 31. Mai 2018. [Online]. Verfügbar unter: https://arxiv.org/pdf/1806.00069.
[32] Duden, *interpretieren*. [Online]. Verfügbar unter: https://www.duden.de/rechtschreibung/interpretieren (Zugriff am: 10. Mai 2022).
[33] S. Masís, *Interpretable machine learning with Python: Learn to build interpretable high-performance models with hands-on real-world examples*. Birmingham: Packt, 2021. [Online]. Verfügbar unter: https://portal.igpublish.com/iglibrary/search/PACKT0005918.html
[34] European Commission. Joint Research Centre., *Robustness and explainability of Artificial Intelligence: from technical to policy solutions*. Publications Office, 2020.
[35] R. Sharma, A. Kumar und C. Chuah, „Turning the blackbox into a glassbox: An explainable machine learning approach for understanding hospitality customer“, *International Journal of Information Management Data Insights*, Jg. 1, Nr. 2, S. 100050, 2021, doi: 10.1016/j.jjimei.2021.100050.

[36] *What are Neural Networks?* [Online]. Verfügbar unter: https://www.ibm.com/cloud/learn/neural-networks (Zugriff am: 13. Juni 2022).
[37] A. Rai, „Explainable AI: from black box to glass box“, *J. of the Acad. Mark. Sci.*, Jg. 48, Nr. 1, S. 137–141, 2020, doi: 10.1007/s11747-019-00710-5.
[38] A. Arora, „Anwendungsgebiete für Machine Learning“, *AIM Agile IT Management*, 8. Feb. 2018, 2018. [Online]. Verfügbar unter: https://www.agile-im.de/2018/02/08/anwendungsgebiete-fuer-machine-learning/. Zugriff am: 11. Mai 2022.
[39] I. Flückiger, „10 Exciting Examples of Machine Learning Applications in Healthcare“, *Towards Data Science*, 4. Okt. 2021, 2021. [Online]. Verfügbar unter: https://towardsdatascience.com/10-exciting-examples-of-machine-learning-applications-in-healthcare-1c4de7b744e6. Zugriff am: 11. Mai 2022.
[40] *RKI - Navigation - Was ist eine Pandemie?* [Online]. Verfügbar unter: https://www.rki.de/SharedDocs/FAQ/Pandemie/FAQ18.html (Zugriff am: 31. Mai 2022).
[41] *RKI - Infektionskrankheiten A-Z - Krankheitsbeschreibung von SARS.* [Online]. Verfügbar unter: https://www.rki.de/DE/Content/InfAZ/S/SARS/Klinik.html;jsessionid=820424C9D8B753812C21DE23C8B6CAFC.internet061 (Zugriff am: 31. Mai 2022).
[42] *Coronavirus kurz erklärt | Zusammen gegen Corona.* [Online]. Verfügbar unter: https://www.zusammengegencorona.de/faqs/covid-19/coronavirus-kurz-erklaert/ (Zugriff am: 31. Mai 2022).
[43] T. Takahashi *et al.*, „Sex differences in immune responses that underlie COVID-19 disease outcomes“ (eng), *Nature*, Jg. 588, Nr. 7837, S. 315–320, 2020, doi: 10.1038/s41586-020-2700-3.
[44] *RKI - Coronavirus SARS-CoV-2 - Epidemiologischer Steckbrief zu SARS-CoV-2 und COVID-19.* [Online]. Verfügbar unter: https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Steckbrief.html (Zugriff am: 31. Mai 2022).
[45] „Exploratory Data Analysis“, 25. Aug. 2020, 2020. [Online]. Verfügbar unter: https://www.ibm.com/cloud/learn/exploratory-data-analysis. Zugriff am: 13. Juni 2022.
[46] K. Potdar, T. S. und C. D., „A Comparative Study of Categorical Variable Encoding Techniques for Neural Network Classifiers“, *IJCA*, Jg. 175, Nr. 4, S. 7–9, 2017, doi: 10.5120/ijca2017915495.
[47] *Data splitting*, 2010. [Online]. Verfügbar unter: https://www.mff.cuni.cz/veda/konference/wds/proc/pdf10/wds10_105_i1_reitermanova.pdf
[48] *Multiple linear regression*, 2008. [Online]. Verfügbar unter: http://hummedia.manchester.ac.uk/institutes/cmist/archive-publications/working-papers/2020/multiple-linear-regression.pdf
[49] A. Botchkarev, „Evaluating Performance of Regression Machine Learning Models Using Multiple Error Metrics in Azure Machine Learning Studio“, *SSRN Journal*, 2018, doi: 10.2139/ssrn.3177507.
[50] *RKI - Infektionskrankheiten A-Z - Risikobewertung zu COVID-19.* [Online]. Verfügbar unter: https://www.rki.de/DE/Content/InfAZ/N/Neuartiges_Coronavirus/Risikobewertung.html;jsessionid=DBEF37347A3EC88A55CE90A61037CC41.internet082?nn=2386228 (Zugriff am: 15. Juni 2022).

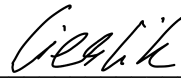
[51] G. Rebala, A. Ravi und S. Churiwala, „Regressions“ in *An Introduction to Machine Learning*, G. Rebala, S. Churiwala und A. Ravi, Hg., Cham: Springer International Publishing, 2019, S. 25–40, doi: 10.1007/978-3-030-15729-6_3.
[52] Jasmina Dj. Novaković, Alempije Veljović, Siniša S. Ilić, Željko Papić und Tomović Milica, „Evaluation of Classification Models in Machine Learning“ (en), *1*, Jg. 7, Nr. 1, 39 – 46-39 – 46, 2017. [Online]. Verfügbar unter: https://uav.ro/applications/se/journal/index.php/tamcs/article/view/158
[53] A. Altmann, L. Toloşi, O. Sander und T. Lengauer, „Permutation importance: a corrected feature importance measure“ (eng), *Bioinformatics*, Jg. 26, Nr. 10, S. 1340–1347, 2010, doi: 10.1093/bioinformatics/btq134.
[54] G. James, D. Witten, T. Hastie und R. Tibshirani, „Tree-Based Methods“ in *Springer Texts in Statistics, An introduction to statistical learning: With applications in R*, G. James, D. Witten, T. Hastie und R. Tibshirani, Hg., New York: Springer, 2017, S. 303–335, doi: 10.1007/978-1-4614-7138-7_8.
[55] P. T. R, „A Comparative Study on Decision Tree and Random Forest Using R Tool“, *International Journal of Advanced Research in Computer and Communication Engineering*, S. 196–199, 2015, doi: 10.17148/IJARCCE.2015.4142.
[56] L. Breiman, „Bagging predictors“, *Mach Learn*, Jg. 24, Nr. 2, S. 123–140, 1996, doi: 10.1007/BF00058655.
[57] G. Rebala, A. Ravi und S. Churiwala, „Random Forests“ in *An Introduction to Machine Learning*, G. Rebala, S. Churiwala und A. Ravi, Hg., Cham: Springer International Publishing, 2019, S. 77–94, doi: 10.1007/978-3-030-15729-6_7.
[58] C. Molnar, *Interpretable machine learning: A guide for making Black Box Models interpretable*. [Morisville, North Carolina]: [Lulu], 2019.
[59] A. E. Roth, *The Shapley Value: Essays in Honor of Lloyd S. Shapley*. Cambridge University Press, 1988.
[60] C. Molnar, *9.5 Shapley Values | Interpretable Machine Learning*. [Online]. Verfügbar unter: https://christophm.github.io/interpretable-ml-book/shapley.html (Zugriff am: 17. Juni 2022).
[61] G. Rebala, A. Ravi und S. Churiwala, „Classification“ in *An Introduction to Machine Learning*, G. Rebala, S. Churiwala und A. Ravi, Hg., Cham: Springer International Publishing, 2019, S. 57–66, doi: 10.1007/978-3-030-15729-6_5.
[62] S. Suthaharan, „Support Vector Machine“ in *Integrated Series in Information Systems Ser*, Bd. 36, *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*, S. Suthaharan, Hg., New York: Springer, 2015, S. 207–235, doi: 10.1007/978-1-4899-7641-3_9.
[63] M. T. Ribeiro, S. Singh und C. Guestrin, „"Why Should I Trust You?": Explaining the Predictions of Any Classifier“, 16. Feb. 2016. [Online]. Verfügbar unter: https://arxiv.org/pdf/1602.04938.
[64] P. Gaspar, J. Carbonell und J. L. Oliveira, „On the parameter optimization of Support Vector Machines for binary classification“, *Journal of Integrative Bioinformatics*, Jg. 9, Nr. 3, S. 33–43, 2012, doi: 10.1515/jib-2012-201.
[65] F. Y. Okay, M. Yildirim und S. Ozdemir, „Interpretable Machine Learning: A Case Study of Healthcare“ in *2021 International Symposium on Networks, Computers and Communications (ISNCC)*, Dubai, United Arab Emirates, 2021, S. 1–6, doi: 10.1109/ISNCC52172.2021.9615727.

[66] J. Petch, S. Di und W. Nelson, „Opening the Black Box: The Promise and Limitations of Explainable Machine Learning in Cardiology“ (eng), *The Canadian journal of cardiology*, Jg. 38, Nr. 2, S. 204–213, 2022, doi: 10.1016/j.cjca.2021.09.004.
[67] N. Barr Kumarakulasinghe, T. Blomberg, J. Liu, A. Saraiva Leao und P. Papapetrou, „Evaluating Local Interpretable Model-Agnostic Explanations on Clinical Machine Learning Classification Models“ in *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*, Rochester, MN, USA, 7/28/2020 - 7/30/2020, S. 7–12, doi: 10.1109/CBMS49503.2020.00009.

Eidesstattliche Erklärung

„Ich versichere an Eides statt durch meine Unterschrift, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt und an allen Stellen, die ich wörtlich oder annähernd wörtlich aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe, mich auch keiner anderen als der angegebenen Literatur oder sonstiger Hilfsmittel bedient habe.
Des Weiteren hat die Arbeit in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen."

Wuppertal, den 23.06.22
Ort, Datum



Unterschrift