Universität Stuttgart
Institut für Automatisierungstechnik
und Softwaresysteme

Konzeption und Entwicklung eines Assistenzsystems für die medizinische Diagnostik mittels maschineller Lernalgorithmen

Bachelorarbeit 3107

An der Universität Stuttgart vorgelegt von
Samira Maleki Pilevar

Medizintechnik

Prüfer: Prof. Dr.-Ing. Dr. h. c. Michael Weyrich

Betreuer: Dr.-Ing. Nasser Jazdi

30.01.2020

Inhaltsverzeichnis

Abbildungsverzeichnis

Formelverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ML	**M**aschinelles **L**ernen
IoMT	**I**nternet **o**f **M**edical **T**hings
MRT	**M**agnet**r**esonanz**t**omographie
KI	**K**ünstliche **I**ntelligenz
SVM	**S**upport **V**ector **M**achine
KNN	**K**-**N**earest-**N**eighbor
RF	**R**andom **F**orest
KNN	**K**ünstliche **N**euronale **N**etze
ANN	**A**rtificial **N**eural **N**etworks
NN	**N**euronales **N**etz
FNN	**F**eedforward-**N**euronales **N**etz
RNN	**R**ecurrentes **N**euronales **N**etz
DL	**D**eep **L**earning
DNN	**D**eep **N**eural **N**etwork
CNN	**C**onvolutional **N**eural **N**etwork
EKG	**E**lektro**k**ardio**g**ramm
BMI	**B**ody-**M**ass-**I**ndex
GUI	**G**raphical **U**ser **I**nterface

Glossar

Intelligente Systeme	Systeme, die über Algorithmen der künstlichen Intelligenz verfügen. Diese Systeme sind lernfähig und verfügen über diese Fähigkeit, unter verschiedenen Aufgaben sichere und fehlertolerante Ergebnisse zu liefern.
Künstliche Intelligenz (KI)	KI wird im Allgemeinen als intelligentes Verhalten in Artefakten definiert. Im Bereich der Informatik lässt sich KI als die Entwicklung von "intelligenten" Computersystemen definieren.
Intelligenz	Die Fähigkeit zu lernen, wahrzunehmen, zu folgern und zu verstehen, um Probleme zu lösen und Entscheidungen zu treffen.
Maschinelle Lernen (ML)	ML ist eine Anwendung der KI, die sich darauf konzentriert, wie ein System selbstständig aus Daten Wissen aufnehmen und sich verbessern kann. Das Verbessern bedeutet in diesem Rahmen, für die gegebene Aufgabe bessere Lösung als vorher zu finden.
Trainingsset	Das Trainingsset wird als Stichprobe aus einer Reihe möglicher Beispiele betrachtet, die es ermöglichen, die statistischen Ähnlichkeiten jeder extrahierten Klasse oder den signifikanten Unterschied zwischen den Klassen zu identifizieren.
Inputs	Inputs, auch Features genannt, sind messbare Eigenschaften oder Kennzeichen von Phänomenen, die für jeden Datensatz Einzelwerte oder Vektoren (mehrere Features) sein können.
Outputs	Outputs, auch Labels genannt, sind Einzelwerte, die als erwartete Ergebnisse der zugehörigen Inputs definiert sind.
Überwachtes Lernen	Überwachtes Lernen bedeutet, ein System aus Inputs mit anzuordnenden bekannten Outputs anzulernen.
Unüberwachtes Lernen	Beim unüberwachten Lernen verfügen die Daten-Sets über kein Label. Dieser Lernstil analysiert die Ähnlichkeit zwischen Input-Daten zur Identifizierung von Clustern oder Gruppen ähnlicher Elemente oder die Ähnlichkeit eines neuen Elements mit einer bestehenden Gruppe.

Semi-überwachten Lernen	In diesem Lernstil verwendeten Daten liegen zwischen dem überwachten und dem unüberwachten Lernen. Die Datasets ohne Label werden zur Identifizierung von Gruppen verwendet. Durch Datasets mit Labels innerhalb jeder identifizierten Gruppe werden Labels für jede Gruppe festgelegt.
Bestärkendes Lernen	Diese Lernmethode hat keine Trainingsdaten zur Verfügung. Der Agent integriert sich in eine Umgebung, indem er im Laufe der Zeit eine Folge von Aktionen erzeugt. Diese Aktionen beeinflussen die Umwelt und führen zu einer positiven oder negativen Belohnung. Die Belohnung wird durch den erreichten Zustand und das Ziel des Agenten bestimmt. Der Agent lernt selbstständig die bestmögliche Aktion unter dem gegebenen Zustand zu ergreifen.
Agent	Der Agent ist ein intelligentes Computerprogramm, das sich ständig an eine bekannte oder unbekannte Umgebung anpasst und in der Lage ist, seine Umgebung im Laufe der Zeit zu verändern.
Regression	Regression kann als eine statistische Standardmethode zur Durchführung von überwachtem Lernen definiert werden, die normalerweise kontinuierliche Werte aus einer Datei voraussagt.
Lineare Regression	Die lineare Regression geht davon aus, dass die Beziehung zwischen den Variablen als lineare Funktion ausgedrückt werden kann. Das Ziel ist es, eine Linie der besten Passform durch alle Datenpunkte zu ziehen, die ein einfaches Verständnis der Vorhersagen ermöglicht, indem Margen- oder Restvariationen minimiert werden.
Klassifikation	Die Klassifikation wird als der Prozess der Kategorisierung von Features aus Trainingssets mit oder ohne Label definiert, und die entsprechenden Modelle werden Klassifikatoren genannt. Die Klassifikation wird verwendet, um Merkmale in mindestens binäre Klassen einzuteilen. Die Genauigkeit der Klassifikation hängt davon ab, wie repräsentativ die Trainingssets für die neuen Daten sind, oder anders formuliert, wie umfangreich alle Kategorien durch Trainingssets abgedeckt sind.
Support Vector Machine (SVM)	SVM ist ein überwachtes Lernmodell, das auf einem nicht-probabilistischen binären linearen Klassifikator basiert, der sowohl

für Klassifizierungs- als auch für Regressionsprobleme verwendet werden kann.

Clustering Clustering ist eine unüberwachte Lernmethode, die Cluster oder Gruppen innerhalb des Datasets anhand ihrer Ähnlichkeiten herausfindet. Beim Clustering ist es entscheidend, dass die Daten innerhalb desselben Clusters enger miteinander verbunden sind als mit Daten aus anderen Clustern.

Neuronales Netz Das neuronale Netz (NN) ist ein Informationsverarbeitungssystem, das aus mehreren miteinander verbundenen Rechenneuronen besteht. Neuronen übertragen Informationen und sind durch Verbindungsglieder miteinander verbunden.

Deep Learning (DL) DL ist eine Klasse von maschinellen Lernmethoden, die aus mehreren Verarbeitungsschichten bestehen. Die verschachtelte und hierarchische Architektur von DL ermöglicht es, aus Rohdaten abstrakte Darstellungen auf hohem Niveau zu lernen. Die Darstellungen auf jeder Ebene sind relativ einfach und die Stärke liegt in der tiefen Architektur.

Assistenzsystem Assistenzsystem ist ein flexibles und anpassungsfähiges System zur betrieblichen Informations- und Entscheidungsunterstützung von Entscheidungsträgern.

Zusammenfassung

Im Umfeld der Medizintechnik müssen in der Zukunft automatisierte Systeme entwickelt werden, die den Menschen bei der Steigerung ihrer Lebensqualität unterstützen können, und zwar durch die Intelligenz der Systeme. Dies kann durch die Integration von künstlicher Intelligenz in die Soft- und Hardware von automatisierten Systemen mit medizinischen Zwecken erreicht werden. Diese Arbeit befasste sich mit der Anwendung von künstlicher Intelligenz zur medizinischen Präventionsdiagnostik. Hierzu wurden in dieser Arbeit zunächst die Begriffe der künstlichen Intelligenz und des maschinellen Lernens ausführlich erläutert. Darauffolgend wurden die Lernstile, Modelltypen und Algorithmen des maschinellen Lernens thematisiert und nach ihren Eigenschaften kategorisiert. Zudem wurde eine Literaturrecherche zu bestehenden Anwendungen der künstlichen Intelligenz im Feld der Medizintechnik in aktuellen Forschungsansätzen und Technologien durchgeführt. Dabei wurde eine Forschungslücke im Stand der Technik hinsichtlich der Anwendung softwarebasierter medizinischer Präventionsdiagnostik unter Einsatz künstlicher Intelligenz identifiziert. Um diese Forschungslücke zu schließen, wurde in der vorliegenden Arbeit ein Konzept vorgestellt, welches auf der Entwicklung eines Assistenzsystems basiert. Dieses Assistenzsystem umfasst drei Vorhersagemodelle zur Bestimmung der Krankheitsrisiken und Präventivmaßnahmen jeweils eine Krankheit, nämlich Erkältungen, Bluthochdruck und Hypercholesterinämie. Die Vorhersagemodelle wurden automatisiert mittels linearer Regression basierend auf simulierten Gesundheitsdaten generiert. Da in dieser Arbeit keine realen Gesundheitsdaten für die Entwicklung der Vorhersagemodelle zur Verfügung standen, wurden diese in MATLAB unter Berücksichtigung des medizinischen Wissens simuliert. Das Assistenzsystem wurde in MATLAB programmiert und kann als Software auf allen Betriebssystemen installiert werden. In diesem Assistenzsystem werden nach Eingabe der Gesundheitsdaten der Benutzer die Krankheitsrisiken sowie die Präventivmaßnahmen zur Verringerung der Risiken ermittelt. Durch den kontinuierlichen Einsatz des Assistenzsystems bei den Benutzern werden die Trainingsdaten ständig erweitert und infolgedessen die Vorhersagemodelle verbessert.

Schlüsselwörter: maschinelles Lernen, Präventionsdiagnostik, künstliche Intelligenz, lineare Regression, überwachtes Lernen, Krankheitsrisiko.

Abstract

In the field of medical technology, automated systems need to be developed in the future that can support people in improving their quality of life with their own intelligence. This can be achieved by integrating artificial intelligence into the software and hardware of automated systems with medical purposes. This thesis dealt with the application of artificial intelligence for medical preventive diagnostics. First of all, the terms artificial intelligence and machine learning were explained in detail in this thesis. Following this, the learning styles, model types and algorithms of machine learning were discussed and categorized according to their characteristics. In addition, a literature research was conducted on existing applications of artificial intelligence in connection with medical technology in current research approaches and technologies. A research gap in the state of the art was identified with regard to the application of software-based medical preventive diagnostics using artificial intelligence. In order to close this research gap, a concept based on the development of an assistance system was presented in this thesis. This assistance system includes three predictive models to determine the disease risk and preventive measures for three diseases, namely colds, hypertension and hypercholesterolemia. The predictive models were automatically generated by linear regression based on simulated health data. Since in this thesis no real health data for the development of the predictive models were available, they were simulated in MATLAB taking into account medical knowledge. The assistance system was programmed in MATLAB and can be installed as software on all operating systems. In this assistance system, after entering the health data of the users, the disease risks as well as the preventive measures to reduce the risks are determined. Due to the continuous use of the assistance system by the users, the training data is constantly expanded and, as a result, the predictive models are further developed.

Key Words: machine learning, preventive diagnostics, artificial intelligence, linear regression, supervised learning, risk of disease.

1 Grundlagen

1.1 Motivation

Heutzutage kann eine große Menge an Gesundheitsdaten der Menschen mittels Medizingeräte, tragbarer und implantierbarer Sensoren erfasst werden. Dazu gehören u. a. Activity Tracker wie Smartwatch, Smart Textiles oder subkutane Sensoren. Gemäß [1] werden die Gesundheitsdaten von Menschen durch 3,7 Millionen aktive Medizingeräte überwacht. Darüber hinaus beinhalten die Patientenakten von Krankenkassen, Krankenhäusern und Arztpraxen zahlreiche Gesundheitsdaten von Menschen.

Hinter diesem Datenvolumen stehen wertvolle Informationen, die die Lebensqualität der Menschen deutlich verbessern könnten. Um aus diesen Daten prädiktive Informationen zu gewinnen, müssen verschiedene Herausforderungen bewältigt werden, wie z.B. die Analyse der großen und vielfältigen Datenmengen. Ein verbreiteter Lösungsansatz dafür ist der Einsatz von maschinellem Lernen (ML).

Das ML hat die Fähigkeit, aus Daten zu lernen und Erfahrungen zu sammeln. ML kann zur Lösung großer Herausforderungen für die Gesundheitsindustrie eingesetzt werden, nämlich der individuellen Therapie, der stratifizierten Patientengruppen sowie der Verschmelzung von Therapie und Diagnose. [2] Wie bedeutsam dieses Thema heutzutage ist, zeigt die weltweite Marktförderung des Internet of Medical Things für 136,8 Milliarden Dollar im Jahr 2021. [1]

Ein Zweck der Verwendung von ML für die Analyse der Gesundheitsdaten kann die Bestimmung des Zusammenhangs zwischen Gesundheitsdaten und bestehenden Krankheiten sein.

Medizinisches Wissen hat gewisse Ursachen von häufigen Krankheiten aufgewiesen. Es gibt jedoch kein Vorhersagemodell in der Medizin, dass alle Einflussfaktoren von Krankheiten mit einem Gewichtungsfaktor definiert und seine Einflussfaktoren im Laufe der Zeit durch die Erfassung neuer Datenmenge erweitert. Die Algorithmen des ML erstellen ein gewisses Modell, um die Beziehung zwischen den Daten zu veranschaulichen. Dadurch können Muster oder Zusammenhänge zwischen den Eigabendaten erkannt werden und damit kann das Ausmaß des Einflusses der einzelnen Krankheitsursachen genauer bestimmt werden. Mit diesen genaueren Informationen können verlässlichere präventive Maßnahmen getroffen werden.

Idealerweise kann dieses Modell gesunde Menschen mit höherem Risiko für bestimmte Krankheiten warnen und die effektivsten Maßnahmen zur Verringerung dieses Risiko empfehlen, was zur Vorbeugung der Krankheit führen kann.

Ein Schritt weiter können die Menschen nach ihrer Anfälligkeit für Krankheiten kategorisiert werden, die die Faktoren wie Vererbung, Stärke des Immunsystems etc. mit einbeziehen.

Hierbei, um ein genaueres Vorhersagemodell zu erstellen, können weitere Einflussfaktoren oder Daten in Betracht gezogen werden. Beispielweise können Wetterdaten, Informationen aus Smart-Home (wie Klimaanlage, Einkaufsliste etc.) oder Kalenderdaten (wie stressige Phase) mitbenutzt werden.

1.2 Automatisierungssysteme in der Medizintechnik

Nach [3] verteilen Automatisierungssysteme sich auf zwei Kategorien. Anlagenautomatisierung und Produktautomatisierung. Anlagenautomatisierung umfasst Automatisierungssysteme, in denen der technische Prozess in verschiedene Systeme aufgeteilt ist. Produktautomatisierung gehört zu dem automatisierten System, bei dem der technische Prozess in einer Einheit durchgeführt wird.

Die am häufigsten eingesetzten Automatisierungssysteme in der Medizintechnik sind im Rahmen der Produktautomatisierung.

Automatisierte medizinische Systeme können nach ihrem Verwendungszweck in drei Gruppen eingeteilt werden:

- Prävention: Systeme, die durch vorgeschriebene Funktionen, die gesunden Menschen vor dem Krankheitsrisiko warnen oder Krankheiten verhindern. Ein Beispiel dafür ist gemäß [4] das stationäre Patientenüberwachungssystem "IntelliVue Guardian System" von Philips Healthcare. Das System erkennt Situationen, in denen eine lebensbedrohliche Krise einen Patienten bedroht, um ein frühzeitiges Eingreifen zu gewährleisten.

- Diagnostik: Systeme, die in Fachbereichen wie Laboratoriumsmedizin, Radiologie, Pathologie usw. zur Erkennung von Krankheiten eingesetzt werden. Ein Beispiel dafür ist die Magnetresonanztomographie (MRT), die ein bildgebendes Verfahren zur Darstellung von Schnittbildern des Körperinneren ist.

- Therapie: Systeme, die Krankheiten beseitigen oder den Heilungsprozess beschleunigen. Als Beispiel dient die Strahlentherapie zur Behandlung von bösartigen Tumoren.

Ein neuer Aspekt der Automatisierungssysteme, der in letzter Zeit erstaunliche Fortschritte erzielt hat, sind intelligente Automatisierungssysteme. Intelligente Systeme können als Systeme definiert werden, die über Algorithmen der künstlichen Intelligenz verfügen. Diese Systeme sind lernfähig und verfügen über diese Fähigkeit, unter verschiedenen Aufgaben sichere und fehlertolerante Ergebnisse zu liefern. Jedes der oben genannten medizinischen Systeme kann intelligent werden, um Aufgaben effektiver, schneller und kostengünstiger zu erledigen. Als Beispiele für intelligente Systeme in Medizin gelten intelligente Implantate, Herzschrittmacher und

Drug-Delivery-Systeme. [5]–[7] Intelligente Systeme sind heute ein unverzichtbarer Bestandteil der Medizin, die die Lebensqualität der Menschen erheblich verbessern.

1.3 Künstliche Intelligenz

Nach [8], [9] wird Künstliche Intelligenz (KI) im Allgemeinen als intelligentes Verhalten in Artefakten definiert. Im Bereich der Informatik lässt sich KI als die Entwicklung von "intelligenten" Computersystemen definieren. Intelligenz ist nach [8], [10] die Fähigkeit zu lernen, wahrzunehmen, zu folgern und zu verstehen, um Probleme zu lösen und Entscheidungen zu treffen. Ein Teilgebiet der künstlichen Intelligenz ist das maschinelle Lernen.

1.4 Maschinelles Lernen

Nach [11]–[13] ist das Maschinelle Lernen eine Anwendung der künstlichen Intelligenz, die sich darauf konzentriert, wie ein System selbstständig aus Daten Wissen aufnehmen und sich verbessern kann. Das Verbessern bedeutet in diesem Rahmen, für die gegebene Aufgabe bessere Lösung als vorher zu finden.

Gemäß [14] wird die Wissensaufnahme (oder Lernen) bei den maschinellen Lernalgorithmen mit Hilfe mathematischer Methoden erfolgt, die basiert auf Beispieldaten (Trainingsdaten) entwickelt werden. Die mathematischen Methoden erkennen Muster zwischen Trainingsdaten, um Vorhersagen oder Entscheidungen zu treffen.

Das Trainingsset wird als Stichprobe aus einer Reihe möglicher Beispiele betrachtet, die es ermöglichen, die statistischen Ähnlichkeiten jeder extrahierten Klasse oder den signifikanten Unterschied zwischen den Klassen zu identifizieren. [15]

1.4.1 Die Lernstile des maschinellen Lernens

Wichtige Lernstile des maschinellen Lernens können in vier Gruppen eingeteilt werden:

- Überwachtes Lernen
- Unüberwachtes Lernen
- Semi-überwachtes Lernen
- Bestärkendes Lernen

Der Unterschied zwischen den Lernmethoden des maschinellen Lernens besteht darin, welche Daten verfügbar sind und welche Strategien zum Lernen verwendet wurden.

1.4.1.1 Überwachtes Lernen

Nach [14], [16], [17] bedeutet überwachtes Lernen, ein System aus Inputs (x_i) mit anzuordnenden bekannten Outputs (y_j) anzulernen. Inputs, auch Features genannt,

sind messbare Eigenschaften oder Kennzeichen von Phänomenen, die für jeden Datensatz Einzelwerte oder Vektoren (mehrere Features) sein können. Outputs, auch Labels genannt, sind Einzelwerte, die als erwartete Ergebnisse der zugehörigen Inputs definiert sind. Im überwachten Lernen wurde das Trainingssets als ein Set bekannter Werte von (x_i, y_j) Paaren definiert. Es ist sehr wichtig, dass die Trainingssets den gesamten Bereich der erwarteten Inputs und gewünschten Outputs umfassen.

Nach [18] ist das Ziel des überwachten Lernens, das Label mit hoher Genauigkeit für Inputs, die nicht im Trainingsset enthalten sind, vorherzusagen.

1.4.1.2 Unüberwachtes Lernen

Im Gegensatz zum überwachten Lernen enthalten die Daten-Sets kein Label beim unüberwachten Lernen. Nach [19], [20] analysiert dieser Lernstil des maschinellen Lernens die Ähnlichkeit zwischen Input-Daten zur Identifizierung von Clustern oder Gruppen ähnlicher Elemente oder die Ähnlichkeit eines neuen Elements mit einer bestehenden Gruppe.

Ein Vorteil des unüberwachten Lernens besteht darin, versteckte Informationen und Strukturen in den Daten zu finden. [17]

1.4.1.3 Semi-überwachtes Lernen

Nach [20], [21] liegen die im semi-überwachten Lernen verwendeten Daten zwischen dem überwachten und dem unüberwachten Lernen, das die relativ wenigen Datasets mit Labels (x_i, y_j) und eine hohe Anzahl von Datasets ohne Labels (x_i) enthält. Datasets ohne Label werden zur Identifizierung von Gruppen oder Clustern verwendet. Durch wenige Datasets mit Labels innerhalb jeder identifizierten Gruppe werden Labels für jede Gruppe festgelegt.

1.4.1.4 Bestärkendes Lernen

Das in Abbildung 1 gezeigte Diagramm ist der grundlegende Entwurf des bestärkenden Lernens (engl. Reinforcement Learning). [22] Erstens wird der Begriff „Agent“ im Rahmen des bestärkenden Lernens definiert:

Nach [20], [23] ist der Agent ein intelligentes Computerprogramm, das sich ständig an eine bekannte oder unbekannte Umgebung anpasst und in der Lage ist, seine Umgebung im Laufe der Zeit zu verändern. Der Agent lernt anhand der vorgegebenen Kriterien aus der Umgebung, welche Maßnahmen zu ergreifen sind.

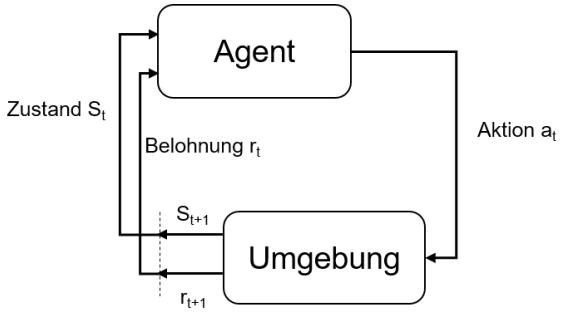


Abbildung 1: Bestärkenden Lernens [22]

Diese Lernmethode hat keine Trainingsdaten zur Verfügung. Der Agent integriert sich in eine Umgebung, indem er im Laufe der Zeit eine Folge von Aktionen a_1, a_2, ..., a_t erzeugt. Diese Aktionen beeinflussen die Umwelt und führen zu einer positiven oder negativen Belohnung in jedem Zeitraum t. Die Belohnung wird durch den erreichten Zustand (S_t) und das Ziel des Agenten bestimmt. Der Agent lernt selbstständig die bestmögliche Aktion unter dem gegebenen Zustand (S_t), in dem sich die Umgebung bei Schritt t befindet. Die Aktion des Agenten ändert wiederum den Zustand der Umgebung von S_t auf S_{t+1} und erzeugt eine Belohnung r_t für den Agenten. Dann ergreift der Agent die bestmögliche Aktion für diesen neuen Zustand (S_{t+1}), dort durch Aufruf einer Belohnung r_{t+1} und so weiter. Über einen Zeitraum von Iterationen versucht der Agent, seine Entscheidungen zu verbessern. Die Rolle der Umgebung besteht darin, dem Agenten verschiedene mögliche/wahrscheinliche Zustände zu präsentieren, die in dem Problem existieren könnten. [16], [22], [24]

1.4.2 Modelltypen und Algorithmen des maschinellen Lernens

1.4.2.1 Regression

Nach [16], [25], [26] kann Regression als eine statistische Standardmethode zur Durchführung von überwachtem Lernen definiert werden, die normalerweise kontinuierliche Werte aus einer Datei voraussagt.

1.4.2.2 Lineare Regression

Die lineare Regression geht davon aus, dass die Beziehung zwischen den Variablen als lineare Funktion ausgedrückt werden kann (siehe Abbildung 2). Das Ziel ist es, eine Linie der besten Passform durch alle Datenpunkte zu ziehen, die ein einfaches Verständnis der Vorhersagen ermöglicht, indem Margen- oder Restvariationen minimiert werden. [20], [25]

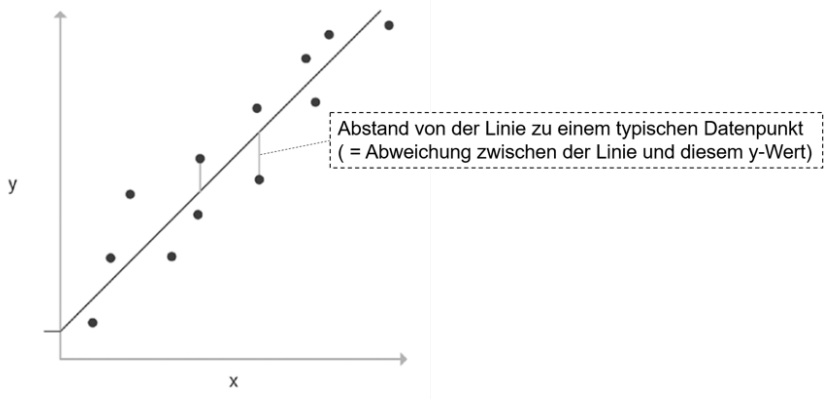


Abbildung 2: Lineare Regression [25]

1.4.2.3 Logistische Regression

Die logistische Regression liefert eine binäre (0 oder 1) Vorhersage, daher ist sie eine Klassifizierungstechnik und kein Regressionsproblem. Die vorhergesagte Ausgabe wird durch logarithmische Transformation der x-Inputs und Verwendung der

Logistikfunktion erzeugt. Ein vorgegebener Schwellenwert wird verwendet, um diese Wahrscheinlichkeit in eine binäre Klassifizierung umzuwandeln. Die logistische Regression verwendet die Sigmoid-Funktion, eine S-förmige Kurve (siehe Abbildung 3 und Formel 1), die jeden beliebigen kontinuierlichen Wert annehmen und in einen Wahrscheinlichkeitswert zwischen 0 und 1 für eine Standardklasse abbilden kann. [20], [25]

$$sig(wert) = \frac{1}{(1 + e^{-wert})}$$

Formel 1: Sigmoid-Funktion [20], [25]

e: Eulersche Zahl

$wert$: Wert, der eine Transformation erfordert.

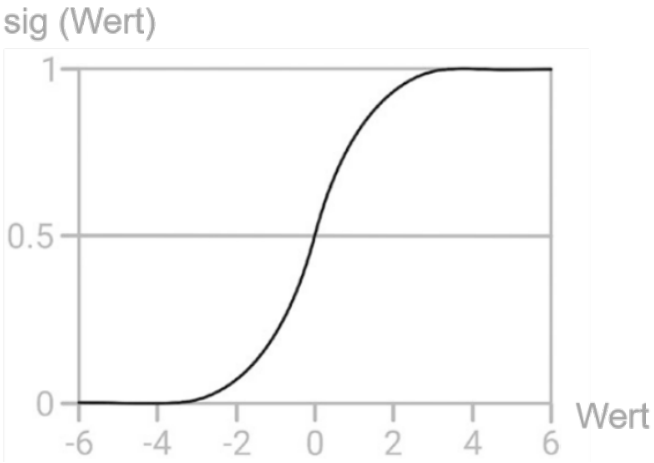


Abbildung 3: Die Darstellung der Sigmoid-Funktion [25]

1.4.2.4 Klassifikation

Nach [25]–[28] wird die Klassifikation als der Prozess der Kategorisierung von Features aus Trainingssets mit oder ohne Label definiert, und die entsprechenden Modelle werden Klassifikatoren genannt. Die Klassifikation wird verwendet, um Merkmale in mindestens binäre Klassen einzuteilen. Die Genauigkeit der Klassifikation hängt davon ab, wie repräsentativ die Trainingssets für die neuen Daten sind, oder anders formuliert, wie umfangreich alle Kategorien durch Trainingssets abgedeckt sind.

Abbildung 4 zeigt die binäre Klassifikation aus dem Trainingssets ohne Label (a) und mit Label (b).

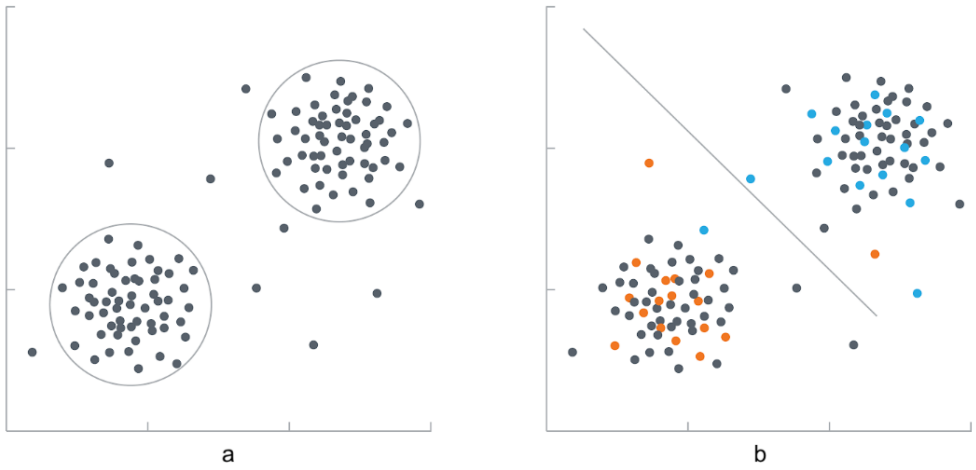


Abbildung 4: Die binäre Klassifikation aus dem Trainingssets ohne Label (a) und mit Label (b) [73]

1.4.2.5 Support Vektor Machine

Gemäß [17], [25] ist Support Vector Machine (SVM) ein überwachtes Lernmodell, das auf einem nicht-probabilistischen binären linearen Klassifikator basiert, der sowohl für Klassifizierungs- als auch für Regressionsprobleme verwendet werden kann.

Um Daten in zwei Klassen zu kategorisieren, wird eine Hyperebene dargestellt, so dass der Abstand der Punkte von beiden Klassen von der Hyperebene maximal ist. Die Hyperebene ist die Trennungsfläche beider Klassen, der eine Dimension kleiner als der Features-Raum hat. Beispielsweise ist die Hyperebene für 2-dimensionale Feautures-Vektor eine Linie. [20], [27]

SVM kann für linear trennbare Daten verwendet werden. Wie in der Abbildung 5 dargestellt, hat die Hyperebene (b) die größte Marge beider Klassen. Die Punkte, die den Linien a und c am nächsten liegen, stellen die Supportvektoren dar. [20]

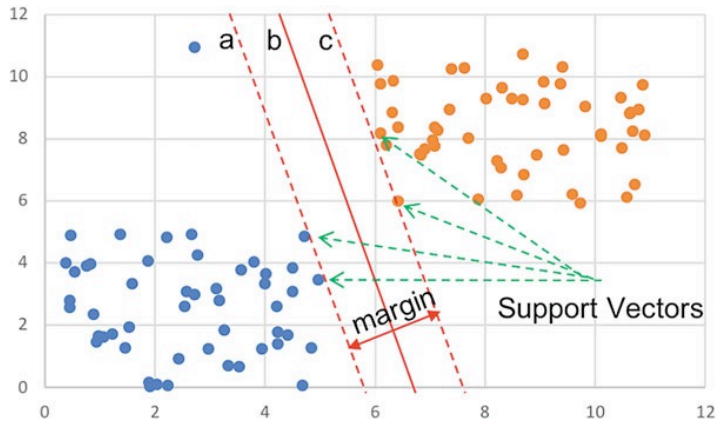


Abbildung 5: Support Vektor Machine [20]

Für nichtlinear trennbare Datenpunkte (linke Seite von Abbildung 6), verwendet SVM die Kernelfunktion, um Datenpunkte in einen höher dimensionalen Raum abzubilden und linear trennbar zu machen (rechte Seite von Abbildung 6). [20], [25]

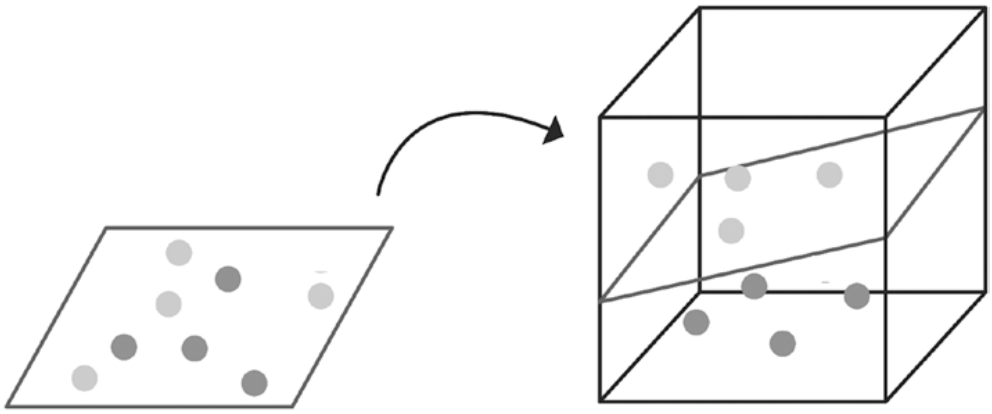


Abbildung 6: Verwendung der Kernelfunktion in SVM [25]

SVMs nehmen numerische Eingaben vor und arbeiten gut mit kleinen Datensets. Mit zunehmender Dimensionalität reduziert sich jedoch die Fähigkeit, das Modell zu verstehen und zu erklären. Mit zunehmender Größe des Datensets kann auch die Trainingszeit steigen. SVMs sind auch bei verrauschten Daten weniger leistungsfähig. [25]

1.4.2.6 Naiver Bayes-Klassifikator

Nach [20], [25], [29] ist naiver Bayes ein probabilistischer Ansatz, der die Unabhängigkeit der Variablen voraussetzt. Naive Bayes verwendet das Bayes-Theorem (Formel 2), um die Wahrscheinlichkeit zu berechnen, dass ein Ereignis eintritt, wenn bereits ein anderes Ereignis eingetreten ist.

$$P(c|x) = \frac{P(x|c) \times P(c)}{P(x)}$$

Formel 2: Bayes-Theorem [20]

P (c | x): Wahrscheinlichkeit von c, wenn x bereits beobachtet wurde.

P (x | c): Wahrscheinlichkeit von x, wenn c passiert ist.

P (c) und P (x) bezeichnen die Wahrscheinlichkeit von c bzw. x.

Anhand von Tabelle 1 als einfachem Beispiel können wir die Wahrscheinlichkeit des Krankheitsrisikos eines Patienten daraus schlussfolgern, ob er ungesund lebt oder nicht. [25]

Tabelle 1: Ergebniswahrscheinlichkeit [25]

AUFZEICHNUNG	UNGESUNG	AUSGANG
PERSON 1	Ja	Hohes Risiko
PERSON 2	Ja	Geringes Risiko
PERSON 3	Ja	Hohes Risiko
PERSON 4	Nein	Geringes Risiko

$$P(\text{hohes Risiko} \mid \text{ungesund}) = \frac{P(\text{ungesund} \mid \text{hohes Risiko}) \times P(\text{hohes Risiko})}{P(\text{ungesund})} = \frac{\frac{2}{2} \times \frac{2}{4}}{\frac{3}{4}}$$

Naive Bayes-Modelle sind einfach zu bauen und besonders nützlich für umfangreiche Datensets. Neben seiner Einfachheit (und damit seiner Naivität) ist naive Bayes dafür bekannt, selbst hoch entwickelte Klassifizierungsmethoden zu übertreffen. [25]

1.4.2.7 K-Nearest-Neighbor

Nach [20], [27], [30] wird der Algorithmus von K-Nearest-Neighbor (KNN) sowohl für Regressionsprobleme als auch für Klassifizierungsprobleme verwendet. Um das Label des neuen Datenpunkts zu bestimmen, werden dessen Koordinaten mit den K-Punkten verglichen, die dem neuen Datenpunkt am nächsten liegen. Die Abbildung 7 veranschaulicht das Prinzip des KNN für Klassifizierungsprobleme (hier ist K 3). Bei Regressionsproblemen wird der Durchschnitt der Labelwerte und bei Klassifizierungsproblemen die Mehrheit berücksichtigt.

Ein Nachteil von KNN liegt darin, dass die beste Wahl von k datenabhängig ist, d.h. für diese Entscheidung sind Vorkenntnisse erforderlich. [30]

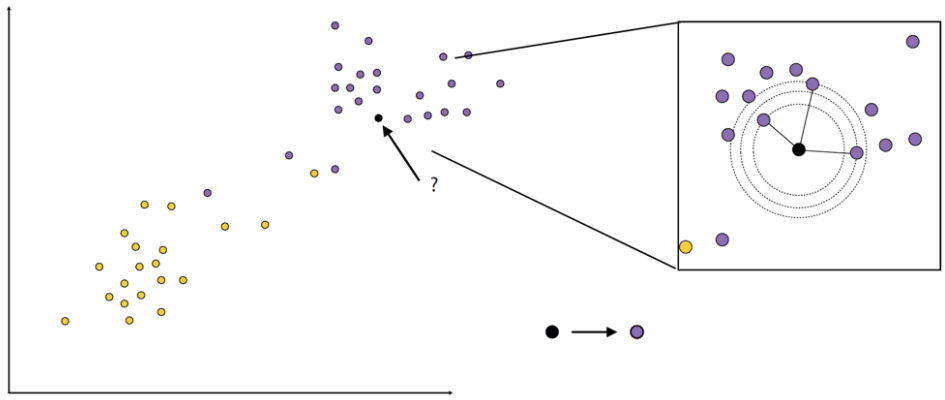


Abbildung 7: KNN-Klassifikator [27]

1.4.2.8 Entscheidungsbaum

Nach [17], [25] sind Entscheidungsbäume Flussdiagramme, die sowohl für Regressionsprobleme als auch für Klassifizierungsprobleme verwendet werden. Bei

Klassifikationsbäumen wird eine kategorische Ausgabe und bei Regressionsbäumen eine numerische Ausgabe erzeugt.

Ein Entscheidungsbaum besteht aus Knoten und Kanten, die die Knoten verbinden. [20] Jede Verzweigung des Baumes entspricht der Beobachtung eines Merkmals. [27] Die Struktur des Entscheidungsbaums basiert auf einer einfachen Reihe von Fragen, die seriell zu einer Antwort führen, die am besten zu den im Training verwendeten Daten passt. Die Fragen, die die Knoten eines Baumes zu einem Pfad führen, wurden mit Hilfe von "if a then x else y" Modellen gestellt. [20] Der oberste Knoten von Entscheidungsbäumen ist die Wurzelknoten. Ein nicht aufgeteilter Zweig ist ein Endknoten oder ein Blattknoten, der ein Klassenlabel als Ausgabe der Vorhersage festlegt. [20], [25]

Die Entscheidungen, die an jedem Knoten getroffen werden, müssen nicht unbedingt binäre Entscheidungen sein, aber aus praktischen Gründen werden sie in der Regel mit binären Entscheidungen konstruiert. [20]

Als Beispiel ist der Entscheidungsbaum in Abbildung 8 für die vereinfachten Datasets in die Tabelle 2 dargestellt. [25]

Die zentrale Frage beim Lernen von Entscheidungsbäumen ist, welche Knoten an welcher Stelle platziert werden sollen. Entscheidungsbäume bieten Vorteile bei der Darstellung großer Datasets und der Priorisierung der wichtigsten Features. [25]

Tabelle 2: Risiko des Typ-2-Diabetes [25]

AUFZEICHNUNG	UNGESUNG	BMI	AUSGANG
PERSON 1	Ja	>25	Hohes Risiko
PERSON 2	Ja	<25	Geringes Risiko
PERSON 3	Ja	>25	Hohes Risiko
PERSON 4	Nein	<25	Geringes Risiko
PERSON 5	Nein	>25	Mittleres Risiko

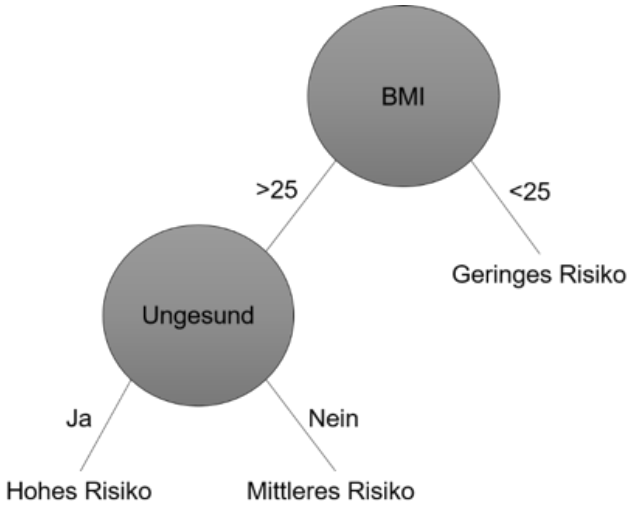


Abbildung 8: Beispiel für einen Entscheidungsbaum für Tabelle 2 [25]

1.4.2.9 Clustering

Gemäß [16], [30] ist Clustering eine unüberwachte Lernmethode, die Cluster oder Gruppen innerhalb des Datasets anhand ihrer Ähnlichkeiten herausfindet. Beim Clustering ist es entscheidend, dass die Daten innerhalb desselben Clusters enger miteinander verbunden sind als mit Daten aus anderen Clustern, oder anders gesagt, dass die Ähnlichkeit im gleichen Cluster maximiert werden soll. K-Means ist ein Clustering-Verfahren, das verwendet wird, wenn die Anzahl der Cluster im Voraus bekannt ist. [24]

K-Means:

Nach [24], [25], [31] zielt K-Means Clustering darauf ab, k ähnliche Gruppen innerhalb der Daten zu finden. Der K-Means-Algorithmus basiert auf einem iterativen Verfahren mit dem Ziel, den Abstand zwischen den Daten und dem Schwerpunkt jedes Clusters zu minimieren. Abbildung 9 zeigt ein Beispiel für K-Means-Verfahren mit K=2. Zunächst werden die beiden Clustermittelpunkte zufällig oder manuell initialisiert (Abbildung 9-a). Dann werden alle Daten zum nächstgelegenen Clustermittelpunkt klassifiziert (Abbildung 9-b) und die neue Clustermittelpunkte werden entsprechend den Daten jedes Clusters definiert (Abbildung 9-c). Dieser Prozess wird wiederholt, bis die Clustermittelpunkte unverändert bleiben (Abbildung 9-d). Es besteht keine Konvergenzgarantie für k-Means-Verfahren, aber die Anzahl der erforderlichen Iterationen ist in der Regel viel geringer als die Anzahl der Datenpunkte.

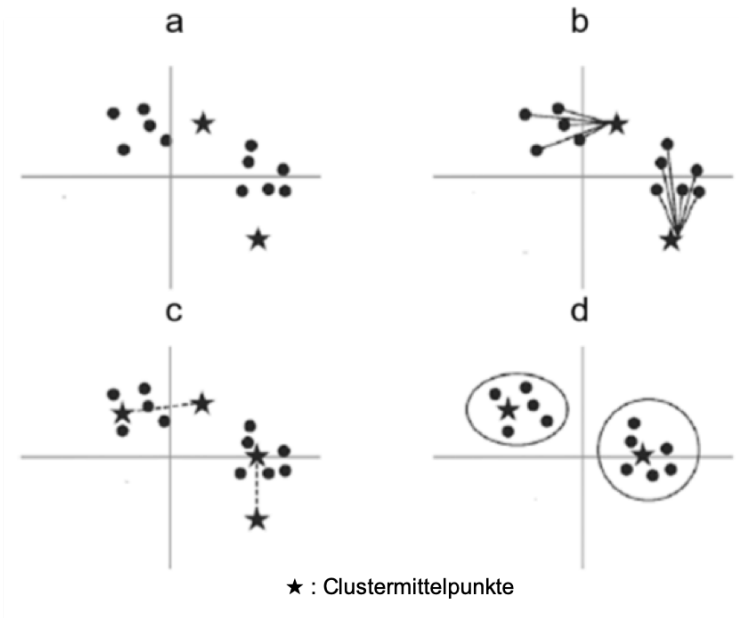


Abbildung 9: K-Means-Clustering [25]

1.4.2.10 Random Forest:

Random Forest (RF) ist eine Ensemblemethode, die mehrere Entscheidungsbäume kombiniert, um eine Überanpassung (engl. Overfitting) zu vermeiden und die Vorhersage über einen einzigen Entscheidungsbaum zu verbessern. [20], [27], [29]

Nach [20], [25], [27] Die Idee von RF ist es, jeden Baum mit einer Stichprobe des Datasets und einer zufälligen Teilmenge von Merkmalen zu erstellen und zu trainieren. RF kann für Klassifizierungs- und Regressionsprobleme verwendet werden. Bei Regressionsproblemen wird ein Durchschnitt der Ergebnisse gebildet, während bei Klassifizierungsproblemen die Mehrheit der Ergebnisse oder Stimmen gebildet wird.

Überanpassung:

Nach [25], [31] ist die Überanpassung eine zu genaue Spiegelung der Trainingssets bei der Entwicklung eines Modells, das gut auf Trainingssets funktioniert und schlechte Leistungen für neue Daten liefert.

Ensemblemethode:

Nach [25], [32] ist die Ensemblemethode definiert als das Erstellen mehrerer Modelle für dieselbe Aufgabe, um einen genaueren Lernenden zu erstellen.

1.4.2.11 Künstliches neuronales Netz:

Das künstliche neuronale Netz (KNN) (engl. artificial neural network) oder das neuronale Netz (NN) ist ein Informationsverarbeitungssystem, das aus mehreren miteinander verbundenen Rechenneuronen besteht, die vom biologischen Neuronalen Netz des Gehirns inspiriert sind. Neuronen übertragen Informationen und sind durch Verbindungsglieder, sogenannte Gewichtungen, miteinander verbunden. Die Anzahl der Neuronen in einem Modell ist gleich der Anzahl der Variablen in den Daten. Die Anordnung der Neuronen in Schichten und die Verbindungsmuster innerhalb und zwischen den Schichten werden allgemein als Netzwerkarchitektur bezeichnet. Abbildung 10-a zeigt ein einfaches einschichtiges KNN mit drei Input-Neuronen (X_1, X_2, X_3) und einem Output-Neuron (Y). Die Gewichte an den Verbindungen von X_1, X_2 und X_3 zum Neuron Y werden durch w_1, w_2 und w_3 angegeben. Ein einschichtiges KNN hat eine Schicht von Verbindungsgewichten. Abbildung 10-b zeigt ein Beispiel für ein mehrschichtiges KNN. Ein mehrschichtiges KNN ist ein Netzwerk mit einer oder mehreren Schichten von Neuronen zwischen der Eingabe- und der Ausgabeschicht. [25], [30], [33]

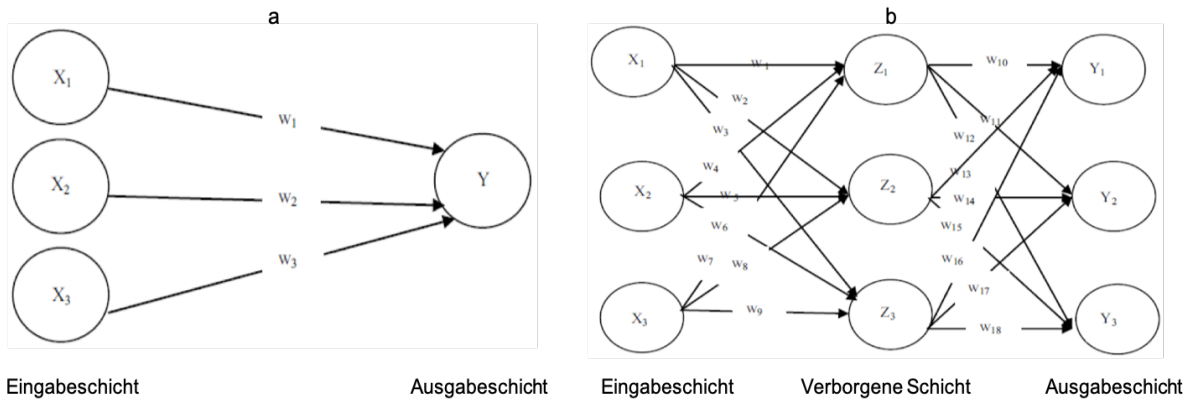


Abbildung 10: Künstliches neuronales Netz [33]

1.4.2.12 Feedforward-Neuronales Netz:

Nach [24], [25], [30] ist das Feedforward-Neuronales Netz (FNN) die einfachste Netzwerkarchitektur von KNN, deren Neuronen ein azyklisches Diagramm aufweisen, in dem sich Informationen nur in eine Richtung von der Eingangsschicht zur Ausgangsschicht bewegen. Die Abbildung 11 zeigt die allgemeine Architektur eines mehrschichtigen FNN mit zwei verborgenen Schichten.

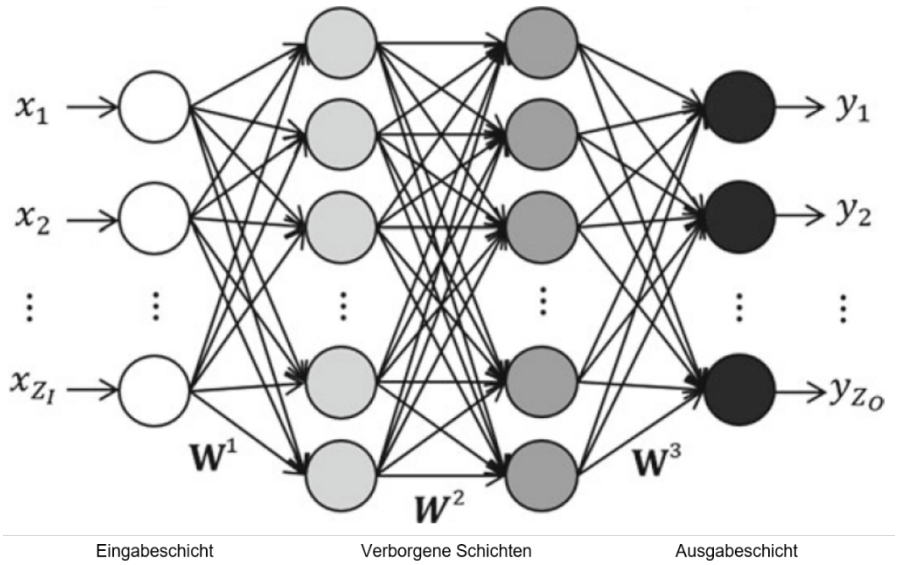


Abbildung 11: Feedforward-Neuronales Netz [30]

1.4.2.13 Recurrentes neuronales Netz

Nach [25], [30] speichert ein Recurrentes neuronales Netz (RNN) die Ausgabe einer Schicht und gibt sie an den Eingang zurück, um die Vorhersagen der Ausgangsschicht zu verbessern (siehe Abbildung 12). Dementsprechend ist der Informationsfluss für RNN nicht nur vorwärts (wie FNN), sondern auch rückwärts. Somit verfügt jeder Knoten über Speicher und verwendet bei der Durchführung von Berechnungen sequentielle Informationen.

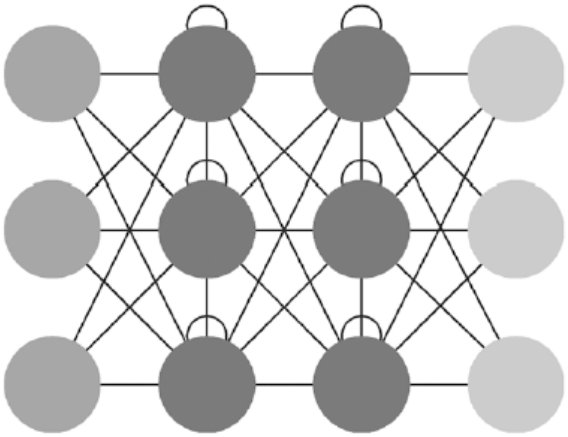


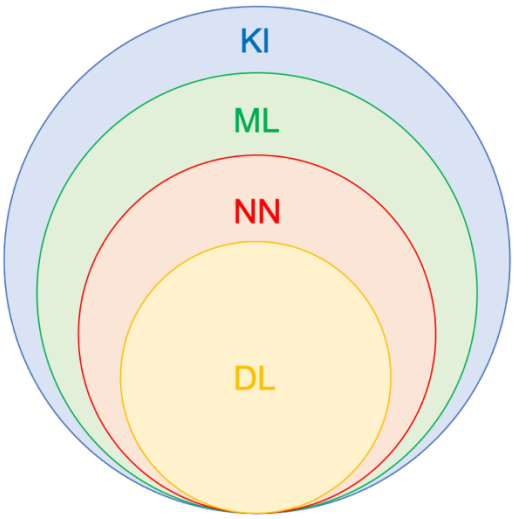
Abbildung 12: Recurrentes neuronales Netz [25]

1.4.2.14 Deep Learning

Gemäß [30], [31] ist Deep Learning (DL) eine Klasse von maschinellen Lernmethoden, die aus mehreren Verarbeitungsschichten bestehen. Die verschachtelte und hierarchische Architektur von DL ermöglicht es, aus Rohdaten abstrakte Darstellungen auf hohem Niveau zu lernen. Die Darstellungen auf jeder Ebene sind relativ einfach und die Stärke liegt in der tiefen Architektur. RNN ist eine Klasse von DL-Modellen.

1.4.3 Kategorisierung von Algorithmen des maschinellen Lernens

Die Abbildung 13 und Abbildung 14 zeigen die Beziehungen und Kategorisierungen, die auf Basis der im Abschnitt 1.4 definierten Begriffe abgeleitet werden.



KI: Künstliche Intelligenz
NN: Neuronale Netz
ML: Maschinelles Lernen
DL: Deep Learning

Abbildung 13: Elemente und Subelemente der KI (in Anlehnung von [34])

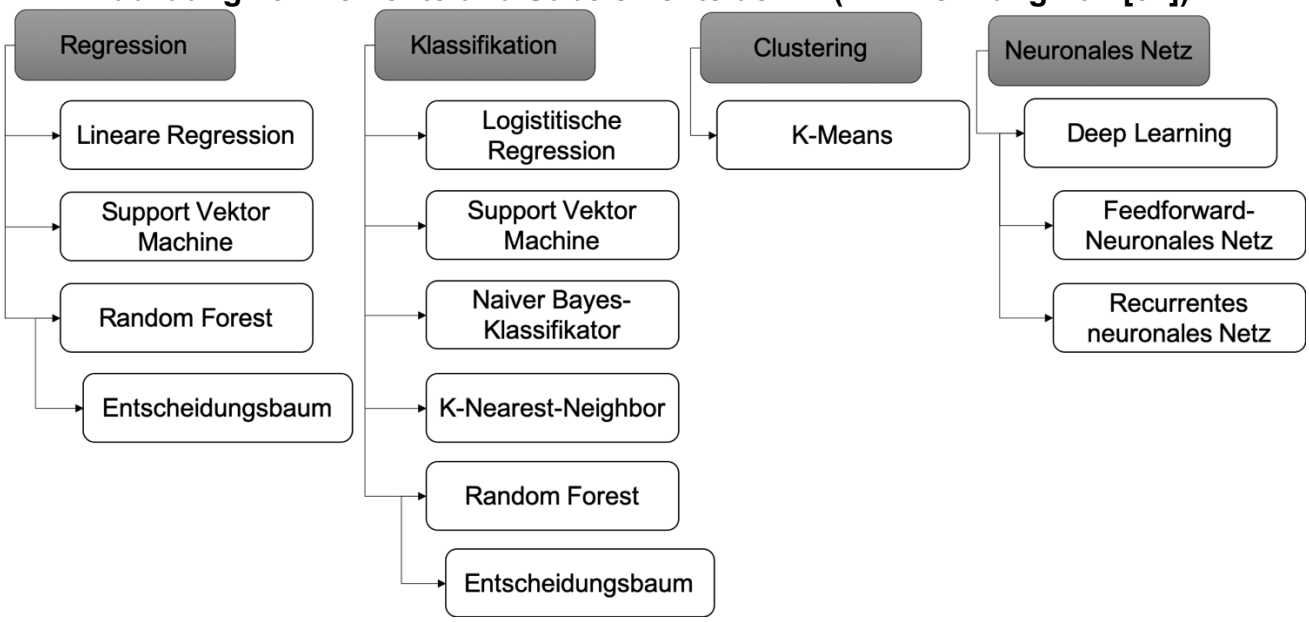


Abbildung 14: Kategorisierung der Modelltypen und Algorithmen von ML

1.5 Anwendungen des maschinellen Lernens in der Medizintechnik

In diesem Abschnitt werden einige Anwendungen des maschinellen Lernens in der Medizintechnik vorgestellt, die nach ihrem Verwendungszweck in die drei Gruppen Diagnostik, Therapie und Prävention eingeteilt sind.

1.5.1 Diagnostik

Erkennung von Lungenentzündungen:

Rajpurkar et al. [35] haben mit Deep Learning einen Algorithmus (CheXNet) zur Erkennung von Lungenentzündungen entwickelt. CheXNet wurde mit über 100.000 Röntgenbilder erstellt, und hat eine höhere Leistung im Vergleich zu einer Gruppe von Radiologen bewiesen.

Elektrokardiogramm-Analyse:

Unter Verwendung eines Deep Neural Networks wurde ein Algorithmus zur Erkennung und Klassifizierung von Herzrhythmusstörungen entwickelt. Die durchschnittliche Genauigkeit des entwickelten Algorithmus hat die Genauigkeit der Diagnose der teilnehmenden Kardiologen in allen Rhythmusklassen übertroffen. [36]

Erkennung von Hautkrebs:

Esteva et al. [37] haben mit klinischen Bildern unter Verwendung des Deep Convolutional Neural Networks (CNN) einen Algorithmus trainiert, der in der Lage ist, Hautkrebs mit einem Qualifikationsniveau zu klassifizieren, das mit Dermatologen vergleichbar ist.

Brustkrebsdiagnose:

Support Vector Machine in Kombination mit der Feature Selection wurde von [38] zur Diagnose von Brustkrebs im Frühstadium verwendet. Der entwickelte Algorithmus hat eine nachgewiesene hohe Genauigkeit (99,5%).

Gesundheitshelferin:

Ada [39] ist eine KI-basierte Applikation für das Smartphone, die Feedback und Beratung auf der Grundlage persönlicher Gesundheitsdaten bietet.

Medizinische Text- und Bildanalyse:

ExB [40] ist eine kognitive Workbench, die KI- und maschinelle Lernalgorithmen zur Analyse von Texten und Bildern verwendet. Die Anwendung von ExB in der Medizin besteht darin, Patientenakten, Laborberichte und medizinische Publikationen zu lesen, zusammenzufassen und auszuwerten.

Medizinische Bildanalyse:

Basierend auf KI und Deep Learning entwickelt Heuro Labs [41] eine modularisierte und skalierbare Plattform mit dem Ziel, gelernte Modelle aus kleineren Datensätzen zu entwickeln. Im Bereich der Medizin wurde diese Plattform für die Analyse und Klassifizierung von radiologischen Bildern genutzt.

Früherkennung und Diagnostik von Krebserkrankungen:

MeVis [42] ist eine Software-Applikation für die Bildverarbeitung in der Medizin. Im Mittelpunkt dieser Software steht die Früherkennung und Diagnose von Brust-, Lungen-, Leber-, Prostata- und Darmkrebs sowie neurologischen Erkrankungen. Diese Software unterstützt alle bildgebenden Verfahren wie Magnetresonanztomographie, digitale Mammographie und Computertomographie.

1.5.2 Therapie

Therapieunterstützung:

Arya [43] ist eine KI-basierte Mood Tracker (Aufzeichnung der Stimmung) zur Unterstützung der Therapie von psychisch kranken Menschen basierend auf ihren Verhaltensmustern.

Individuelle Therapie für Diabetiker:

Xbird [44] ist eine medizinische KI-basierte Software, die eine individuelle Therapie für Diabetiker auf Basis von Daten von Smartphones und tragbaren Sensoren anbietet.

1.5.3 Prävention

Prävention von Hypoxämie während der Operation:

Nach [45] ist Hypoxämie definiert als Sauerstoffmangel im arteriellen Blut. Basierend auf Algorithmen des maschinellen Lernens haben Lundberg et al. [46] ein System entwickelt, das das Risiko einer Hypoxämie vorhersagt und die Risikofaktoren während der Vollnarkose in Echtzeit erkennt. Das System, das aus der elektronischen

Patientenakte von über 50.000 Operationen trainiert wurde, verbesserte die Leistungsfähigkeit der Anästhesisten.

In den vorherigen Kapiteln wurden die Grundlagen des maschinellen Lernens und seine Anwendungen in der Medizintechnik erläutert, was hilft, im nächsten Kapitel ein Assistenzsystem für die medizinische Diagnostik zu konzipieren und einen Algorithmus des maschinellen Lernens basierend auf den zu definierenden Anforderungen des Assistenzsystems auszuwählen.

2 Konzeption

In dieser Arbeit wird ein Assistenzsystem konzipiert, welches eine intelligente medizinische Diagnostik ermöglicht. Hiermit wird zunächst die notwendige Charakteristik des Systems sowie der Charakter eines Assistenzsystems in Unterkapitel 2.1 ermittelt. Das Assistenzsystem muss die Krankheitsrisiken und Präventivmaßnahmen basierend auf den Gesundheitsdaten der Benutzer ermitteln. Zu diesem Zweck wird für jede Krankheit ein Vorhersagemodell mittels maschineller Lernalgorithmen und Trainingsdaten in das Assistenzsystem integriert. Welche Eigenschaften das Vorhersagemodell besitzt, wird im Unterkapitel 2.2 erläutert. Das Unterkapitel 2.2 besteht aus vier Abschnitten.

Die Vorhersagemodelle werden basierend auf Trainingsdaten erstellt. Da in dieser Arbeit keine echten Gesundheitsdaten vorliegen, müssen sie simuliert werden. Im ersten Abschnitt wird erläutert, welche Abgrenzungen für die Simulation der Gesundheitsdaten verwendet werden. Im zweiten Abschnitt wird festgelegt, welcher Lernstil und welche Algorithmen zur Erstellung der Vorhersagemodelle verwendet werden. Im dritten Abschnitt wird die Funktionsweise des Vorhersagemodells näher betrachtet. Anschließend werden im vierten Abschnitt einige Möglichkeiten zur Verbesserung und Weiterentwicklung des Vorhersagemodells aufgeführt.

Im Unterkapitel 2.3 werden zwei Szenarien zur Nutzung des Assistenzsystems konzipiert. Schließlich befasst sich das Unterkapitel 2.4 mit der Konzeption der Schnittstelle zwischen dem Benutzer und dem Assistenzsystem

2.1 Konzeption eines Assistenzsystems mittels maschineller Lernalgorithmen

Nach [47] ist ein Assistenzsystem ein flexibles und anpassungsfähiges System zur betrieblichen Informations- und Entscheidungsunterstützung von Entscheidungsträgern. In einem Assistenzsystem können die Regeln zur Entscheidungsunterstützung unter Verwendung von maschinelles Lernalgorithmen erweitert und optimiert werden. Wie bereits für die Grundlagen in Unterkapitel 1.4 angekündigt, kann sich ein System durch die Verwendung von ML-Algorithmen erweitern und verbessern.

Im Rahmen dieser Arbeit soll ein Konzept für eine softwarebasierte intelligente Diagnostik entwickelt werden. Dieses Softwaresystem kann ein Assistenzsystem sein, welches die Algorithmen des maschinellen Lernens beinhaltet und basierend auf gegebenen Gesundheitsdaten die Krankheitsrisiken und Präventivmaßnahmen für den Benutzer veranschaulicht.

Dieses Assistenzsystem wird automatisiert für jede Krankheit ein Vorhersagemodell basierend auf Trainingsdaten generieren können. Diese Vorhersagemodelle müssen immer in der Lage sein, sich zu erweitern bzw. zu verbessern, wenn im System weitere

Gesundheitsdaten von Benutzern eingegeben werden. Dieses System kann auf einem Rechner als Software installiert werden und die Benutzer können ihre Gesundheitsdaten über die Benutzeroberfläche eingeben. Dementsprechend werden den Benutzern die Krankheitsrisiken und Präventivmaßnahmen als Output über das Assistenzsystem zur Verfügung gestellt.

Im nächsten Unterkapitel wird die Konzeption der Vorhersagemodelle ausführlich beschrieben.

2.2 Konzeption der Vorhersagemodelle

Mit Hilfe von Algorithmen des maschinellen Lernens kann, basierend auf Trainingsdaten, für jede Krankheit ein Modell erstellt werden. Jedes Modell vergleicht nach der Eingabe der Gesundheitsdaten des Benutzers die Ähnlichkeit zwischen den Gesundheitsdaten und den Trainingsdaten und stellt das Risiko für jede Krankheit fest. Darüber hinaus kündigen die Modelle priorisierte Präventivmaßnahmen an, die auf dem Ausmaß des Einflusses von einzelnen Gesundheitsdaten auf den gegebenen Risiken basieren. Die Modelle werden aufgrund ihrer präventiven Funktionalität als Vorhersagemodelle bezeichnet. Um das Konzept in einem Prototyp umzusetzen, konzentriert sich diese Arbeit auf die Erstellung von Vorhersagemodellen für drei häufige Krankheiten.

Laut [48] hatte in den letzten 6 Monaten des Jahres 2018 fast die Hälfte der Befragten ein- oder zweimal eine Erkältung, was dazu führte, dass die Erkältung als erste Krankheit ausgewählt wurde. Gemäß [49] leiden in Deutschland 24,5% der Männer und 21,5% der Frauen unter Bluthochdruck. Aus diesem Grund wurde er als zweite Krankheit ausgewählt. Als dritte Krankheit wurde die Hypercholesterinämie gewählt, weil nach [49] 32,2% der Männer und 34,9% der Frauen in Deutschland einen zu hohen Cholesterinspiegel haben.

Zur Begrenzung der Einflussfaktoren auf die ausgewählten Krankheiten im zu entwickelnden Prototyp wurden zehn identische Gesundheitsfaktoren, nämlich (1) Geschlecht, (2) Körpertemperatur, (3) Ruheherzfrequenz, (4) Aktivität, (5) Alter, (6) BMI (Body-Mass-Index), (7) Schlafdauer, (8) Wasserkonsum, (9) Nikotinkonsum und (10) Alkoholkonsum, als Einflussfaktoren betrachtet, die nach [49]–[53] die größten Auswirkungen auf diese Krankheiten haben.

Darüber hinaus werden die Gesundheitsdaten innerhalb von 10 Tagen vor Auftreten der Krankheit berücksichtigt, um den Einfluss der einzelnen Gesundheitsdaten im Laufe der Zeit zu untersuchen. Tabelle 3 zeigt, wie beispielsweise die Trainingsdaten der Vorhersagemodelle aussehen können.

Um die Vorhersagemodelle mittels maschineller Lernalgorithmen im Assistenzsystem zu erstellen, muss eine hohe Anzahl von Trainingsdaten in der Form von Tabelle 3 verfügbar sein. Im nächsten Abschnitt geht es um das Konzept der Erstellung von Trainingsdaten.

Tabelle 3: Trainingsdaten der Vorhersagemodelle

		x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	y_1	y_2	y_3
		Geschlecht	Körper temperatur	Ruhe herzfrequenz	Aktivität	Alter	BMI	Schlaf dauer	Wasser konsum	Nikotin konsum	Alkohol konsum	Erkält-ung	Blut hoch-druck	Hyper choles-terinämie
Patient 1	T1	W	37	83	60 M	30 J	18,6	7 S	1 L	nein	nein	✓		
	T2	W	38	80	30 M	30 J	18,6	8 S	0,5 L	nein	0,5 L Bier			
	…	…	…	…	…	…	…	…	…	…	…			
	T10	W	37	81	10 M	30 J	18,6	7 S	1,5 L	nein	2 Schnaps			
Patient 2	T1	M	36,7	96	30 M	45 J	21	7 S	1,3 L	ja	nein		✓	✓
	T2	M	37,2	94	10 M	45 J	21	7 S	1,5 L	ja	1,5 L Bier			
	…	…	…	…	…	…	…	…	…	…	…			
	T10	M	36,9	95	8 M	45 J	21	8 S	1 L	ja	1 L Bier			
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
Patient n	T1	W	37	89	15 M	19 J	23	8 S	1 L	nein	nein		✓	
	T2	W	36,8	91	22 M	19 J	23	8 S	0,8 L	nein	nein			
	…	…	…	…	…	…	…	…	…	…	…			
	T10	W	37,3	90	40 M	19 J	23	7 S	1,4 L	nein	nein			

Abkürzungen: T= Tag M= Minute S= Stunde L= Liter J= Jahr W= weiblich M= männlich

2.2.1 Trainingsdaten der Vorhersagemodelle

Da die realen Gesundheitsdaten in dieser Arbeit für die Erstellung der Vorhersagemodelle nicht verfügbar sind, werden sie unter Berücksichtigung der folgenden Punkte simuliert.

- Da die Gesundheitsdaten für alle drei Vorhersagemodelle gleich sind ($x_1, x_2, \dots, x_{10}$ in Tabelle 3), werden für jede Krankheit relevante und irrelevante Gesundheitsdaten ermittelt.
- Positive und negative Einflüsse der Gesundheitsdaten auf jede Krankheit werden untersucht.
- Die normalen und abnormalen Grenzwerte der Gesundheitsdaten für jede Krankheit werden festgelegt.
- Für jedes Vorhersagemodell werden zwei Gruppen von Gesundheitsdaten, eine mit Gesund-Labels und eine weitere mit Krank-Labels, simuliert. Die Werte von y_1, y_2 und y_3 in Tabelle 3 wurden als Gesund- und Krank-Labels bezeichnet.
- Für alle Gesund-Labels wird null und für alle Krank-Labels wird eins als Wert eingesetzt.
- Gesundheitsdaten werden in mehreren Dekaden von Datasets simuliert, wobei jedes Dataset die Gesundheitsdaten eines Tages darstellt. Eine Dekade der Datasets präsentiert den zehntägigen Verlauf (T1, T2, …, T10 in Tabelle 3) von Gesundheitsdaten einer Person.
- Bei der Simulation von Gesundheitsdaten in den Dekaden der Datensets wird davon ausgegangen, dass Alter, Geschlecht und BMI innerhalb von 10 Tagen unverändert bleiben.
- Die Simulation von Gesundheitsdaten wird im Rahmen der festgelegten Grenzwerte randomisiert durchgeführt.
- Irrelevante Gesundheitsdaten einer Krankheit werden so simuliert, dass sie keine abnormalen Grenzwerte für andere Krankheiten aufweisen.

2.2.1.1 Untersuchung der relevanten Einflussfaktoren und deren Grenzwerte für jede Krankheit

Im Folgenden werden die Einflüsse der Gesundheitsdaten ($x_1, x_2, \dots, x_{10}$ in Tabelle 3) auf Erkältungen, Bluthochdruck und Hypercholesterinämie erforscht.

Bei der Untersuchung der Einflussfaktoren auf die Erkältung ist zu berücksichtigen, dass diese Krankheit nach [54], [55] eine virusbedingte Entzündung ist. Folglich liegt der Fokus bei der Bestimmung der Einflussfaktoren darauf, wie sich Gesundheitsdaten auf die Funktion des Immunsystems auswirken.

Geschlecht

Gemäß [48] zeigten die Ergebnisse einer Umfrage, dass die Anzahl der Frauen und Männer, die in einem bestimmten Zeitraum eine Erkältung hatten, fast gleich war. Basierend darauf wurde in dieser Arbeit angenommen, dass das Geschlecht ein irrelevanter Faktor bei Erkältungen ist.

Nach [49] steigt die Zahl der Männer ab 55 Jahren und der Frauen ab 65 Jahren mit Bluthochdruck am schnellsten im Vergleich zu anderen Lebensabschnitten. Außerdem werden Anzahl von Männern ab 45 und Frauen ab 55 Jahre alt mit Hypercholesterinämie fast verdoppelt.

Körpertemperatur

Die normale Körpertemperatur beträgt nach [56] von 36,6 bis 37,5 °C. Forscher der Yale University haben herausgefunden, dass mit abnehmender Körpertemperatur die angeborene Immunantwort auf Viren schwächer wird.[57]

Aufgrund fehlender Informationen in den Literaturen über den Einfluss der Körpertemperatur auf den Bluthochdruck und die Hypercholesterinämie wurde sie in dieser Arbeit als irrelevanter Faktor für diese Krankheiten angesehen.

Ruheherzfrequenz

Nach [50], [53], [58]–[60] ist die Ruheherzfrequenz der Herzschlag in der körperlichen Ruhe, der im Normalzustand zwischen 60 und 100 Schlägen pro Minute liegt. Die Annäherung des Wertes an die untere Grenze zeigt die höhere Aktivität des Körpers und die weniger stressige Phase, was zur Verringerung des Risikos für alle betrachteten Krankheiten führt.

Aktivität

Nach [53] sind mindestens 30 Minuten Bewegung pro Tag (wie ein einfacher Spaziergang) erforderlich, um die Funktion des Immunsystems aufrechtzuerhalten. Darüber hinaus bewirkt die Erhöhung der Aktivität zu einer besseren Funktion des Immunsystems. Laut [61], [62] führt Bewegungsmangel zu Bluthochdruck und Hypercholesterinämie.

Alter

Es wurde angenommen, dass das Assistenzsystem von Jugendlichen, jungen Erwachsenen und Erwachsenen benutzt wird, was einem Alter von 14 bis 64 Jahren

nach [63] entspricht. Gemäß [53] nimmt die Immunabwehr ab dem 60. Lebensjahr ab, was die Anfälligkeit für Erkältungen erhöht. Der Einfluss des Alters auf Bluthochdruck und Hypercholesterinämie wurde zusammen mit dem Einfluss des Geschlechts auf diese Krankheiten untersucht.

BMI

Gemäß [64] wird der BMI nach Formel 3 berechnet. Ein BMI von weniger als 18,5 kg/m^2 ist als Untergewicht, von 18,5 bis 24,9 kg/m^2 als Normalgewicht und von mehr als 25 kg/m^2 als Übergewicht definiert.

$$BMI(kg/m^2) = Gewicht\ (kg)/Größe^2(m^2)$$

Formel 3: BMI-Berechnung

Unter- und Übergewicht haben laut [53] einen negativen Einfluss auf die Erkältung. Gemäß [61], [62] haben über die Hälfte der übergewichtigen Menschen einen hohen Blutdruck. Außerdem erhöht ein Übergewicht das Risiko einer Hypercholesterinämie.

Schlafdauer

Die normale Schlafdauer wurde nach [65] von 6 bis 10 Stunden pro Tag definiert. Der Schlafmangel führt nach [53], [66], [67] zu einer Schwächung des Immunsystems. Es erhöht auch das Risiko von Hypercholesterinämie und Bluthochdruck.

Wasserkonsum

Der Körper benötigt nach [68] mindestens 1,5 Liter Flüssigkeit pro Tag. Eine Abnahme des Wasserhaushalts des Körpers vermindert laut [53] die Funktionalität des Immunsystems. Wegen des Informationsdefizits über den Einfluss des Wasserkonsums auf die Hypercholesterinämie und den Bluthochdruck in der Literatur wurde er in dieser Arbeit als irrelevanter Faktor für beide Krankheiten angesehen.

Nikotinkonsum

Das Rauchen hat nach [49], [53], [61] negative Auswirkungen auf die Immunabwehr. Es erhöht auch das Risiko von Bluthochdruck und Hypercholesterinämie.

Alkoholkonsum

Laut [53] hat mehr als 25 g Alkoholkonsum pro Tag negative Folgen für das Immunsystem. Der Begriff "chronischer Alkoholabusus" wurde nach [53], [69] definiert als täglicher Alkoholkonsum von mehr als 40 g bei Frauen und mehr als 60 g bei Männern, was das Risiko von Bluthochdruck und Hypercholesterinämie erhöht. In dieser Arbeit wurde 50 g Alkoholkonsum als die Grenze des schlechten Einflusses auf den Bluthochdruck und die Hypercholesterinämie angesehen.

Die relevanten und irrelevanten Gesundheitsdaten und deren Grenzwerte wurden für jede Krankheit in Tabelle 4 zusammengefasst.

Tabelle 4: Die Zusammenfassung der Grenzwerte von Gesundheitsdaten

	Erkältung	**Bluthochdruck**	**Hypercholesterinämie**
Geschlecht	irrelevant	↑ Männer ≥ 55 Jahre alt ↑ Frauen ≥ 65 Jahre alt	↑ Männer ≥ 45 Jahre alt ↑ Frauen ≥ 55 Jahre alt
Körpertemperatur	↑ 35°C - 36,5 °C	irrelevant	irrelevant
Ruheherzfrequenz	↑ >100 bpm	↑ >100 bpm	↑ >100 bpm
Aktivität	↓ >30 min pro Tag	↓ >30 min pro Tag	↓ >30 min pro Tag
Alter	↑ >60 Jahre alt	↑ Männer ≥ 55 Jahre alt ↑ Frauen ≥ 65 Jahre alt	↑ Männer ≥ 45 Jahre alt ↑ Frauen ≥ 55 Jahre alt
BMI	↑ >25 kg/m^2 ↑ <18,5 kg/m^2	↑ >25 kg/m^2	↑ >25 kg/m^2
Schlafdauer	↑ <6 Stunden pro Tag	↑ <6 Stunden pro Tag	↑ <6 Stunden pro Tag
Wasserkonsum	↑ <1500 ml pro Tag	irrelevant	irrelevant
Nikotinkonsum	↑ ≥ 1 Zigarette pro Tag	↑ ≥ 1 Zigarette pro Tag	↑ ≥ 1 Zigarette pro Tag
Alkoholkonsum	↑ >25 g Alkoholgehalt pro Tag	↑ >50 g Alkoholgehalt pro Tag	↑ >50 g Alkoholgehalt pro Tag
Dieser Gesundheitsfaktor erhöht (↑) oder verringert (↓) das Risiko der Krankheit innerhalb der festgelegten Grenzwerte			

Nach der Einführung der Methodik zur Erstellung von Trainingsdaten für das Assistenzsystem werden im nächsten Abschnitt die Auswahlkriterien der Lerngorhythmen zur Verarbeitung dieser Trainingsdaten ausführlich beschrieben.

2.2.2 Auswahlkriterien des Lernalgorithmus

Der Entscheidungsbaum in Abbildung 15 wurde erstellt, um die Entscheidung über den Lernstil des maschinellen Lernens zu treffen, der in dieser Arbeit verwendet wird.

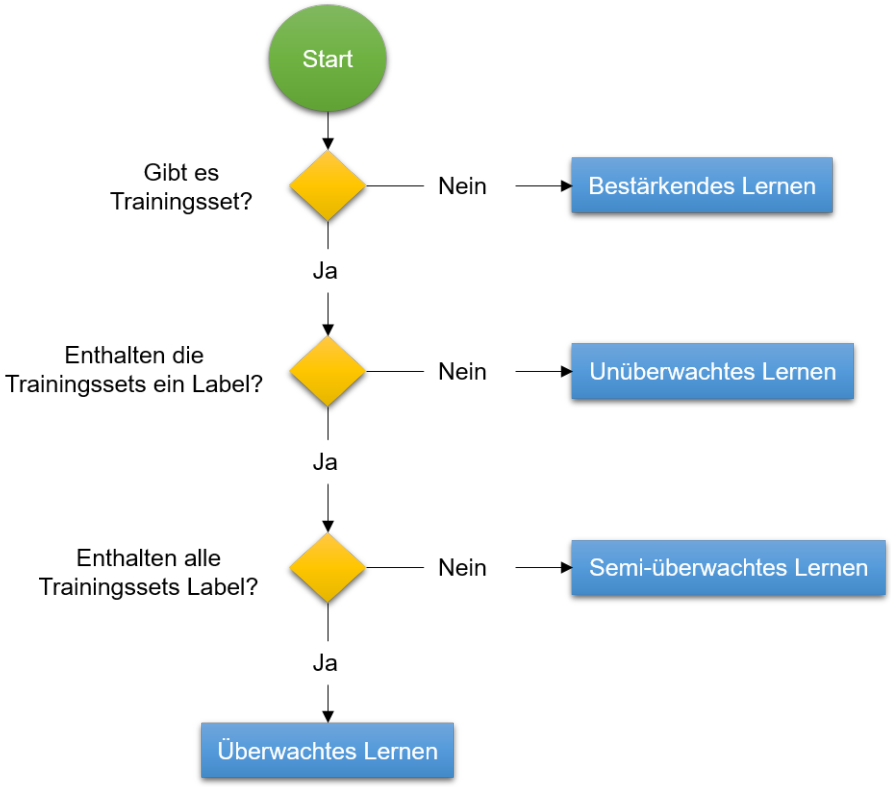


Abbildung 15: Die Auswahl des Lernstils von ML

Bei der Entwicklung der Vorhersagemodelle liegt der Fokus auf dem Einfluss von Gesundheitsdaten auf bestimmten Krankheiten. Wie in Tabelle 3 dargestellt, werden alle Trainingsdaten mit den Labels (y_1, y_2 und y_3) simuliert. Daraus lässt sich schließen, dass überwachtes Lernen in dieser Arbeit als Lernstil verwendet werden soll.

Zur Auswahl des passenden Algorithmus wurde ein Entscheidungsbaum entwickelt, der in Abbildung 16 dargestellt ist.

Bei der Auswahl des Lernalgorithmus muss zunächst das gewünschte Format der Ergebnisse festgelegt werden. Wie in die Grundlagen (im Abschnitt 1.4.2.1) ausführlich beschreiben, werden kontinuierliche Werte durch eine Regression vorhergesagt. Andererseits kategorisiert die Klassifikation die Ergebnisse in verschiedene Klassen. Beispielsweise können durch die Anwendung der Klassifikation in dieser Arbeit die Krankheitsrisiken in 3 Klassen kategorisiert werden, wie z.B. geringes Risiko, mittleres Risiko und hohes Risiko. Auf der anderen Seite können die Krankheitsrisiken durch den Einsatz der Regression in Prozent bestimmt werden. Da ein kontinuierlicher Wert in dieser Arbeit einen genaueren Einblick für die Benutzer schaffen kann, wird ein Regressionsalgorithmus (roter Bereich der Abbildung 16) verwendet.

Zur Auswahl einer der vier Optionen (Random Forest, Neuronales Netz, Lineare Regression und Entscheidungsbaum) sind zwei kritische Punkte zu unterscheiden:

Geschwindigkeit für die Erstellung des Vorhersagemodells und Genauigkeit des Vorhersagemodells. Alle diese Algorithmen können theoretisch im Anwendungsfall dieser Arbeit verwendet werden. Da jedoch die Gesundheitsdaten in dieser Arbeit mit hoher Genauigkeit simuliert werden, ist es optimaler, einen Algorithmus zu verwenden, der mit höherer Geschwindigkeit das Vorhersagemodell erzeugt. Nach [70] werden die Modelle mit den linearen Regressions- und Entscheidungsbaum-Algorithmen schneller trainiert als mit dem neuronalen Netz und Random Forest-Algorithmen.

In diesem Schritt stehen nur zwei Algorithmen zur Auswahl, nämlich die lineare Regression und der Entscheidungsbaum. Um einen geeigneten Algorithmus zwischen diesen beiden zu wählen, ist es notwendig zu prüfen, ob eine lineare Approximation akzeptabel ist. Nach [25] ist linearer Approximation akzeptabel, wenn y aus einer linearen Kombination von Eingangsgrößen $(x_1, x_2, \dots, x_n)$ berechnet werden kann. Formel 4 dient dazu, einen allgemeinen Überblick über diese Bedingung zu schaffen.

$$h(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

Formel 4: Hypothese Funktion

$h(x)$: geschätzter Wert von y

$\theta_0, \theta_1, \dots, \theta_n$: Parameter des Modells

$x_1, x_2, \dots, x_n$: Features oder Eingangsgrößen

n : Anzahl der Features

Die Hypothese Funktion, die später näher ermittelt wird, schätzt anhand von Parametern des Modells $(\theta_0, \theta_1, \dots, \theta_n)$, wie sich jedes Feature in y auswirkt. Das bedeutet, dass mit einem trainierten Modell, das auf linearer Regression basiert, das Ausmaß des Einflusses von Gesundheitsdaten auf jede Krankheit bestimmt werden kann, was genau das gewünschte Ergebnis dieser Arbeit ist. Daher wird unter der Annahme der linearen Approximation die lineare Regression als ML-Algorithmus für die Erstellung von Vorhersagemodellen ausgewählt.

Die bessere Leistung der linearen Regression im Vergleich zum Entscheidungsbaum für kontinuierliche Variablen wurde auch in [71] bestätigt.

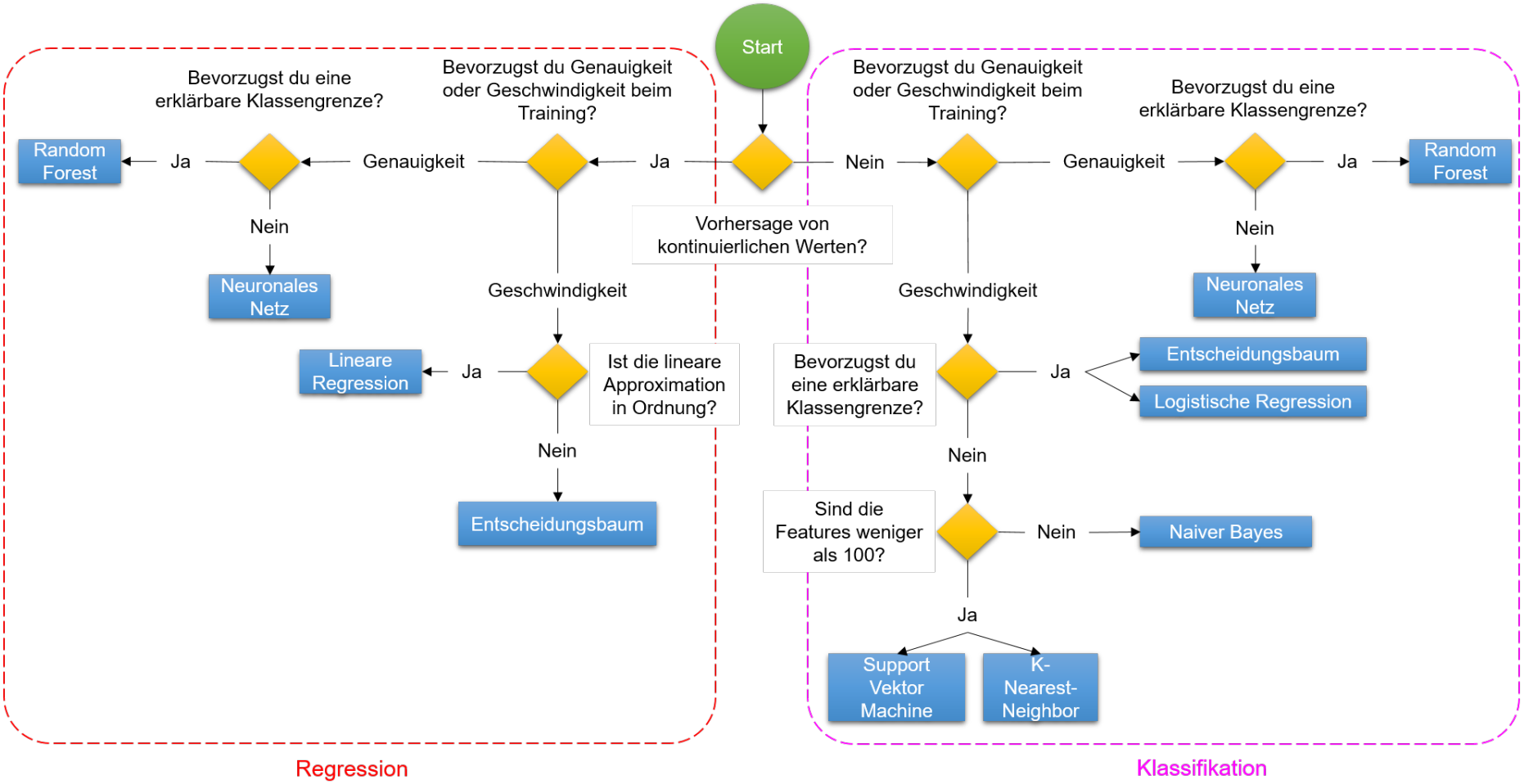


Abbildung 16: Die Auswahl des Algorithmus von ML (in Anlehnung von [70], [74])

2.2.3 Funktionsweise des Vorhersagemodells

Das Vorhersagemodell jeder Krankheit sollte gemäß der Aufgabenstellung dieser Arbeit nach der Eingabe der Gesundheitsdaten das Krankheitsrisiko und die hierarchischen Präventionsmaßnahmen bestimmen. Um das Vorhersagemodell dieser Eigenschaften zu erzeugen, wird für jedes Vorhersagemodell eine mathematische Funktion basierend auf den Trainingsdaten unter Verwendung der linearen Regression mit mehreren Variablen erstellt. Wie diese mathematischen Funktionen bei jedem Vorhersagemodell aussehen, zeigt Formel 5.

$$h_1(x) = \theta_{1,0} + \theta_{1,1}x_1 + \theta_{1,2}x_2 + \cdots + \theta_{1,10}x_{10}$$

$$h_2(x) = \theta_{2,0} + \theta_{2,1}x_1 + \theta_{2,2}x_2 + \cdots + \theta_{2,10}x_{10}$$

$$h_3(x) = \theta_{3,0} + \theta_{3,1}x_1 + \theta_{3,2}x_2 + \cdots + \theta_{3,10}x_{10}$$

Formel 5: Die mathematischen Funktionen der Vorhersagemodelle

$h_1(x)$: Risiko von Erkältung

$h_2(x)$: Risiko von Bluthochdruck

$h_3(x)$: Risiko von Hypercholesterinämie

$\theta_{1,0}, \theta_{1,2}, \ldots, \theta_{3,10}$: lineare Regressionsparameter

x_1 : Geschlecht x_2 : Körpertemperatur x_3 : Ruheherzfrequenz x_4 : Aktivität

x_5 : Alter x_6 : BMI x_7 : Schlafdauer x_8 :Wasserkonsum

x_9 : Nikotinkonsum x_{10}: Alkoholkonsum

Zur Vereinfachung der Darstellung und Berechnung wurden die linearen Regressionsparameter der einzelnen Vorhersagemodelle als Vektor ϑ_1, ϑ_2 und ϑ_3 präsentiert. Formel 6 zeigt, wie die Vektoren aussehen.

$$\vartheta_1 = [\theta_{1,0}\ \theta_{1,1}\ \theta_{1,2}\ \ldots \theta_{1,10}]$$

$$\vartheta_2 = [\theta_{2,0}\ \theta_{2,1}\ \theta_{2,2}\ \ldots \theta_{2,10}]$$

$$\vartheta_3 = [\theta_{3,0}\ \theta_{3,1}\ \theta_{3,2}\ \ldots \theta_{3,10}]$$

Formel 6: Lineare Regressionsparameter

Die linearen Regressionsparameter (ϑ_1, ϑ_2 und ϑ_3) wurden basierend auf Trainingsdaten ermittelt. Die Größe der einzelnen Elemente von ϑ_1, ϑ_2 und ϑ_3

verdeutlicht das Ausmaß des Einflusses einzelner Gesundheitsdaten $(x_1, x_2, \dots, x_{10})$ auf das Krankheitsrisiko. Anhand linearer Regressionsparameter können die Risiken der Krankheiten für neue Gesundheitsdaten wie Formel 7 berechnet werden. X_{neu} steht für die Vektordarstellung neuer Gesundheitsdaten $(x_{1,neu}, x_{2,neu}, \dots, x_{10,neu})$. Zur Anpassung der Matrixdimensionen wurde X_{neu} um ein Element $(x_0 = 1)$ erweitert.

$$X_{neu} = \begin{bmatrix} x_0 \\ x_{1,neu} \\ x_{2,neu} \\ \dots \\ x_{10,neu} \end{bmatrix}$$

$$h_1(x) = \vartheta_1 \times X_{neu} = \left[\theta_{1,0}\ \theta_{1,1}\ \theta_{1,2}\ \dots \theta_{1,10}\right]_{(1\times 11)} \times \begin{bmatrix} x_0 \\ x_{1,neu} \\ x_{2,neu} \\ \dots \\ x_{10,neu} \end{bmatrix}_{(11\times 1)}$$
$$= \theta_{1,0}x_0 + \theta_{1,1}x_1 + \theta_{1,2}x_2 + \dots + \theta_{1,10}x_{10}$$

$$h_2(x) = \vartheta_2 \times X_{neu}$$

$$h_3(x) = \vartheta_3 \times X_{neu}$$

$$h_n(x) = \vartheta_n \times X_{neu} = \theta_{n,0}x_1 + \theta_{n,1}x_1 + \theta_{n,2}x_2 + \dots + \theta_{n,10}x_{10} \qquad (n = 1,2,3)$$

Formel 7: Berechnung der Krankheitsrisiken

Wenn die Elemente der Formel des Krankheitsrisikos $(\theta_{n,0}x_1 + \theta_{n,1}x_1 + \theta_{n,2}x_2 + \dots + \theta_{n,10}x_{10})$ nach Größe geordnet werden, werden die das Krankheitsrisiko beeinflussenden Faktoren hierarchisch dargestellt. Folglich werden Präventivmaßnahmen als Maßnahmen zur Reduzierung der Werte dieser geordneten Größe definiert.

2.2.4 Verbesserung und Weiterentwicklung des Vorhersagemodells

Die simulierten Trainingsdaten sind die Basis für die Erstellung der Vorhersagemodelle. Durch den Einsatz des Assistenzsystems bei den Benutzern werden die Trainingsdaten der zugehörigen Vorhersagemodelle ständig erweitert. Bei der Erweiterung der Trainingsdaten ist zu berücksichtigen, dass aufgrund der Auswahl des überwachten Lernens als Lernstil der Vorhersagemodelle alle Trainingsdaten über Labels verfügen müssen. Aus diesem Grund können die Gesundheitsdaten als weitere Trainingsdaten verwendet werden, die Informationen zum Gesundheitszustand (y_1, y_2 und y_3 in Tabelle 3) enthalten. Um die Vorhersagemodelle weiterzuentwickeln, werden die Regressionsparameter (ϑ_1, ϑ_2 und ϑ_3 in Formel 6) bei der Erweiterung der Trainingsdaten aktualisiert.

Ein weiterer Aspekt, der die Genauigkeit von Vorhersagemodellen erhöhen kann, ist die Umwandlung von Gesundheitsdaten $(x_1, x_2, \dots, x_{10})$ in einer vergleichbaren Größenordnung. So liegen beispielsweise die Werte von x_8 (Wasserkonsum) zwischen 1000 und 2500 ml, die Werte von x_7 (Schlafdauer) zwischen 6 bis 10 Stunden und die Werte von x_2 (Körpertemperatur) zwischen 35 bis 38 °C. Eine Methode zur Umwandlung von Gesundheitsdaten in vergleichbarer Größenordnung ist die Standardisierung (Formel 8).

$$x_i' = \frac{x_i - \mu}{\sigma}$$

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma = \frac{1}{n-1}\sum_{i=1}^{n} |x_i - \mu|^2$$

Formel 8: Standardisierung der Gesundheitsdaten [30]

x_i': Standardisierter Wert von x_i

μ: Mittelwert von $x_1, x_2, \dots, x_{10} \quad (n = 10)$

σ: Standardabweichung von $x_1, x_2, \dots, x_{10} \quad (n = 10)$

2.3 Konzeption der Nutzung des Assistenzsystems

Es wurden zwei Szenarien definiert, die es den Benutzern ermöglichen, das Assistenzsystem sowohl zu nutzen als auch seine Vorhersagemodelle zu erweitern. Das erste Szenario wurde für Benutzer definiert, die das Assistenzsystem selten nutzen, und das zweite Szenario für Benutzer, die das Assistenzsystem permanent verwenden. Im ersten Szenario gibt der Benutzer die Gesundheitsdaten von maximal 5 Tagen zusammen in das Assistenzsystem ein und erhält Krankheitsrisiken und Präventivmaßnahmen. Beim zweiten Szenario muss der Benutzer vor der ersten Nutzung des Assistenzsystems erstellt werden. Anschließend gibt der bekannte Benutzer seine kontinuierlichen Gesundheitsdaten und auch die aufgetretenen Erkrankungen in das Assistenzsystem ein. Einerseits sind die Ergebnisse des Assistenzsystems in diesem Szenario genauer, da die Gesundheitsdaten über einen längeren Zeitraum betrachtet werden, andererseits verfügen die Gesundheitsdaten in diesem Szenario Informationen über aufgetretene Erkrankungen (y_1, y_2 und y_3 in Tabelle 3), die zur Weiterentwicklung der Vorhersagemodelle anonym gespeichert werden können.

Bei der Eingabe der aufgetretenen Krankheiten werden folgende Punkte berücksichtigt:

- Wenn keine Symptome einer Erkältung vorliegen, bestätigt der Benutzer, dass er keine Erkältung hat. Nach [72] sind Kopfschmerzen, Gliederschmerzen, Halsschmerzen, Schnupfen, Husten und Fieber die Hauptsymptome der Erkältung.
- Wurde Bluthochdruck oder kein Bluthochdruck durch eine ärztliche Untersuchung oder ein mobiles Blutdruckmessgerät bestätigt, trägt der Benutzer im Assistenzsystem ein, ob Bluthochdruck vorhanden ist oder nicht.
- Wenn durch eine Blutuntersuchung eine Hypercholesterinämie oder keine Hypercholesterinämie nachgewiesen wurde, gibt der Benutzer in das Assistenzsystem ein, ob diese Krankheit vorliegt oder nicht.

Bei der Speicherung von Gesundheitsdaten als weitere Trainingsdaten werden folgende Punkte berücksichtigt:

- Wenn der Benutzer bestätigt, dass eine Krankheit nicht vorliegt und vor dieser Bestätigung die Gesundheitsdaten der letzten 10 Tage eingegeben wurden, werden die Gesundheitsdaten der letzten 10 Tage mit Gesund-Label in den Trainingsdaten des Vorhersagemodells dieser Krankheit gespeichert.
- Nach der Bestätigung einer Krankheit müssen auch die Gesundheitsdaten der letzten 10 Tage vorhanden sein, um die Gesundheitsdaten dieses Zeitraums mit Krank-Label in Trainingsdaten des zugehörigen Vorhersagemodells zu speichern.

Der Ablauf der Nutzung des Assistenzsystems im ersten und zweiten Szenario wurde in Sequenzdiagrammen in den Abbildung 17 und Abbildung 18 dargestellt. Im zweiten Sequenzdiagramm wurde nur ein Vorhersagemodell und eine Bestätigung der aufgetretenen Krankheit durch den Benutzer gezeigt. Bei x_{10} (Alkoholgehalt) ist zu beachten, dass der Alkoholgehalt in Gramm in das System eingegeben werden muss. Da dieser Wert für den Benutzer unklar sein kann, wird er vom System selbst berechnet. Dafür wird der Benutzer die getrunkene Menge des Bieres, Whiskeys und Weins ($x_{10,1}, x_{10,2}$ und $x_{10,3}$) in das System eingeben. Der Alkoholgehalt-Rechner berechnet x_{10} basierend auf $x_{10,1}, x_{10,2}$ und $x_{10,3}$ unter Berücksichtigung der Volumenprozente der einzelnen Getränke. Das System wurde nur auf diese drei alkoholischen Getränke beschränkt, weil sie nach Ansicht der Autorin die am häufigsten konsumierten Getränke sind.

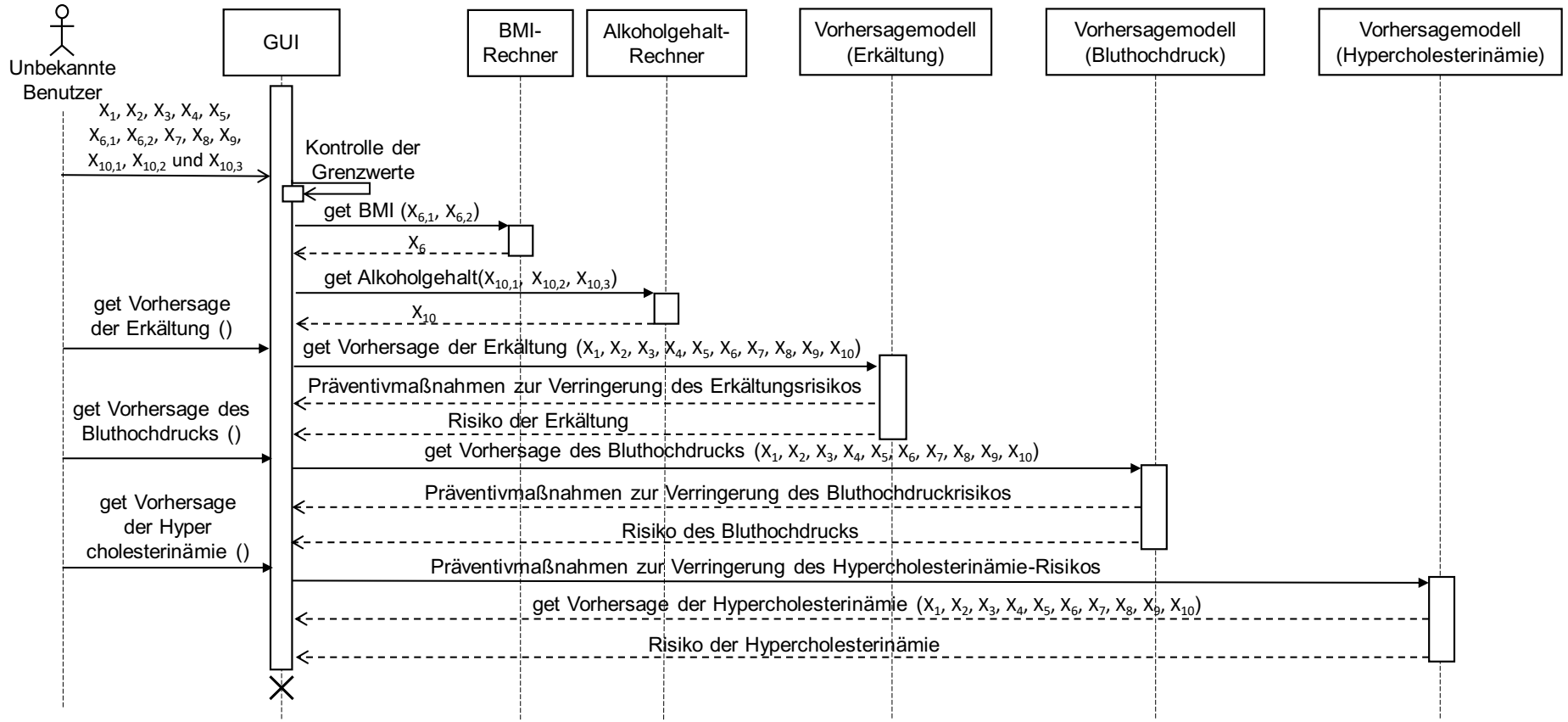


Abbildung 17: Das Sequenzdiagramm des ersten Szenarios

X_1: Geschlecht
X_2: Körpertemperatur
X_3: Ruheherzfrequenz
X_4: Aktivität
X_5: Alter
$X_{6,1}$: Körpergröße
$X_{6,2}$: Gewicht
X_6: BMI
X_7: Schlafdauer
X_8: Wasserkonsum
X_9: Anzahl der gerauchten Zigaretten
$X_{10,1}$: Menge des Bierkonsums
$X_{10,2}$: Menge des Whiskey-Konsums
$X_{10,3}$: Menge des Wein-Konsums
X_{10}: Alkoholgehalt

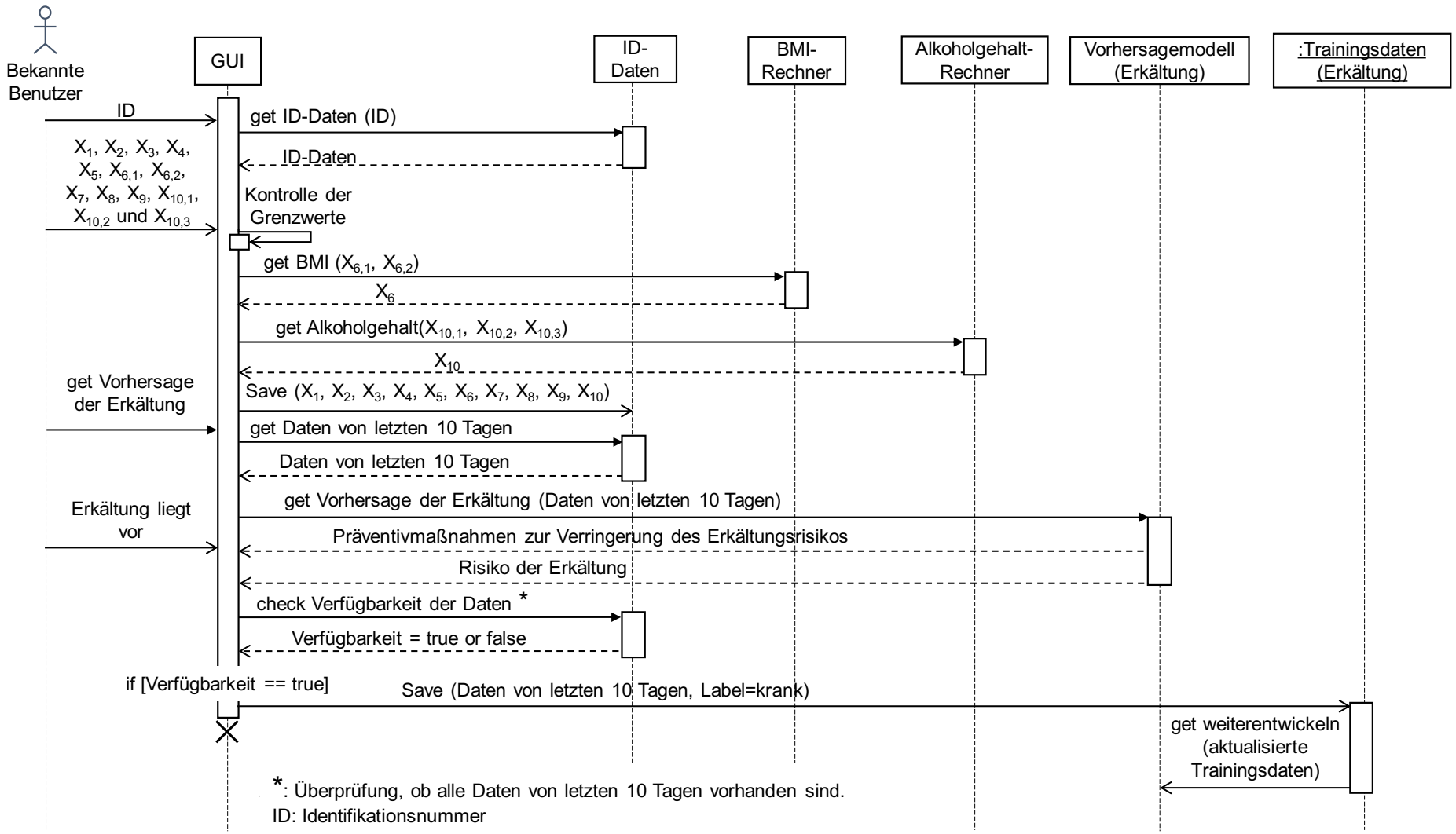


Abbildung 18: Das Sequenzdiagramm des zweiten Szenarios

2.4 Konzeption der Schnittstelle

Um die Gesundheitsdaten des Benutzers in das Assistenzsystem einzugeben und die Ergebnisse darzustellen, muss über eine Schnittstelle eine Verbindung zwischen dem Benutzer und dem Assistenzsystem hergestellt werden.

In dieser Arbeit wurde eine GUI (Graphical User Interface) als Schnittstelle verwendet, die die Gesundheitsdaten des Benutzers nach Überprüfung der Grenzwerte an den Vorhersagenmodellen des Assistenzsystems weiterleitet und die Ergebnisse der einzelnen Vorhersagemodelle nach Auswahl der Krankheit durch den Benutzer veranschaulicht.

Zur Unterstützung der im Unterkapitel 2.3 definierten Szenarien ist die GUI in drei Fenster unterteilt: Vorhersage-, Benutzerdefinition- und kontinuierliche Vorhersage und Weiterentwicklung-Fenster.

Vorhersage-Fenster

Dieser Teil wird im ersten Nutzungsszenario verwendet und enthält 3 Bereiche. Im ersten Bereich werden die über einen kurzen Zeitraum konstant bleibenden Gesundheitsfaktoren wie Geschlecht, Alter und BMI eingegeben. Die restlichen Gesundheitsdaten, welche maximal über Daten von 5 Tagen verfügen, wurden im zweiten Bereich eingegeben. Im dritten Bereich wird der Benutzer eine Krankheit auswählen und das Assistenzsystem wird die damit verbundenen Krankheitsrisiken und Präventivmaßnahmen veranschaulichen.

Benutzerdefinition-Fenster

Um das Assistenzsystem basierend auf dem zweiten Szenario nutzen zu können, muss der Benutzer in diesem Bereich zunächst erstellt werden und eine Identifikationsnummer vom Assistenzsystem erhalten.

Kontinuierliche Vorhersage und Weiterentwicklung-Fenster

Dieses Fenster besteht aus 5 Bereichen. Der erste Bereich dient zur Eingabe der Identifikationsnummer. Im zweiten Bereich werden bereits eingegebene Gesundheitsdaten der Identifikationsnummer dargestellt. Neue Gesundheitsdaten können im dritten Bereich eingegeben werden. Im vierten Bereich wird der Benutzer nach dem im Unterkapitel 2.3 beschriebenen Prinzip eingeben, ob eine Krankheit vorliegt oder nicht. Anschließend wird im fünften Bereich eine Krankheit ausgewählt und das damit verbundene Krankheitsrisiko und die Präventionsmaßnahmen dargestellt.

3 Prototyp

Zur Evaluierung des beschriebenen Konzepts in dieser Arbeit wurde ein Assistenzsystem als Prototype in MATLAB entwickelt. Zur Entwicklung des Assistenzsystems wurde zunächst die Softwarearchitektur des Assistenzsystems auf Basis der geschilderten Eigenschaften, Funktionalitäten und Einschränkungen in der Konzeption entworfen. Im Unterkapitel 3.1 wird die Softwarearchitektur, die Softwarekomponenten sowie die Beziehungen zwischen den Komponenten erläutert. Anschließend wird im Unterkapitel 3.2 die Funktionalität der einzelnen Softwarekomponenten näher beschrieben. Schließlich wird im Abschnitt 3.3 die Installation und Benutzungsanleitung des Assistenzsystems beschrieben.

3.1 Softwarearchitektur

Abbildung 19 gibt einen Gesamtüberblick über die Softwarearchitektur des Assistenzsystems, die aus folgenden Klassen und Daten besteht.

- Klassen: *„GUI“*, *„gradient descent“*, *„feature normalization“*, *„cost function“*, Vorhersagemodelle für Erkältung, Bluthochdruck und Hypercholesterinämie.
- Daten: Daten des Benutzers (Falls Benutzer im System definiert ist), Trainingsdaten für Erkältung, Bluthochdruck und Hypercholesterinämie.

Wie bereits im Abschnitt 2.3 der Konzeption vorgestellt wurde, gibt es zwei Anwendungsszenarien für das Assistenzsystem. Im ersten Szenario gibt der Benutzer der GUI seine Gesundheitsdaten und wählt eine Krankheit zwischen Erkältung, Bluthochdruck und Hypercholesterinämie aus. Das GUI gibt diese Daten an das entsprechende Vorhersagemodell weiter. Das Vorhersagemodell hat die Aufgabe, das Krankheitsrisiko und die Präventivmaßnahmen für die erhaltenen Gesundheitsdaten zu bestimmen. Zur Ermittlung dieser Ergebnisse ruft das Vorhersagemodell *„gradient descent“* auf. Diese Klasse ist für die Anpassung der Regressionsparameter an die entsprechenden Trainingsdaten zuständig. Um die Genauigkeit des Regressionsparameters zu erhöhen, müssen die Trainingsdaten zunächst auf einer vergleichbaren Größenordnung angeordnet werden. Aus diesem Grund ruft die Klasse *„gradient descent“ die Klasse „feature normalization“* auf. *„Feature normalization“* ruft die Trainingsdaten der ausgewählten Krankheit auf und ordnet sie den Trainingsdaten zu. Schließlich gibt sie die zugeordneten Trainingsdaten an *„gradient descent“ weiter*. Eine weitere Klasse im Zusammenhang mit *„gradient descent“* ist die *„cost function“*. Diese Klasse prüft die Genauigkeit des Regressionsparameters, der durch *„gradient descent“* erzeugt wurde. Nachdem *„gradient descent“* den Regressionsparameter berechnet hat, gibt sie diesen dem Vorhersagemodell an. Anhand des erhaltenen Regressionsparameters bestimmt das Vorhersagemodell das Krankheitsrisiko und die Präventionsmaßnahmen. Das Vorhersagemodell gibt dann diese Ergebnisse an die GUI weiter. Im Folgenden veranschaulicht die GUI die Ergebnisse für den Benutzer.

Wenn der Benutzer eine andere Krankheit auswählt, wird der gesamte Vorgang wiederholt.

Im zweiten Szenario muss der Benutzer zunächst seine persönlichen Daten bekannt machen (oder geben). Die GUI erstellt ein neues Objekt zur Speicherung der Benutzerdaten, dass denselben Namen wie der erhaltene Benutzername besitzt. Der im System definierte Benutzer kann nach Eingabe des Benutzernamens die Gesundheitsdaten, das Datum und aufgetretene Krankheiten in die GUI eingeben. Wie im ersten Szenario wählt der Benutzer auch hier eine Krankheit zwischen Erkältung, Bluthochdruck und Hypercholesterinämie aus. Die GUI ruft die historischen Daten des Benutzers auf und speichert in ihr die neuen Gesundheitsdaten mit dem erhaltenen Datum. In diesem Szenario gibt die GUI dem Vorhersagemodell die vorhandenen Gesundheitsdaten innerhalb der letzten 10 Tage weiter. Das Verfahren zur Bestimmung des Krankheitsrisikos und der Präventivmaßnahmen für die ausgewählte Krankheit ist dasselbe wie im ersten Szenario. Falls der Benutzer das Vorhandensein oder Nichtvorhandensein einer Krankheit bestätigt hat (sogenannte Labels), ruft die GUI die entsprechenden Trainingsdaten auf und speichert die Gesundheitsdaten als weitere Trainingsdaten darin.

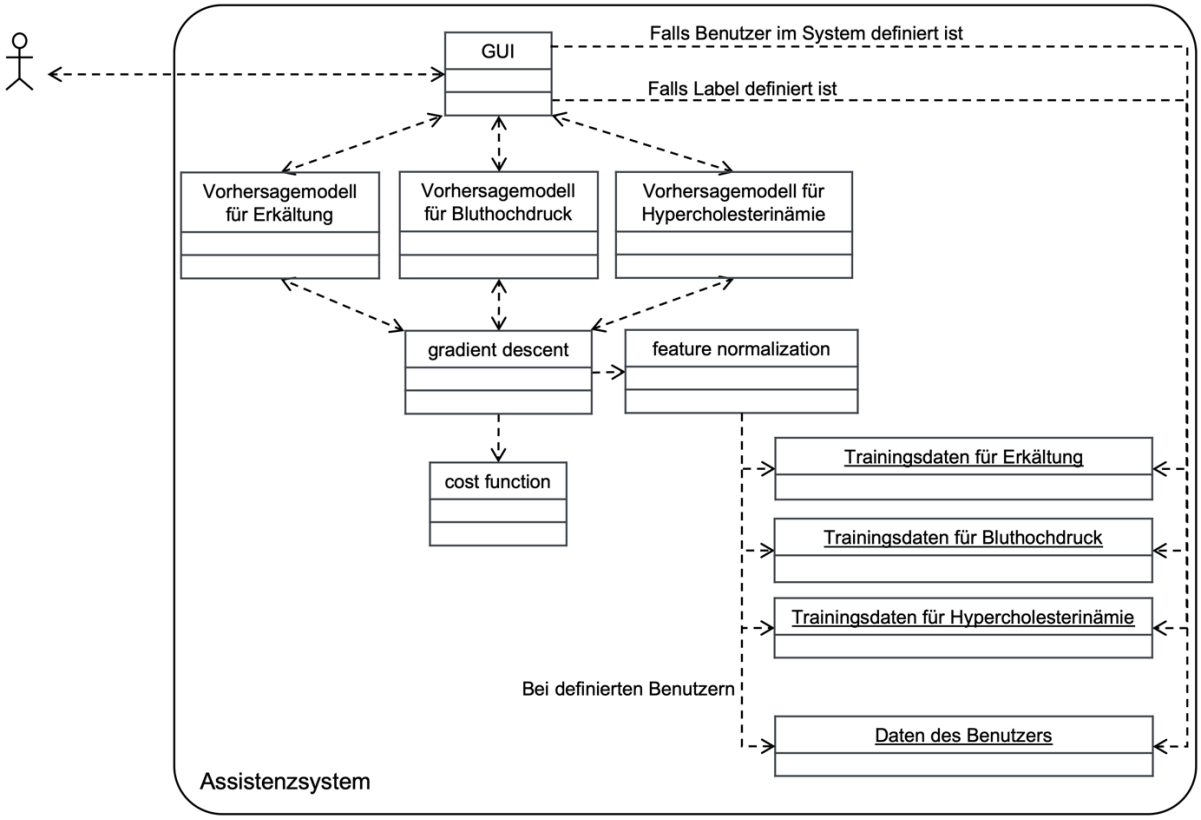


Abbildung 19: Softwarearchitektur

Die Funktionalität der einzelnen Komponenten wird im nächsten Abschnitt im Detail erläutert.

3.2 Beschreibung der Systemkomponenten

In diesem Abschnitt wird ein detaillierter Blick auf die einzelnen Softwarekomponenten des Assistenzsystems geworfen. Außerdem wird dargestellt, wie jede Komponente in MATLAB entwickelt wurde.

3.2.1 GUI

Die GUI des Assistenzsystems besitzt folgende Funktionalitäten:

- Sie schafft die Möglichkeit der Verbindung zwischen dem Benutzer und dem Assistenzsystem. Durch diese Verbindung können die Benutzerdaten empfangen und die Ergebnisse des Assistenzsystems dem Benutzer präsentiert werden.
- Erstellung einer neuen Textdatei bei der Definition des Benutzers im zweiten Anwendungsszenario des Assistenzsystems. Die Eigenschaften dieser Textdatei werden in Abschnitt 3.2.7 ausführlich erläutert.
- Erweiterung der Trainingsdaten bei erhaltenen Gesundheitsdaten mit Labels im zweiten Anwendungsszenario.
- Kontrolle der Grenzen der vom Benutzer erhaltenen Gesundheitsdaten auf Basis definierter normaler und abnormaler Grenzwerte im Unterkapitel 2.2.1.1 der Konzeption.
- Einsetzen des normalen Wertes bei fehlenden Eingaben von Gesundheitsdaten durch den Benutzer.

Die GUI wurde in MATLAB mit dem App-Designer entwickelt. Wie im Unterkapitel 2.4 der Konzeption vorgestellt wurde, verfügt die GUI über 3 Fenster, nämlich das Vorhersage-, das Benutzerdefinition- und das kontinuierliche Vorhersage- und Weiterentwicklungs-Fenster. Wie die einzelnen GUI-Fenster aussehen und wie sie verwendet werden, wird im Abschnitt 3.3 detailliert erklärt.

3.2.2 Gradient descent

Wie in Abschnitt 2.2.3 der Konzeption beschrieben wurde, wird für jedes Vorhersagemodell eine Hypothesenfunktion ($h_n(x)$) basierend auf Trainingsdaten generiert. Die linearen Regressionsparameter (ϑ_n) der Hypothesenfunktionen müssen so gewählt werden, dass die Werte von $h_n(x)$ zumindest nahe an den y-Werten der Trainingsdaten liegen. *„Gradient descent“* dient dazu, den Regressionsparameter zu finden, der zu einem Mindestabstand zwischen $h_n(x)$ und y-Werten führt. Abbildung 20 veranschaulicht, wie diese Klasse in MATLAB entwickelt wurde. Im Folgenden werden die Ein- und Ausgänge dieser Klasse geschildert.

- Eingänge:
  - „X“ ist eine Matrix bestehend aus 10 Spaltenvektoren (x_1 bis x_{10}in Tabelle 3 der Konzeption).

- „y“ ist ein Vektor von y_1, y_2 oder y_3 (siehe Tabelle 3 der Konzeption)
- „theta“ ist ein Nullvektor mit den Dimensionen der Regressionsparameter.
- „alpha“ ist „learning rate“ der linearen Regression. Diese Variable bestimmt, wie weit ϑ_n in jeder Iteration von for-Schleife verändert wird. Im Abschnitt 3.2.6 wird erklärt, wie den Wert von „alpha“ festgelegt wird.
- „num_iters“ ist die Anzahl der Iterationen von for-Schleife. Im Abschnitt 3.2.6 wird erklärt, wie „num_iters“ den Berechnungsablauf beeinflusst und wie sein Wert bestimmt wird.

- Ausgänge:
  - *„theta“* repräsentiert den Regressionsparameter (ϑ_n)
  - *„J_history“* ist das Ergebnis von *„CopmputerCost“* bei jeder Iteration der for-Schleife. *„J_history“* wird später dazu beitragen, geeignete *„alpha“* und *„num_iters“* zu bestimmen. *„CopmputerCost“* ist eine weitere Klasse, die in dieser Klasse aufgerufen wurde. Die Funktionalität des *„CopmputerCost“* wird im Abschnitt 3.2.3 erläutert.

```
function [theta, J_history] = GradientDescent(X, y, theta, alpha, num_iters)
m = length(y);
J_history = zeros(num_iters, 1);
for iter = 1:num_iters
delta = ((theta' * X' - y')*X)';
theta = theta - alpha / m * delta;
J_history(iter) = ComputeCost(X, y, theta);

end

end
```

Abbildung 20: Gradient descent

3.2.3 Cost function

Um den Abstand zwischen der Hypothesenfunktion ($h_n(x)$) und dem y-Werten bei jedem Regressionsparameter (ϑ_n) zu berechnen, wird die Klasse *„cost function“* verwendet. Je kleiner der Abstand zwischen $h_n(x)$ und y ist, desto genauer wurde das Vorhersagemodell an die Trainingsdaten angepasst. Aus diesem Grund wurde die Aufgabe von *„cost function“* als Überprüfung der Genauigkeit des *„gradient descent“* beschrieben. Abbildung 21 zeigt, wie *„cost function“* in MATLAB entwickelt wurde.

```
function J = ComputeCost(X, y, theta)
m = length(y);
J = 0;
J = sum((X*theta - y).^2)/(2*m);

end
```

Abbildung 21: Cost function

3.2.4 Trainingsdaten für Vorhersagemodelle

Die Trainingsdaten wurden nach den im Unterkapitel 2.2.1 der Konzeption beschriebenen Richtlinien generiert. Für jedes Vorhersagemodell wurden 1000 Trainingssets mit Krank- und 1000 Trainingssets mit Gesund-Label generiert. Die Trainingssets mit Krank-Label wurden so generiert, dass in jedem Trainingsset nur ein Faktor abnormale und die anderen normalen Grenzwerte aufweisen. Zur Erhöhung der Genauigkeit der Vorhersagemodelle wurden alle Trainingsdaten mit dem Ursprung Null generiert. So wurden Beispielweise die im Abschnitt 2.2.1.1 festgelegten Grenzwerte der Körpertemperatur, nämlich 35 bis 37,5 °C, durch die Grenzwerte zwischen 0 und 2,5 ersetzt. Anschließend wurden die Trainingsdaten für jedes Vorhersagemodell in separaten Textdateien gespeichert. Abbildung 22 zeigt, wie die Trainingsdaten für das Vorhersagemodells der Erkältung aussehen.

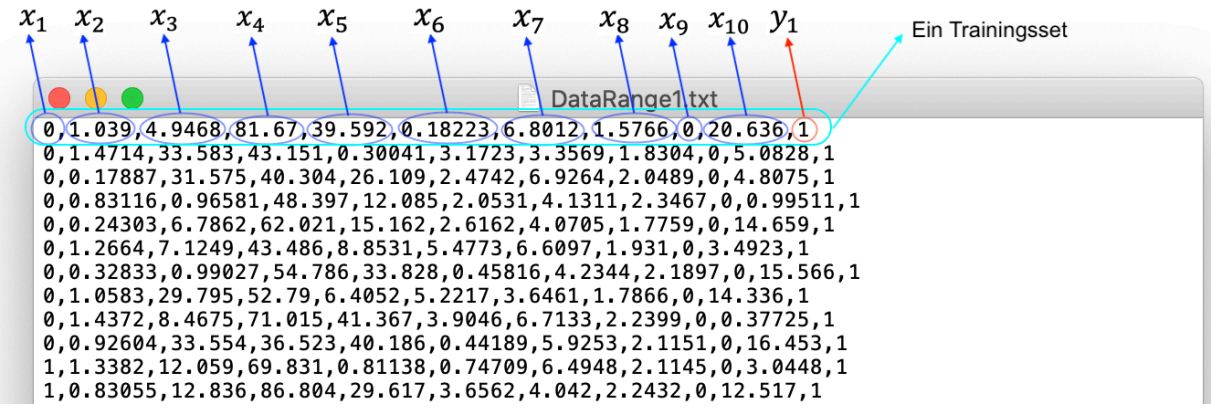


Abbildung 22: Trainingsdaten für das Vorhersagemodell der Erkältung

3.2.5 Feature normalization

Wie im Abschnitt 2.2.4 der Konzeption erläutert, dient die Standardisierung von Gesundheitsdaten als ein weiterer Schritt zur Erhöhung der Genauigkeit von Vorhersagemodellen. Abbildung 23 veranschaulicht, wie die Formel 8 der Konzeption in MATLAB entwickelt wurde.

```
function [X_norm,mu ,sigma] = FeatureNormalization(X)
X_norm = X;
mu = zeros(1, size(X, 2));
sigma = zeros(1, size(X, 2));
if size(X,1)== 1
    X_norm = (X-mean(X))./ std(X);
else
    for i=1:size(X,2)
        mu(i)=mean(X(:,i));
        sigma(i)=std(X(:,i));
        X_norm(:,i) = (X(:,i) - mu(i)) ./ sigma(i);
    end
end

end
```

Abbildung 23: Feature normalization

3.2.6 Vorhersagemodelle

Anhand der Klassen, die in den vorherigen Abschnitten vorgestellt wurden, und der generierten Trainingsdaten werden die Regressionsparameter der Vorhersagemodelle festgelegt. Abbildung 24 zeigt, wie dieser Teil in MATLAB entwickelt wurde.

```
data = load('DataRange1.txt');
X = data(:, 1:10);
y = data(:, 11);
m = length(y);
[X,mu,sigma] = FeatureNormalization(X);
X = [ones(m, 1) X];
alpha = 0.01;
num_iters = 500;
theta = zeros(11, 1);
[theta, J_history] = GradientDescent(X, y, theta, alpha, num_iters);
```

Abbildung 24: Vorhersagemodell

Bei der Bestimmung der Werte von *„alpha“* und *„num_iters“* wurden folgende Punkte berücksichtigt:

- Je kleiner der Wert von *„alpha“* wird, desto langsamer wird der minimale Wert der *„cost function“* erreicht. Mit anderen Worten, ein kleinerer Wert von *„alpha“* erfordert größere *„num_iters“* und folglich eine längere Berechnungszeit.

- Der Wert von *„alpha“* darf nicht zu groß gewählt werden, um sicherzustellen, dass die minimale *„cost function“* für jede Trainingsdaten berechnet werden kann.

- *„alpha“* und *„num_iters“* müssen so gewählt werden, dass die Werte von *„J_history“* zunächst ein langsames Absinken aufweisen und dann unverändert bleiben. Ein langsames Absinken von *„J_history“* zeigt an, dass der Abstand zwischen $h_n(x)$ und dem y-Wert mit jeder Iteration kleiner geworden ist. Gleiche Werte am Ende

des Ablaufs von *„J_history“* verdeutlicht, dass der letzte Wert von *„J_history“* minimal ist und auch nach mehreren Iterationen unverändert bleibt.

Nach der Prüfung des Ablaufs von *„J_history“* mit unterschiedlichen Werten von *„alpha“* und *„num_iters“* wurde der Wert von *„alpha“* auf 0.01 und *„num_iters“* auf 500 gesetzt. Abbildung 25 zeigt den Unterschied verschiedener Alpha-Werte bei gleichem *„num_iters“*. In Abbildung 25-a beträgt der Alpha-Wert 0.001, was ein zu langsames Absinken der *„J_history“* aufweist. In Abbildung 25-b ist der Alpha-Wert 0,1, was zu einem zu schnellen Absinken des Wertes von *„J_history“* führt. Abbildung 25-c gehört zu alpha=0,01, was einen weder zu langsamen noch zu schnellen Abstieg darstellt. Zusätzlich kann man sicherstellen, dass sich die mit alpha=0.01 erreichte minimale *„J_history“* bei weiteren Iterationen nicht verändert. Abbildung 26 zeigt den Unterschied zwischen verschiedenen *„num_iters“* mit dem gleichen Wert von *„alpha“*. In Abbildung 26-a ist der *„num_iters“* gleich 100. Da die Kurve am Ende der Iterationen keinen flachen Teil besitzt, kann man nicht davon ausgehen, dass der letzte Wert von *„J_history“* ein Minimalwert ist. In Abbildung 26-b ist *„num_iters“* gleich 1000, was bedeutet, dass die Berechnungszeit viel länger geworden ist. Außerdem bleibt der Wert von *„J_history“* für die beinahe letzten 700 Iterationen unverändert. Abbildung 26-c zeigt den Abstieg der *„J_history“* mit *„alpha“* gleich 500, die nicht nur einen sicheren Minimalwert lieferte, sondern auch keine hohe Rechenzeit verursachte.

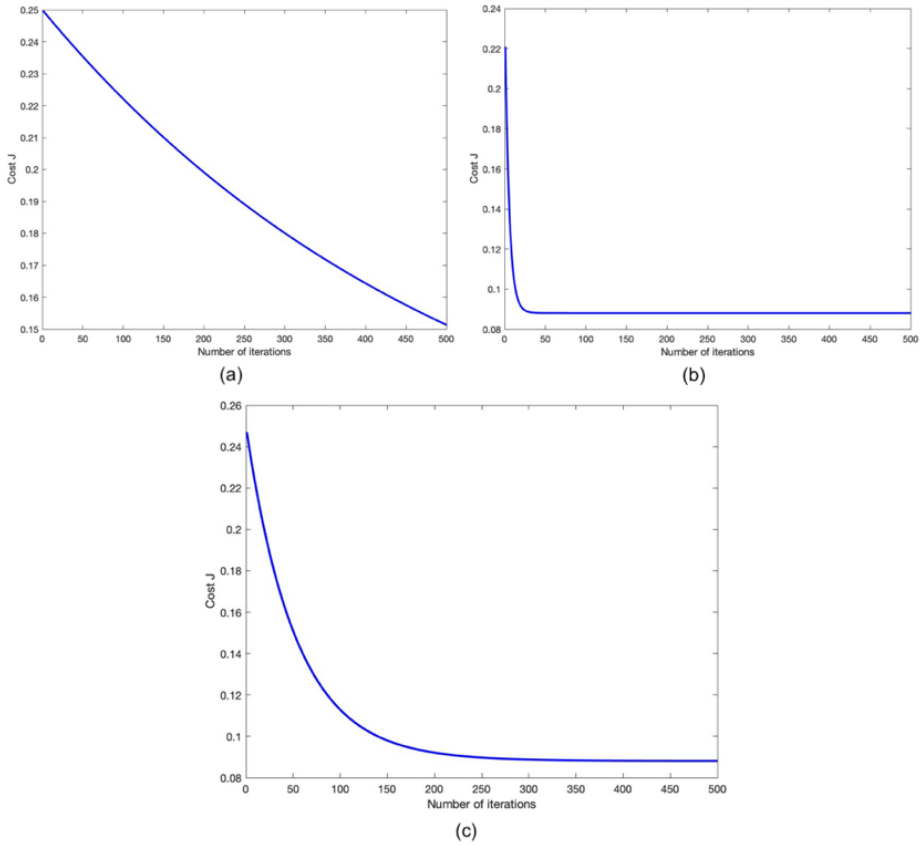


Abbildung 25: Bestimmung des alpha

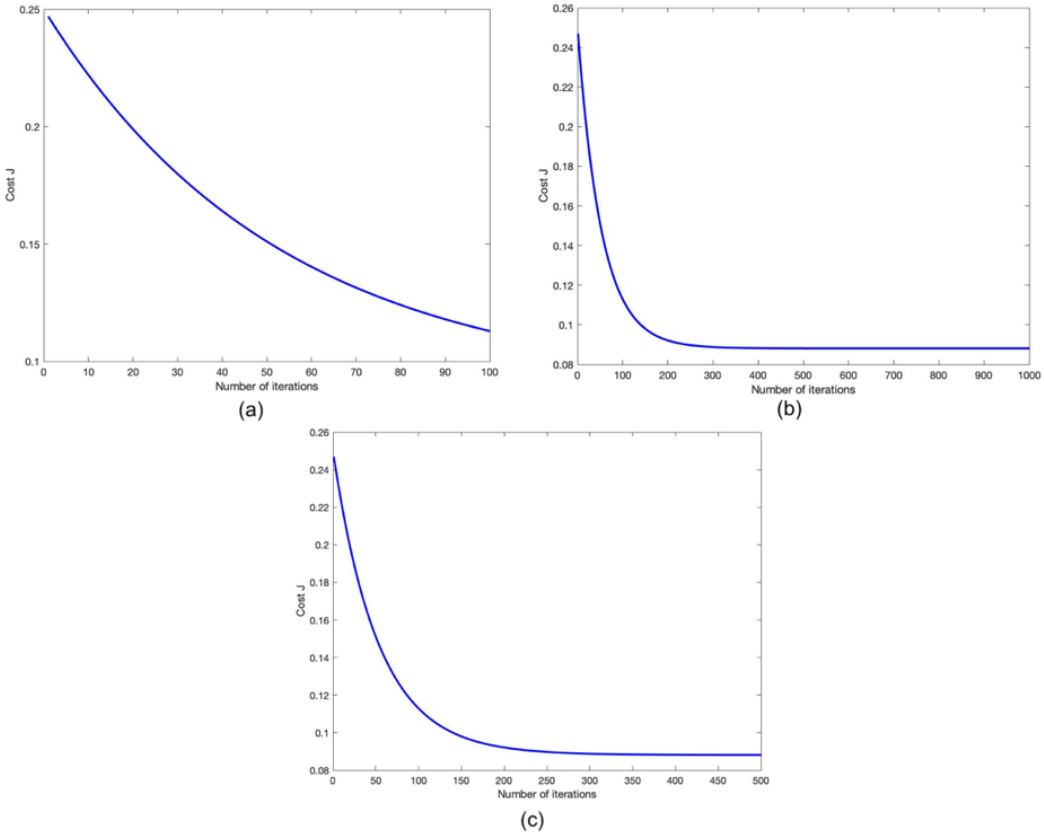


Abbildung 26: Bestimmung des num-iters

3.2.7 Daten des Benutzers

Wie im Unterkapitel 2.4 der Konzeption beschrieben wurde, muss für das zweite Anwendungsszenario der Benutzer zunächst im Assistenzsystem angelegt werden. Zu diesem Zweck gibt der Benutzer einen Benutzernamen und eine Geheimzahl in das System ein. Somit erstellt das Assistenzsystem eine neue Textdatei mit diesen beiden Einträgen. Die erstellte Textdatei wird im zweiten Anwendungsszenario zu drei Zwecken aufgerufen:

- Zur Speicherung neuer Gesundheitsdaten des Benutzers
- Zur Entnahme der vorhandenen Gesundheitsdaten der letzten 10 Tage des Benutzers, um das Krankheitsrisiko und die Präventivmaßnahmen zu bewerten.
- Bei der Bestätigung der Existenz oder Nichtexistenz einer Krankheit durch den Benutzer wird die Textdatei auf die Verfügbarkeit von Gesundheitsdaten der letzten 10 Tage geprüft. Im Falle, dass alle geprüften Daten verfügbar sind, werden die Daten zur Speicherung als neue Trainingsdaten entnommen.

Abbildung 27 zeigt, wie die gespeicherten Daten eines Benutzers aussehen.

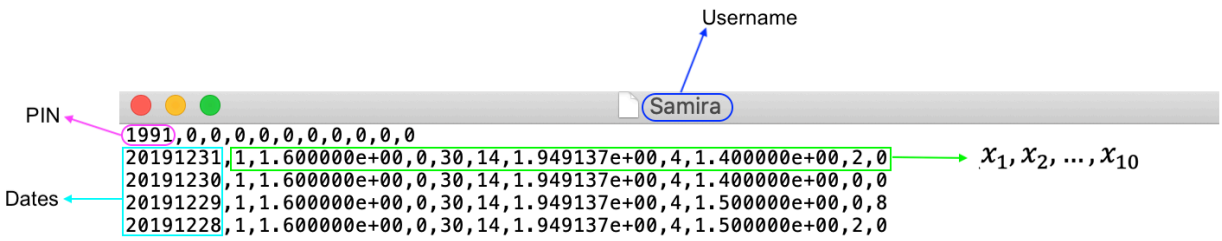


Abbildung 27: Daten des Benutzers

Der nächste Abschnitt erklärt im Detail, wie das Assistenzsystem installiert und vom Benutzer verwendet werden kann.

3.3 Installations- und Benutzungsanleitung

Zur Nutzung des Assistenzsystems wurde eine komplette MATLAB-Datei und ein Package zur Installation des Assistenzsystems als Applikation zur Verfügung gestellt. Für diese beiden Optionen muss MATLAB auf dem Computer installiert sein. Die MATLAB-Datei enthält eine .mlapp-Datei, die die GUI des Assistenzsystems ist, sowie alle anderen Klassen und Objekte des Assistenzsystems. Durch Klicken auf die .mlapp-Datei wird die GUI geöffnet und kann vom Benutzer verwendet werden. Abbildung 28 veranschaulicht die Schritte zur Installation des Assistenzsystems als neue Applikation in MATLAB. Zuerst muss die .mlappinstall-Datei geöffnet werden. Im zweiten Schritt muss die Installation der Applikation bestätigt werden. Anschließend wird das Assistenzsystem, wie der dritte Teil von Abbildung 28 zeigt, als neue Applikation in der Registerkarte APPS zugänglich sein. Mit dem Klick auf installierte APP wird die GUI geöffnet.

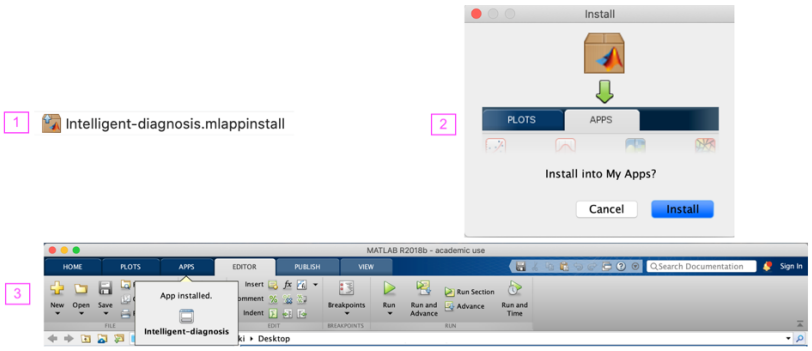


Abbildung 28: Installation des Assistenzsystems

Abbildung 29 gibt einen Überblick über das erste Fenster der GUI. Im ersten Teil wählt der Benutzer das Geschlecht aus. In den zweiten Bereich wird das Gewicht in Kilogramm und die Höhe in Zentimetern eingetragen. Basierend auf den Angaben in Teil 2 berechnet das System den BMI des Benutzers und veranschaulicht ihn in Teil 5. In Teil 3 stellt der Benutzer fest, ob er Raucher ist oder nicht. Als Ergebnis dessen, was in Teil 3 ausgewählt wurde, wird Teil 16 veränderbar und nicht veränderbar sein. In Teil 4 wird das Alter eingetragen. Der Benutzer kann Gesundheitsdaten von 1 bis 5 Tagen in das System eingeben. Dazu wählt der Benutzer in Teil 6 die gewünschte Anzahl von Tagen aus. In Abschnitten 7, 8, 9, 10 und 11 werden die Körpertemperatur

in °C, Anzahl der Ruheherzfrequenz pro Minute, Aktivität in Minuten, Schlafdauer in Stunden und Wasserkonsum in Milliliter eingetragen. In Teil 12 wird angegeben, ob an dem Tag alkoholische Getränke konsumiert wurde oder nicht, und folglich werden die Teile 13, 14 und 15 veränderbar oder nicht veränderbar sein. Falls der Benutzer auf Teil 12 mit Ja geantwortet hat, kann die getrunkene Menge des Bieres, Weins und Whiskeys in den Teilen 13, 14, 15 eingegeben werden. Die Anzahl der an dem Tag gerauchten Zigaretten kann in Teil 16 eingetragen werden. Nachdem der Benutzer alle Daten eingegeben hat, kann er in Teil 17 wählen, das Risiko und die Präventivmaßnahmen von welcher Krankheit in den Teilen 18 und 19 dargestellt werden müssen.

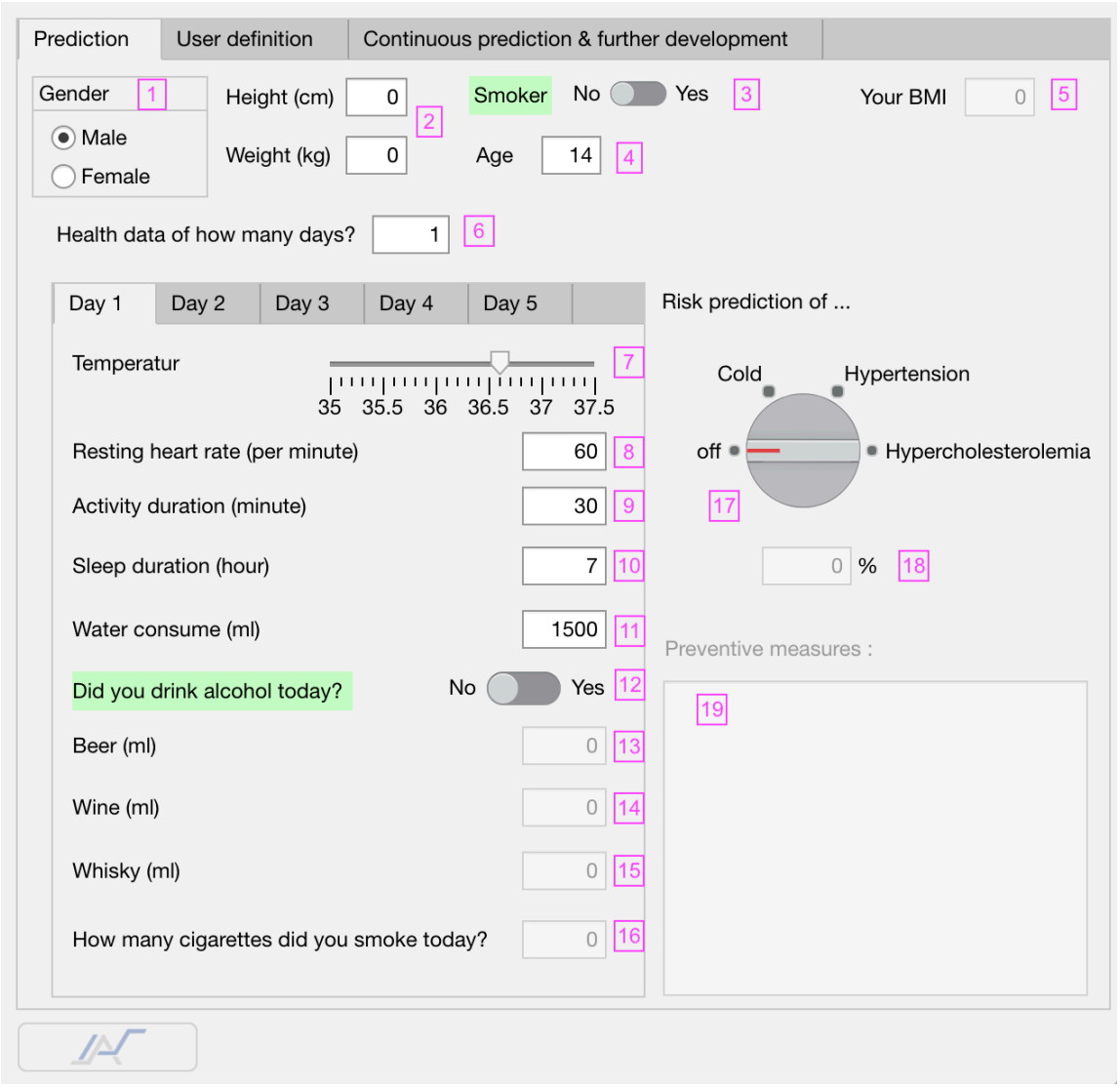


Abbildung 29: Vorhersage-Fenster

In Abbildung 30 wurde das Benutzerdefinition-Fenster veranschaulicht. In Teil 1 dieses Fensters kann der Benutzer einen nur aus Buchstaben bestehenden Benutzernamen eintragen. In Teil 2 wird eine vierstellige PIN eingegeben. Anschließend wird beim

Drücken der Taste 3 der Benutzer mit den in Teil 1 und 2 eingegebenen Daten im System definiert.

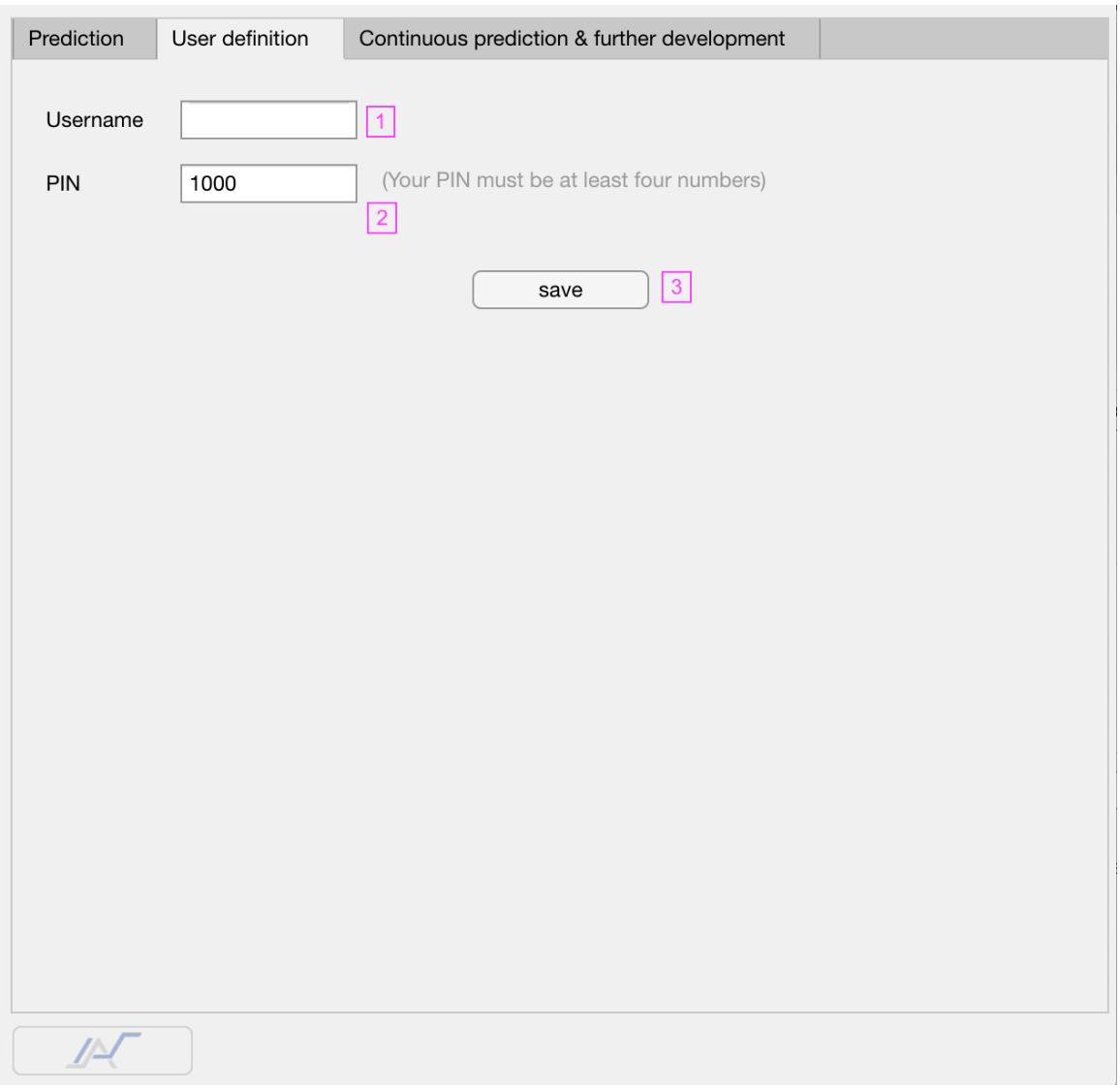


Abbildung 30: Benutzerdefinition-Fenster

In Abbildung 31 ist zu sehen, wie das Kontinuierliche Vorhersage und Weiterentwicklungs-Fenster aussieht. In Teil 1 und 2 werden der Benutzername und die PIN des Benutzers eingetragen. Weitere Funktionen dieses Fensters können nur dann verwendet werden, wenn die in Teil 1 und 2 eingetragenen Eingaben korrekt sind. Durch Bestätigung von Teil 3 wird in Teil 4 veranschaulicht, die Gesundheitsdaten von welchen Tagen im System gespeichert sind. Die Teile 5, 6 und 7 sind drei weitere Teile, die dieses Fenster von dem ersten Fenster unterscheiden. In Teil 5 kann der Benutzer bestätigen, ob eine Krankheit, gemäß der in Abschnitt 2.3 der Konzeption beschriebenen Richtlinie, vorhanden ist oder nicht. In Teil 6 wird das zugehörige Datum eingetragen. Durch Drücken der Taste 7 werden folgende Prozessschritte in dem System ausgeführt:

- Die neuen Gesundheitsdaten werden mit dem dazugehörigen Datum in der Datei des Benutzers gespeichert.

- Wenn der Benutzer das Vorhandensein oder Nichtvorhandensein einer Krankheit in Teil 5 bestätigt hätte und die Gesundheitsdaten des Benutzers von den letzten 10 Tagen verfügbar wären, werden die Gesundheitsdaten der letzten 10 Tage als neue Trainingsdaten gespeichert.

Hierbei ist zu beachten, dass Informationen über aufgetretene Krankheiten nicht in der Datei des Benutzers gespeichert werden und nur für die Weiterentwicklung des Assistenzsystems verwendet werden. Außerdem werden bei der Speicherung der Gesundheitsdaten des Benutzers als zusätzliche Trainingsdaten nur x_1 bis x_{10} (siehe Tabelle 3 der Konzeption) und keine weiteren Informationen im System gespeichert.

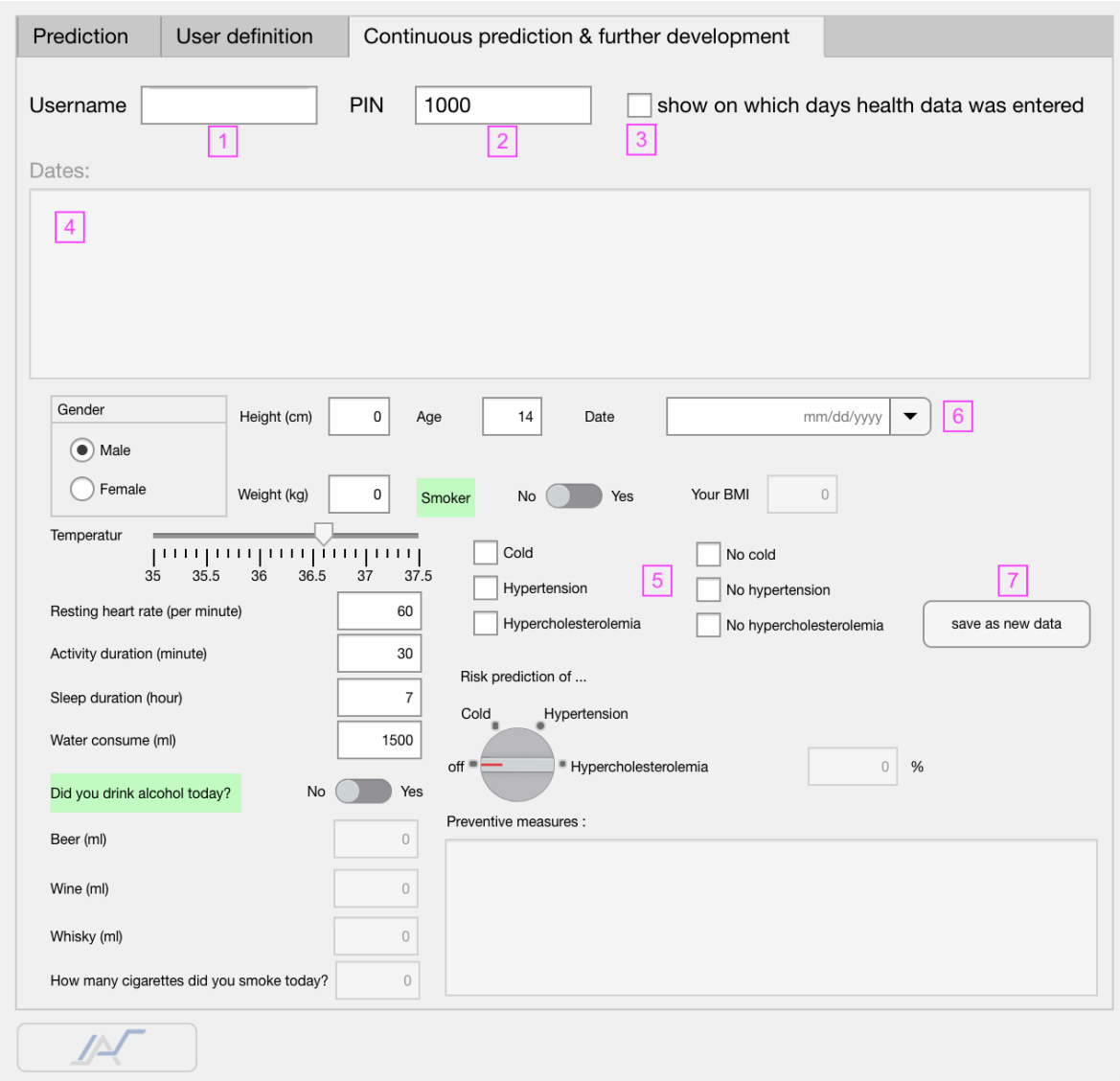


Abbildung 31: Kontinuierliche Vorhersage und Weiterentwicklung-Fenster

Literature

[1] “Allied Market Research.” [Online]. Available: https://www.alliedmarketresearch.com/.

[2] P. Loskill, “Digitalisierung, Miniaturisierung, Personalisierung.”

[3] T. Heimbold, *Einführung in die Automatisierungstechnik: Automatisierungssysteme, Komponenten, Projektierung und Planung*. Carl Hanser Verlag GmbH Co KG, 2014.

[4] “IntelliVue Guardian.” [Online]. Available: https://www.usa.philips.com/healthcare/clinical-solutions/early-warning-scoring/intellivue-guardian-ews.

[5] T. Ramge, “Intelligente herzschrittmacher.” .

[6] M. Faschingbauer and K. Seide, “Das intelligente Implantat,” *Trauma und Berufskrankheit*, vol. 11, no. 1, pp. 60–64, 2009.

[7] C. Alvarez-Lorenzo, L. Bromberg, and A. Concheiro, “Light-sensitive intelligent drug delivery systems,” *Photochem. Photobiol.*, vol. 85, no. 4, pp. 848–860, 2009.

[8] N. J. Nilsson and N. J. Nilsson, *Artificial intelligence: a new synthesis*. Morgan Kaufmann, 1998.

[9] M. Siepermann, “Künstliche Intelligenz.” [Online]. Available: https://wirtschaftslexikon.gabler.de/definition/kuenstliche-intelligenz-ki-40285.

[10] M. Negnevitsky, *Artificial intelligence: a guide to intelligent systems*. Pearson education, 2005.

[11] R. Lackes and M. Siepermann, “Maschinelles Lernen.” .

[12] T. Hastie, R. Tibshirani, and J. Friedman, “Unsupervised learning,” in *The elements of statistical learning*, Springer, 2009, pp. 485–585.

[13] Y. S. Abu-Mostafa, M. Magdon-Ismail, and H.-T. Lin, *Learning from data*, vol. 4. AMLBook New York, NY, USA:, 2012.

[14] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[15] B. D. Ripley and N. L. Hjort, *Pattern recognition and neural networks*. Cambridge university press, 1996.

[16] R. E. Neapolitan and X. Jiang, *Artificial Intelligence: With an introduction to machine learning*. Chapman and Hall/CRC, 2018.

[17] M. Paluszek and S. Thomas, *MATLAB machine learning recipes: A problem-solution approach, Second edition*. 2019.

[18] R. F. de Mello and M. A. Ponti, *Machine Learning - A Practical Approach on the Statistical Learning Theory*. 2018.

[19] M. E. Celebi and K. Aydin, *Unsupervised learning algorithms*. Springer, 2016.

[20] G. Rebala, A. Ravi, and S. Churiwala, *An Introduction to Machine Learning*. Springer, 2019.

[21] X. Zhu, J. Lafferty, and R. Rosenfeld, "Semi-supervised learning with graphs. Diss." Carnegie Mellon University, Language Technologies Institute, School of …, 2005.

[22] M. Sewak, *Deep Reinforcement Learning: Frontiers of Artificial Intelligence*. Springer, 2019.

[23] A. Nandy and M. Biswas, *Reinforcement Learning: With Open AI, TensorFlow and Keras Using Python*. Apress, 2017.

[24] W. Ertel and N. T. Black, *Grundkurs Künstliche Intelligenz*. Springer, 2018.

[25] A. Panesar, *Machine Learning and AI for Healthcare*. 2019.

[26] D. Forsyth, *Applied Machine Learning*. 2019.

[27] S. Papp, W. Weidinger, M. Meir-Huber, B. Ortner, G. Langs, and R. Wazir, *Handbuch Data Science: Mit Datenanalyse und Machine Learning Wert aus Daten generieren*. Carl Hanser Verlag GmbH Co KG, 2019.

[28] T. T. Pham, *Applying Machine Learning for Automated Classification of Biomedical Data in Subject-Independent Settings*. 2019.

[29] O. Niggemann, *Machine Learning for Cyber Physical Systems*. 2017.

[30] P. P. Angelov, *Empirical Approach to Machine Learning*. 2017.

[31] G. A. Tsihrintzis, D. N. Sotiropoulos, and L. C. Jain, *Intelligent Systems Reference Library 149 Machine Learning Paradigms Advances in Data Analytics*. 2019.

[32] F. Hutter, *Automated Machine Learning*. 2019.

[33] S. Chakraverty, D. M. Sahoo, and N. R. Mahato, *Concepts of soft computing: Fuzzy and ANN with programming*. Springer, 2019.

[34] R. Kumar Agrawal, "Difference between Machine Learning, Deep Learning and Artificial Intelligence." [Online]. Available: https://medium.com/@UdacityINDIA/difference-between-machine-learning-deep-learning-and-artificial-intelligence-e9073d43a4c3.

[35] P. Rajpurkar *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv Prepr. arXiv1711.05225*, 2017.

[36] A. Y. Hannun *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nat. Med.*, vol. 25, no. 1, p. 65, 2019.

[37] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.

[38] M. F. Akay, "Support vector machines combined with feature selection for breast

cancer diagnosis," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3240–3247, 2009.

[39] "Ada Health GmbH." [Online]. Available: https://ada.com/.

[40] "ExB Group." [Online]. Available: https://www.exb.de/gesundheit/.

[41] "HeuroLabs." [Online]. Available: http://heurolabs.com/.

[42] "MeVis Medical Solutions AG." [Online]. Available: https://www.mevis.de/.

[43] "Arya." [Online]. Available: http://www.aryaapp.co/.

[44] "xbird." [Online]. Available: http://www.xbird.io/.

[45] W. Blaum, "Hans Walter Striebel: Anästhesie, Intensivmedizin, Notfallmedizin– für Studium und Ausbildung," *GMS Z. Med. Ausbild.*, vol. 31, no. 2, 2014.

[46] S. M. Lundberg *et al.*, "Explainable machine-learning predictions for the prevention of hypoxaemia during surgery," *Nat. Biomed. Eng.*, vol. 2, no. 10, p. 749, 2018.

[47] R. Obermaier, "Handbuch Industrie 4.0 und Digitale Transformation: Betriebswirtschaftliche, technische und rechtliche Herausforderungen." Springer, 2019.

[48] "Umfrage zur Häufigkeit von Erkältungen in Deutschland nach Geschlecht 2019." [Online]. Available: https://de.statista.com/statistik/daten/studie/661674/umfrage/umfrage-zur-haeufigkeit-von-erkaeltungen-in-deutschland-nach-geschlecht/#statisticContainer.

[49] W. Thefeld, "Verbreitung der Herz-Kreislauf-Risikofaktoren Hypercholesterinämie, Übergewicht, Hypertonie und Rauchen in der Bevölkerung," *Bundesgesundheitsblatt-Gesundheitsforschung-Gesundheitsschutz*, vol. 43, no. 6, pp. 415–423, 2000.

[50] M. Middeke, E. Pospisil, and K. Völker, "Bluthochdruck senken ohne Medikamente." Thieme, Stuttgart, 2000.

[51] H. H. Kornhuber, "Bluthochdruck und Alkoholkonsum," in *Arterielle Hypertonie*, Springer, 1984, pp. 149–162.

[52] H. H. Kornhuber and G. Lisson, "Bluthochdruck, Übergewicht und Alter: für Frühbehandlung der Hypertonie*: Eine Betriebsuntersuchung," *DMW-Deutsche Medizinische Wochenschrift*, vol. 106, no. 50, pp. 1692–1696, 1981.

[53] L. Rink, A. Kruse, and H. Haase, *Immunologie für Einsteiger*. Springer, 2012.

[54] "Erkältung." [Online]. Available: https://www.apotheken-umschau.de/Erkaeltung.

[55] R. Hänsel, *Phytopharmaka: Grundlagen und Praxis*. Springer-Verlag, 2013.

[56] S. Kushimoto, S. Gando, and D. Saitoh, "Körpertemperatur und Outcome bei schwerer Sepsis," *J. Club AINS*, vol. 3, no. 01, pp. 31–33, 2014.

[57] "Why do we catch more colds when the temperature drops?" [Online]. Available: https://nationalpost.com/health/why-do-we-catch-more-colds-when-the-

temperature-drops-blame-our-immune-system.

[58] E. Bassenge, H. T. Schneider, and A. Daiber, “Oxidativer Stress und kardiovaskuläre Erkrankungen,” *DMW-Deutsche Medizinische Wochenschrift*, vol. 130, no. 50, pp. 2904–2909, 2005.

[59] “Was ist die Ruheherzfrequenz?” [Online]. Available: https://www.philips.de/c-f/XC000005478/was-ist-die-ruheherzfrequenz.

[60] B. Skala, “Ruheherzfrequenz.”

[61] K. Janhsen, “9 Hypertonie,” 2008.

[62] M. Schmidt, “Bewegungstherapie und Rehabilitation,” *Man. Medizin*, vol. 54, no. 1, pp. 46–49, 2016.

[63] M. Hirshkowitz *et al.*, “National Sleep Foundation’s sleep time duration recommendations: methodology and results summary,” *Sleep Heal.*, vol. 1, no. 1, pp. 40–43, 2015.

[64] D. Volkert, “Der Body-Mass-Index (BMI)-ein wichtiger Parameter zur Beurteilung des Ernährungszustands,” *Aktuel. Ernahrungsmed.*, vol. 31, no. 03, pp. 126–132, 2006.

[65] A. Heidbreder and P. Young, “Auch tagsüber immer schläfrig,” *DNP-Der Neurol. und Psychiater*, vol. 13, no. 10, pp. 67–74, 2012.

[66] A. N. Vgontzas, D. Liao, E. O. Bixler, G. P. Chrousos, and A. Vela-Bueno, “Insomnia with objective short sleep duration is associated with a high risk for hypertension,” *Sleep*, vol. 32, no. 4, pp. 491–497, 2009.

[67] J. E. Gangwisch *et al.*, “Short sleep duration as a risk factor for hypercholesterolemia: analyses of the National Longitudinal Study of Adolescent Health,” *Sleep*, vol. 33, no. 7, pp. 956–961, 2010.

[68] U. Siedentopp, “Wasser—Lebensquelle und Heilmittel,” *Dtsch. Zeitschrift für Akupunkt.*, vol. 59, no. 4, pp. 45–49, 2016.

[69] W. Böcker, H. Denk, P. U. Heitz, H. Moch, G. Höfler, and H. Kreipe, *Lehrbuch Pathologie*. Elsevier Health Sciences, 2019.

[70] “Microsaft Azure.” [Online]. Available: https://docs.microsoft.com/de-de/azure/machine-learning/algorithm-cheat-sheet.

[71] Y. S. Kim, “Comparison of the decision tree, artificial neural network, and linear regression methods based on the number and types of independent variables and sample size,” *Expert Syst. Appl.*, vol. 34, no. 2, pp. 1227–1234, 2008.

[72] “Die Symptome einer Erkältung.” [Online]. Available: https://www.erkaeltung-online.de/symptome/.

[73] “supervised-vs-unsupervised-learning.” [Online]. Available: https://lawtomated.com/supervised-vs-unsupervised-learning-which-is-better/.

[74] “SAS.” [Online]. Available: https://blogs.sas.com/content/subconsciousmusings/2017/04/12/machine-

learning-algorithm-use/?imm_mid=0f1128&cmp=em-data-na-na-newsltr_ai_20170424.

Erklärung

Ich erkläre, die Arbeit selbständig verfasst und bei der Erstellung dieser Arbeit die einschlägigen Bestimmungen, insbesondere zum Urheberrechtsschutz fremder Beiträge, eingehalten zu haben. Soweit meine Arbeit fremde Beiträge (z. B. Bilder, Zeichnungen, Textpassagen) enthält, erkläre ich, dass diese Beiträge als solche gekennzeichnet sind (z. B. Zitat, Quellenangabe) und ich eventuell erforderlich gewordene Zustimmungen der Urheber zur Nutzung dieser Beiträge in meiner Arbeit eingeholt habe.

Unterschrift:

Stuttgart, den 05.02.2020