UNIVERSITÄT PASSAU
Fakultät für Informatik und Mathematik

Machine Learning Methoden von Corona Labor Daten

Bachelorarbeit von Johannes Böhm

November 2021

Lehrstuhl für Mathematik mit Schwerpunkt Digitale Bildverarbeitung
Prof. Dr. Tomas Sauer

Studienfach: Bachelor Internet Computing

Prüfer/Betreuer: Prof. Dr. Tomas Sauer

Abstract

Methoden des maschinellen Lernens können in vielen Bereichen unterstützen. Diese Arbeit stellt drei Algorithmen vor und vergleicht ihre Ergebnisse darin, wie gut sie Corona PCR Daten des Corona-Wildtyps von der Alpha-Mutation B.1.1.7 unterscheiden und ungesehene Daten den Klassen zuordnen können. Die verwendeten Algorithmen sind der Decision Tree Algorithmus, Random Forest Algorithmus und ein Neuronales Netz. Der Random Forest Algorithmus schneidet dabei mit einer mittleren Genauigkeit von 84% am besten ab, gefolgt vom Decision Tree Algorithmus mit 79% und dem Neuronalen Netz mit 77%.

Contents

1 Einleitung

Als "Machine Learning", auf Deutsch "maschinelles Lernen", bezeichnet man die Computer gestützte Weiterentwicklung der statistischen Auswertung. Der Begriff wurde in den 1950er Jahren von Arthur Samuel, einem Mitarbeiter von IBM, geprägt und in den sechziger Jahren weiter für Algorithmen der Mustererkennung verwendet. In den Siebzigern kamen die ersten künstlichen neuronalen Netze hinzu, die durch ihre Schichtstruktur in der Lage sind, komplexe Muster abzubilden und zu "lernen" [Foote, 2019, Samuel, 1959]. Heutzutage werden Machine Learning Methoden vor allem verwendet, um auf Basis von gesammelten Daten Voraussagen treffen zu können. Firmen analysieren ihre Verkaufszahlen, erstellen dynamische Produktempfehlungen oder versuchen anhand von Mustern Betrug zu verhindern [Foote, 2019]. Neben diesem finanziellen Aspekt bieten Machine Learning Algorithmen auch viel Potential für die Medizin. Einem von Google mit entwickeltem Modell gelang es 2019 auf Basis von Mammographiedaten Brustkrebs zu erkennen. Das Modell lag mit über 80% Genauigkeit deutlich höher als die Erkennung durch Ärzte (25% Genauigkeit) [Silverio, 2020, McKinney et al., 2021]. Mit der Verbreitung des Virus SARS-CoV-2 ist eine neues wichtiges Forschungsfeld entstanden. Ein für die Menschheit relevantes Problem sind die Mutationen des Virus, die sich ansteckender oder tödlicher verhalten. Diese Arbeit beschäftigt sich mit der Klassifizierung von Corona PCR-Daten nach dem Vorliegen des Wildtyps des Virus, der sich zu Beginn der Pandemie verbreitet hat, oder der Alpha-Mutation B.1.1.7 von SARS-CoV-2 mit Hilfe von Machine Learning Algorithmen. Zuerst werden die verwendeten Daten und Methoden vorgestellt, dann werden die Ergebnisse der verschiedenen Algorithmen gezeigt, verglichen und diskutiert.

2 Verwandte Arbeiten

Methoden des maschinellen Lernens wurden für verschiedene Bereiche untersucht und verglichen. Geoffry K. F. Tso und Kelvin K. W. Yau veröffentlichten eine Studie, in der sie Regressionsanalyse, Decision Tree und ein Neuronales Netz darin verglichen, wie gut die Methoden den Verbrauch elektrischer Energie vorhersagen können [Tso and Yau, 2007]. Emmanuel Awoin, Peter Appiahene, Frank Gyasi und Abdulai Sabtiwu1 der *University of Energy and Natural Resources* in Sunyani, Ghana verglichen in ihrer Studie

drei Decision Tree Algorithmen: C5.0, C4.5 und CART. Sie versuchten mit den Algorithmen die Entwicklung und Leistung ländlicher Banken in Ghana vorherzusagen [Awoin et al., 2020]. T.R. Prajwala vom *CMR Institute of Technology* ver-öffentlichte eine Arbeit, in der sie die den Decision Tree Algorithmus und Random Forest darin verglich, wie genau sie Regen aufgrund von Wetterdaten vorhersagen können [Prajwala, 2015]. Thomas Langer et al. entwickelten Modelle für maschinelles Lernen zur Vorhersage von RT-PCR Ergebnissen für das SARS-CoV-2-Virus bei Patienten mit grippeähnlichen Symptomen unter Verwendung grundlegender klinischer Daten [Langer et al., 2020].

3 Methodik

In dieser Arbeit werden drei verschiedene Methoden verglichen, die die Daten in die beiden Klassen "Wildtyp" und "Alpha-Mutation" einordnen: Der "Decision Tree" Algorithmus, zu Deutsch Entscheidungsbaum, der "Random Forest" Algorithmus und ein Neuronales Netz.

3.1 Verwendete Daten

3.1.1 Quelle der Daten

Die Daten, die zur Untersuchung herangezogen werden, stammen aus den Geräten "Quant Studio5" von ThermoFisher und "VERSANT kPCR" von Siemens bzw. der zugehörigen Software. Zur Verfügung gestellt wurden sie vom Labor des Passauer Klinikums. Die Geräte messen mit dem PCR-Verfahren Viruslasten einer Probe. PCR steht für Polymerase-Chain-Reaction, auf Deutsch Polymerase-Kettenreaktion. Für jede Probe wird im Gerät in 40 Zyklen die Fluoreszenz eines Farbstoffes gemessen. An das Zielmolekül (das Virus) wird der Farbstoff mit einem Primer-Protein angehängt und das Molekül wird in mehreren Phasen mit verschiedenen Temperaturen mit Hilfe eines Reagenz vervielfacht sodass die Fluoreszenz steigt. Je höher die Viruslast zu Beginn war, desto schneller wird ein Grenzwert erreicht, ab dem die Fluoreszenz des Farbstoffes höher als die Hintergrundfluoreszenz ist und das Ergebnis wird als positiv gewertet [BioRad, 2021]. Der Zyklus, in dem der Grenzwert erreicht wird, gibt den CT-Wert an, ein Index für die Viruslast. CT ist Englisch für Cycle-Threshold, zu Deutsch Zyklus Schwellwert. Ein niedriger CT-Wert bedeutet also eine hohe Viruslast in der Probe zu Beginn.

Ein CT-Wert von über 39 wird nicht mehr als positives Ergebnis gewertet, da davon auszugehen ist, dass sich der Farbstoff nach einiger Zeit auch an ein anderes Molekül angehängt haben kann. Insgesamt wird mit 6 Filtern gemessen (siehe Figure 1). Da der Farbstoff FAM verwendet wurde, werden nur die Daten des ersten Filters für die Datenanalyse herangezogen. Die Messwerte stehen für jeden Durchlauf in der Software bereit und können als .csv Datei exportiert werden. Da die Kontrollen, die bei jeder Messung mitlaufen, und der Großteil der negativen Resultate für die Untersuchung uninteressant sind, bestand der erste Schritt daraus, Datenblätter mit positiven Ergebnissen aus der Software zu exportieren, die relevanten herauszufiltern und für die Machine Learning Algorithmen aufzubereiten. Die Daten, die als Wildtyp eingestuft wurden, stammen von Messungen im November 2020, die der Mutation B.1.1.7 aus dem Zeitraum März bis Mai 2021. Zum Zeitpunkt des Datenexports lagen 194 verwendbare Messungen der Mutation vor. Um die Algorithmen nicht über eine unterschiedliche Anzahl der beiden Klassen zu beeinflussen, wurden ebenfalls 194 Wildtyp Messungen verwendet. Als Kontrollgruppe diente die gleiche Anzahl negativer Messungen vom November 2020.

3.1.2 Beschreibung des Datensatzes

Für ein besseres Verständnis der Datenstruktur wurden die drei relevanten unterschiedlichen Messtypen als Liniendiagramme erstellt: eine negative Messung, eine positive Messung des Wildtyps und eine positive Messung der Alpha-Mutation B.1.1.7 (siehe Figures 2-4). Die Y-Achse zeigt den Messwert der Fluoreszenz, die X-Achse die Zyklen. Die negativen Messungen sehen in den Diagrammen alle recht ähnlich aus, bei den positiven Messungen gibt es sichtbare Unterschiede zwischen den Kurvenverläufen. Die hier abgebildeten zeigen die Gemeinsamkeiten: erst stagnieren die Werte, dann kommt es zu einem exponentiellen Anstieg und am Schluss einem Abflachen der Kurve. Unterschiedlich ist der Übergang zur exponentiellen Phase, der bei einigen schon um den 15. Zyklus, bei anderen erst um den 30. beginnt. (Anmerkung: Relevant ist die gestrichelt dargestellte Kurve des Filters x1-m1, da diese die Fluoreszenzwerte des verwendeten Farbstoff FAM darstellt.) Für die Algorithmen wurden die Daten wie folgt aufbereitet: ein Datensatz besteht aus der Markierung für die Klasse (0 = Wildtyp, 1 = Mutation) und den Fluoreszenzwerten des Filters x1-m1, der den Farbstoff FAM herausfiltert. Verwendet werden die Zyklen 11 bis 40. Die Zyklen 1 bis 10

Instrument filters and supported dyes

System dyes

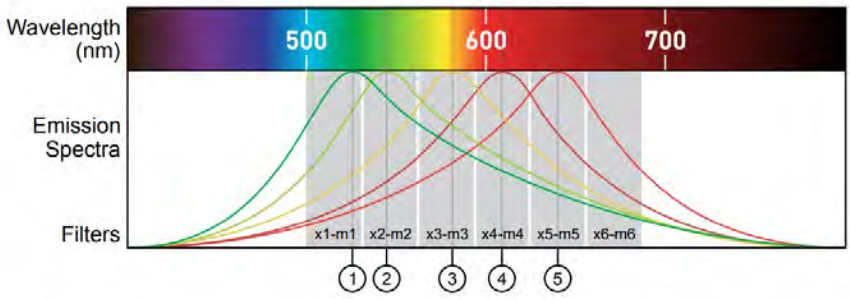
The QuantStudio™ 3 and 5 Systems use a coupled four-color, coupled five-color, or decoupled six-color filter set that supports the dyes shown in the following table and figure. For more information about the spectral dye calibration kits available for the QuantStudio™ 3 and 5 Systems, contact Support.

Peak channel	Color	Filter wavelength (nm)[1]		Pre-calibrated dyes	Example custom dyes
		Excitation	Emission		
x1-m1	Blue	470 ± 15	520 ± 15	FAM™ and SYBR™ Green	SYT09
x2-m2	Green	520 ± 10	558 ± 12	VIC™	HEX™, TET™, and JOE™[2]
x3-m3	Yellow	550 ± 10	587 ± 10	ABY™, NED™, and TAMRA™	Cy®3
x4-m4	Orange	580 ± 10	623 ± 14	JUN™ and ROX™	Texas Red™
x5-m5	Red	640 ± 10	682 ± 14	Cy®5 and MUSTANG PURPLE™	LIZ™
x6-m6	Deep-Red	662 ± 10	711 ± 12	None[3]	Cy®5.5

[1] The central wavelengths are the optimized wavelengths.

[2] The HEX™ and TET™ dyes from Thermo Fisher Scientific fall within the emission wavelength range of the system, therefore they can be added and adapted for use in experiments on the system. To add any of these dyes to the Dye Library, perform a custom dye calibration.

[3] This filter set currently does not support any dyes supplied by Thermo Fisher Scientific.



① x1-m1 — FAM™, SYBR™ Green
② x2-m2 — VIC™
③ x3-m3 — ABY™, NED™, Cy®3, and TAMRA™
④ x4-m4 — JUN™, ROX™, and Texas Red™
⑤ x5-m5 — Cy®5 and MUSTANG PURPLE™

Figure 1: Instrument filters and supported dyes [ThermoFisher, 2015]

wurden herausgenommen, da sich die Werte hier nur sehr gering voneinander unterscheiden und die Algorithmen diese marginalen Unterschiede, auch "Rauschen" genannt, nicht berücksichtigen sollen (siehe Table 1).

Feature	**Wert**	**Feature**	**Wert**	**Feature**	**Wert**
Mutation	0	21-x1-m1	226454	31-x1-m1	1575388
11-x1-m1	199473	22-x1-m1	250654	32-x1-m1	1685012
12-x1-m1	199603	23-x1-m1	295152	33-x1-m1	1774147
13-x1-m1	199462	24-x1-m1	373352	34-x1-m1	1846732
14-x1-m1	201388	25-x1-m1	496513	35-x1-m1	1905401
15-x1-m1	201368	26-x1-m1	665361	36-x1-m1	1951812
16-x1-m1	202095	27-x1-m1	866493	37-x1-m1	1992663
17-x1-m1	202755	28-x1-m1	1076394	38-x1-m1	2022842
18-x1-m1	204726	29-x1-m1	1271770	39-x1-m1	2048596
19-x1-m1	207477	30-x1-m1	1438709	40-x1-m1	2070604
20-x1-m1	214594				

Table 1: Ein Datensatz

3.2 Decision Tree

Der Decision Tree Algorithmus zur Klassifikation erstellt ein Modell, einen Entscheidungsbaum, anhand dessen Daten Klassen zugeordnet werden. Es gibt verschiedene Versionen des Algorithmus, in dieser Arbeit wird der "Klassifizierungs- und Regressionsbaum", auf Englisch "Classification and Regression Tree", kurz "CART" verwendet. Die Bäume sind Binärbäume, die an jedem Knoten den Wert eines "Features" mit einem berechneten Wert vergleichen. Der Pfad durch den Baum endet mit Ziel-"Blättern", die eine Wahrscheinlichkeit für die Klassifizierung angeben. Berechnet wird dieses Modell, indem der Machine-Learning Algorithmus anhand von Trainingsdaten Zusammenhänge zwischen dem Zielwert, der Klassifizierung, und anderen Daten, "Features", lernt [Drakos, 2019a]. In dieser Arbeit entsprechen die Features den Fluoreszenzwerten der Zyklen. Im ersten Schritt wird für jedes Feature die beste Aufteilung gesucht. Die Qualität der Aufteilung wird anhand eines Kriteriums, dem Gini-Index, berechnet. Der Gini Index ist ein statistisches Maß zur Berechnung und Darstellung von

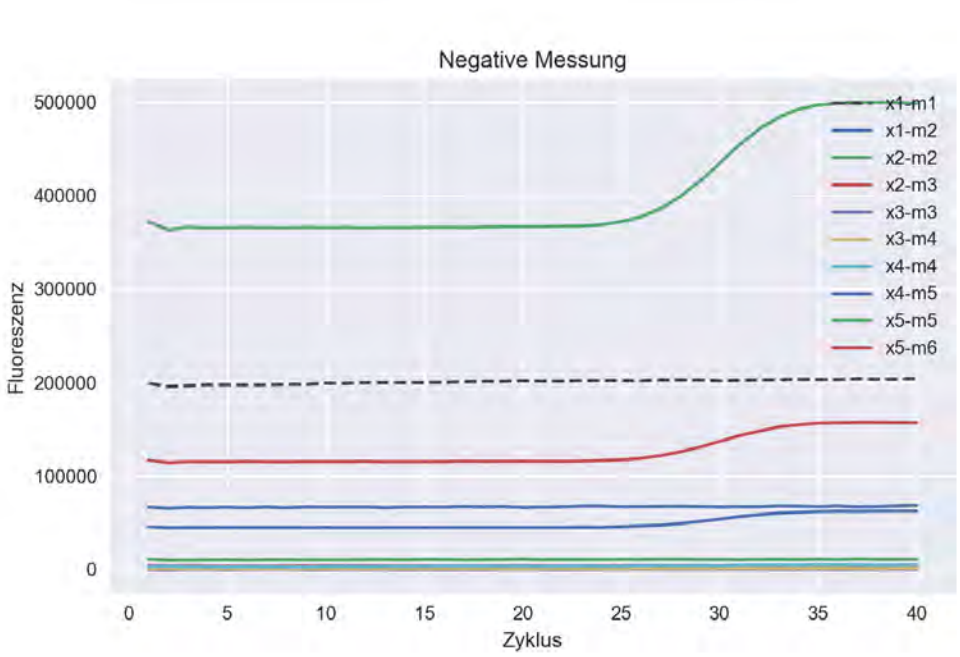


Figure 2: Verlauf der Messwerte aller 10 Filter über 40 Zyklen einer negativen Messung

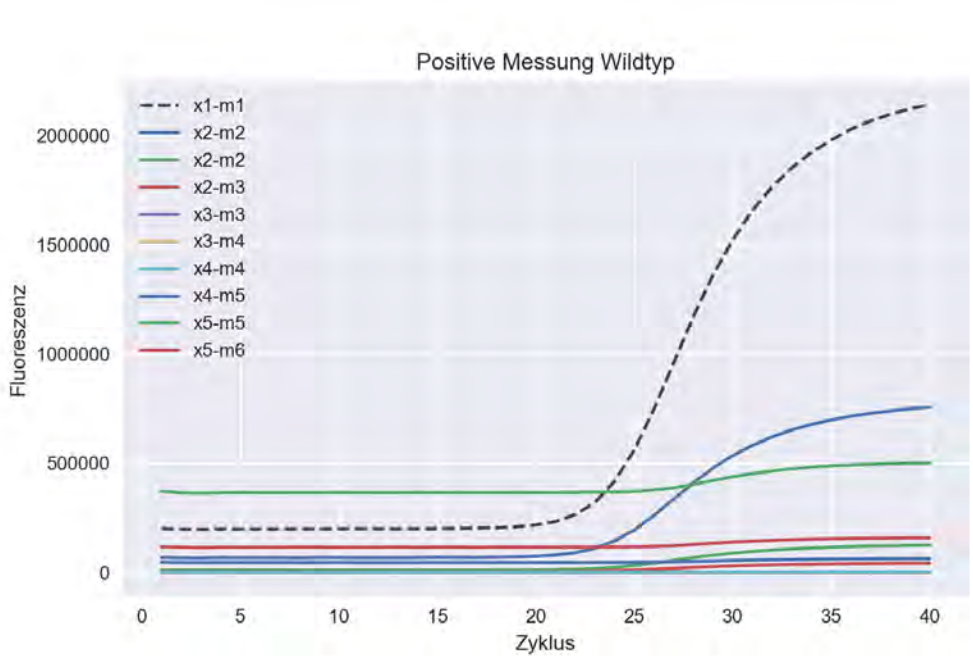


Figure 3: Verlauf der Messwerte aller 10 Filter über 40 Zyklen einer positiven Messung (Wildtyp)

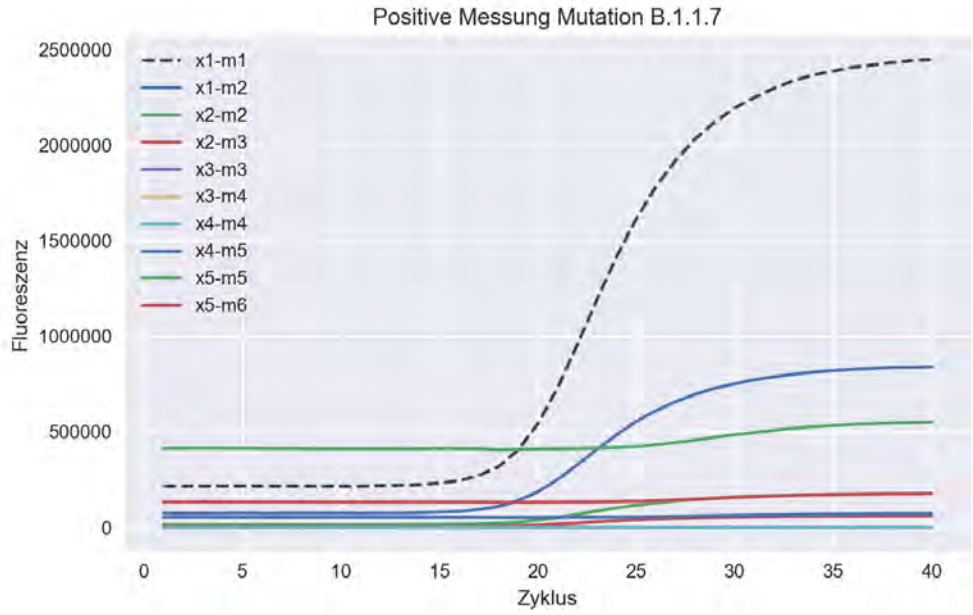


Figure 4: Verlauf der Messwerte aller 10 Filter über 40 Zyklen einer positiven Messung (Alpha-Mutation B.1.1.7)

Ungleichverteilungen, auch Gini-Verunreinigungsmaß genannt. Man erhält für jedes Feature die Aufteilung mit dem höchsten Gini-Wert, also mit möglichst ungleichen Gruppen. Das Feature mit dem höchsten Gini-Wert dient als Wurzelknoten, für die Teilgruppen wird rekursiv wieder die Gini Berechnung durchgeführt, bis ein Zielkriterium erreicht ist. Dieses Kriterium ist zum Beispiel eine Baumtiefe oder Zielgruppengröße. Diese Gruppen entsprechen dann vollständig oder zum Großteil einer Klasse.

Die Formel für den Gini-Index eines Knoten t ist:

$$i(t) = \sum_{i,j} C(i|j)p(i|t)p(j|t)$$

bei dem $C(i|j)$ die Kosten einer Fehlklassifizierung eines Klasse-j-Falls als Klasse-i-Fall sind.

[Kuzilek et al., 2014, scikit learn, 2021b]

3.2.1 Optimierung und Overfitting

Ein Nachteil der Decision Trees ist, dass sie zum "Overfitting" neigen. Das bedeutet, dass sie Zusammenhänge aus den Trainingsdaten in den Baum aufnehmen, die nicht relevant sind und in den Evaluierungs- oder Zieldaten nicht vorkommen. Solche Zusammenhänge werden als "Noise" (Rauschen) bezeichnet. Um das zu vermeiden, kann "Pruning" (Beschneiden) angewendet werden, ein Verfahren bei dem Teilbäume durch Blätter ersetzt werden, um die Komplexität des Modells zu reduzieren. Es gibt zwei Arten von Pruning: Pre-Pruning wird schon während der Erstellung des Baumes durchgeführt, Post-Pruning reduziert den Baum erst nach der vollständigen Berechnung.

Pre-Pruning, auch vorzeitiges Stoppen ("Early Stopping Rule"), ist eine Methode, bei dem die Konstruktion des Baumes beim Erreichen einer bestimmten Baumtiefe angehalten wird.

Pruning nach Kostenkomplexität ist ein Algorithmus der Post-Pruning-Gruppe. Die Reduktion der Kostenkomplexität wird nach einer Teilbaumbewertung nach der "Residual Sum of Squares", Residuenquadratsumme (RSS), berechnet. RSS ist eine statistische Technik, um Varianz in einem Datensatz zu ermitteln. Neben RSS wird mit dem
Baumkomplexitäts-Strafwert, eine Funktion über die Anzahl der Blätter im Teilbaum, die Größe der Teilbäume ausgeglichen.

Numerisch ist der Baumwert wie folgt definiert:

$$Baumwert = RSS + aT$$

(a, alpha, ist ein Hyperparameter, den man durch "Cross-Validation" findet und T ist die Anzahl der Blätter im Unterbaum.)

Man berechnet den Baumwert für alle Teilbäume im Decision Tree und wählt dann den Teilbaum mit dem niedrigsten Baumwert. Aus der Gleichung lässt sich erkennen, dass der Wert von alpha die Wahl des Teilbaums bestimmt. Alpha wird durch "Cross-Validation" ermittelt. Man wiederholt den Vorgang für verschiedene alpha-Werte, was eine Folge von Bäumen ergibt. Der alpha-Wert, der im Durchschnitt den niedrigsten Baumwert ergibt, ist der endgültige alpha-Wert. Je größer er ist, umso stärker wird der Baum beschnitten.

[Monum, 2021, Mohan, 2021]

3.2.2 Modell Einblicke und partielle Abhängigkeitsdiagramme

Um das Modell verbessern zu können, sind Einblicke in die Struktur des Modells wichtig. So lassen sich Fragen wie "Nach welchem Schema werden die Vorhersagen erstellt?" und "Welche Features der Daten sind besonders relevant?" beantworten. Der Baum kann als Diagramm erstellt werden, um den Pfad, der der Voraussage zu Grunde liegt, nachvollziehen zu können. Der Baum ist von oben nach unten zu lesen. In jedem Knoten wird der Wert eines Features überprüft und wenn die Bedingung erfüllt wird, also der Wert kleiner als der Vergleichswert ist, wird der Pfad nach links unten gewählt, andernfalls rechts unten [Becker, 2021d] (Figure 5).

Um den Effekt eines einzelnen Features auslesen zu können, können sogenannte "Partial Dependence Plots", zu Deutsch "Partielle Abhängigkeitsdiagramme" erstellt werden. Dieses Diagramm zeigt für ein Feature den Effekt der Veränderung des Wertes auf das vorausgesagte Ergebnis. Die X-Achse zeigt die Werte des Features an und die Y-Achse die Auswirkung auf die Wahrscheinlichkeit, dass die Vorhersage zutrifft, es sich also um die Mutation und nicht den Wildtypen handelt [Becker, 2021a] (Figure 6).

3.3 Random Forest

Random Forest ist eine Machine Learning Technik, welche wie der Decision Tree Algorithmus sowohl Regressions-, als auch Klassifizierungsaufgaben

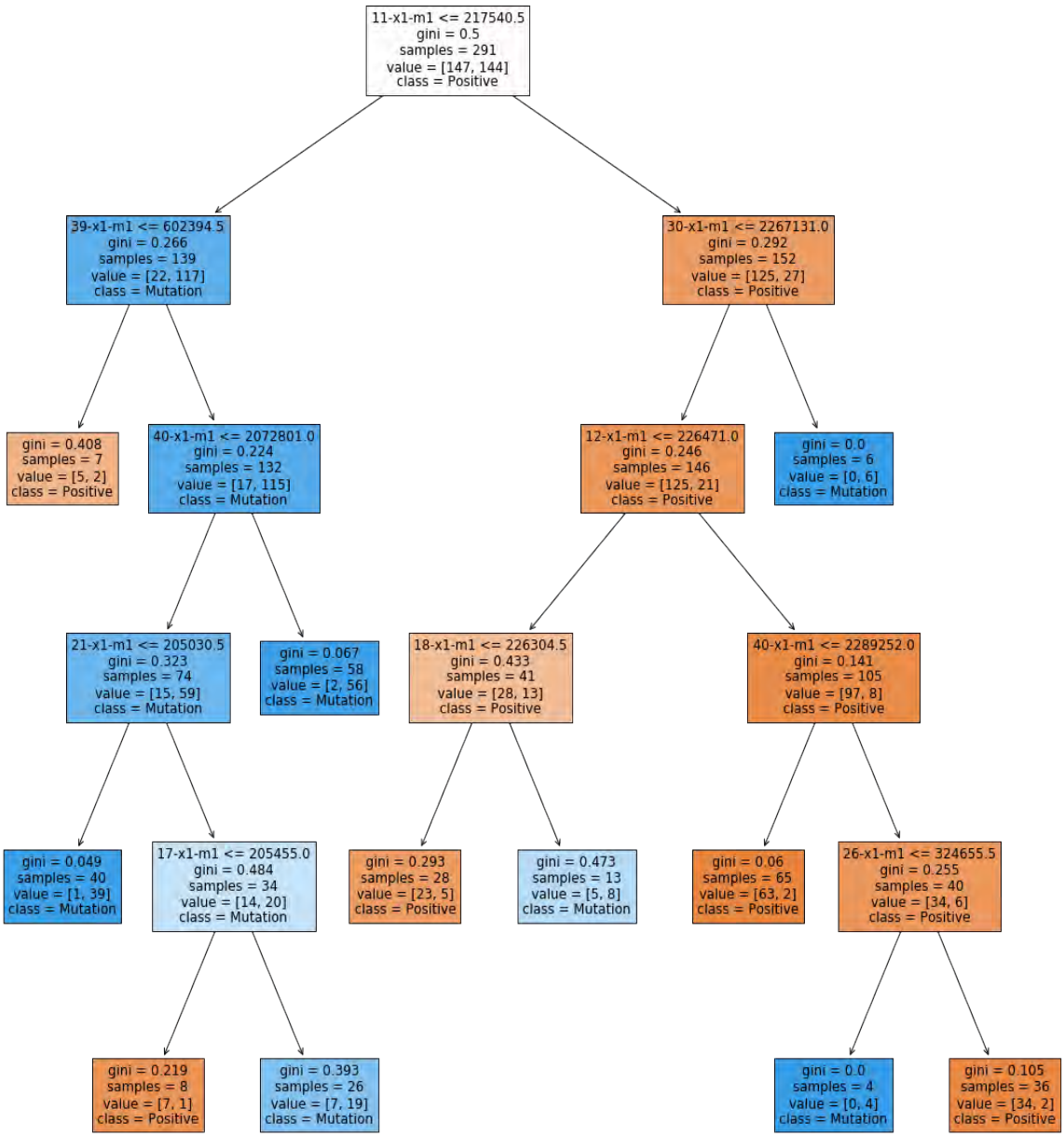


Figure 5: Modell eines Decision Trees

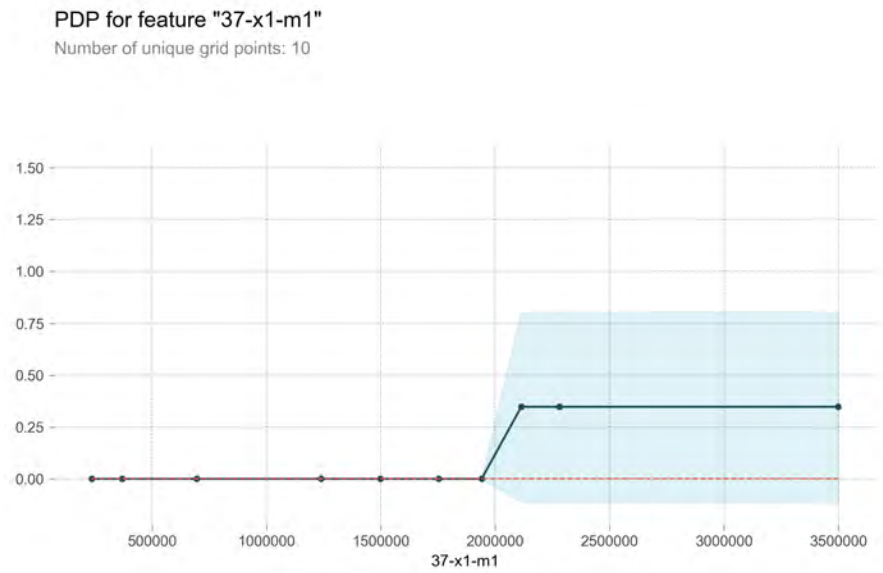


Figure 6: Partielles Abhängigkeitsdiagramm für den Wert des 37. Zyklus

lösen kann. Ein Random Forest Modell besteht aus mehreren Decision Trees, die zufällige Unterschiede in ihren Knoten und Entscheidungen aufweisen. Das Random Forest Modell verwendet diese Entscheidungsbäume, um auf deren Basis bessere Voraussagen treffen zu können. Dazu wird die "Ensemble" Lernmethode verwendet. Jeder Baum des Random Forest lernt von einem zufällig ausgewählten Teil der Trainingsdaten und bewertet die Verifizierungsdaten. Das Modell des Random Forest übernimmt die Voraussage der Mehrheit der Bäume. Die Idee ist, dass so über die Varianz der einzelnen Bäume eine bessere Klassifizierung und weniger Bias (Verzerrung) erreicht wird. Falsche Zusammenhänge in den einzelnen Modellen der Decision Trees werden durch die anderen ausgeglichen und so das "Overfitting" reduziert [Yiu, 2019, Drakos, 2019b, Donges, 2021].

3.3.1 Cross-Validation

Beim Erstellen eines Modells werden die Daten in zwei Teile aufgeteilt, üblicherweise sind etwa 70% des Datensatzes Trainingsdaten und 30% werden für die Validierung verwendet. Um die Qualität und Genauigkeit des Modells

zu bewerten, werden die Voraussagen betreffend den Validierungsteil untersucht. Für eine bessere Einschätzung gerade bei kleinen Datensätzen gibt es die Möglichkeit, den Algorithmus mehrmals mit unterschiedlichen Trainings- und Validierungsdatensätzen auszuführen. Dazu werden die gesamten Daten in fünf gleiche Teile geteilt und jeder Teil dient bei fünf Durchläufen einmal zur Validation, die anderen vier zusammen als Trainingsdaten. Die Genauigkeit des Algorithmus wird dann aus dem Durchschnitt der fünf Modelle berechnet [Cook, 2021].

3.3.2 Permutation Importance

Eine weitere Technik, um festzustellen, welche Features den größten Einfluss auf unser Modell haben, ist "Permutation Importance", also die Bedeutung der Anordnung der Werte eines Features. Permutation Importance ist eine gute Möglichkeit, um Verständnis für die Relevanz von Features zu erhalten, da die Berechnung konsistent und einfach zu verstehen ist. Um die Relevanz der Features zu berechnen und zu vergleichen, mischt der Algorithmus alle Werte eines Features (also die Werte einer Spalte) und überprüft, wie stark dies die Vorhersage des Modells verschlechtert. Bei relevanten Features führt dies zu einer deutlichen Verschlechterung der Vorhersage, während sie für das Modell unwichtige Daten die Genauigkeit kaum beeinflussen. Das Ergebnis kann man sich als "Ranking", d.h. gewichtet ausgeben lassen. Die Zahlen zeigen dazu an, um wie viel Prozent sich das Modell bei der Durchmischung verschlechtert hat [Becker, 2021b] (Figure 7).

3.3.3 SHAP Values

Als letzte verwendete Methode des Random Forest Algorithmus bleiben noch "SHAP-Values" vorzustellen. Der Name steht für "SHapley Additive exPlanations", in etwa "Shapleys zusätzliche Erläuterungen", eine Möglichkeit, die Zusammensetzung einer Voraussage genauer aufzuschlüsseln, und so den Einfluss eines bestimmten Wertes des Features auf das Gesamtergebnis zu sehen. SHAP-Values können beispielsweise verwendet werden, um eine Ablehnung eines Kredits auf Basis eines Machine Learning Modells zu begründen; Banken sind Kunden gegenüber verpflichtet, dies offenzulegen. Um den Einfluss des Wertes eines Features zu bestimmen, wird die Genauigkeit der Voraussage mit einer anderen Voraussage unter Verwendung eines Basiswertes des Features verglichen [Lundberg, 2021a, Becker, 2021c, Lundberg, 2021b].

Weight	Feature
0.0186 ± 0.0240	12-x1-m1
0.0062 ± 0.0101	11-x1-m1
0.0021 ± 0.0154	38-x1-m1
0.0021 ± 0.0154	14-x1-m1
0 ± 0.0000	37-x1-m1
0.0000 ± 0.0130	36-x1-m1
0 ± 0.0000	13-x1-m1
0 ± 0.0000	19-x1-m1
0 ± 0.0000	20-x1-m1
0 ± 0.0000	22-x1-m1
0 ± 0.0000	23-x1-m1
0 ± 0.0000	24-x1-m1
0.0000 ± 0.0130	39-x1-m1
-0.0021 ± 0.0082	31-x1-m1
-0.0021 ± 0.0240	40-x1-m1
-0.0041 ± 0.0101	34-x1-m1
-0.0041 ± 0.0101	15-x1-m1
-0.0041 ± 0.0101	17-x1-m1
-0.0062 ± 0.0101	21-x1-m1
-0.0062 ± 0.0101	16-x1-m1
… 10 more …	

Figure 7: Permutation Importance des Random Forest Algorithmus

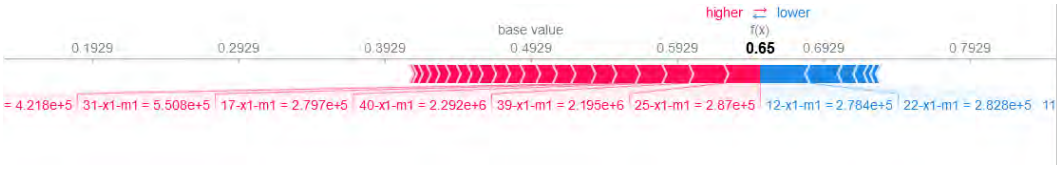


Figure 8: Grafische Darstellung der SHAP-Values

3.4 Neuronales Netz

Das verwendete Neuronale Netz basiert auf Keras aus der Python-Bibliothek Tensorflow. Ein Neuronales Netz besteht aus einzelnen Neuronen. Ein Neuron ist in der einfachsten Form eine lineare Funktion:

$$y = wx + b$$

Diese Funktion verarbeitet eine Eingabe x, indem sie sie gewichtet, also mit w multipliziert und mit b addiert. b ermöglicht es, das Ergebnis unabhängig von der Eingabe zu verändern. Ein Neuron kann mehrere Eingaben haben, bei drei Eingaben verändert sich die Formel entsprechend zu:

$$y = w_0x_0 + w_1x_1 + w_2x_2 + b$$

Das verwendete sequentielle Modell eines Neuronalen Netzes verbindet die Neuronen in mehreren Schichten. Die erste Schicht erhält die Eingaben, die Zwischenschichten verarbeiten diese weiter und die letzte Schicht gibt die Ergebnisse aus. Die inneren Schichten werden auch als "versteckte Schichten" ("hidden Layer") bezeichnet, da man ihre Ergebnisse, die Zwischenergebnisse des Netzes, nicht sieht. Um dem Neuronalen Netz die Möglichkeit zu geben, komplexere Zusammenhänge abzubilden, werden Schichten mit einer Aktivierungsfunktion ausgestattet. Die einfachste ist die "ReLU"-Funktion, "Rectified linear unit". Diese korrigiert negative Werte zu 0: $max(0, x)$.

Zu Beginn sind die Gewichtungen der Schichten auf zufällige Werte gesetzt. Um die Trainingsdaten möglichst gut abbilden zu können, wird das Netz trainiert und die Werte werden mit jedem Durchlauf angepasst. Dafür wird eine "Verlust-Funktion" ("Loss-Function") verwendet, die den Unterschied zwischen dem berechneten und tatsächlichen Wert z.B. als mittleren absoluten Fehler ("mean absolute Error - MAE") ermittelt. Eine andere Funktion, passend für das vorliegende Klassifizierungsproblem, ist die Binäre-Kreuzentropie ("Binary cross-entropy"), die die Differenz der vorhergesagten Wahrscheinlichkeit für eine Klasse 0 oder 1 mit der tatsächlichen angibt. Würde beim verwendeten Datensatz eine Mutation (Klasse 1) also mit einer Wahrscheinlichkeit von 0.95, 95%, dieser zugeordnet, wäre das eine geringe Differenz und die Gewichtung würde in etwa so beibehalten werden. Die Veränderung der Gewichte zugunsten geringer Verluste übernimmt ein Optimierungsalgorithmus. Dieser Algorithmus gehört zur Klasse des Stochastischen Gradientenabstiegs ("Stochastic Gradient Descent"). Für das Training werden mehrere Schritte ausgeführt und wiederholt:

1. Das Netz wird mit einem Teil der Trainingsdaten gefüttert und Vorhersagen werden getroffen.

2. Die Differenz (Verlust-Funktion) zwischen den Vorhersagen und tatsächlichen Werten wird berechnet.

3. Die Gewichtungen werden angepasst, sodass die Differenz geringer wird.

Diese Schritte werden so oft wiederholt, bis der Verlust nicht mehr geringer wird.

Eine Batch-Normalisierungsschicht dient dazu, die Werte neu zu skalieren, sodass das Netzwerk besser mit ihnen arbeiten kann. Die Fluoreszenzwerte der Messungen reichen von 200.000 bis über 3.000.000 und werden durch diese Schicht angepasst und normalisiert. Beim Trainieren passiert es oft, dass bestimmte Wertkombinationen erreicht werden, die sich zwar in den Testdaten finden, aber nicht in den Validierungsdaten ("Overfitting"). Um Overfitting zu vermeiden, wird ein "Dropout-Layer" verwendet, eine Schicht die zufällig verschiedene Verbindungen löscht, sodass nur stabile Muster zur Klassifizierung verwendet werden können [Fortuner, 2019, Holbrook, 2021].

Es ergibt sich folgende Schichtstruktur für das verwendete Neuronale Netzwerk:

Layer	Output Shape	
Dense	(None, 32)	activation='relu'
BatchNormalization	(None, 32)	
Dropout	(None, 32)	0.2
Dense	(None, 1)	activation='sigmoid'

Table 2: Schichtstruktur des Neuronalen Netzes

3.5 Metriken

Um die Effektivität von Klassifizierungsalgorithmen zu bestimmen, lassen sich verschiedene Metriken anwenden. Die erste verwendete Metrik ist Genauigkeit (Accuracy). Genauigkeit berechnet sich aus dem Verhältnis der Anzahl der richtig eingeordneten Messungen zur Gesamtanzahl der Messungen. Für diese Metrik ist es wichtig, dass die Klassen ausgewogen sind,

da auch ein hoher Wert erzielt wird, wenn das Modell viele Messungen der quantitativ höheren Gruppe zuordnet. Einen detaillierten Einblick gibt die Confusion Matrix. Die Confusion Matrix ist eine Tabelle, die zeigt wie viele Messungen das Modell korrekt den Klassen zugeordnet hat und wo es falsch lag. In diesem Fall, mit zwei Klassen, ergibt sich eine 2x2-Matrix mit den Feldern "korrekt positiv", "falsch positiv", "korrekt negativ" und "falsch negativ". Die Confusion Matrix lässt sich auch visuell als Heatmap darstellen und bildet die Grundlage für weitere Metriken.

	Actual positive	**Actual negative**
Predicted positive	True positive	False positive
Predicted negative	False negative	True negative

Table 3: Confusion Matrix

Mit den aus der Confusion Matrix ablesbaren Werten lassen sich Precision, zu Deutsch Präzision und Recall berechnen. Precision ist der Anteil der korrekt zugeordneten einer Klasse im Verhältnis zu allen Zuordnungen dieser Klasse. Recall ist der Anteil der richtig positiv eingeordneten im Verhältnis zu allen positiven Messungen.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Der F1-Wert ist das harmonische Mittel von Precision und Recall und dient dazu, ein Verhältnis der beiden Werte anzugeben. Er berechnet sich nach folgender Formel:

$$F1Wert = \frac{2 * Precision * Recall}{Precision + Recall}$$

[Mishra, 2021, scikit learn, 2021a, Markham, 2014]

4 Ergebnisse

Für die Algorithmen liegen unterschiedliche Ergebnisse vor. Zum einen wird für den Decision Tree und Random Forest die mittlere Genauigkeit

der cross-validierten Berechnungen und für das Neuronale Netz die mittlere Genauigkeit mehrfacher Berechnungen angegeben. Bei den cross-validierten Berechnungen wurde der Datensatz, bestehend aus insgesamt 388 Messungen zu gleichen Teilen Wildtyp und Alpha-Mutation, in fünf gleiche Teile aufgeteilt und jeder Teil diente bei fünf Durchläufen einmal zur Validation und viermal zum Training. Für das Neuronale Netz ist die Aufteilung 75% Trainingsdaten, also 291 Datensätze, und 25% bzw. 97 Datensätze zur Validation.

Um die im Vergleich weniger aussagekräftigen, aber trotzdem interessanten Metriken wie Precision und Recall berechnen zu können, wird ein Modell der Algorithmen verwendet und die Confusion Matrix betrachtet. Hierbei ist die Anzahl der Trainingsdatensätze ebenfalls 291 und 97 dienen zur Validation. Als Kontroll- und Vergleichsgruppe dient für jeden Algorithmus die Genauigkeit der Zuordnung in die Klassen *Wildtyp* oder *negative Messung* mit gleicher Anzahl und Aufteilung.

4.1 Decision Tree

Ergebnisse für die Klassifizierung von Wildtyp oder Mutation:

Genauigkeit unter Verwendung von Cross-Validation: **0.789**
Genauigkeit unter Verwendung von Pruning: **0.825**

Confusion Matrix eines Decision Tree Modells:

	Actual positive	**Actual negative**
Predicted positive	38	9
Predicted negative	6	44

Table 4: Confusion Matrix eines Decision Tree Modells

$$Accuracy_{DT} = \frac{82}{97} \approx 0.85$$

$$Precision_{DT} = \frac{38}{38+9} = \frac{38}{47} \approx 0.81$$

$$Recall_{DT} = \frac{38}{38+6} = \frac{38}{44} \approx 0.86$$

$$F1Wert_{DT} = \frac{2 * \frac{38}{47} * \frac{38}{44}}{\frac{38}{47} + \frac{38}{44}} \approx 0.84$$

Ergebnisse für die Klassifizierung von negativer Messung oder Wildtyp:

Genauigkeit unter Verwendung von Cross-Validation: **0.974**

4.2 Random Forest

Ergebnisse für die Klassifizierung von Wildtyp oder Mutation:

Genauigkeit unter Verwendung von Cross-Validation: **0.840**

Confusion Matrix eines Random Forest Modells:

	Actual positive	**Actual negative**
Predicted positive	40	7
Predicted negative	7	43

Table 5: Confusion Matrix eines Random Forest Modells

$$Accuracy_{RF} = \frac{83}{97} \approx 0.86$$

$$Precision_{RF} = \frac{40}{40 + 7} = \frac{40}{47} \approx 0.85$$

$$Recall_{RF} = \frac{40}{40 + 7} = \frac{40}{47} \approx 0.85$$

$$F1Wert_{RF} = \frac{2 * \frac{40}{47} * \frac{40}{47}}{\frac{40}{47} + \frac{40}{47}} \approx 0.85$$

Ergebnisse für die Klassifizierung von negativer Messung oder Wildtyp:

Genauigkeit unter Verwendung von Cross-Validation: **0.985**

4.3 Neuronales Netz

Ergebnisse für die Klassifizierung von Wildtyp oder Mutation:

Das neuronale Netz verhielt sich bei jeder neuen Berechnung sehr unterschiedlich. Dabei war es unerheblich, ob die Berechnung mit unterschiedlichen oder gleichen Trainingsdaten ausgeführt wurde.

Mittlere Genauigkeit des Algorithmus: **0.765**

	Actual positive	**Actual negative**
Predicted positive	43	4
Predicted negative	22	28

Table 6: Confusion Matrix eines Neuronalen Netzes

$$Accuracy_{NN} = \frac{71}{97} \approx 0.73$$

$$Precision_{NN} = \frac{43}{43+4} = \frac{43}{47} \approx 0.91$$

$$Recall_{NN} = \frac{43}{43+22} = \frac{43}{65} \approx 0.66$$

$$F1Wert_{NN} = \frac{2 * \frac{43}{47} * \frac{43}{65}}{\frac{43}{47} + \frac{43}{65}} \approx 0.77$$

Ergebnisse für die Klassifizierung von negativer Messung oder Wildtyp:

Mittlere Genauigkeit des Algorithmus: **0.977**

5 Diskussion

Die Ergebnisse lassen sich folgendermaßen interpretieren: Die Metrik, die für den Vergleich der Algorithmen am besten geeignet ist, ist die Genauigkeit, da sie zeigt, wie gut die Algorithmen mit den Daten arbeiten können. Hier hat klar Random Forest das beste Ergebnis mit 84%, gefolgt von Decison Tree

mit 79% und dem Neuronalen Netz mit 77% auf dem letzten Platz. Einzelne Modelle der Algorithmen liefern höhere Genauigkeitswerte, aufgrund der geringen Anzahl an Validierungsdaten sind diese Zahlen jedoch weniger aussagekräftig als die Genauigkeit im cross-validierten Mittel. Diese Ergebnisse zeigen trotz des kleinen Datensatzes eine gute Leistung der Algorithmen in der Klassifizierung. Die Unterscheidung zwischen negativen und Wildtyp-Messungen ist wie erwartet sehr gut ausgefallen. Es ist durchaus möglich, dass bei einer Datensatzgröße jenseits der 5000 die Struktur eines Neuronalen Netzes die Zusammenhänge besser modelliert. Grundsätzlich scheinen alle drei Algorithmen geeignet, bei der Klassifizierung zu unterstützen.

6 Fazit

Maschinelles Lernen bietet ein großes Potential, wenn es um die Unterstützung von Klassifizierungen medizinischer Daten geht. Die Arbeit hat gezeigt, dass bereits kleine Datensätze von wenigen Hundert Einträgen ausreichen, um Modelle zu erhalten, die mit 80-prozentiger Genauigkeit eine Corona Mutation vom Wildtyp unterscheiden können, also bei der Untersuchung unterstützen. Mit größeren Trainingsdatensätzen sind Modelle denkbar, die diese Zuordnungen verlässlich vornehmen können.

References

[Awoin et al., 2020] Awoin, E., Ap-piahene, P., Gyasi, F., and Sabtiwu1, A. (2020). Predicting the performance of rural banks in ghana using machine learning approach. Retrieved 22.10.2021 from https://www.hindawi.com/journals/afs/2020/8028019/.

[Becker, 2021a] Becker, D. (2021a). Partial plots. Retrieved 30.08.2021 from https://www.kaggle.com/dansbecker/partial-plots.

[Becker, 2021b] Becker, D. (2021b). Permutation importance. Retrieved 30.08.2021 from https://www.kaggle.com/dansbecker/permutation-importance.

[Becker, 2021c] Becker, D. (2021c). Shap values. Retrieved 02.09.2021 from https://www.kaggle.com/dansbecker/shap-values.

[Becker, 2021d] Becker, D. (2021d). Use cases for model insights. Retrieved 16.08.2021 from https://www.kaggle.com/dansbecker/use-cases-for-model-insights.

[BioRad, 2021] BioRad (2021). Introduction to pcr primer probe chemistries. Retrieved 03.09.2021 from https://www.bio-rad.com/de-de/applications-technologies/introduction-pcr-primer-probe-chemistries?ID=LUSOJW3Q3.

[Cook, 2021] Cook, A. (2021). Cross-validation. Retrieved 26.08.2021 from https://www.kaggle.com/alexisbcook/cross-validation.

[Donges, 2021] Donges, N. (2021). A complete guide to the random forest algorithm. Retrieved 09.10.2021 from https://builtin.com/data-science/random-forest-algorithm.

[Drakos, 2019a] Drakos, G. (2019a). Decision tree regressor explained in depth. Retrieved 03.07.2021 from https://gdcoder.com/decision-tree-regressor-explained-in-depth/.

[Drakos, 2019b] Drakos, G. (2019b). Random forest regressor explained in depth. Retrieved 03.07.2021 from https://gdcoder.com/random-forest-regressor-explained-in-depth/.

[Foote, 2019] Foote, K. D. (2019). A brief history of machine learning. Retrieved 16.08.2021 from https://www.dataversity.net/a-brief-history-of-machine-learning/.

[Fortuner, 2019] Fortuner, B. (2019). Loss functions. Retrieved 09.10.2021 from https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html.

[Holbrook, 2021] Holbrook, R. (2021). Intro to deep learning - lessons 1 to 6. Retrieved 30.09.2021 from https://www.kaggle.com/learn/intro-to-deep-learning.

[Kuzilek et al., 2014] Kuzilek, J., Kremen, V., Soucek, F., and Lhotska, L. (2014). Independent component analysis and decision trees for ecg holter recording de-noising. *PLOS ONE*, 9(6):1–9.

[Langer et al., 2020] Langer, T., Favarato, M., Giudici, R., Bassi, G., Garberi, R., Villa, F., Gay, H., Zeduri, A., Bragagnolo, S., Molteni, A., Beretta, A., Corradin, M., Moreno, M., Vismara, C., Perno, C. F., Buscema, M., Grossi, E., and Fumagalli, R. (2020). Development of machine learning models to predict rt-pcr results for severe acute respiratory syndrome coronavirus 2 (sars-cov-2) in patients with influenza-like symptoms using only basic clinical data. Retrieved 22.10.2021 from https://link.springer.com/article/10.1186/s13049-020-00808-8.

[Lundberg, 2021a] Lundberg, S. (2021a). Shap. Retrieved 02.09.2021 from https://github.com/slundberg/shap.

[Lundberg, 2021b] Lundberg, S. (2021b). Shap documentation. Retrieved 02.09.2021 from https://shap.readthedocs.io/en/latest/index.html.

[Markham, 2014] Markham, K. (2014). Simple guide to confusion matrix terminology. Retrieved 09.10.2021 from https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/.

[McKinney et al., 2021] McKinney, S. M., Sieniek, M., Godbole, V., Godwin, J., Antropova, N., Ashrafian, H., Back, T., Chesus, M., Corrado, G. S., Darzi, A., Etemadi, M., Garcia-Vicente, F., Gilbert, F. J., Halling-Brown, M., Hassabis, D., Jansen, S., Karthikesalingam, A., Kelly, C. J., King, D., Ledsam, J. R., Melnick, D., Mostofi, H., Peng, L., Reicher, J. J., Romera-Paredes, B., Sidebottom, R., Suleyman, M., Tse, D., Young,

K. C., De Fauw, J., and Shetty, S. (2021). International evaluation of an ai system for breast cancer screening. Retrieved 03.09.2021 from https://www.nature.com/articles/s41586-019-1799-6.pdf.

[Mishra, 2021] Mishra, A. (2021). What is decision tree pruning and how is it done? Retrieved 09.10.2021 from https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234.

[Mohan, 2021] Mohan, A. (2021). Pruning decision trees - tutorial. Retrieved 30.09.2021 from https://www.kaggle.com/arunmohan003/pruning-decision-trees-tutorial.

[Monum, 2021] Monum, A. (2021). What is decision tree pruning and how is it done? Retrieved 30.09.2021 from https://www.educative.io/edpresso/what-is-decision-tree-pruning-and-how-is-it-done.

[Prajwala, 2015] Prajwala, T. R. (2015). A comparative study on decision tree and random forest using r tool. Retrieved 22.10.2021 from https://www.researchgate.net/profile/Prajwala-T-R/publication/272425234_A_Comparative_Study_on_Decision_Tree_and_Random_Forest_Using_R_Tool/links/5fb4bfeca6fdcc9ae05ef42b/A-Comparative-Study-on-Decision-Tree-and-Random-Forest-Using-R-Tool.pdf.

[Samuel, 1959] Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229.

[scikit learn, 2021a] scikit learn (2021a). Precision-recall. Retrieved 09.10.2021 from https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html.

[scikit learn, 2021b] scikit learn (2021b). Tree algorithms: Id3, c4.5, c5.0 and cart. Retrieved 20.08.2021 https://scikit-learn.org/stable/modules/tree.html.

[Silverio, 2020] Silverio, M. (2020). Google ai for breast cancer detection beats doctors. Retrieved 16.08.2021 from

https://towardsdatascience.com/google-ai-for-breast-cancer-detection-beats-doctors-65b8983352e0.

[ThermoFisher, 2015] ThermoFisher (2015). Quantstudio™ 3 and 5 real-time pcr systems - installation, use, and maintenance. Retrieved 27.09.2021 from https://tools.thermofisher.com/content/sfs/manuals/MAN0010407_QuantStudio3_5_InstallUseMaint_UG.pdf, Page 11.

[Tso and Yau, 2007] Tso, G. K. and Yau, K. K. (2007). Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy*, 32(9):1761–1768.

[Yiu, 2019] Yiu, T. (2019). Understanding random forest. Retrieved 17.08.2021 from https://towardsdatascience.com/understanding-random-forest-58381e0602d2.

Eigenständigkeitserklärung

Hiermit erkläre ich, Johannes Böhm, dass ich die vorliegende Arbeit eigenständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.
Alle sinngemäß und wörtlich übernommenen Ausführungen aus fremden Quellen wurden kenntlich gemacht. Die Bachelorarbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Ort, Datum	Unterschrift