

Probabilistisches maschinelles Lernen von Entscheidungs- bäumen für Unsicherheit in medizinischen Daten

SEMINARARBEIT

eingereicht bei

Prof. Dr. Bernd Heinrich

Institut für Wirtschaftsinformatik

Fakultät für Informatik und Data Science

Universität Regensburg

von

Alina Weichselgartner

Nelkenweg 9, 94351 Feldkirchen

MR: 2148270

Wirtschaftsinformatik (M. Sc.)

Lucas Luttner

Im Gewerbegebiet 34a, 94369 Rain

MR: 2029371

Wirtschaftsinformatik (M. Sc.)

Regensburg, 24.02.2023

Inhaltsübersicht

Abkürzungsverzeichnis	iii
Tabellenverzeichnis	iv
1 Einleitung und Motivation	1
2 Literaturüberblick	3
3 Methodik zur Berücksichtigung von Unsicherheit in Entscheidungsbäumen	6
3.1 Vorverarbeitung der Daten.....	6
3.2 Erzeugung von Wahrscheinlichkeiten	6
3.3 Erstellung des Entscheidungsbaumes	8
3.4 Erstellung des probabilistischen Random Forest	12
4 Evaluation der Methodik	16
5 Diskussion der Ergebnisse.....	22
6 Zusammenfassung und Ausblick.....	25
Anhang A: Weitere Unterlagen und Ausführungen	26
7 Literatur	31

Inhaltsverzeichnis

Abkürzungsverzeichnis	iii
Tabellenverzeichnis	iv
1 Einleitung und Motivation	1
2 Literaturüberblick	3
3 Methodik zur Berücksichtigung von Unsicherheit in Entscheidungsbäumen	6
3.1 Vorverarbeitung der Daten.....	6
3.2 Erzeugung von Wahrscheinlichkeiten	6
3.2.1 Clustering.....	7
3.2.2 k-nächste-Nachbarn Verfahren	8
3.3 Erstellung des Entscheidungsbaumes	8
3.3.1 Training des Entscheidungsbaumes.....	8
3.3.2 Klassifikation mit dem Entscheidungsbaum	11
3.4 Erstellung des probabilistischen Random Forest	12
3.4.1 Bootstrapping.....	12
3.4.2 Training.....	13
3.4.3 Klassifikation.....	14
4 Evaluation der Methodik	16
5 Diskussion der Ergebnisse.....	22
6 Zusammenfassung und Ausblick.....	25
Anhang A: Weitere Unterlagen und Ausführungen	26
A.0 Heart-Disease: Ergänzende Auswertung.....	26
A.1 Heart-Disease: Confusion Matrix – Entscheidungsbaumverfahren	27
A.2 Heart-Disease: Confusion Matrix – Random Forest	28
A.3 Breast-Cancer: Confusion Matrix – Entscheidungsbaumverfahren	29
A.4 Breast-Cancer: Confusion Matrix – Random Forest	30
7 Literatur	31

Abkürzungsverzeichnis

k-nächste-Nachbarn Algorithmus	KNN
Mean Squared Error	MSE
Probabilistischer Random Forest	PRF

Tabellenverzeichnis

Tabelle 2-1 Literaturüberblick	4
---	---

Abbildungsverzeichnis

Abbildung 1-1 Anzahl veröffentlichter Artikel über Unsicherheit in medizinischen Daten.....	1
Abbildung 3-1 Datensatz nach Wahrscheinlichkeitserzeugung	7
Abbildung 3-2 Entropieberechnung	9
Abbildung 3-3 Berechnung des Informationsgewinns	9
Abbildung 3-4 Aktualisierung des Dataframes	10
Abbildung 3-5 Klassifikation einer Instanz.....	11
Abbildung 3-6 Pseudocode Fit Methode	13
Abbildung 3-7 Pseudocode PRF Classifier	14
Abbildung 3-8 Klassifizierung mit Random Forest	15
Abbildung 4-1 Bestimmung des nächsten Split-Attributes	16
Abbildung 4-2 Aufbau trainierter Entscheidungsbaum (Ausschnitt).....	17
Abbildung 4-3 Klassifikation der ersten 7 Instanzen	17
Abbildung 4-4 Accuracy von Datensatz 1.....	18
Abbildung 4-5 Mean Squared Error von Datensatz 1	19
Abbildung 4-6 Accuracy von Datensatz 2.....	20
Abbildung 4-7 Mean Squared Error von Datensatz 2	20

1 Einleitung und Motivation

Hohes Cholesterin, Vorerkrankungen und Bluthochdruck – alles das sind Faktoren, die auf eine Herzerkrankung hindeuten können. Aber was wäre, wenn sich Krankheiten schon im Voraus vorhersagen lassen würden? Der Einsatz von Machine-Learning-Algorithmen macht dies möglich. So können künftig Erkrankungen früher entdeckt werden und es besteht zusätzlich die Möglichkeit, schneller geeignete Maßnahmen zu ergreifen. Durch Früherkennungsuntersuchungen können relevante Daten gesammelt werden, um Ärzte bei der Krankheitsdiagnose zu unterstützen. Ein sehr großes Hindernis daran stellt allerdings die oftmals mangelnde Datenqualität von medizinischen Daten dar. So können beispielsweise Werte fehlen oder falsch sein. Werden inkonsistente Daten zur Prognose verwendet, so führt dies in klassischen Machine-Learning-Verfahren auch zu ungenauen Resultaten. Besonders in kritischen Bereichen, wie beispielsweise der Medizin, können unsichere Daten direkte Auswirkungen auf die Gesundheit der Patienten haben. Klinische Studien könnten so bei mangelnder Datenqualität zu unzuverlässigen Ergebnissen führen. Dies hat starke negative Auswirkungen auf die Erkennung von neuen und bereits bestehenden Krankheiten eines Patienten. Die hohe Relevanz des Themas spiegelt sich auch in der folgenden Abbildung von Alizadehsani et al. (2021) wider.

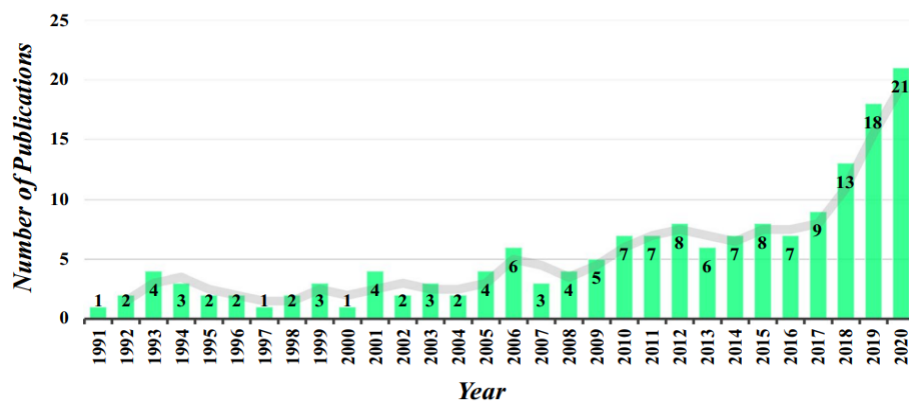


Abbildung 1-1 Anzahl veröffentlichter Artikel über Unsicherheit in medizinischen Daten

Quelle: Alizadehsani et al. (2021, S. 4)

Besonders seit dem Jahre 2017 steigt die Anzahl an wissenschaftlichen Artikeln zu Unsicherheit in medizinischen Daten nahe zu exponentiell. Dieser Trend wird zum einen vorangetrieben durch die Entwicklung neuer Algorithmen, sowie durch die rasant steigenden Datenmengen und zugleich sinkenden Speicherkosten. Neben dem Umgang mit Unsicherheit spielt auch in der Medizin besonders die Erklärbarkeit des Verfahrens eine große Rolle. Ergebnisse müssen für Ärzte interpretierbar bleiben, damit eine Diagnose auch sinnvoll begründet werden kann. Für solch eine Problemstellung eignet sich das Grundgerüst des Entscheidungsbaumverfahrens besonders gut. Durch Modifikation des klassischen Entscheidungsbaumes können Wahrscheinlichkeiten berücksichtigt werden. Somit wird bei der Klassifizierung einer Instanz nicht nur eine einzige Klasse ausgegeben, sondern für jede Zielklasse die Wahrscheinlichkeit, mit der die

Instanz dieser Klasse zugeordnet ist. Da Entscheidungsbäume jedoch anfällig für Overfitting sind, sind Ensemble-Methoden wie beispielsweise der Random Forest eine sinnvolle Erweiterung zur Maximierung der Performance gegenüber neuen, unbekannten Daten.

Da ein solches Verfahren noch nicht existiert, ist es Ziel dieser Arbeit, folgende Forschungsfrage zu beantworten: **Wie können unsichere Inputdaten in baumbasierten maschinellen Lernverfahren berücksichtigt werden?**

Hierfür wird im ersten Teil der Arbeit zunächst auf bereits bestehende Literatur zum Thema Unsicherheit in Baum-basierten Verfahren eingegangen. Im Hauptteil der Arbeit wird die gewählte Methodik zur Berücksichtigung von unsicheren Inputdaten im Entscheidungsbaum-Verfahren, als auch im Random Forest erläutert. Dabei wird zuerst auf die Vorverarbeitung der Daten und das Erzeugen der Wahrscheinlichkeiten für die unsicheren Werte mittels Clustering und k-nächste-Nachbarn-Verfahren näher eingegangen. Darauf folgend wird die Methodik des Trainings und der Klassifizierung von Instanzen bei Entscheidungsbäumen und dem Random Forest auf Basis von Wahrscheinlichkeitsverteilungen erörtert. Anschließend werden die Ergebnisse anhand von zwei medizinischen Datensätzen evaluiert, um den Erfolg der Methode zu quantifizieren. Hierbei liegt der Fokus auf den beiden Gütemaßen Accuracy und Mean Squared Error. Zusätzlich beinhaltet dieses Kapitel beispielhafte Python-Outputs eines ausgewählten Datensatzes. Im fünften Kapitel werden die Ergebnisse diskutiert und im gleichen Zuge Vorteile, sowie Limitationen dieser Methodik aufgezeigt. Abschließend erfolgt eine Zusammenfassung und ein Ausblick über weitere potenzielle Forschungsfelder im Bereich Datenunsicherheit bei Entscheidungsbäumen.

2 Literaturüberblick

Ziel dieses Kapitels ist es, verwandte Arbeiten zu Datenunsicherheit bei Entscheidungsbäumen vorzustellen. Hierfür ist eine systematische Literaturanalyse durchgeführt worden. Als Grundlage für die Analyse sind die Datenbanken ACM, IEEE, Science Direct und zur Ergänzung Google Scholar verwendet worden. Durch Vorauswertung der Artikel über Abstract und Titel sind 12 Arbeiten als relevant identifiziert worden. Davon sind vier Quellen besonders relevant und werden aus diesem Grund in diesem Kapitel genauer erörtert. Dabei werden zunächst alle relevanten Artikel beschrieben und anschließend in einer Übersichtstabelle zusammengefasst. Diese Übersicht soll einen Überblick über die verschiedenen Ansätze und Methoden im Bereich der Datenunsicherheit bei Entscheidungsbäumen geben und dazu beitragen, die Stärken und Schwächen der einzelnen Ansätze zu verdeutlichen, sowie die Relevanz der entwickelten probabilistischen Random Forest Methode (PRF) aufzuzeigen.

Im Artikel von Tsang et al. (2009) wird die Verwendung von Entscheidungsbäumen zur Klassifikation von Test-Tupeln im Kontext unsicherer Attribute untersucht. Das Klassifikationsmodell wird durch eine Funktion M dargestellt, die einen Merkmalsvektor, bestehend aus Wahrscheinlichkeitsdichtefunktionen für jedes Attribut, auf eine Wahrscheinlichkeitsverteilung über eine Klassenmenge C abbildet. Um ein Test-Tupel zu klassifizieren, werden Zwischen-Tupel mit Gewichten verbunden und die Wahrscheinlichkeit, dass das Tupel zu einer bestimmten Klasse gehört, wird anhand des Unterbaums berechnet, der an einem bestimmten Knoten verwurzelt ist. Zwei Ansätze zur Verarbeitung unsicherer Daten bei dem Aufbau von Entscheidungsbäumen werden dabei unterschieden und nachfolgend präsentiert. Dazu gehört zum einen der „Averaging“-Ansatz, bei dem die Wahrscheinlichkeitsdichtefunktionen jedes Attributs durch deren Mittelwerte ersetzt werden, und der „Distribution-based“-Ansatz, bei dem alle Stichprobenpunkte berücksichtigt werden, aus denen sich die Wahrscheinlichkeitsdichtefunktionen zusammensetzen. Der „Distribution-based“-Ansatz ist zwar genauer, jedoch weniger effizient als die Mittelwertbildung (Tsang et al. 2009).

Der PRF-Ansatz von Reit et al. (2019) behandelt Features und Labels als Wahrscheinlichkeitsverteilungsfunktionen anstelle von deterministischen Werten, um Unsicherheiten in den Daten zu berücksichtigen. Die PRF-Features werden zu Wahrscheinlichkeitsdichtefunktionen mit Erwartungswerten, die den bereitgestellten Feature-Werten angehören, und Varianzen, die dem entsprechenden Quadrat der Unsicherheiten entsprechen. Die Labels werden zu Massenfunktionen, das heißt, dass jedes Objekt behandelt wird, als hätte es jedes Label zu einer Wahrscheinlichkeit zugeordnet. Die Unterschiede zwischen PRF und Random Forest ergeben sich aus dieser Behandlung, da das PRF im Grenzfall bei geringeren Unsicherheiten zum ursprünglichen Random Forest konvergiert. Das PRF kann dazu beitragen, die Vorhersagegenauigkeit von Random Forests zu verbessern, indem Unsicherheiten in den Eingabedaten berücksichtigt werden. Aufgrund der damit verbundenen zusätzlichen Komplexität kann es jedoch auch weniger effizient sein (Reis et al. 2019).

Ein weiterer verwandter Ansatz von Qin et al. (2009) beschreibt einen Algorithmus zur Erstellung von Entscheidungsbäumen für Klassifizierungsaufgaben. Der Algorithmus beginnt mit

dem Aufbau eines einzelnen Knotens, der die Trainingsstichproben repräsentiert. Wenn die Proben alle der gleichen Klasse angehören, wird der Knoten zu einem Blatt, das mit dieser Klasse beschriftet ist. Gehören die Stichproben nicht alle derselben Klasse an, so wählt der Algorithmus das Attribut aus, das die Stichproben am besten in die einzelnen Klassen einteilt, indem dieser das auf probabilistischer Entropie basierendes Maß, das probabilistische Informationsgewinnverhältnis, verwendet. Dieses Attribut wird das Test-Attribut für den Knoten. Wenn das Test-Attribut numerisch oder unsicher numerisch ist, werden die Daten an einer ausgewählten Position geteilt. Ist das Test-Attribut kategorisch oder unsicher kategorisch, werden die Daten in mehrere Zweige für jeden Wert des Test-Attributs aufgeteilt. Der Vorgang wird anschließend rekursiv wiederholt, bis alle Stichproben für einen bestimmten Knoten derselben Klasse angehören oder es keine verbleibenden Attribute mehr gibt, nach denen die Stichproben aufgeteilt werden können. Im letzteren Fall wird die Klasse mit der höchsten Gewichtung verwendet und der Knoten wird zu einem Blatt, das mit dieser Klasse gekennzeichnet ist. Der Algorithmus kann zur Vorhersage von Klassentypen verwendet werden, indem dieser am Wurzelknoten beginnt, die Testbedingung an jedem Knoten anwendet und dem entsprechenden Zweig auf der Grundlage des Testergebnisses folgt (Qin et al. 2009).

Einen anderen Ansatz spiegelt das Paper von Hristova (2014) wieder, bei der eine Methode zur Berücksichtigung der Aktualität bei der Entscheidungsbaumklassifizierung für Big Data vorgestellt wird. Der Fokus liegt hierbei auf dem Klassifizieren und nicht auf dem Trainieren eines Entscheidungsbaumes bei unsicheren Daten. Die Methode umfasst die Ableitung einer Wahrscheinlichkeit für jeden Knoten im Entscheidungsbaum auf der Grundlage der Struktur des Baumes und der verfügbaren historischen Daten, sowie die Verfeinerung dieser Wahrscheinlichkeit mithilfe zusätzlicher Daten. Die Methode wird an drei verschiedenen Datensätzen demonstriert, von denen zwei aus dem Kontext von Big Data stammen. Zudem zeigt die Methodik eine verbesserte Klassifizierungsgenauigkeit im Vergleich zur Nichtberücksichtigung von Aktualität der Daten und eine verbesserte Effizienz im Vergleich zu einem anderen Ansatz. Die Methode hat einige Einschränkungen, darunter die Tatsache, dass diese speziell für bereits trainierte Entscheidungsbäume entwickelt worden ist und dass möglicherweise nicht immer historischen Daten zur Verfügung stehen (Hristova 2014).

Quelle	Machine Learning Approach		Classification Type		Input Data Type		Uncertainty Type	
	Random Forest	Decision Tree	Binary	Multiclass	Numerical	Categorical	Features	Labels
(Tsang et al. 2009)		x	x		x		x	
(Reis et al. 2019)	x		x	x	x		x	x
(Hristova 2014)		x	x	x	x	x	x	
(Qin et al. 2009)		x	x		x	x	x	

Tabelle 2-1 Literaturüberblick

In Tabelle 2-1 sind alle im oberen Abschnitt erwähnten Artikel nochmals kategorisiert dargestellt. Wie aus der Abbildung zu entnehmen ist, ist nur ein Ansatz von Qin et al. identifiziert worden, welcher sich auf die Verwendung eines Random Forest zur Klassifikation von unsicheren Daten bezieht. Hierbei ist jedoch zu erkennen, dass dieser nicht geeignet ist für kategoriale Daten, welche besonders im medizinischen Bereich der Anamnese sehr dominant sind. Allgemein kann man sagen, dass es große Defizite im Bereich der kategorialen Daten gibt, da auch der Ansatz von Hristova nicht das Trainieren eines Entscheidungsbaums berücksichtigt. Aus den oben genannten Gründen soll eine Methode entwickelt werden, die diese Forschungslücke schließt. Der Ansatz soll sowohl zur binären als auch zur Multi-Label-Klassifikation verwendet werden können und das sowohl für binäre beziehungsweise kategoriale Inputdaten als auch für Daten, die sich sinnvoll in Kategorien zerlegen lassen.

3 Methodik zur Berücksichtigung von Unsicherheit in Entscheidungsbäumen

In Kapitel 3 wird nachfolgend die gewählte Methodik zur Berücksichtigung von unsicheren Inputdaten in Entscheidungsbaumverfahren und Random Forests erläutert und anhand von Beispielen dargelegt. Hierbei wird auf die Vorverarbeitung der Daten, sowie auf die Erzeugung der Wahrscheinlichkeiten eingegangen. Zusätzlich wird die Funktionsweise des Trainings und der Klassifizierung mittels des Entscheidungsbaumverfahrens und des Random Forests thematisiert.


3.1 Vorverarbeitung der Daten

Bevor das eigentliche Baumverfahren angewendet werden kann, muss der Datensatz vorverarbeitet werden. Da diese Methode ausschließlich kategoriale Merkmale voraussetzt, müssen im ersten Schritt der Vorverarbeitung kategoriale und numerische Attribute identifiziert werden. Alle numerischen Attribute werden anschließend in kategoriale Attribute umgewandelt. Dazu gibt es zwei Möglichkeiten. Die erste besteht darin, numerisch verteilte Merkmale manuell in Ausprägungen einzuteilen. Dies ist besonders sinnvoll, wenn es bereits vordefinierte Intervalle gibt, wie es beispielsweise bei dem Body-Mass-Index der Fall ist. Zusätzlich lässt sich die optimale Anzahl an Ausprägungen auch mithilfe des k-Means Clustering bestimmen, unter der Voraussetzung, dass sich das jeweilige Attribut sinnvoll in Cluster unterteilen lässt. Bei der Bestimmung der Anzahl an Ausprägungen ist zudem zu beachten, dass die numerischen Attribute in so wenig Ausprägungen wie möglich unterteilt werden, da eine zu große Zahl an Ausprägungen ansonsten zu Overfitting bei der Anwendung des Entscheidungsbaumverfahrens führen kann. Gelten Daten als unsicher, müssen diese im letzten Schritt aus dem Datensatz entfernt werden.

3.2 Erzeugung von Wahrscheinlichkeiten

Nachdem unsichere Inputdaten mit ausschließlich kategorial- beziehungsweise binär-verteilten Attributen vorliegen, müssen die fehlenden Werte durch ausgewählte Verfahren mit Wahrscheinlichkeiten ersetzt werden. In dieser Arbeit wird dazu das k-nächste-Nachbarn Verfahren (KNN) und das Clustering angewendet. Bevor beide Methoden verwendet werden können, werden alle gelöschten Werte durch eine Mean-Imputation aufgefüllt. Ziel beider Verfahren ist es, dass die Attribute in ihre jeweiligen Ausprägungen unterteilt sind und die unsicheren Werte durch Wahrscheinlichkeiten ersetzt sind.

Attribut1	Attribut2
NaN	0.0
1.0	1.0
1.0	2.0
0.0	NaN
1.0	2.0



Attribut1 (Ausprägung=0)	Attribut1 (Ausprägung=1)	Attribut2 (Ausprägung=0)	Attribut2 (Ausprägung=1)	Attribut2 (Ausprägung=2)
0.2	0.8	1.0	0.0	0.0
0.0	1.0	0.0	1.0	0.0
0.0	1.0	0.0	0.0	1.0
1.0	0.0	0.1	0.5	0.4
0.0	1.0	0.0	0.0	1.0

Abbildung 3-1 Datensatz nach Wahrscheinlichkeitserzeugung

Abbildung 3-1 zeigt dabei, in welchem Format der Datensatz nach der Erzeugung der Wahrscheinlichkeiten mit dem KNN beziehungsweise dem Clustering vorliegen sollen. Hierzu werden die einzelnen Attribute in ihre jeweiligen Ausprägungen aufgeteilt. So wird beispielsweise aus einem Attribut2 mit 3 Ausprägungen Attribut2(Ausprägung=0), Attribut2(Ausprägung=1) und Attribut2(Ausprägung=2). Die erzeugten Wahrscheinlichkeiten für die gelöschten Werte sind in der obigen Abbildung rot eingefärbt.

Im nächsten Abschnitt soll näher auf die beiden Verfahren zur Erzeugung der Wahrscheinlichkeiten eingegangen werden.

3.2.1 Clustering

Eine Möglichkeit, die fehlenden Werte in Wahrscheinlichkeitsverteilungen umzuwandeln, ist hierbei das Clustering. Hintergrund dieser Methode ist es, dass davon ausgegangen wird, dass alle Instanzen sich sinnvoll in mehrere Cluster unterteilen lassen. Ziel ist es, innerhalb der Cluster möglichst wenig Abweichung und außerhalb der Cluster die Distanz zu den anderen Clustern zu maximieren. Bei großen multidimensionalen Datensätzen mit vielen Attributen bietet sich zur Vorverarbeitung eine Hauptkomponentenanalyse (PCA) an. Anschließend können die Instanzen in sinnvolle Cluster gruppiert werden, um im nächsten Schritt zur Generierung der Wahrscheinlichkeiten die Anzahl an vorkommenden Ausprägungen pro Attribut aufzusummieren und durch die Summe aller Instanzen im Datensatz zu teilen. Die berechneten Mittelwerte pro Cluster und Attribut werden anschließend für die fehlenden Werte eingesetzt, davon abhängig in welchem Cluster sich eine Instanz befindet. Dieses Verfahren ist besonders für Datensätze mit wenigen Attributen geeignet oder für Datensätze, die sich im medizinischen Kontext sinnvoll in Cluster gruppieren lassen. Hauptvorteil ist dafür eine kürzere Rechenlaufzeit.

3.2.2 k-nächste-Nachbarn Verfahren

Eine noch akkuratere Methode zur Erzeugung von Wahrscheinlichkeiten aus den gelöschten Werten ist das k-nächste-Nachbarn Verfahren. Hierbei wird durch jede Spalte des Dataframes iteriert und gleichzeitig werden die Werte in Wahrscheinlichkeiten für jede Ausprägung eines Attributs umgewandelt. Tritt ein gelöschter Wert auf, so wird im nächsten Schritt die euklidische Distanz zwischen der aktuellen Testreihe zu allen anderen Reihen im Dataframe mit Durchschnittswerten berechnet. Um die nächsten Nachbarn für die Testreihe zu erhalten, werden die Distanzen nach aufsteigender Reihenfolge sortiert. Im zweiten Schritt wird durch die k nächsten-Nachbarn im Dataframe mit den gelöschten Werten iteriert. Um den gelöschten Wert durch eine neu berechnete Wahrscheinlichkeit zu ersetzen, werden anschließend die Werte der k nächsten Nachbarn gezählt. Hierfür wird die Anzahl von gleichen Ausprägungen pro Attribut gezählt und durch die Anzahl an k-nächsten Nachbarn geteilt. Die resultierenden Wahrscheinlichkeiten für jede Ausprägung eines Attributs werden nun für den gelöschten Wert ersetzt. Um die aus dem KNN erzeugten Wahrscheinlichkeiten in das neue Dataframe einfügen zu können, müssen im nächsten Schritt zusätzlich die nicht gelöschten Daten in Wahrscheinlichkeiten dargestellt werden. Hierfür müssen die Spaltenbezeichnungen um die Ausprägungen erweitert werden.

Zusammenfassend lässt sich festhalten, dass beide Verfahren die Wahrscheinlichkeiten auf eine gleiche Weise berechnen, jedoch das KNN meist zu deutlich präziseren Ergebnissen führt durch Instanz bezogenes Clustern. Eine Einzelfallprüfung auf dem ausgewählten Datensatz ist jedoch immer vorzunehmen.

3.3 Erstellung des Entscheidungsbaumes

Nachdem die unsicheren Werte durch neu berechnete Wahrscheinlichkeiten ersetzt worden sind, kann in den nächsten Schritten das Trainieren eines Entscheidungsbaums auf probabilistischen Inputdaten vorgestellt werden, sowie das danach folgende Klassifizieren durch Multiplikation der Wahrscheinlichkeiten pro Ast. Diese mathematischen Grundlagen dienen als Basis für den PRF-Ansatz, wodurch mehrere probabilistische Entscheidungsbäume durch Bootstrapping erzeugt werden.

3.3.1 Training des Entscheidungsbaumes

Bevor das Training des Entscheidungsbaumes erfolgt, wird zunächst ein Dictionary mit den Attributen als Key und den zugehörigen Ausprägungen als Value erstellt. Dies ist notwendig, um die Zugehörigkeit von Ausprägungen zu ihren jeweiligen Attributen abzubilden. Die Trainingsfunktion benötigt als Inputwerte das Dictionary, den Trainingsdatensatz in Form eines Dataframes, den minimal berücksichtigten Informationsgewinn und die maximale Baumtiefe als Hyperparameter, sowie die y-Spalte mit den Klassenzugehörigkeiten. Um den Wurzelknoten zu erhalten, müssen im ersten Schritt des Trainings die an die Wahrscheinlichkeiten angepassten Entropien berechnet werden.

Erster Schritt der gewichteten Entropieberechnung auf Ausprägungsebene a:

$$E_a = - \sum_{y=1}^Y \frac{p_y}{p_a} * \log_2 \left(\frac{p_y}{p_a} \right)$$

Zweiter Schritt der gewichteten Entropieberechnung auf Attributebene A:

$$E_A = \sum_{a=1}^A \frac{p_a}{p_A} * E_a$$

mit:

Y : Anzahl Klassen

A : Anzahl Ausprägungen

p_y : Summe der Wahrscheinlichkeiten pro Ausprägung für Klasse y

p_a : Summe der Wahrscheinlichkeiten pro Ausprägung für alle Klassen y

p_A : Summe der Wahrscheinlichkeiten aller Ausprägungen a über alle Klassen y

Abbildung 3-2 Entropieberechnung

Abbildung 3-2 zeigt, wie die klassische Entropieberechnung an die Wahrscheinlichkeiten angepasst wird durch eine relative Gewichtung der jeweiligen Ausprägungen. Die Entropieberechnung gliedert sich in insgesamt zwei Teilschritte. Im ersten Schritt wird für jede Ausprägung die Entropie berechnet. Anschließend daran, wird im zweiten Schritt die anhand der Ausprägungen gewichtete Entropie für jedes Attribut berechnet.

Nachdem die Entropien für jedes Attribut berechnet worden sind, muss im nächsten Schritt der Informationsgewinn ermittelt werden. Dieser Schritt dient als Entscheidungsgrundlage für die Wahl des Split-Attributs beim Trainieren des probabilistischen Entscheidungsbaums.

$$I_e = \arg \max(I_{A-1} - I_A, 0)$$

mit:

I_e : Informationsgewinn durch Split in Ebene e

I_{A-1} : Informationsgehalt in der vorherigen Ebene

I_A : Informationsgehalt in der aktuellen Ebene

Abbildung 3-3 Berechnung des Informationsgewinns

Abbildung 2-3 zeigt, wie der Informationsgewinn berechnet wird. Es wird für jedes potenzielle Attribut die gewichtete Entropie über alle Ausprägungen berechnet und schließlich das Attribut A als Split-Punkt ausgewählt, welches den höchsten Informationsgehalt aufweist. Hierfür wird die Differenz aus dem Informationsgehalt der vorherigen Ebene e berechnet durch das Minimieren des Ergebnisses aus I_{A-1} und I_A . Das Attribut mit dem höchsten Informationsgewinn wird anschließend als Wurzelknoten festgelegt, sofern weder die beiden Hyperparameter der maximalen Baumtiefe, noch die Unterschreitung des minimalsten Informationsgewinns greifen. Für die weitere Untergliederung des Baumes werden die jeweiligen Ausprägungen des Attributs

mit dem höchsten I_e zur weiteren Unterteilung der Klassen verwendet. Somit kann jedes Attribut nur in jeweils in einem Pfad des Baumes auftreten.

Um das nächste Split-Attribut berechnen zu können, muss das aktuelle Dataframe zunächst aktualisiert werden.

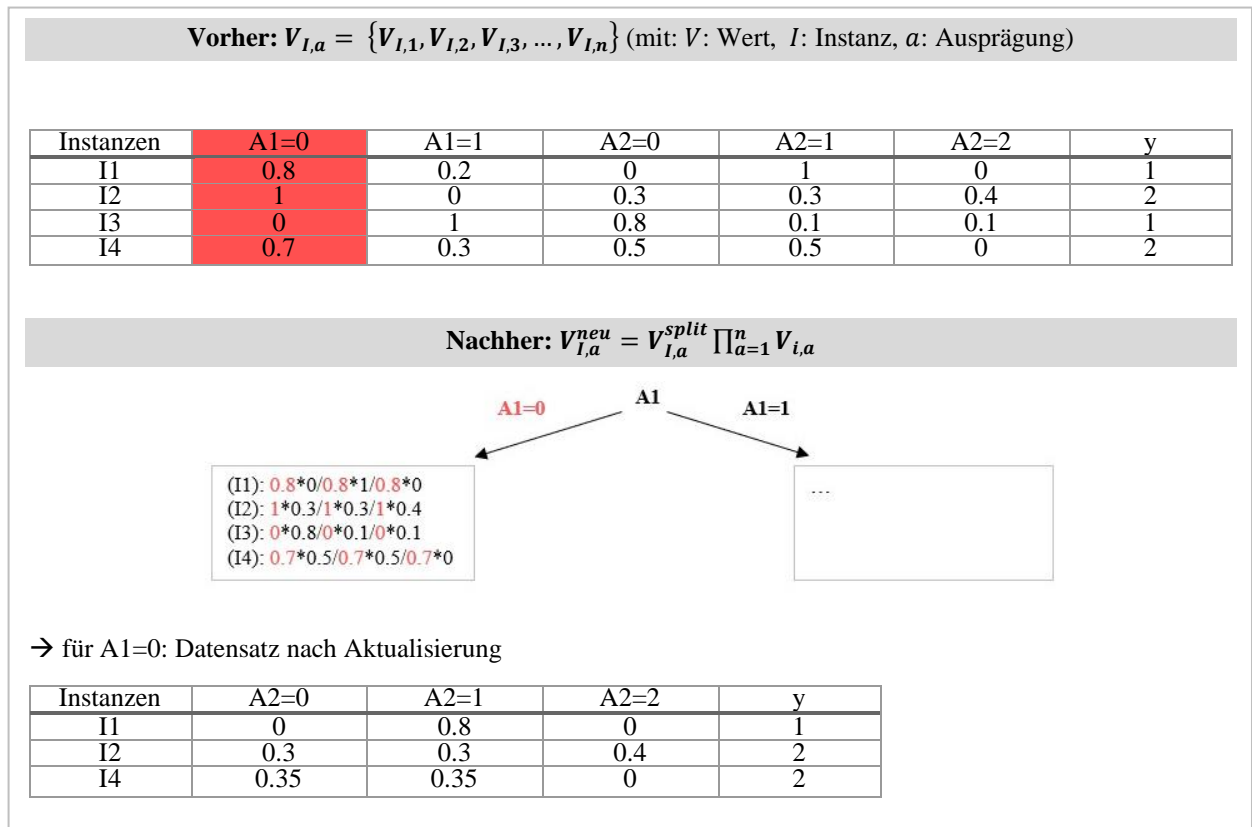


Abbildung 3-4 Aktualisierung des Dataframes

Abbildung 3-4 zeigt, welche Schritte abgeschlossen werden müssen, um das Dataframe aktualisieren zu können. Angenommen wurde, dass Attribut1 den höchsten Informationsgewinn generieren konnte und steht somit auch auf der obersten Ebene des Entscheidungsbaums. Hierbei werden zunächst alle Werte des aktuellen Astes mit allen anderen Ausprägungen von anderen Attributen multipliziert. Die neu berechneten Werte werden anschließend in einem neuen Dataframe abgespeichert. Anschließend daran, werden alle Instanzen, welche nur 0-Werte beinhalten gelöscht, da sie für den nächsten Split nicht relevant sind. Bevor das nächste Split-Attribut berechnet werden kann, wird das aktuelle Attribut mit allen dazugehörigen Ausprägungen aus dem Dataframe und dem Dictionary gelöscht. Nun kann anhand des aktualisierten Dataframes und Dictionary die Entropie für die verbleibenden Attribute berechnet werden, um für die nächste Ebene e für jede Ausprägung A das neue Split-Attribut zu berechnen. Diese Berechnungen werden so lange weitergeführt, bis ein Abbruchkriterium greift. In dieser Arbeit werden drei Abbruchbedingungen definiert. Sobald der Informationsgewinn nicht mehr stark genug anwächst, die maximale Baumtiefe erreicht ist oder alle Attribute abgearbeitet worden sind, bricht der Algorithmus ab. Sobald das Ende eines Pfades erreicht ist, folgt die Bestimmung der Klassenwahrscheinlichkeiten. Hier liegt ein weiterer Unterschied zum klassischen Entschei-

Entscheidungsbaum, welcher am Ende jeden Pfades nur eine Klasse nennt. Für die weitere Klassifizierung auf Wahrscheinlichkeitsdaten werden jedoch alle Klassen berücksichtigt, welche nach dem letzten Split noch vorhanden sind, durch Gewichtung der relativen Häufigkeit. Die Baumstruktur wird anschließend in einem verschachtelten Array abgespeichert.

3.3.2 Klassifikation mit dem Entscheidungsbaum

Nachdem der Entscheidungsbaum mit den Trainingsdaten trainiert worden ist, erfolgt die Klassifikation von Daten. Hierfür werden alle Wahrscheinlichkeiten einer Instanz entlang den Pfaden des trainierten Baums multipliziert. Das Verfahren der Klassifizierung basiert auf dem Artikel von Hristova (2014). Die folgende Abbildung visualisiert die einzelnen Schritte der Klassifikation.

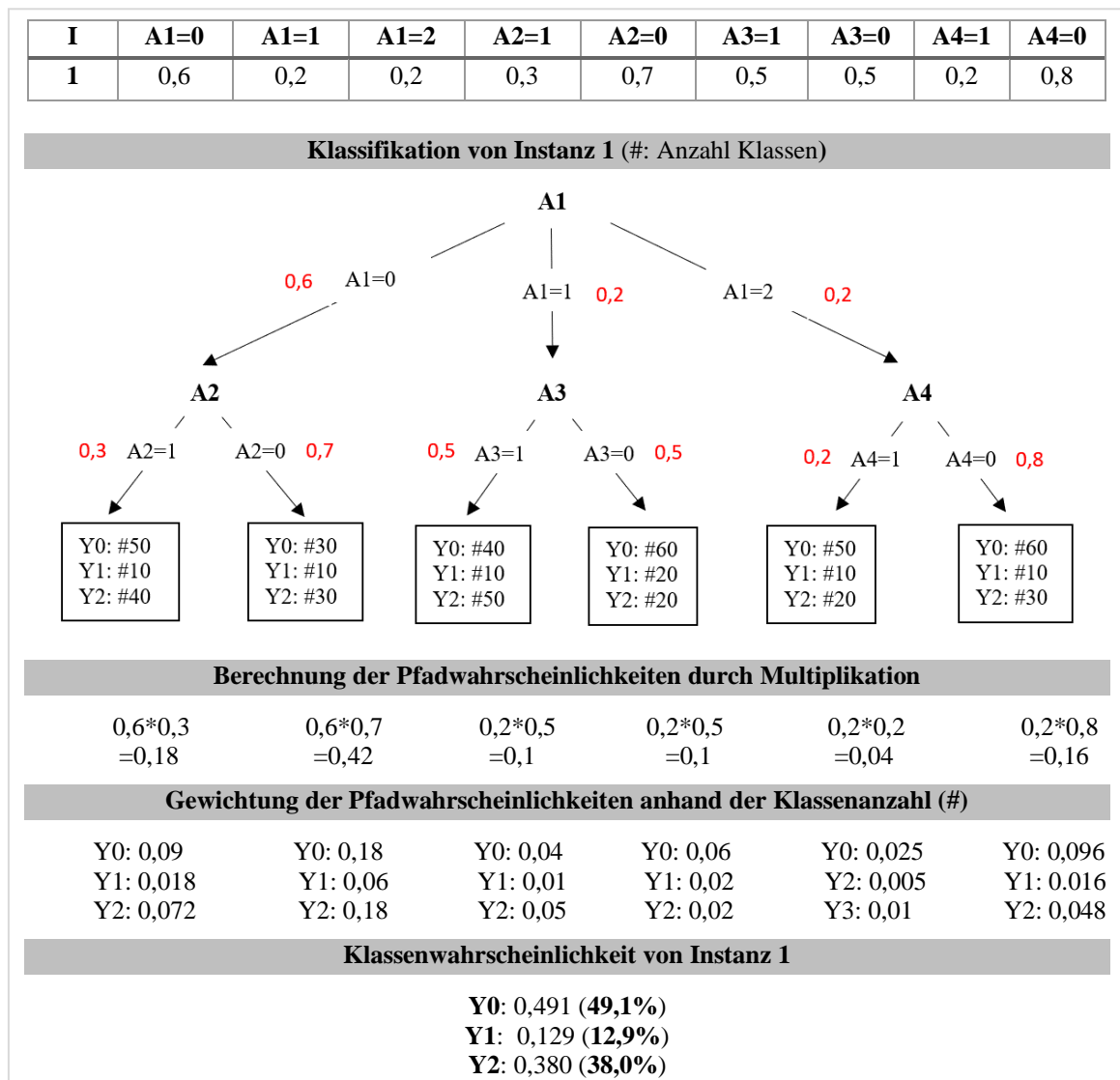


Abbildung 3-5 Klassifikation einer Instanz

Quelle: in Anlehnung an Hristova (2014, S. 10)

Abbildung 3-5 stellt die Klassifizierung anhand einer beispielhaften Instanz dar. Hierbei werden alle Ausprägungen der Instanz im Baum durchlaufen. Die Werte auf jedem Pfad werden

im nächsten Schritt miteinander multipliziert. Anschließend werden die berechneten Pfadwahrscheinlichkeiten anhand der Anzahl der vorkommenden Zielklassen gewichtet. Im letzten Schritt werden die gewichteten Pfadwahrscheinlichkeiten jeder Klasse aufsummiert. Das Klassifikationsverfahren liefert so am Ende für jede Instanz statt einer einzigen Klassenzuordnung für jede Instanz Wahrscheinlichkeiten für alle vorkommenden Klassen.

3.4 Erstellung des probabilistischen Random Forest

Im Anschluss an das Trainieren und Klassifizieren eines einzelnen Entscheidungsbaumes werden die oben genannten Ansätze in einem probabilistischen Random Forest gebündelt. Der Einsatz eines Random Forest Ansatzes gehört zu den Ensemble-Learning Methoden und bringt viele wichtige Vorteile für eine präzisere Klassifikation gegenüber einfachen Entscheidungsbäumen mit sich. Zum einen kann so gewährleistet werden, dass die Klassifikation robust gegenüber stark korrelierten Features ist, zum anderen ermöglicht ein Random Forest durch das Trainieren mehrere Bäume auf Basis unterschiedlicher Features und Instanzen, dass durch die neuen Split-Attribute auch andere Wahrscheinlichkeitspaare, sowohl beim Trainieren, als auch beim Klassifizieren, berücksichtigt werden. Diese wichtigen Aspekte werden in dieser PRF-Methode gepaart mit dem Grundverständnis von unsicheren Daten und Wahrscheinlichkeitsberechnungen. In den Abschnitten werden die einzelnen Schritte des PRF detailliert beschrieben und zusätzlich wird auf die Unterschiede zu einem klassischen Random Forest Ansatz auf vollständigen Daten eingegangen.

3.4.1 Bootstrapping

Ein wichtiges Kernelement beim Erstellen eines Random Forests ist hierbei das Bootstrapping. Hierbei handelt es sich um eine Methode der Stichprobenziehung, bei der zufällig eine bestimmte Anzahl an Instanzen und Features aus einem Datensatz gezogen werden und dem Random Forest Algorithmus übergeben werden, um darauf basierend einen Entscheidungsbaum zu erstellen. Dieser Schritt wird n-mal wiederholt, was zur Folge hat, dass bei jedem Bootstrapping immer leicht unterschiedliche Entscheidungsbäume entstehen. Grundlage für das Bootstrapping ist ein vorangehender Train-Test-Split der Daten. Die Trainingsdaten werden anschließend an den Bootstrapping Algorithmus übergeben. Dieser löscht für jede Iteration eine zufällige Anzahl an Attributen und übergibt dem Random Forest für die nächsten Schritte der Entscheidungsbaumerstellung und Auswahl sowohl einen Trainingsdatensatz als auch einen Validierungsdatensatz. Wichtige Hyperparameter sind hierbei die Dropoutrate (Prozentualer Anteil an gelöschten Features), sowie die Größe des Validierungsdatensatzes. Der wichtigste Unterschied zum klassischen Bootstrapping liegt in der Auswahl der Features. Ein klassisches Bootstrapping wählt zufällig Spalten aus dem Datensatz aus, berücksichtigt hierbei allerdings keine Zusammenhänge zu anderen Ausprägungen, welche sowohl für die Klassifikation als auch für die spätere Interpretation von elementarer Bedeutung ist. Bei dem PRF werden nicht zufällig Spalten gelöscht, sondern es wird eine zufällige Anzahl an n Attributen, inklusive den dazugehörigen Ausprägungen gelöscht. So wird verhindert, dass in einem Datensatz bestehend aus Attribut A1 mit den Ausprägungen (A1=0, A1=1, A1=2) nur Ausprägung A1=0 gelöscht wird. Im Falle, dass A1 gelöscht werden soll, werden alle dazugehörigen Ausprägungen (A1=0, A1=1, A1=2)

mitgelöscht. Dies ermöglicht später ein besseres Interpretieren des Entscheidungsbaums, was im medizinischen Kontext von elementarer Bedeutung ist.

3.4.2 Training

Das Trainieren des PRF stellt den Kern der Methode dar. Ziel ist es, durch variieren der Instanzen, beziehungsweise Attributen unterschiedliche Entscheidungsbäume zu generieren und anschließend jeden Baum auf Basis der Validierungsdaten zu evaluieren, um nur die n besten Bäume für die spätere Klassifizierung zu verwenden.

Algorithm 1: Probabilistic Random Forest - Fit

Function *RandomForestFit* (*columnDictionary*, *dfTrain*, *nTrees*, *nTreesFinal*, *minInfoGain*, *maxDepth*, *validationSize*, *nFeaturesAfterDrop*)

```

    fittedTreesDict = [ ]
    foreach Tree in range (0, nTrees) do
        //for every tree do a bootstrapping of rows and columns,
        //split given train dataframe (df) into train and validation
        //sample
        dfBtTrain, dfBtValidation, columnDictBt = bootstrapping
        (columnDictionary, dfTrain, validationSize, nFeaturesAfterDrop)
        //train every tree based on the bootstrapping
        trainedTree = trainTree (columnDictBt, dfBtTrain, minInfoGain,
        maxDepth)
        //make a prediction for every instance based on the trained
        //tree
        predictionTrain = predict (trainedTree, dfBtTrain)
        predictionTest = predict (trainedTree, dfBtValidation)
        //calculate mean squared error for the given tree for train
        //and validation data
        validationMse = meanSquaredErrorTree (dfBtValidation,
        prediction_test)
        fittedTreesDict.update({ trainedTree: validationMse })
    end
    //Finding n = nTreesFinal number of Trees with smallest MSE on
    //validation dataset
    finalTreesArray = nFittedTrees (nTreesFinal, fittedTreesDict)
    return finalTreesArray
end

```

Abbildung 3-6 Pseudocode Fit Methode

Quelle: eigene Darstellung

Abbildung 3-6 zeigt einen vereinfachten Aufbau des PRF – Fit Algorithmus als Pseudocode. Als Return der Funktion werden die $nTreesFinal$ Anzahl an trainierten Bäumen mit dem niedrigsten Mean Squared Error ausgewählt und der Reihe nach in einem Array gespeichert. Neben den im Bootstrapping genannten Hyperparametern kann zusätzlich die maximale Tiefe des Baumes, sowie der geringste zugelassene Informationsgewinn nach einem Split als Hyperparameter zur Vermeidung von Overfitting übergeben werden. Column Dictionary beinhaltet die Verbindung zwischen Attribut und Ausprägung und *dfTrain* ist der Datensatz nach dem Train-Test-Split. Zusammengefasst beinhaltet der Algorithmus alle vorher genannten Schritte des Trainierens und Klassifizieren eines Entscheidungsbaumes auf Basis des Bootstrapping.

3.4.3 Klassifikation

Nachdem die der PRF – Fit Algorithmus die bestmöglichen Bäume in trainierter Form zurückgibt, übernimmt der Classifier-Algorithmus die Aufgabe der finalen Vorhersage einer Klasse in Form von einem Mittelwert der Wahrscheinlichkeitswerten der einzelnen Bäume.

Algorithm 2: Probabilistic Random Forest – Classifier (for 3 Classes)

```

Function RandomForestClassifier (finalTreesArray, dfTest, columnDictionary)

    singleTreePrediction = [ ]
    PrfPrediction = [ ]

    foreach tree in finalTreesArray do
        predictionTree = predict (tree, dfTest)
        singleTreePrediction.append(predictionTree)
    end

    foreach yPredDf in range (0, len(dfTest)) do
        y0 = 0
        y1 = 0
        y2 = 0

        foreach yPredTree in range (0, len(singleTreePrediction)) do
            //iterate for one instance through every probability
            //prediction from the given trees and calculate the mean
            y0 += singleTreePrediction [yPredDf][yPredTree][0][0]
            y1 += singleTreePrediction [yPredDf][yPredTree][0][1]
            y2 += singleTreePrediction [yPredDf][yPredTree][0][2]
        end

        y0PredMean = y0/len(singleTreePrediction)
        y1PredMean = y1/len(singleTreePrediction)
        y2PredMean = y2/len(singleTreePrediction)
        PrfPrediction.append([y0PredMean, y1PredMean, y2PredMean])
    end

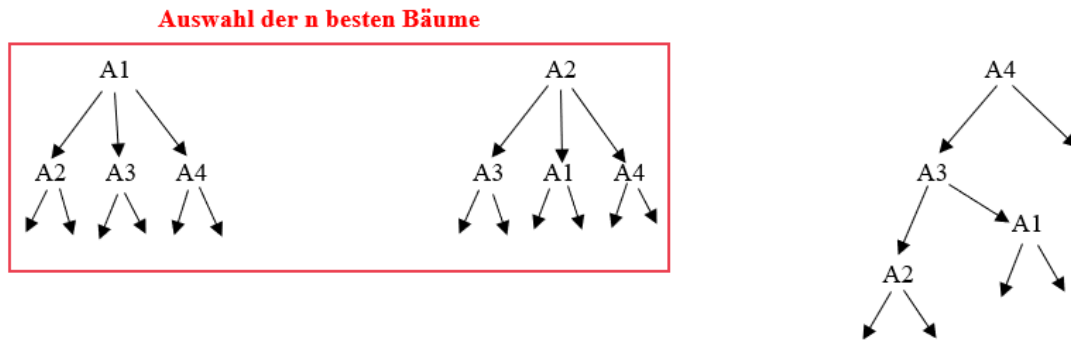
    //final Random Forest prediction after calculating the mean of
    //every single class probability of the given best n number of trees
    //from the fit function

    return PrfPrediction
end

```

Abbildung 3-7 Pseudocode PRF Classifier

In Abbildung 3-7 wird dieser Algorithmus vereinfacht am Beispiel von einer Multi-Label Klassifikation mit drei Klassen veranschaulicht. Hierfür dient als Input das Array mit den trainierten Bäumen. Es wird für jeden dieser Bäume eine Klassifizierung durchgeführt und anschließend die Klassenwahrscheinlichkeiten jeder Instanz in einem Array zwischengespeichert. Nachdem die Klassifizierung für jeden Baum auf Basis der Testdaten durchgeführt worden ist, werden für eine Instanz alle Klassenwahrscheinlichkeiten von jedem Entscheidungsbaum aufsummiert und anschließend der Mittelwert für jede Klassenwahrscheinlichkeit berechnet. Als finale Vorhersage wird für jede Instanz somit ein einzelnes Array zurückgegeben, bestehend aus den Klassenwahrscheinlichkeiten, welche sich zu 100 Prozent aufsummieren lassen.



Berechnung der Klassenwahrscheinlichkeit und Auswahl der n besten Bäume			
P:	[0,2 0,8 0,0]	[0,1 0,7 0,2]	[0,3 0,5 0,2]
MSE:	0,15	0,2	0,35
Berechnung der Klassenwahrscheinlichkeit anhand der n besten Bäume			
	[0,15 0,75 0,1]		

Abbildung 3-8 Klassifizierung mit Random Forest

Abbildung 3-8 zeigt den Ablauf der Klassifizierung mittels des Random Forest Verfahrens nochmals detaillierter. Anfangs erzeugt der Algorithmus mehrere unterschiedliche Entscheidungsbäume. Aus diesen werden anschließend anhand des MSE (Mean Squared Error) der Validierungsdaten die n besten Bäume ausgewählt. In diesem konkreten Beispiel werden aus drei Entscheidungsbäumen zwei davon aufgrund des geringeren MSE zur Klassifizierung verwendet. Die Klassenwahrscheinlichkeit für eine bestimmte Klasse kann wie folgt berechnet werden:

$$P_{y_i} = \frac{1}{N} * \sum_{n=1}^N y_{i,n}$$

Wobei P_{y_i} die Wahrscheinlichkeit für eine Klasse y_i darstellt, N die Anzahl der Bäume und $y_{i,n}$ die Wahrscheinlichkeit eines Entscheidungsbaumes n für eine Klasse y_i . Zusammengefasst werden die Mittelwerte über alle Klassenwahrscheinlichkeiten der n besten Entscheidungsbäume gebildet.

4 Evaluation der Methodik

Das folgende Kapitel dient der Evaluation des probabilistischen Entscheidungsbaumes und PRF-Ansatzes. Als Evaluationsmaße werden nachfolgend die Accuracy und der MSE verwendet. Bei der Berechnung der Accuracy wird dabei diejenige Klasse mit der höchsten Wahrscheinlichkeit als Klassenzuordnung angenommen. Die insgesamt Methodik der Evaluation besteht daraus, zufällig 0, 10, 20, 30, 40, 50, 70 und 90 Prozent der Daten zu löschen, die Daten gemäß der in Kapitel 3 beschriebenen Methode zu verarbeiten und die Ergebnisse anschließend anhand der beiden Gütemaße zu vergleichen. Für beide Datensätze ist eine Train-Test-Split Aufteilung in 70-30 vorgenommen worden. Damit die Performance des Verfahrens besser bewertet werden kann, werden drei weitere Machine-Learning Ansätze verwendet, welche aber im Gegensatz zu diesem Verfahren auf vollständigen Datensätzen angewendet werden. Verwendet worden sind hierfür der klassische Entscheidungsbaum und das Random Forest mit default-Werten aus der Sklearn-Bibliothek. Zusätzlich werden die Verfahren durch ein Neuronales Netz mit zwei Hidden-Layern ergänzt. Diese drei Verfahren dienen als Baseline, wie die Ergebnisse bei vollständigen Daten aussehen würden.

In diesem Kapitel werden nachfolgend zwei medizinische Datensätze evaluiert. Der erste Datensatz besteht aus binären und kategorialen Daten mit binär verteilten Klassen zur Diagnose einer Herzerkrankung (Pytlak 2021). Zur besseren Evaluation ist der Datensatz zufällig von 253.000 auf 10.000 Instanzen gekürzt worden und zusätzlich ist eine Harmonisierung der Klassenhäufigkeiten auf eine gleiche Aufteilung vorgenommen worden. Somit handelt es sich um einen mittelgroßen Datensatz. Dieser Schritt dient dem Zweck, dass dadurch bei einer zufälligen Vorhersage eine Wahrscheinlichkeit von 50 Prozent daraus resultiert. Neben den 10.000 Instanzen umfasst der Datensatz 17 Attribute. Bei der Definition der Parameter für den probabilistischen Entscheidungsbaum und PRF ist die maximale Baumtiefe auf 6, der maximale Informationsgewinn auf 0,03 und eine Dropout-Rate von 30 Prozent festgelegt worden. Diese Einstellungen haben bei einer iterativen GridSearch die besten Ergebnisse erzielt.

```
1 key_min,info_min = info_min_attribute(column_dict,df,"HeartDisease")
2 print("The Split-Attribute: ",key_min," leads to the smallest entropy of: ",info_min)

The Split-Attribute:  AgeCategory_cat  leads to the smallest entropy of:  0.8564048514381923
```

Abbildung 4-1 Bestimmung des nächsten Split-Attributes

Abbildung 4-1 zeigt dabei, wie das Attribut für den Wurzelknoten im Herzkrankheiten-Datensatz bestimmt wird. Durch das Aufrufen der `info_min_attribute` Funktion wird mithilfe der Entropieberechnung das Attribut bestimmt, dass die kleinste Entropie besitzt. In diesem konkreten Fall weist das Attribut „Age“ die geringste auf und ist somit der Wurzelknoten des Baumes. Im nächsten Schritt werden die Ausprägungen des Wurzelattributs als Entscheidungsregeln festgelegt. Die nächsten Berechnungen ergeben, dass das nächste Attribut bei der ersten Ausprägung von „Age“ das Attribut „GenHealth“ ist.

```

[['AgeCategory_cat_0',
  [['GenHealth_Excellent',
    [['SleepTime_cat_0', [[0.8601398601398601, 0.13986013986013987, 0.0]]],
    ['SleepTime_cat_1', [[0.9070175438596492, 0.09298245614035087, 0.0]]],
    ['SleepTime_cat_2', [[0.8789954337899544, 0.12100456621004566, 0.0]]],
    ['SleepTime_cat_3', [[0.8108108108108109, 0.1891891891891892, 0.0]]],
    ['SleepTime_cat_4',
      [[0.9655172413793104, 0.034482758620689655, 0.0]]]]],
  ['GenHealth_Fair',
    [['Race_American Indian/Alaskan Native', [[0.375, 0.625, 0.0]]],
    ['Race_Asian', [[0.725, 0.275, 0.0]]],
    ['Race_Black', [[0.5252525252525253, 0.47474747474747475, 0.0]]],
    ['Race_Hispanic', [[0.7142857142857143, 0.2857142857142857, 0.0]]],
    ['Race_Other', [[0.6379310344827587, 0.3620689655172414, 0.0]]],
    ['Race_White', [[0.6028571428571429, 0.39714285714285713, 0.0]]]]],
  ['GenHealth_Good',
    [['Stroke_No', [[0.7926078028747433, 0.20739219712525667, 0.0]]],
    ['Stroke_Yes', [[0.4339622641509434, 0.5660377358490566, 0.0]]]]],
  ['GenHealth_Poor',
    [['PhysicalHealth_cat_0',
      [[0.6274509803921569, 0.37254901960784315, 0.0]]],
    ['PhysicalHealth_cat_1',
      [[0.5862068965517241, 0.41379310344827586, 0.0]]],
    ['PhysicalHealth_cat_2',
      [[0.4936708860759494, 0.5063291139240507, 0.0]]],
    ['PhysicalHealth_cat_3', [[0.64, 0.36, 0.0]]]]],
  ['GenHealth_Very good',
    [['DiffWalking_No', [[0.8568872987477638, 0.14311270125223613, 0.0]]],
    ['DiffWalking_Yes',
      [[0.7554347826086957, 0.24456521739130435, 0.0]]]]]]],
['AgeCategory_cat_1',

```

Abbildung 4-2 Aufbau trainierter Entscheidungsbaum (Ausschnitt)

Abbildung 4-2 zeigt, wie die in Abbildung 4-1 beschriebenen Ergebnisse der Entropieberechnung konkret in einem trainierten Baum dargestellt werden. Bei dem Python-Output handelt es sich lediglich um einen beispielhaften Ausschnitt aus der linken Seite des Baumes. Der Entscheidungsbaum wird dabei je Ebene in einer eigenen Verschachtelung wie in der obigen Abbildung dargestellt. So befindet sich beispielsweise das Attribut „Age“ in Ebene 0, das Attribut „GenHealth“ in Ebene 1 und die Attribute „SleepTime“, „Race“, „Stroke“, „PhysicalHealth“ und „DiffWalking“ in Ebene 2. Wenn eine der Abbruchbedingungen erfüllt ist, werden die Wahrscheinlichkeit für jede Klasse am Ende angehängt.

```

[[[0.7648757730050258, 0.23512422699497404, 0.0]],
 [[0.8392554991539763, 0.16074450084602368, 0.0]],
 [[0.302247191011236, 0.697752808988764, 0.0]],
 [[0.5820092455732377, 0.41799075442676215, 0.0]],
 [[0.6774566473988439, 0.3225433526011561, 0.0]],
 [[0.49661181026137463, 0.5033881897386253, 0.0]],
 [[0.13584439641864773, 0.8641556035813523, 0.0]]]

```

Abbildung 4-3 Klassifikation der ersten 7 Instanzen

Abbildung 4-3 zeigt das Ergebnis aus der Klassifikation der ersten sieben Instanzen. Demnach hat Instanz beziehungsweise Person 1 mit einer Wahrscheinlichkeit von 76,49 Prozent eine Herzerkrankung und mit 23,51 Prozent keine.

Bei Anwendung des PRF-Verfahrens sind 20 Bäume trainiert worden und davon die sechs Entscheidungsbäume mit dem niedrigsten MSE zur Klassifikation der Instanzen verwendet worden.

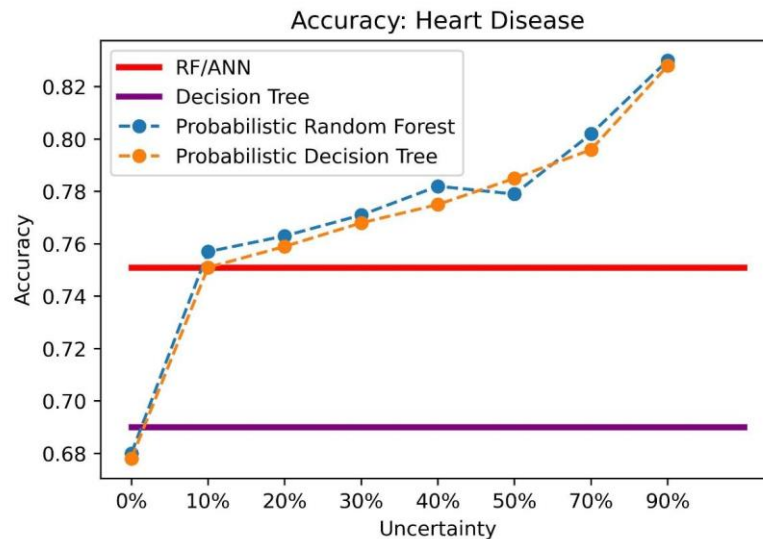


Abbildung 4-4 Accuracy von Datensatz 1

Abbildung 4-4 zeigt die Ergebnisse der Accuracy des ersten Herzkrankheiten-Datensatzes. Die violette Baseline stellt hierbei die Accuracy eines unangepassten Entscheidungsbaumes und die rote Baseline die eines Random Forests und einem Neuronalem Netz mit zwei Hidden-Layer auf vollständigen Daten. Die orangene Linie zeigt die Ergebnisse des in dieser Arbeit entwickelten probabilistischen Entscheidungsbaumes und die blaue Linie des PRF-Ansatzes. Bei Betrachtung der Abbildung ist auffällig, dass die angepasste Methodik bei 0 Prozent gelöschten Daten eine deutlich schlechtere Accuracy im Vergleich zu den Baselines erzielt. Auf gelöschten Daten steigt die Accuracy dieser Methodik mit der Anzahl an gelöschten Werten jedoch in fast allen Fällen stetig an. Der PRF ist zudem fast immer besser als der probabilistische Entscheidungsbaum. Diese Tatsache ist mit der Klassifizierung anhand der besten n Bäume zu begründen.

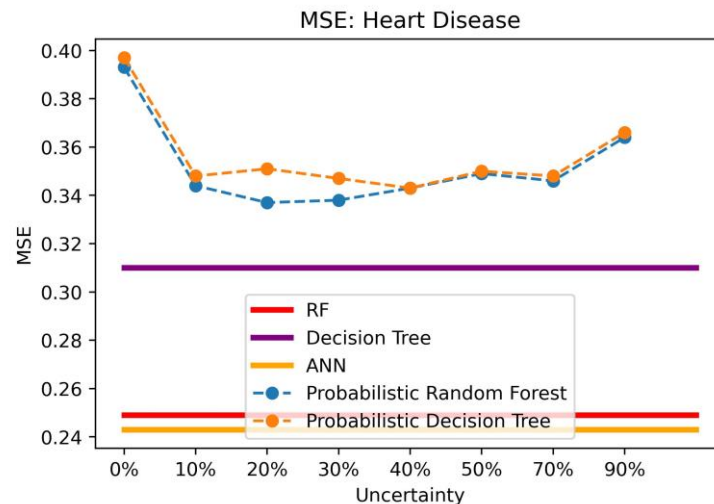


Abbildung 4-5 Mean Squared Error von Datensatz 1

Wird nun der MSE in Abbildung 4-5 betrachtet, so ist auffällig, dass dieser mit der Anzahl an gelöschten Werten im Vergleich zu ungelöschten Daten sinkt. Ab einem hohen Anteil gelöschter Werte nimmt der MSE jedoch wieder zu. Auch hier fällt erneut auf, dass der probabilistische Random Forest insgesamt einen geringeren MSE aufweist. Im Vergleich zu den Baseline-Verfahren auf vollständigen Daten ist jedoch zu erkennen, dass beide Ansätze einen deutlich höheren MSE aufweisen. Mögliche Interpretationen der Ergebnisse werden im nächsten Kapitel behandelt.

Die Ergebnisse wurden ergänzt durch eine weitere Auswertung mit Fokus auf das Verhalten der Baseline Methoden (siehe Anhang A. 0). Hierfür wurde ein weiterer Vorverarbeitungsschritt an die Ergebnisse des KNN-Algorithmus angereicht, welcher jeweils die Ausprägungen als sicher gewertet hat, welche die höchste Wahrscheinlichkeit nach dem KNN-Verfahren hatte. Anschließend wurden auf dieser Datenbasis die transferierten kategorialen Daten verwendet, um die drei Baseline Methoden zu trainieren und zu evaluieren gemäß des MSE und der Accuracy. Die Resultate verhalten sich sehr ähnlich zu denen auf vollständigen Daten mit nur minimalen Schwankungen. Diese Ergebnisse unterstreichen die Annahme, dass das KNN-Verfahren sehr gut geeignet ist für diesen Datensatz, um die unsicheren Daten wiederherzustellen.

Der zweite Datensatz ist deutlich kleiner und besteht, wie der vorherige aus kategorialen/binären Daten mit binär verteilten Klassen zur Diagnose von Brustkrebs (Maji 2020). Aufgrund des geringeren Umfangs ist dieser Datensatz weder gekürzt noch bezüglich der Klassen harmonisiert worden. Insgesamt besteht er aus 681 Instanzen, welche sich in 9 Attribute unterscheiden lassen. Bei der Definition der Parameter für den Entscheidungsbaum und des PRFs ist die maximale Baumtiefe auf 7, der maximale Informationsgewinn auf 0,03 und eine Dropout-Rate von 0,125 festgelegt worden. Auch hier sind aus 20 Bäumen die besten 30 Prozent für die Klassifizierung verwendet worden.

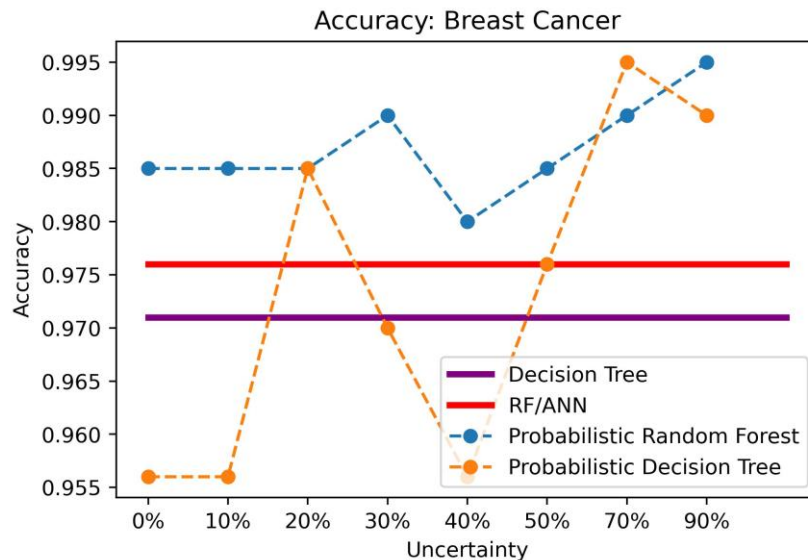


Abbildung 4-6 Accuracy von Datensatz 2

Abbildung 4-6 zeigt die Ergebnisse der Accuracy Berechnungen für den Brustkrebs-Datensatz. Hier fällt sofort ins Auge, dass der probabilistische Entscheidungsbaum eine deutlich schlechtere Accuracy als der probabilistische Random Forest besitzt. Wird allerdings nur der probabilistische Random Forest betrachtet, so lässt sich auch hier erneut der Trend erkennen, dass die Accuracy besser wird, je mehr Unsicherheit in den Daten vorliegt. Insgesamt sind auch hier die Ergebnisse ähnliche zur Baseline.

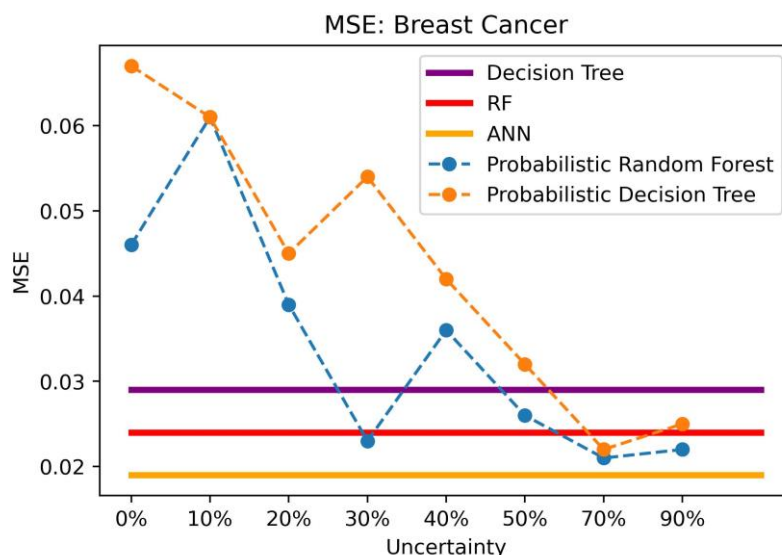


Abbildung 4-7 Mean Squared Error von Datensatz 2

Wird nun in Abbildung 4-7 der MSE des zweiten Datensatzes betrachtet, so fällt auf, dass bei dieser Datengrundlage der MSE deutlich geringer als beim ersten Datensatz ist. Insgesamt nimmt aber auch hier der MSE bei bis zu 70 Prozent gelöschten Werten deutlich ab, bevor er anschließend steigt und sich ähnlich zur Baseline verhält.

Zusammenfassend für beide Datensätze lässt sich also sagen, dass die Accuracy mit Anzahl der gelöschten Daten zunimmt, während der MSE bei einem höheren Anteil an gelöschten Werten abnimmt, jedoch in beiden Datensätze deutlich schlechter ist als die Baselines. Die beste Accuracy liefert der entwickelte Algorithmus gemäß der beiden evaluierten Datensätze bei 30 bis 70 Prozent gelöschten Daten. Insgesamt liefert der probabilistische Random Forest bei beiden Evaluationsmetriken bessere Ergebnisse. Detailliertere Confusion-Matrizen der Ergebnisse bei 20, 40, 70 und 90 Prozent gelöschten Daten befinden sich in Anhang A.1 bis A.4.

5 Diskussion der Ergebnisse

Nachdem in den vorherigen Kapiteln sowohl die Relevanz der Methode als auch der mathematische Aufbau und die Evaluation der Algorithmen thematisiert worden sind, dient Kapitel fünf der Diskussion der Ergebnisse.

Unsichere Inputdaten sind in medizinischen Daten keine Seltenheit. Umso wichtiger ist es, diese Wahrscheinlichkeiten für eine präzise Klassifizierung zu berücksichtigen. Das einfache Einfügen der Daten im Standard-KNN Algorithmus führt zu einem starken Informationsverlust und ist besonders bei kategorialen Daten nicht sinnvoll. Je unsicherer die Inputdaten sind, desto wichtiger ist es auch die Wahrscheinlichkeiten der anderen Ausprägungen zu berücksichtigen, da dies sonst zu falschen Klassifikationen führen kann. Die hierfür entwickelte PRF-Methode baut genau auf dieser Grundlage auf. Durch die gewichtete Entropieberechnung können alle Wahrscheinlichkeiten eines Attributs berücksichtigt werden. Somit kann das Verfahren besonders gut mit sehr unsicheren Daten arbeiten und erzielt gute Ergebnisse. Aus der Evaluation lassen sich drei Kernelemente identifizieren. Beim Blick auf die Accuracy und den MSE bei keinerlei Unsicherheit lässt sich erkennen, dass diese zu Beginn deutlich schlechter als die Baselines sind. Das liegt daran, dass das in dieser Arbeit entwickelte Verfahren keine numerischen Daten verarbeiten kann, jedoch beide Datensätze mehrere numerische Daten beinhalten. Diese müssen zuerst in Intervalle und Kategorien transferiert werden. Dieser Schritt ist mit einem Informationsverlust verbunden, wodurch sich die anfänglich schlechte Performance rechtfertigen lässt, da bei den Baseline-Verfahren keine Umwandlung von numerischen Daten stattgefunden hat.

Die zweite Hauptthese lässt sich erkennen bei steigender Unsicherheit der Daten bis zu 90 Prozent. Hierbei wurde besonders im ersten großen Datensatz beobachtet, dass die im gesamten iterativen Prozess der steigenden Unsicherheit zwar zum einen die Accuracy zunimmt, allerdings der MSE deutlich oberhalb der Baseline liegt. Ähnliches ist auch im kleinen Breast Cancer Datensatz zu beobachten, welcher jedoch auf Grund der Größe der Daten sich ungenauer interpretieren lässt. Diese könnte der mathematischen Funktionsweise des PRF-Algorithmus geschuldet sein. Ein Hauptgrund für die steigende Accuracy liegt dabei sicherlich in der Berücksichtigung aller Wahrscheinlichkeiten einer Instanz. Insgesamt werden die Entscheidungen immer knapper und verlieren an Eindeutigkeit, allerdings kann durch das Berücksichtigen der einzelnen Wahrscheinlichkeiten einer Instanz deutlich feingranularer klassifiziert werden. Dies hat jedoch zur Folge, dass der MSE deutlich unter den unsicheren Daten leidet, weil selbst eigentlich eindeutig zu klassifizierende Klassen immer mehr Richtung 50-50 Prozent Entscheidungen konvergieren. Dies kann besonders gut an dem mittelgroßen Datensatz 1 beobachtet werden. Zusammenfassend lässt sich jedoch festhalten, dass der PRF sowohl für kleine als auch mittelgroße Datensätze besonders im Bereich der Accuracy, also dem korrekten Erkennen der Klasse sehr gut performt. Die dritte Interpretationsebene fokussiert sich auf den Schritt der Vorverarbeitung. Hierfür wurde ein speziell modifizierter KNN-Ansatz entwickelt, um die Wahrscheinlichkeiten für die fehlenden Daten zu generieren. Dadurch, dass das KNN-Verfahren die neuen Wahrscheinlichkeiten für die fehlenden Werte basierend auf den k ähnlichsten

Instanzen berechnet, kann die Wahrscheinlichkeit einer Ausprägung besonders präzise berechnet werden und gegebenenfalls Werte, die vorher eher zu einem schlechteren Klassifikationsergebnis beigetragen hätten, im Falle des Löschens mit einer Wahrscheinlichkeit versehen werden, welche zu einer besseren Einteilung von Klassen führen kann. Erst ab einem gewissen Grad von 90 Prozent gelöschten Daten nimmt die Leistungsfähigkeit des KNN-Verfahrens deutlich ab, weil der KNN-Algorithmus keine optimale Gruppierung mehr vornehmen kann. Neben dem Gesichtspunkt der guten Performance trotz fehlender Daten, ist zusätzlich zu nennen, dass die Methodik am Ende der Klassifikation einer Instanz nicht nur eine Klasse liefert, sondern für jeden einzelne Zielklasse die Wahrscheinlichkeit, mit der eine Instanz dieser angehören könnte. Da manche Klassenentscheidungen zudem knapp ausfallen können, ist die Berücksichtigung von anderen Klassenwahrscheinlichkeiten in der finalen Klassifikation besonders wichtig. So kann zusätzlich eingesehen werden, mit welcher Wahrscheinlichkeit ein Patient beispielsweise eine Krankheit doch haben könnte, obwohl die eigentliche Klassifizierung keine Krankheit klassifiziert hätte. Die Tatsache, dass ein einzelner Entscheidungsbaum Wahrscheinlichkeitsverteilungen statt einer einzigen Klasse liefert, bringt zusätzlich einen großen Vorteil für den PRF mit sich. Bei Verwendung von klassischen Entscheidungsbäumen für das Random Forest Verfahren würden lediglich die Anzahl der Klassen je Instanz gezählt werden. Das probabilistische Random Forest Verfahren bekommt allerdings mehrere Wahrscheinlichkeitsverteilungen von den probabilistischen Entscheidungsbäumen übergeben. Durch diese Funktion können zusätzlich Klassen berücksichtigt werden, die eine nicht so hohe Wahrscheinlichkeit besitzen. Knappe Entscheidungen können so deutlich besser erkannt werden.

Im Gegensatz zu den oben genannten Stärken, weist die Methode allerdings auch Limitationen auf. Inputdaten müssen in einem bestimmten Format vorliegen, um gute Resultate im probabilistischen Entscheidungsbaum zu erzielen. Die entwickelte Methodik kann zwar auch verschiedenen verteilte Merkmale wie beispielsweise numerische Attribute verarbeiten, jedoch müssen diese anfangs in kategoriale Daten umgewandelt werden. Das entwickelte probabilistische Baumverfahren sollte allerdings bestenfalls so wenig Ausprägungsunterteilungen in jedem Attribut beinhalten wie möglich. Bei der oben dargestellten Evaluation ist der Ausprägungsraum auf maximal sechs beschränkt. Da in diesem Verfahren jede Ausprägung einen Splitpunkt darstellt, würde eine zu hohe Anzahl an Ausprägungen je Attribut zu Overfitting und Ausartung des Baumes führen. Werden jedoch aus einem sehr divers numerisch verteilten Attribut nur wenige Kategorien gebildet, so geht daraus zwangsweise ein Informationsverlust einher. Wird beispielsweise der Herzerkrankungs-Datensatz betrachtet, so führt die Umwandlung der numerisch verteilten Attributen BMI und Alter zu einem gewissen Informationsverlust. Mit der angepassten Methodik geht allerdings zusätzlich auch ein Verlust der menschlichen Interpretierbarkeit mit einher. Es ist zwar anhand der Baumstruktur leicht ersichtlich, welche Attribute einen hohen Einfluss auf die Klassenzuordnung haben, weil diese sehr weit oben im Baum positioniert sind, jedoch ist die Klassifizierung einer Instanz aufgrund der Ausprägungen komplexer. Eine Instanz kann erst klassifiziert werden, nachdem vorher verschiedene Multiplikations- und Additionsberechnungen durchgeführt worden sind. Wird bei möglichen Limitationen auf die Anpassung eines anderen Datensatzes Bezug genommen, so lässt sich feststellen, dass

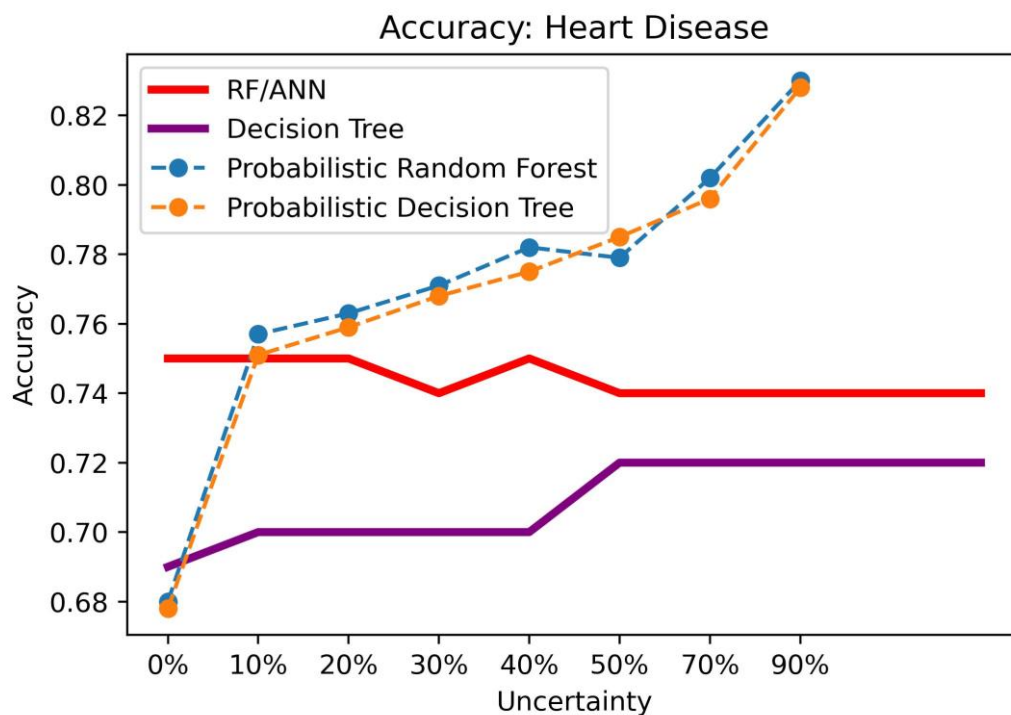
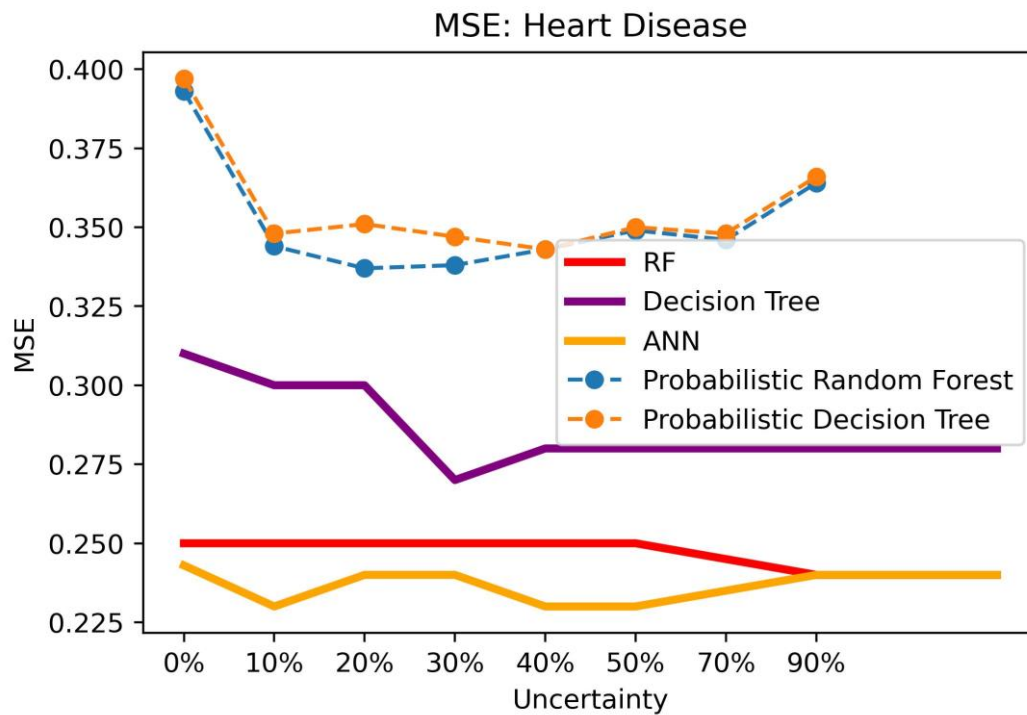
Datenquellen mit vielen Attributen in der Anpassung im Code auch dementsprechend mit steigendem Aufwand einhergeht. So muss beispielsweise das Dictionary mit den Ausprägungen und deren zugehörigen Attributen deutlich zeitaufwändiger angepasst werden. Zu viele Attribute können zudem in Entscheidungsbaumverfahren zu einer Überanpassung führen. Diese Tatsache führt zu einem weiteren Kritikpunkt: Die Rechenlaufzeit. Je nach Variation der Datensatzgröße und der Wahl der Hyperparameter steigen die Berechnungen bei der Klassifikation nahezu exponentiell an, weil das in dieser Arbeit entwickelte Verfahren für eine Instanz sämtliche Pfade des trainierten Entscheidungsbaums mit den Wahrscheinlichkeiten der Instanz multipliziert. Im Gegensatz zu einem normalen Entscheidungsbaum, welcher nur eine Pfad entlang bis zu einer Klasse folgt, können beim probabilistischen Entscheidungsbaum mehrere hundert Pfade berücksichtigt werden. Als eine letzte Limitation ist zu nennen, dass der Algorithmus nur Unsicherheit in den Inputdaten berücksichtigen kann und nicht in den Labels.

6 Zusammenfassung und Ausblick

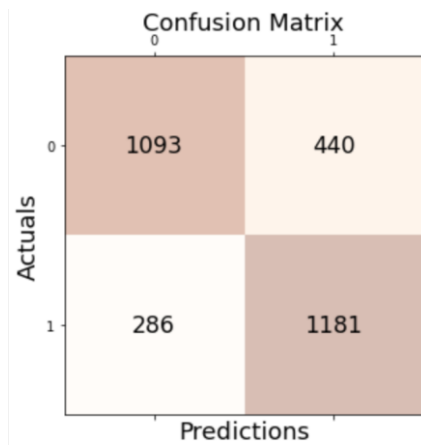
Zusammenfassend lässt sich sagen, dass in dieser Arbeit ein sehr guter Ansatz entwickelt worden ist, der unsichere Inputdaten sowohl im Entscheidungsbaumverfahren als auch im Random Forest Verfahren berücksichtigt. Die Möglichkeit der entwickelten Methodik, Wahrscheinlichkeiten anstelle von einfachen Werten übergeben zu können, bringt viele Vorteile mit sich. Am Ende der Klassifizierung einer Instanz wird nicht nur eine Klasse zurückgegeben, sondern für jede Zielklasse die Wahrscheinlichkeit, mit der die Instanz dieser Klasse angehören könnte. Somit werden in diesem Verfahren auch Klassen berücksichtigt, die eine eher kleinere Wahrscheinlichkeit besitzen. Aus diesem Grund performt der entwickelte Algorithmus auch bei sehr knappen Klassenentscheidungen deutlich besser als die zugehörigen klassischen Verfahren. Trotz der teilweise vielen unsicheren Daten liefert das Verfahren sehr gute Ergebnisse. Insgesamt muss jedoch bei Auswahl der Anzahl an unsicheren Daten ein Trade-Off zwischen Accuracy und MSE erfolgen. Dieser Trade-Off ist je nach gegebenen Inputdaten und Zielsetzung der Analyse unterschiedlich zu bewerten. Ist es wichtig, die stets die wahrscheinlichste Zielklasse richtig zu erkennen, sollte die Accuracy maximiert werden. Wird viel Wert auf eine hohe Accuracy gelegt, so sollten gemäß den evaluierten Datensätzen die Daten einen eher hohen Grad an Unsicherheit besitzen. Ist es im Gegensatz dazu besonders wichtig, auch niedrige Klassenwahrscheinlichkeiten einer Instanz möglichst akkurat abzubilden, so sollte der MSE minimiert werden. Die Evaluation der beiden Datensätze hat insgesamt ergeben, dass eine eher geringe Unsicherheit in den Inputdaten zu einem geringen MSE führt. Eine steigende Accuracy geht in dieser Arbeit mit einem steigenden MSE einher und ein sinkender MSE mit einer sinkenden Accuracy. Dennoch unterliegen der Methodik Limitationen, an denen sich zukünftige Forschungsfelder ableiten lassen. So könnte das Verfahren hinsichtlich der validen Inputattributen erweitert werden, sodass sich auch Datensätze mit beispielsweise ordinal- oder nominal skalierten Merkmalen klassifizieren lassen. Zusätzlich könnte die Methode insofern verändert werden, dass dem Verfahren statt Wahrscheinlichkeiten je Zielklasse, Wahrscheinlichkeitsdichtefunktionen übergeben werden. Das in dieser Arbeit entwickelte Verfahren kann lediglich unsichere Inputwerte bei den Attributen verarbeiten, jedoch nicht Unsicherheit in den Zielklassen. Hinsichtlich dessen Gesichtspunkt könnte die Methodik folglich erweitert werden. Da die Methodik durch die im Hintergrund laufenden Multiplikations- und Additionsberechnungen bei der Klassifizierung von Instanzen mit einem Interpretierbarkeitsverlust im Vergleich zu klassischen Entscheidungsbaumverfahren einhergeht, könnte hier als Thema für neue Forschungsfelder die Transparenz für den Nutzer erhöht werden. In dieser Arbeit ist lediglich mit medizinischen Datensätzen gearbeitet worden. Aus diesem Grund könnte das entwickelte Verfahren zusätzlich auf andere Arten von Daten, wie beispielsweise Maschinendatensätzen angewandt werden. Hier könnte ein Vergleich von vielen heterogenen Datensätzen verschiedener Themenbereiche Rückschlüsse über die Eignung des Verfahrens für bestimmte Daten geben.

Anhang A: Weitere Unterlagen und Ausführungen

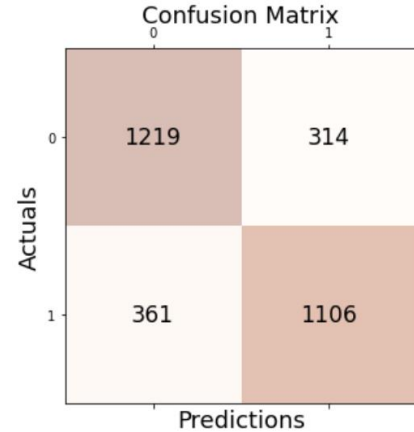
A. 0 Heart-Disease: Ergänzende Auswertung



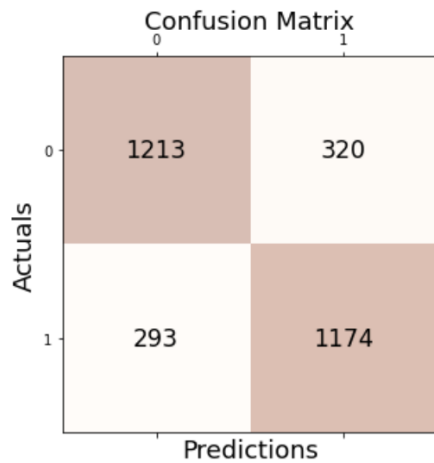
A.1 Heart-Disease: Confusion Matrix – Entscheidungsbaumverfahren

20% gelöschte Daten

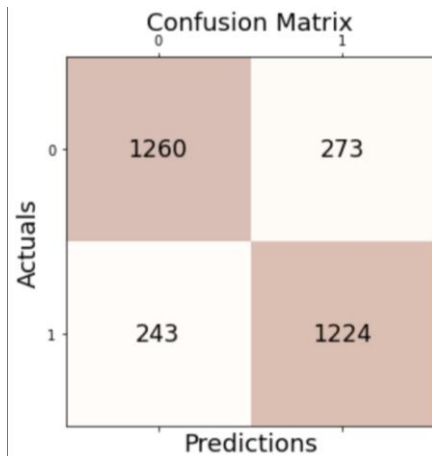
Recall: 0.805
Accuracy: 0.758
F1 Score: 0.765
MSE: 0.35145454692354244

40% gelöschte Daten

Recall: 0.754
Accuracy: 0.775
F1 Score: 0.766
MSE: 0.3434310393001963

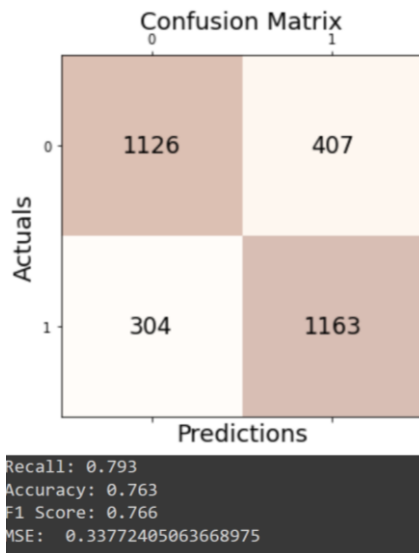
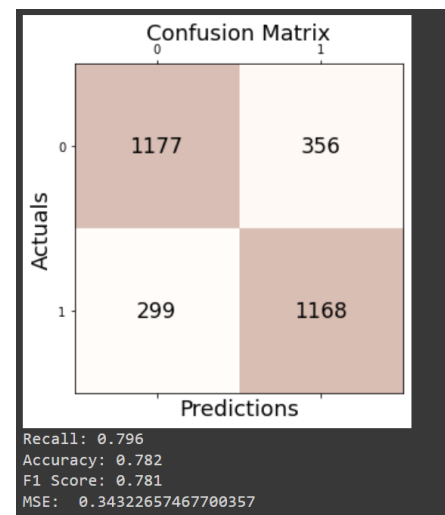
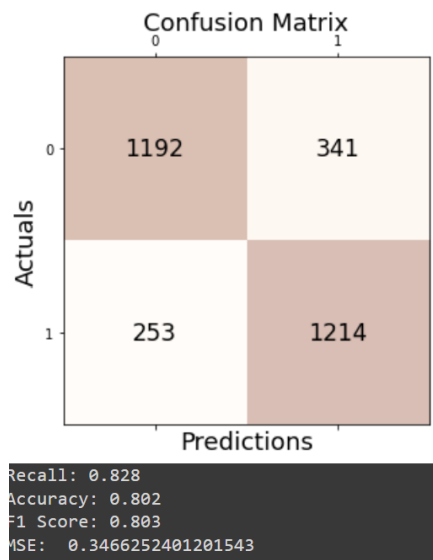
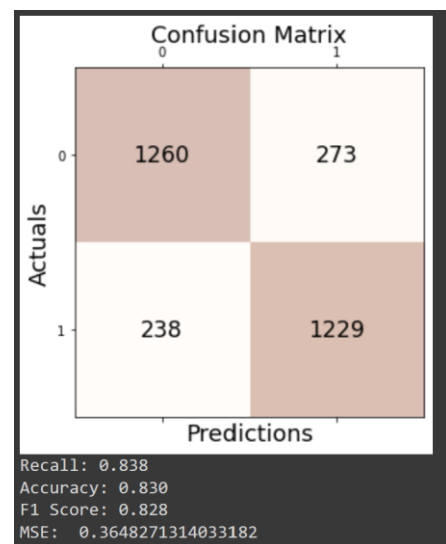
70% gelöschte Daten

Recall: 0.800
Accuracy: 0.796
F1 Score: 0.793
MSE: 0.3488931665134948

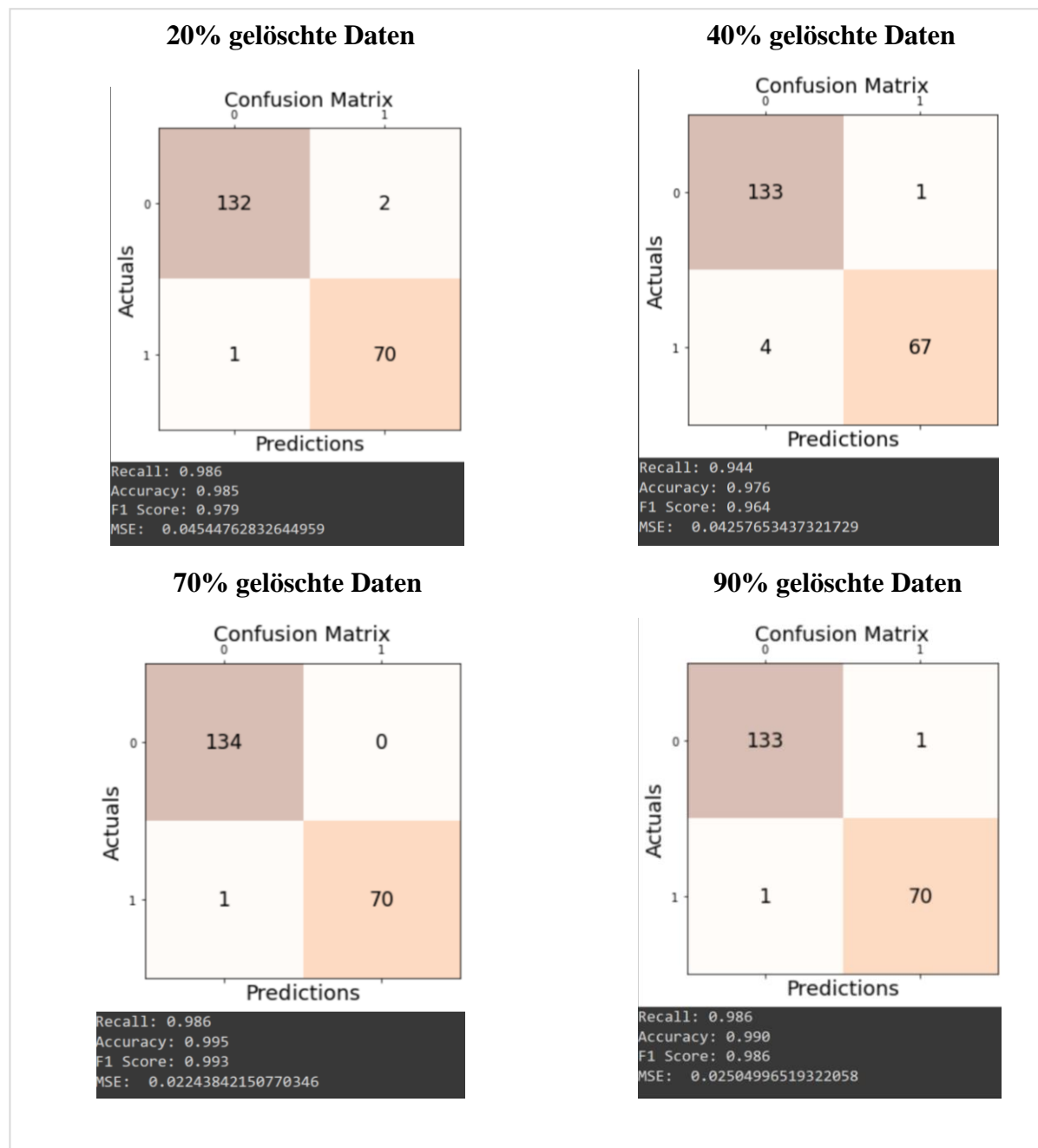
90% gelöschte Daten

Recall: 0.834
Accuracy: 0.828
F1 Score: 0.826
MSE: 0.36606102958762254

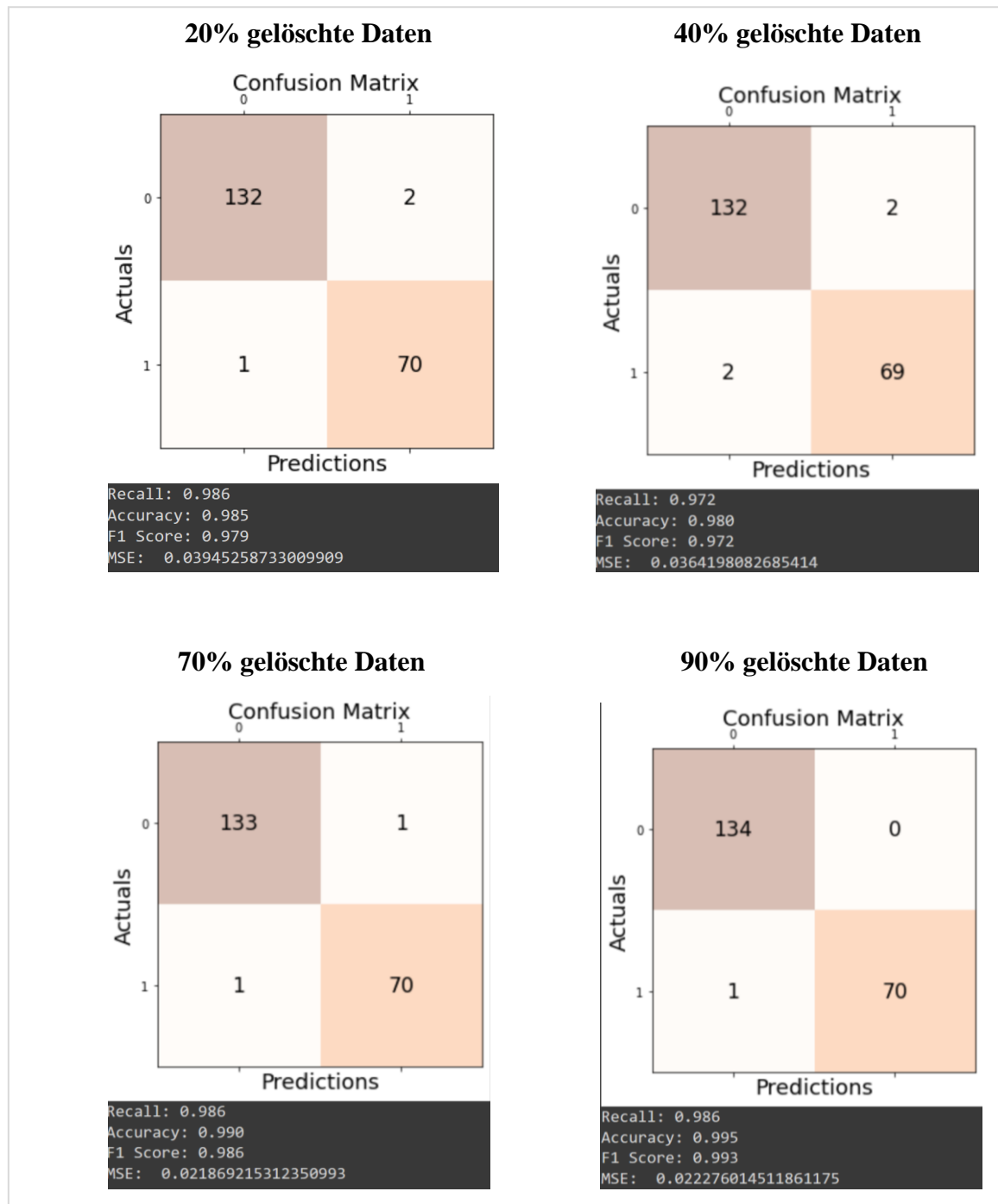
A.2 Heart-Disease: Confusion Matrix – Random Forest

20% gelöschte Daten**40% gelöschte Daten****70% gelöschte Daten****90% gelöschte Daten**

A.3 Breast-Cancer: Confusion Matrix – Entscheidungsbaumverfahren



A.4 Breast-Cancer: Confusion Matrix – Random Forest



7 Literatur

- Alizadehsani, Roohallah, et al.* (2021): Handling of uncertainty in medical data using machine learning and probability theory techniques: A review of 30 years (1991–2020). In: *Annals of Operations Research*, S. 1-42.
- Hristova, Diana* (2014): Considering Currency in Decision Trees in the Context of Big Data. In: *Thirty Fifth International Conference on Information Systems*. Auckland, S. 1-21.
- Maji, Adhyan* (2020): Breast Cancer Prediction.
<https://www.kaggle.com/datasets/adhyanmaji31/breast-cancer-prediction>, Abruf am 2022-12-30
- Pytlak, Kamil* (2021): Personal Key Indicators of Heart Disease.
<https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease>, Abruf am 2022-12-27.
- Qin, Biao; Xia, Yuni; Li, Fang* (2009): DTU: A Decision Tree for Uncertain Data. In: *T. Theeramunkong, T.; Kijirikul, B.; Cercone, N.; Ho, T.-B* (Hrsg.): *Advances in Knowledge Discovery and Data Mining*. Springer, Heidelberg, S. 4-15.
- Reis, Itamar; Baron, Dalya; Shahaf, Sahar* (2019): Probabilistic Random Forest: A Machine Learning Algorithm for Noisy Data Sets. In: *The Astronomical Journal* 157 (1), S. 1-17.
- Tsang, Smith; Kao, Ben; Yip, Kevin Y.; Ho, Wai-Shing; Lee, Sau Dan* (2009): Decision Trees for Uncertain Data. In: *2009 IEEE 25th International Conference on Data Engineering*. Shanghai, China, S. 441-444.

Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Die Arbeit wurde bisher an keiner anderen Hochschule zur Erlangung eines akademischen Grades eingereicht. Die vorgelegten Druckexemplare und die dem Prüfer/der Prüferin zur Verfügung gestellte elektronische Version (PDF-Datei) der Arbeit sind identisch. Von den in §13 Abs. 3 PO 2015 vorgesehenen Rechtsfolgen habe ich Kenntnis.

Regensburg, 24.02.2022

Ort, Datum



Unterschrift

Ich habe die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt. Die Arbeit wurde bisher an keiner anderen Hochschule zur Erlangung eines akademischen Grades eingereicht. Die vorgelegten Druckexemplare und die dem Prüfer/der Prüferin zur Verfügung gestellte elektronische Version (PDF-Datei) der Arbeit sind identisch. Von den in §26 Abs. 6 BPO 2021 vorgesehenen Rechtsfolgen habe ich Kenntnis.

Regensburg, 24.02.2023

Ort, Datum



Unterschrift