Vorhersage der Fließgrenze bei Rotorbauteilen unter Einfluss fehlender Daten

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur / Master of Science

im Rahmen des Masterstudiums

Statistik-Wirtschaftsmathematik (066 395)

eingereicht von

Patrick Freund, BSc

Matrikelnummer 11805793

Ausgeführt bei:	Institut für Stochastik und Wirtschaftsmathematik
Betreuer:	Univ.-Prof. Dipl.-Ing. Dr.techn. Peter Filzmoser

Wien, am 26. August 2025

(Unterschrift Verfasser) (Unterschrift Betreuer)

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, am 26. August 2025

Patrick Freund

Einverständniserklärung zur Plagiatsprüfung

Ich nehme zur Kenntnis, dass die vorgelegte Arbeit mit geeigneten und dem derzeitigen Stand der Technik entsprechenden Mitteln (Plagiat-Erkennungssoftware) elektronisch-technisch überprüft wird. Dies stellt einerseits sicher, dass bei der Erstellung der vorgelegten Arbeit die hohen Qualitätsvorgaben im Rahmen der ausgegebenen der an der TU Wien geltenden Regeln zur Sicherung guter wissenschaftlicher Praxis - „Code of Conduct“ (Mitteilungsblatt 2007, 26. Stück, Nr. 257 idgF.) an der TU Wien eingehalten wurden. Zum anderen werden durch einen Abgleich mit anderen studentischen Abschlussarbeiten Verletzungen meines persönlichen Urheberrechts vermieden.

Wien, am 26. August 2025

Patrick Freund

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Danksagung

Einen besonderen Dank möchte ich der Firma **MTU Aero Engines AG** aussprechen, ohne deren Unterstützung dieses Projekt nicht möglich gewesen wäre. Besonders hervorheben möchte ich die umfassende technische Unterstützung sowie die Bereitstellung der notwendigen Daten. Mein besonderer Dank gilt außerdem meinem Kollegen und Betreuer, **Christoph Lier**, der mich jederzeit mit hilfreichen Ratschlägen, seinem tiefgehenden Fachwissen in der Werkstoffkunde und seiner engagierten Betreuung unterstützt hat.

Ebenso danke ich der **TU Wien** und meinem Betreuer **Peter Filzmoser**. Die exzellente Ausbildung an der Universität und die Unterstützung der Lehrkräfte haben es mir ermöglicht, in den vergangenen Jahren wertvolles Wissen im Bereich der Wirtschaftsmathematik im Rahmen meines Masterstudiums zu erwerben.

Darüber hinaus bedanke ich mich bei all jenen Personen, die mich insbesondere in den letzten Monaten bei der Durchführung des Projekts unterstützt und begleitet haben. Ihre Hilfe, Ratschläge und Ermutigungen haben maßgeblich dazu beigetragen, dass ich dieses Vorhaben erfolgreich abschließen konnte.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Inhaltsverzeichnis

Kurzfassung **5**

Abstract **6**

1 Einführung **7**

2 Datensatz und Problemformulierung **9**
2.1 Datensatz . . . 9
2.2 Notation und Problemformulierung . . . 11

3 Modellpipeline **16**

4 Imputation fehlender Werte **31**
4.1 Muster und Mechanismen fehlender Werte . . . 32
4.2 Imputationsmethoden . . . 37
4.2.1 Mittelwertimputation . . . 38
4.2.2 KNN-Imputation . . . 39
4.2.3 MissForest . . . 41
4.3 Imputationstest . . . 44

5 Vorhersagemodelle **48**
5.1 Lineares Regressionsmodell . . . 48
5.1.1 Gewichtung . . . 50
5.1.2 Ergebnisse . . . 51
5.2 Ridge Regression . . . 52
5.2.1 Gewichtung . . . 52
5.2.2 Ergebnisse . . . 53
5.3 Lasso Regression . . . 53
5.3.1 Gewichtung . . . 55
5.3.2 Ergebnisse . . . 55
5.4 k-Nearest-Neighbor . . . 55
5.4.1 Gewichtung . . . 56
5.4.2 Ergebnisse . . . 56
5.5 Random Forest . . . 57
5.5.1 Gewichtung . . . 60
5.5.2 Ergebnisse . . . 60
5.6 XGBoost . . . 61
5.6.1 Gewichtung . . . 65
5.6.2 Ergebnisse . . . 66

6 Resultate **67**
6.1 Parametertuning . . . 69
6.2 Feature Importance . . . 71
6.3 Entscheidungsprozess . . . 73

7 Conclusio **77**

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

In der Luftfahrt ist eine präzise Bewertung von Materialeigenschaften essenziell, um die notwendigen Sicherheitsstandards gewährleisten zu können. Mit dem Ziel, Prüfversuche und damit verbundene Kosten zu reduzieren sowie experimentell nicht zugängliche Kennwerte im Inneren eines Bauteils zu bestimmen, gewinnen sowohl die Vorhersage von Werkstoffkennwerten wie der Fließgrenze metallischer Rotorscheiben als auch der Umgang mit unvollständigen Datensätzen an Bedeutung.

Um diesen praxisnahen Problemstellungen nachzugehen, wird in dieser Arbeit eine Modellpipeline zur detaillierten Datenaufbereitung und präzisen Vorhersage entwickelt. Der Fokus liegt dabei insbesondere auf der Analyse und Imputation fehlender Werte. Zur Simulation ausgelassener Zugversuche werden künstlich unterschiedliche Anteile fehlender Werte erzeugt.
Zur Minimierung des Vorhersagefehlers werden verschiedene Imputationsmethoden - Mittelwertimputation sowie Methoden basierend auf k-Nearest-Neighbor (KNN-Imputation) und Random Forest (MissForest) - und unterschiedliche Vorhersagemodelle - Lineare Regression, Ridge Regression, Lasso Regression, k-Nearest-Neighbor, Random Forest, XGBoost - sowie deren Zusammenspiel verglichen. Zusätzlich wird ein Gewichtungsansatz untersucht, bei dem der Einfluss einzelner Beobachtungen abhängig von der Anzahl ihrer fehlenden Werte gesteuert wird. Dieser zeigt jedoch nur geringe Vorteile gegenüber den Standardmodellen.

Die KNN-Imputation erzielt über alle Fehldatenanteile hinweg die geringsten Imputationsfehler und liefert in Kombination mit Random Forest die höchste Vorhersagegenauigkeit. Durch die Optimierung der Modellparameter konnte ein mittlerer absoluter Fehler (MAE) von 17,6 MPa erreicht werden.

Diese vielversprechenden Ergebnisse geben Anlass zur Integration in einen datenbasierten Entscheidungsprozess zur Qualitätsbewertung von Rotorbauteilen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abstract

In the aerospace industry, the precise evaluation of material properties is essential to ensure the required safety standards. In order to reduce the number of experimental tests and the associated costs, as well as to estimate material parameters that cannot be measured in the interior of components, both the prediction of material properties such as the yield strength of metallic rotor discs and the handling of incomplete datasets are gaining increasing importance.

To address these practical challenges, this thesis presents a modeling pipeline for detailed data preprocessing and accurate prediction. The main focus is on the analysis and imputation of missing values. To simulate omitted tensile tests, various levels of missing data are artificially introduced into the dataset.
To minimize the prediction error, different imputation techniques - mean imputation as well as methods based on k-Nearest-Neighbor (KNN-Imputation) and Random Forest (MissForest) - and regression models - Linear Regression, Ridge Regression, Lasso Regression, k-Nearest-Neighbor, Random Forest, XGBoost - as well as their combinations are compared. In addition, a weighting approach is investigated, in which the influence of individual observations is controlled based on the number of their missing values. However, this approach shows only limited improvements over the standard models.

KNN-Imputation yields the lowest imputation error across all levels of missing data and achieves the highest prediction accuracy when combined with the Random Forest model. By optimizing the model parameters, a mean absolute error (MAE) of 17.6 MPa is achieved.

These promising results suggest that the proposed pipeline can be integrated into a data-driven decision-making process for the quality assessment of rotor components.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

1 Einführung

In Zeiten, in denen Daten branchenunabhängig einen wachsenden Stellenwert einnehmen und der Zugang zu historischen Daten durch den technischen Fortschritt der letzten Jahre um ein Vielfaches vereinfacht wurde, steht das Ziel, diese entsprechend verstehen und verwerten zu können, im Vordergrund. Besonders für die Qualitätssicherung metallischer Werkstoffe sind datengetriebene Methoden ein wertvolles Werkzeug, um Zusammenhänge zwischen Messgrößen feststellen und quantifizieren zu können sowie darauf basierende Entscheidungen zu treffen. Dadurch lassen sich nicht nur Praxisannahmen empirisch überprüfen, sondern auch Prüfaufwand und Kosten potenziell reduzieren.

Ein besonders sicherheitskritischer Anwendungsbereich ist die Luftfahrt. Um den Transport von Menschen reibungslos gewährleisten und Katastrophen verhindern zu können, ist die präzise Bewertung von Materialeigenschaften essenziell, da bereits minimale Abweichungen zu sicherheitsrelevanten Problemen führen können. Eine zentrale Kenngröße in diesem Kontext ist die Fließgrenze von Rotorscheiben aus Titan- und Nickellegierungen. Diese beschreibt jene Spannung, unter der ein Material dauerhaft plastisch verformt wird, sodass eine Rückkehr in die ursprüngliche Form nicht mehr möglich ist. Diese Definition unterstreicht die Bedeutung des Kennwerts für die Flugsicherheit. Die Fließgrenze wird üblicherweise durch Zugversuche ermittelt. Da diese experimentelle Messung nur mit hohem Kostenaufwand realisierbar ist und bei inneren Lagen des Werkstoffs eine Zerstörung des Bauteils erfordert, ist die Nutzung vorhandener Daten zur Vorhersage dieses Wertes von großem Interesse - wie auch in den weiter unten diskutierten Arbeiten der Literatur gezeigt wird.

Aus diesem Grund wird in dieser Arbeit untersucht, inwieweit Vorhersagemodelle auf Basis historischer Daten und verfügbarer Kennwerte eines Bauteils die Fließgrenze einer metallischen Legierung vorhersagen können. Im Zuge dessen sollen Korrelationen und Muster innerhalb des Datensatzes identifiziert werden, um ein tieferes Verständnis der einfließenden Kennwerte eines solchen Metalls zu gewinnen. Darauf aufbauend sollen getroffene Annahmen auf Basis physikalischer Eigenschaften gestützt werden.
Da Datensätze in der Regel nicht unmittelbar für den Einsatz mathematischer Methoden geeignet sind, ist in den meisten Fällen eine sorgfältige Datenaufbereitung erforderlich. Dieser Schritt nimmt häufig den Großteil des Zeitaufwands ein, wird jedoch selten entsprechend beleuchtet [1]. Insbesondere fehlende Werte verkomplizieren den Einsatz von Vorhersagemodellen und führen zu Unsicherheiten, die die Interpretation der Ergebnisse beeinträchtigen. Der Beitrag dieser Arbeit besteht darin, nicht nur die Methodik für Vorhersagen vorzustellen, sondern auch eine vollständige Modellpipeline mit besonderem Fokus auf den Umgang mit fehlenden Werten zu implementieren. Insbesondere der Umgang mit fehlenden Werten ist ein wichtiger Aspekt, da dadurch Optimierungspotenzial im Fertigungsprozess aufgezeigt werden kann. Vielversprechende Ergebnisse könnten nämlich zu einer Reduktion der experimentellen Versuche führen.

In der Literatur finden sich diverse Arbeiten zur Vorhersage von Werkstoffeigenschaften. So beschreibt etwa Bhandari et al. in [2] eine Fließgrenzenvorhersage für Hoch-Entropie-Legierungen (Metalle mit mehreren Legierungselementen in nahezu gleicher Konzentra-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

tion [3]) bei hohen Temperaturen auf Basis eines Random-Forest-Modells. Dabei konnten Ergebnisse erzielt werden, die mit den durchgeführten Experimenten vergleichbar sind. Haden et al. stellen in [4] ein Konfidenzintervall für die Vorhersage der Fließgrenze der Titanlegierung Ti-6Al-4V basierend auf einer statistischen Schätzung vor. Auch in diesem Fall konnte eine Übereinstimmung zwischen Modellschätzung und experimentell gewonnenen Daten erzielt werden. Ein Vergleich von Machine-Learning-Algorithmen zur Vorhersage der Zugfestigkeit (ein Kennwert, der stark mit der Fließgrenze korreliert) von plastischen Werkstoffen wird in [5] von Mahajan et al. durchgeführt. Dabei werden auf Basis einer Regression mit Gaußprozessen vielversprechende Ergebnisse erzielt. Einen Ansatz, der nicht auf dem Einsatz konventioneller Vorhersagemodelle basiert, sich aber dennoch als erfolgreich erweist, stellen Li et al. in [6] mit einem temperaturabhängigen Modell zur Vorhersage der Fließgrenze metallischer Materialien vor.

Diese Studien haben jedoch gemeinsam, dass der aufwendigen Datenaufbereitung nur wenig Aufmerksamkeit geschenkt wird. Da in den meisten Fällen fehlende Werte nicht näher behandelt werden, wird außerdem angenommen, dass die für die Vorhersage zugrunde liegenden Datensätze keine fehlenden Werte enthalten. Dieses in der Praxis weitgehend unrealistische Szenario wird vermutlich durch Löschung von Beobachtungen mit fehlenden Werten realisiert. Solch ein Vorgehen führt jedoch im Allgemeinen zu einem erheblichen Informationsverlust. Diese Arbeit soll diese Lücke schließen, indem sie die Auswirkungen imputierter Daten auf die Vorhersagequalität analysiert und unterschiedliche Imputationsmethoden vergleicht.

Es finden sich jedoch auch Arbeiten, in denen die Auswirkung fehlender Werte auf die Genauigkeit von Vorhersagemodellen untersucht wird. Dazu zählt etwa die Arbeit [7] von Lyngdoh et al., in der ein Vorstoß zur Untersuchung der Auswirkungen fehlender Daten auf die Vorhersage im Rahmen der Werkstoffüberprüfung unternommen wird. Für die Vorhersage der Druck- und Zugfestigkeit von Beton wird im Vorfeld ein unvollständiger Datensatz mit verschiedenen Imputationsmethoden behandelt. Ein ähnliches Ziel verfolgt Yin in [8], wobei der Einsatz unterschiedlicher Imputationsmethoden untersucht und die Auswirkung auf ein Lineares Regressionsmodell analysiert wird.

In diesen Arbeiten wird jedoch nicht näher auf die Integration der Ergebnisse in den Entscheidungsprozess eingegangen. Auch diese Lücke soll durch die Modellpipeline dieser Arbeit geschlossen werden, indem ein kompakter Überblick zur Verwendung der aus den mathematischen Ergebnissen gewonnenen Informationen gegeben wird. Ein Vergleich mit diesen Arbeiten erweist sich als schwierig, da die resultierenden Ergebnisse auf unterschiedlichen Datensätzen basieren. Diese unterscheiden sich insbesondere im Anteil fehlender Daten, den Gründen für deren Fehlen sowie in der zugrunde liegenden Struktur. Darüber hinaus ist die Vorhersagequalität stark von den erklärenden Informationen abhängig, die ebenfalls zwischen den Arbeiten variieren.

Im nächsten Kapitel wird der verwendete Datensatz vorgestellt und das vorliegende Problem genauer beschrieben.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

2 Datensatz und Problemformulierung

2.1 Datensatz

Eines der Ziele einer Rotorteilüberprüfung ist die Bestimmung der Fließgrenze des Bauteils bei Durchführung eines Zugversuchs. Diese Überprüfung findet in der Regel an verschiedenen Lagen des Prüfkörpers statt. Fester Bestandteil einer solchen Überprüfung ist die Feststellung der Fließgrenze an der Lage 1 - dem sogenannten Testring. Während ein Zugversuch am Testring keine Zerstörung des Bauteils erfordert, ist dies bei den anderen Lagen nicht der Fall. Daher wird ein Zugversuch an diesen Lagen meist vermieden und nur in Ausnahmefällen durchgeführt.
Daher ist es sinnvoll, sich Gedanken zu machen, wie sich aus den Ergebnissen am Testring Vorhersagen für die weiteren Lagen ableiten lassen.

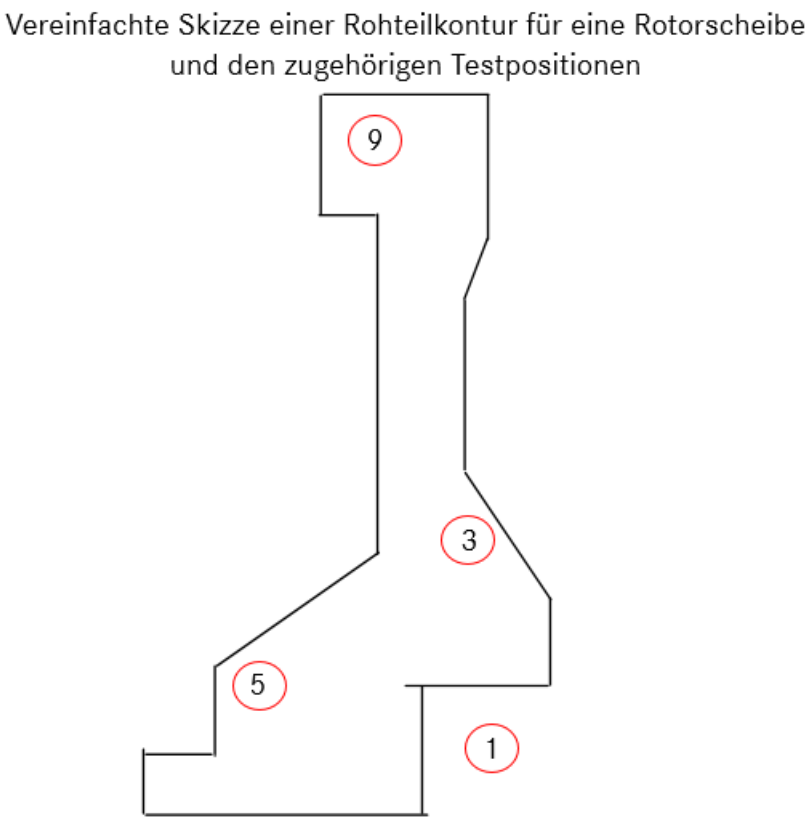

Abbildung 1: Skizze einer Rotorscheibe mit eingezeichneten Testlagen

Abbildung 1 zeigt die unterschiedlichen Lagen einer beispielhaften Rotorscheibe. Der Testring befindet sich an Lage 1, bei dem für jede Beobachtung die Ergebnisse des Zugversuchs vorliegen. Auf Basis dieser Information soll die Fließgrenze an den weiteren eingezeichneten Lagen des Bauteils bestimmt werden.

Der dieser Aufgabenstellung zugrunde liegende Datensatz besteht aus einer Ansammlung von Testergebnissen, bei denen Prüfkörper außerhalb des Testrings auf ihre Fließgrenze geprüft wurden. Insgesamt enthält der Datensatz 21.396 solcher Tests an unterschiedlichen

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Lagen. Die zu erklärende Fließgrenze eines Prüfkörpers stellt eine quantitative Zielgröße dar, deren Ausprägung mithilfe verschiedener Variablen - einschließlich der Testergebnisse am Testring des jeweiligen Bauteils - modelliert werden soll.
Der Rohdatensatz umfasst insgesamt 18 erklärende Variablen. Diese Anzahl wird sich durch nachfolgende Überlegungen innerhalb der Datenaufbereitung noch verändern. Tabelle 1 gibt einen ersten Überblick über die im Datensatz vorkommenden Variablen.

Variable	**Beobachtung 1**	**Beobachtung 2**
Fließgrenze	1015	915
Temperatur	600	350
Jahr	2017	2024
Lage	2	2
Durchmesser Bolzen	10	8
Materialspezifikation	Spezifikation 1	Spezifikation 5
Serialnummer	Nummer 1	Nummer 8
Prozessspezifikation	Spezifikation 3	Spezifikation 8
Material Name	Nickellegierung	Titanlegierung
Triebwerksprogramm	TWP 1	TWP 2
Teilnummer	TN 3	TN 18
Fließgrenze Lage 1	1180	840
Bruchdehnung A4 Lage 1	15,3	16,1
Bruchdehnung A5 Lage 1	NA	16,7
Brucheinschnürung Lage 1	28,9	36,0
Zugfestigkeit Lage 1	1290	1001
Wärmebehandlungsmethode	A	A
Schmelzmethode	Triple Melt	Triple Melt
Zulieferer	Route 3	Route 8

Tabelle 1: Übersicht der Daten anhand von zwei beispielhaften Beobachtungen

Die Zielvariable **Fließgrenze** ist in der Einheit Megapascal (MPa) angegeben und nimmt in der Regel Werte zwischen 700 und 1.300 MPa an. Damit ein Bauteil im weiteren Verlauf eingesetzt werden kann, muss dessen Fließgrenze über einem, je nach Lage, durch eine Norm vorgegebenen Schwellenwert liegen.
Besonders kritisch ist der Fall, in dem ein Rotorbauteil fälschlicherweise als akzeptabel

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

eingestuft wird, also wenn der prognostizierte Wert über dem Schwellenwert liegt, der tatsächliche Wert jedoch darunter. Ein solches Fehlurteil gilt es möglichst zu verhindern.

Die am Testring ermittelten Testergebnisse werden durch die Variablen **Fließgrenze Lage 1**, **Bruchdehnung A4 Lage 1**, **Bruchdehnung A5 Lage 1**, **Brucheinschnürung Lage 1** und **Zugfestigkeit Lage 1** dargestellt. Eine hohe Korrelation zwischen erklärender Variable und Zielgröße ist insbesondere bei der Fließgrenze an der Testringlage zu erwarten. Dies zeigt sich bereits anhand der beiden Beispielbeobachtungen deutlich. Neben der Fließgrenze wird auch von der Bruchdehnung, der Brucheinschnürung sowie der Zugfestigkeit erwartet, dass sie wesentlich zur Güte des Vorhersagemodells beitragen. Die Tests werden unter unterschiedlichen Temperaturbedingungen durchgeführt - diese Information ist in der Variable **Temperatur** enthalten. Die Variable **Lage** gibt an, an welcher Lage des Bauteils die Fließgrenze überprüft wurde. Initial wird nicht davon ausgegangen, dass die Varianz der Zielvariable durch unterschiedliche Zeitpunkte der Tests erklärbar ist. Insofern wird die Variable **Jahr** eher als Hilfsvariable für den gesamten Workflow betrachtet - hierzu später mehr. Die weiteren erklärenden Variablen enthalten allgemeine Informationen zur sogenannten CutUpID, also zum Prüfkörper. Dazu zählen die **Materialspezifikation**, die eine Normierung der Testumgebung darstellt, die **Serialnummer**, die im weiteren Verlauf eine untergeordnete Rolle spielt, die **Prozessspezifikation**, die den Prozess beschreibt, das Material in der Variable **Material Name**, das **Triebwerksprogramm**, das angibt, in welcher Turbine das Bauteil später verbaut wird, die **Teilnummer**, die bereits durch andere erklärende Variablen beschrieben wird, sowie die **Wärmebehandlungsmethode** und die **Schmelzmethode**. Die Variable **Zulieferer** enthält zudem Informationen über die sogenannte Zuliefererroute, die sich aus dem Zulieferer des Gusses sowie der Schmelze zusammensetzt. Diese hat bei statistischen Untersuchungen der Testergebnisse eine hohe Bedeutung, da ermittelt werden soll, ob es Routen gibt, bei denen die Fließgrenze regelmäßig einen niedrigen Wert annimmt. Auf Basis dieser Information können Zulieferer mit strengeren Normierungen konfrontiert oder - sofern möglich - gänzlich vermieden werden.

2.2 Notation und Problemformulierung

Da die Zielvariable im untersuchten Datensatz quantitativ ist und mehrere erklärende Variablen die Varianz der Zielvariable beschreiben sollen, handelt es sich um ein **multiples Regressionsmodell**.

Es wird die in der Statistik gängige Notation von James et al. aus [9] übernommen und

- n = Anzahl der Beobachtungen
- p = Anzahl der Merkmale (auch erklärende Variablen oder Prädiktoren genannt)

definiert.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die Zielvariable wird im Folgenden durch den Vektor

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \in \mathbb{R}^n \tag{1}$$

dargestellt, wobei die Komponenten y_i mit $i = 1, \ldots, n$ den Zielwert der i-ten Beobachtung angeben. Der Mittelwert der Komponenten des Zielvariablenvektors ergibt sich gemäß der Notation von Kuhn und Johnson in [10] zu

$$\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i. \tag{2}$$

Die erklärenden Variablen sind in der Regressormatrix

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \tag{3}$$

zusammengefasst. Dabei bezeichnet der Eintrag x_{ij} den Wert der j-ten erklärenden Variable der i-ten Beobachtung. Die Spalten der Regressormatrix, also die Beobachtungen einer erklärenden Variable, werden mit

$$\mathbf{x}_j = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix} \quad \text{für } j = 1, \ldots, p \tag{4}$$

notiert.
Analog zur Zielvariable $\mathbf{y}$ ergibt sich der Durchschnitt einer Spalte $\mathbf{x}_j$ zu

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^{n} x_{ij}. \tag{5}$$

Wird im Folgenden ein konkretes Merkmal oder die Zielvariable als Zufallsvariable betrachtet, erfolgt die Notation mit Großbuchstaben $\mathbf{X}_j$ bzw. $\mathbf{Y}$. Deren Realisierungen basierend auf den beobachteten Datensatz werden wie zuvor eingeführt mit Kleinbuchstaben $\mathbf{x}_j$ bzw. $\mathbf{y}$ notiert.

Das multiple Regressionsmodell lässt sich wie folgt formulieren:

$$\mathbf{y} = f(\mathbf{X}) + \boldsymbol{\varepsilon}. \tag{6}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die Funktion f modelliert den Zusammenhang der Prädiktoren mit der Zielvariable $\mathbf{y}$ und $\boldsymbol{\varepsilon}$ stellt einen zufälligen Fehlerterm dar. Da f unbekannt ist, besteht das Ziel darin, eine Schätzung $\hat{f}$ zu ermitteln, um auf Basis beobachteter Prädiktoren Vorhersagen treffen zu können:

$$\hat{\mathbf{y}} = \hat{f}(\mathbf{X}). \tag{7}$$

Wie in [9] beschrieben, ist die exakte Form der Funktion f für viele Anwendungen - auch für den vorliegenden Datensatz - von untergeordneter Bedeutung.

Zur Schätzung der Funktion f stehen verschiedene Vorhersagemodelle zur Verfügung, die sich in ihrer Komplexität, der Interpretierbarkeit und den zugrunde liegenden Annahmen unterscheiden. Die meisten Modelle haben jedoch gemeinsam, dass Ausreißer, fehlende Werte und kategoriale Variablen zu Problemen oder verzerrten Schätzungen führen können.
Tabelle 2 veranschaulicht diese, in der Praxis bei Datensätzen häufig vorkommenden Probleme, anhand der vorliegenden Daten:

Variable	**Anzahl NAs**	**Anzahl Ausreißer**	**Anzahl seltene Kategorien**	**Variablentyp**	**Anzahl Kategorien**
Fließgrenze	0	11	-	stetig	-
Temperatur	0	243	3	kategorisch	9
Jahr	629	12	5	kategorisch	16
Lage	0	0	0	kategorisch	11
⌀Bolzen	0	52	3	kategorisch	8
Materialspezifikation	0	0	0	kategorisch	45
Serialnummer	0	0	0	kategorisch	2195
Prozessspezifikation	0	0	0	kategorisch	343
Material	0	0	0	kategorisch	9
Triebwerksprogramm	7	0	0	kategorisch	13
Teilnumber	0	0	0	kategorisch	111
Fließgrenze Lage 1	0	10	-	stetig	-
Dehnung A4 Lage 1	8839	44	-	stetig	-
Dehnung A5 Lage 1	6730	137	-	stetig	-
Einschnürung Lage 1	0	0	-	stetig	-
Zugfestigkeit Lage 1	0	22	-	stetig	-
Wärmebehandlungsmethode	0	0	0	kategorisch	6
Schmelzmethode	433	0	0	kategorisch	2
Zulieferer	0	0	0	kategorisch	30

Tabelle 2: Übersicht der Variablen hinsichtlich Ausreißer, fehlende Werte und Datentypen

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Zunächst sind in der dritten Spalte die Ausreißer jeder Variable dargestellt. Diese wurden für numerische Variablen initial mit der sogenannten modifizierten Z-Score-Methode ermittelt, auf die später noch näher eingegangen wird. Bei kategorialen Variablen wurden mithilfe der relativen Häufigkeit „seltene“ Kategorien ermittelt (siehe vierte Spalte), und alle zugehörigen Beobachtungen wurden als Ausreißer markiert. Die Summe dieser „Ausreißerbeobachtungen“ ist in der dritten Spalte aufgeführt. Die türkis markierten Zeilen sind davon nicht betroffen - seltene Kategorien sind in diesen Variablen durchaus möglich und stellen kein Grund zur Achtsamkeit dar. Auslöser dafür können etwa neu eingeführte Zulieferer oder Materialspezifikationen sein. Seltene Kategorien infolge von Tippfehlern bei der Datenangabe wurden bereits im Voraus kontrolliert und können ausgeschlossen werden.
In der fünften Spalte ist zu erkennen, dass ein Großteil der Variablen qualitativ ist. Durch gängige Encoding-Methoden wie One-Hot-Encoding stellt dies grundsätzlich kein großes Problem dar. Da aber einige Variablen eine hohe Anzahl an Kategorien besitzen, würde One-Hot-Encoding die Dimension der Regressormatrix vervielfachen. Deshalb wird im weiteren Verlauf der Arbeit nach alternativen Lösungen gesucht. Insbesondere die drei grau markierten Zeilen enthalten eine Vielzahl an Kategorien und müssen dementsprechend separat behandelt werden. Da die Serialnummer bzw. die Teilnummer bereits durch andere erklärende Variablen beschrieben werden und angenommen wird, dass sie nur wenig zum Fit des Vorhersagemodells beitragen, werden sie aus dem Datensatz entfernt. Somit ist nur noch die hohe Anzahl an Kategorien in der Variable Prozessspezifikation zu berücksichtigen.
Darüber hinaus weisen einige Variablen im Datensatzes fehlende Werte auf. Da das einfache Entfernen von Beobachtungen mit fehlenden Werten - insbesondere wegen der Variablen Bruchdehnung A4 Lage 1 und Bruchdehnung A5 Lage 1 - zu einem erheblichen Informationsverlust führen würde, wird von dieser Option abgesehen. Stattdessen wird der Datensatz mithilfe geeigneter Imputationsmethoden vervollständigt, bevor die bereits erwähnten Vorhersagemodelle zur Anwendung kommen.

Um Vorhersagen auf Basis von Daten zu treffen, die zumindest einen dieser kritischen Aspekte aufweisen, wird ein Verfahren benötigt, das robust gegenüber Ausreißern, fehlenden Werten und kategorialen Variablen mit hoher Kardinalität ist. Ziel ist es, eine Modellpipeline zu konstruieren, die diese Robustheit besitzt und mit einem geeigneten Vorhersagemodell die Fließgrenze im Inneren des Bauteils basierend auf Messwerten am Testring zuverlässig prognostiziert. Resultierend daraus ergeben sich Möglichkeiten für die spätere Integration in den Entscheidungsprozess zur Qualitätssicherung von Rotorbauteilen.
Wie bereits erläutert, liegt der Fokus dieser Arbeit insbesondere auf dem Umgang mit fehlenden Werten, um Reduktionen experimenteller Versuche zu ermöglichen. Da Optimierungspotenzial und damit verbundene Kosteneinsparungen von großer Bedeutung sind, richtet sich das besondere Interesse auf die Versuche am Testring - konkret auf jene fünf Variablen mit der Endung „Lage 1“. Ziel ist es zu untersuchen, ob diese reduziert und durch Imputation ersetzt werden können, ohne die Genauigkeit zukünftiger Vorhersagen signifikant zu beeinflussen. Teilziel dieser Arbeit ist daher die Analyse des Auftretens fehlender Werte sowie die Bewertung verschiedener Imputationsmethoden in Kombination mit den eingesetzten Vorhersagemodellen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Üblicherweise würde eine umfassende grafische Analyse der Daten vor der Modellierung helfen, die Sinnhaftigkeit der Modelle besser einzuschätzen. Da der Datensatz jedoch strengen Datenschutzrichtlinien unterliegt, ist dies in der vorliegenden Arbeit nicht möglich. Im Folgenden werden daher nur einige ausgewählte Visualisierungen ohne Achsenskalierung dargestellt.

Der Aufbau der Arbeit ist wie folgt:

- Kapitel 3 stellt die vollständige Modellpipeline vor. Hier soll ein allgemeiner Überblick über den Aufbau des Verfahrens, inklusive der für die Robustheit notwendigen Datenaufbereitungsschritte (Ausreißererkennung, Kodierung kategorialer Variablen, Skalierung, Imputation) und der verwendeten Vorhersagemodelle sowie die Validierung dieser Modelle gegeben werden.

- Kapitel 4 gibt einen Einblick in die Theorie fehlender Werten und beschreibt die theoretischen Hintergründe der drei angewendeten Imputationsmethoden zur Ersetzung dieser. Ihre generelle Leistungsfähigkeit unabhängig von der späteren Vorhersagequalität wird bei unterschiedlich hohem Anteil fehlender Werte anhand eines Vergleichs zwischen den bekannten wahren Werten und den imputierten Werten bewertet.

- Kapitel 5 behandelt die zur Auswahl stehenden linearen und nichtlinearen Vorhersagemodelle. Für jedes Modell wird ein Ansatz vorgestellt, mit dem Beobachtungen mit hoher Unsicherheit (etwa durch eine hohe Anzahl an Ausreißern oder fehlender Werte) im Modelltraining geringer gewichtet werden. Dadurch soll der Einfluss von imputierter Information abgeschwächt werden. Es wird zum einen die Abhängigkeit der Wahl der Imputationsmethode aus Kapitel 4 im Hinblick der Vorhersagegüte untersucht, andererseits sollen Rückschlüsse über die Leistung der Vorhersagemodelle abhängig vom Anteil fehlender Werte gezogen werden.

- Kapitel 6 präsentiert die finale Kombination aus Imputationsverfahren und Vorhersagemodell sowie die daraus resultierenden Möglichkeiten zur Einbindung in den Entscheidungsprozess. Zudem werden die Auswirkungen von Modellparametern zur weiteren Optimierung der Vorhersagegüte diskutiert. Prognosseintervalle werden konstruiert, um Unsicherheiten einzufangen und so Entscheidungen über die Akzeptanz eines Rotorbauteils zu untersützen. Darüber hinaus wird die sogenannte Feature Importance analysiert, um zu überprüfen, ob die Annahmen aus der Praxis über die Relevanz einzelner Variablen bestätigt werden können.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3 Modellpipeline

Dieses Kapitel gibt einen Überblick über die gesamte Modellpipeline und deren Funktionsweise. Die Implementierung ermöglicht es, verschiedene Imputationsmethoden und Vorhersagemodelle zu kombinieren, um so für die gegebene Problemstellung und den jeweiligen Anteil an fehlenden Werten die optimale Modellkonfiguration zu identifizieren. Bereits vor der Ausführung kann zudem festgelegt werden, ob ein gewichteter Modellfit verwendet werden soll. Um der Frage nachzugehen, inwieweit sich Versuche am Testring einsparen lassen, wird der Anteil künstlich eingestreuter fehlender Werte variiert. Dieses Vorgehen wird im Folgenden als „Imputationstest“ bezeichnet. Dadurch wird eine Analyse der Auswirkungen unvollständiger Informationen auf die Vorhersagequalität ermöglicht. Die Pipeline ist daher zunächst darauf ausgelegt, für jeden Imputationstest ein optimales Zusammenspiel aus Imputation, Gewichtung und Vorhersage zu erzielen. Sobald diese Konfiguration bestimmt ist, erfolgen darauf basierend Vorhersagen und Intervallbildungen, um künftig Entscheidungen über Bauteile treffen zu können.

Abbildung 2: Aufbau der gesamten Modellpipeline

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die Modellpipeline wurde in Python mittels PyCharm implementiert. Für allgemeine Befehle wurden die Pakete **NumPy** [11], **Pandas** [12] und **SciPy** [13] verwendet - für die Visualisierung kam das Paket **Matplotlib** [14] zum Einsatz.
Abbildung 2 visualisiert den Ablauf der Pipeline. Die hellorange eingefärbten Felder kennzeichnen die Datenaufbereitung (Feature Engineering). Dieser Teil ist essenziell, da viele Vorhersagemodelle strenge Anforderungen an die Features stellen. Feature Engineering stellt nicht nur die Anwendbarkeit sicher: Eine geeignete Aufbereitung der Daten kann laut verschiedener Quellen wie [9, 1] maßgeblich die Qualität des Vorhersagemodells beeinflussen. Der Einfluss ist dabei stark vom angewendeten Modell abhängig.
Obwohl robuste Methoden grundsätzlich eine Möglichkeit bieten, den zeitaufwändigen Prozess der Datenaufbereitung zu umgehen, werden diese in der vorliegenden Arbeit nicht berücksichtigt. Grund dafür ist die hohe Anzahl kategorialer Variablen im Datensatz, bei denen robuste Verfahren in der Regel nur eingeschränkt geeignet sind.

Ausreißer: Zu Beginn der Datenaufbereitung steht das Auffinden von Ausreißern im Mittelpunkt. Barnett und Lewis definieren in [15] einen Ausreißer als eine Beobachtung, die inkonsistent mit dem restlichen Datensatz ist. Typische Ursachen für Ausreißer sind manuelle Eingabefehler (z.B. die Eingabe von 10.000 MPa statt 1.000 MPa, siehe Abbildung 3 links) oder Messfehler (z.B. Umrechnungsfehler zwischen Grad Celsius und Kelvin, siehe Abbildung 3 rechts). Aufgrund der Struktur des Datensatzes ist anzunehmen, dass sich diese Fehlerquellen auf einzelne Variablen beschränken und gezielt identifizieren lassen können. Daher wird eine weniger komplexe, univariate Ausreißeranalyse als methodisch ausreichend erachtet. Da die kategorialen Variablen vor der Ausreißeranalyse noch nicht kodiert wurden, erfolgt eine Unterscheidung zwischen qualitativen und quantitativen Variablen.

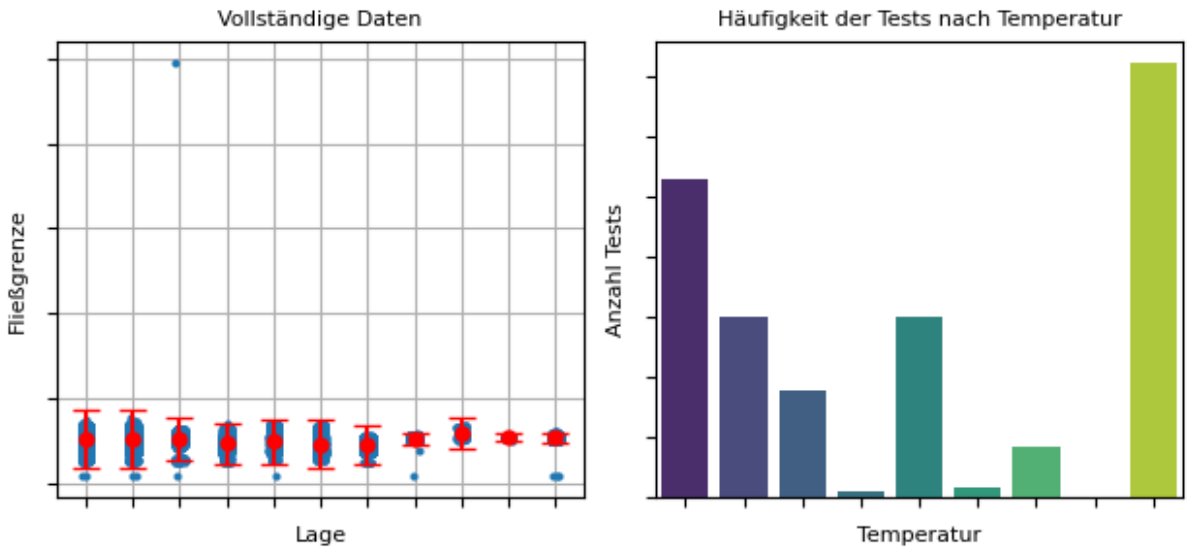

Abbildung 3: Darstellung der im Rohdatensatz vorkommenden Ausreißer

Für die Ausreißererkennung in den qualitativen Variablen (alle stetigen Variablen aus Tabelle 2) stellt die Z-Score-Methode eine gängige Möglichkeit dar.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Dabei wird für jede Beobachtung der Variable $\mathbf{X}_j$ der Z-Score

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j} \tag{8}$$

berechnet, wobei $\bar{x}_j$ dem in (5) definierten Mittelwert entspricht und

$$s_j = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)^2} \tag{9}$$

die Standardabweichung von $\mathbf{x}_j$ ist. Beobachtungen mit einem absoluten Z-Score größer als dem gängigen Schwellenwert $\tau = 3$ gelten dabei als Ausreißer [16]:

$$|z_{ij}| > 3 \quad \Rightarrow \quad \text{Ausreißer.} \tag{10}$$

Da jedoch sowohl der Mittelwert als auch die Standardabweichung ausreißerempfindlich sind und die Z-Score-Methode zusätzlich eine Normalverteilung der Variablen voraussetzt [16], wird stattdessen die modifizierte Z-Score-Methode verwendet. In der ersetzt der robustere Median den Mittelwert, und die Standardabweichung wird durch die Median Absolute Deviation (MAD) ersetzt. Dieses robuste Streuungsmaß ist für die Realisierung $\mathbf{x}_j$ der Variable $\mathbf{X}_j$ definiert als:

$$\text{MAD}_j = \text{Median}_{i=1,\ldots,n} \left(|x_{ij} - \text{Median}(\mathbf{x}_j)| \right) \tag{11}$$

Es ist der Median der absoluten Abstände von jedem Wert x_{ij} zum Median von $\mathbf{x}_j$. Damit ergibt sich der modifizierte Z-Score als

$$z_{ij}^{(\text{MOD})} = \frac{0{,}6745 \cdot (x_{ij} - \text{Median}(\mathbf{x}_j))}{\text{MAD}_j}. \tag{12}$$

Iglewicz und Hoaglin empfehlen in [16] sowohl die Konstante 0,6745, da $\mathbb{E}[\text{MAD}] = 0{,}6745 \cdot \sigma$ für großes n gilt, als auch den Schwellenwert $\tau = 3{,}5$ als Schwellenwert zur Identifikation eines Ausreißers:

$$\left| z_{ij}^{(\text{MOD})} \right| > 3{,}5 \quad \Rightarrow \quad \text{Ausreißer.} \tag{13}$$

Sei $L_j = \{l_j^{(1)}, l_j^{(2)}, \ldots, l_j^{(s)}\}$ die Menge aller Kategorien einer kategorialen Variable $\mathbf{X}_j$ mit s Kategorien. Zur besseren Übersichtlichkeit wird in der theoretischen Darstellung der Ausreißererkennung für kategoriale Variablen auf die explizite Verwendung der Variablenindizes verzichtet. Die Kategorie $l_j^{(k)}$ wird dabei verkürzt als k notiert.

Für jede Kategorie k wird die relative Häufigkeit

$$h(k) = \frac{H(k)}{n} \tag{14}$$

bestimmt, wobei $H(k)$ die absolute Häufigkeit der Kategorie k angibt. Seltene Kategorien k werden auf Basis des Schwellenwertes $\tau = 0{,}01$ identifiziert:

$$h(k) < 0{,}01 \quad \Rightarrow \quad k \quad \text{Ausreißerkategorie der Variable } \mathbf{X}_j. \tag{15}$$

Beobachtungen, bei denen die Variable $\mathbf{X}_j$ den Wert k annimmt, werden infolgedessen als Ausreißer markiert.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Nach der Identifikation von Ausreißern folgt der angemessene Umgang mit diesen auffälligen Daten.

Num. Variable
1000
10000
953
890
1200

→

Num. Variable
1000
953
890
1200

Kat. Variable
Triple Melt
Triple Melt
Triple Melt
Single Melt
Triple Melt

→

Kat. Variable
Triple Melt
Triple Melt
Triple Melt
Triple Melt

Abbildung 4: Ausreißer werden mittels modifizierter Z-Score-Methode und relativer Häufigkeit identifiziert und aus dem Datensatz entfernt

Da im Datensatz bereits vor der Ausreißeranalyse fehlende Werte vorkommen, werden die Einträge in den markierten Ausreißerzellen entfernt (siehe Abbildung 4) und im weiteren Verlauf durch Imputation ersetzt. Eine Ausnahme bildet die erklärende Variable Jahr: Für Beobachtungen mit seltenen Jahresangaben - ausschließlich einzelne Versuche vor dem Jahr 2015 - wird eine neue Kategorie gebildet, in der diese zusammengefasst werden.

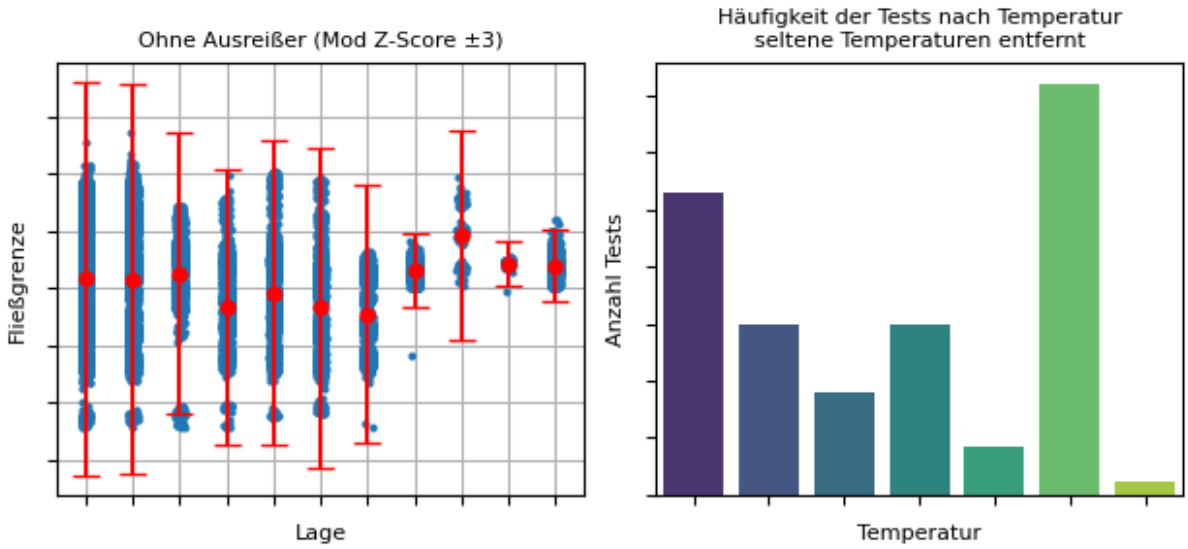

Abbildung 5: Darstellung wie in Abbildung 3 nach Entfernen der Ausreißer

Abbildung 5 zeigt den bereinigten Datensatz aus Abbildung 3. In der numerischen Variable Fließgrenze wurde mithilfe der modifizierten Z-Score-Methode erfolgreich der offensichtliche Eingabefehler von 10.000 MPa entfernt. Die durch Umrechnungsfehler entstandenen seltenen Kategorien in der kategorialen Variable Temperatur wurden ebenfalls entfernt und sollen im weiteren Verlauf mithilfe der Imputation ersetzt werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Encoding: Allgemein hängt die Notwendigkeit, kategoriale Variablen in numerische Features zu transformieren, vom gewählten Vorhersagemodell ab. Während baumbasierte Methoden mit kategorialen Merkmalen in der Regel gut umgehen können, setzen lineare Modelle numerische Eingabedaten voraus. Da im Rahmen des Modellvergleichs auch lineare Modelle berücksichtigt werden, erfolgt eine entsprechende Transformation der qualitativen Variablen. Dabei wird, abhängig von der Anzahl der Kategorien einer kategorialen Variable und dem Vorkommen fehlender Werte, entschieden, ob Target-Encoding oder One-Hot-Encoding aus dem **Pandas**-Paket angewendet wird.

One-Hot-Encoding ist ein Verfahren, bei dem jede Kategorie einer kategorialen Variable durch eine eigene sogenannte Dummy-Variable abgebildet wird. Eine Variable mit s Kategorien wird dadurch in s binäre Merkmale überführt, wobei pro Beobachtung genau eine den Wert 1 annimmt und alle übrigen den Wert 0. In Abbildung 6 ist dieses Vorgehen für die Testlage mit $s = 2$ abgebildet.
One-Hot-Encoding ist ein gängiges Verfahren für den Umgang mit kategorialen Variablen in Vorhersagemodellen und eignet sich besonders zur Interpretation einzelner Kategorien (z.B. eigener Koeffizient bei linearen Modellen), vorausgesetzt die Variable weist nur wenige Kategorien auf. Andernfalls würde die Abbildung auf s neue Merkmale die Dimension der Regressormatrix $\mathbf{X}$ erheblich vergrößern.

Für kategoriale Merkmale mit hoher Kardinalität sollte eine Alternative gewählt werden. In der Modellpipeline werden daher Variablen mit mehr als 15 Kategorien mittels eines gewichteten Target-Encoding, wie es Micci-Barreca in [17] vorschlägt, in numerische Merkmale überführt.
Analog zur Ausreißeranalyse wird für die Kategorien die verkürzte Schreibweise k angesetzt. Die Grundidee beim Target-Encoding besteht darin, eine Kategorie k auf den bedingten Erwartungswert der Zielvariable abzubilden. Da der Erwartungswert unbekannt ist, wird ein Schätzer S_k angesetzt:

$$k \to S_k \approx \mathbb{E}[Y|\mathbf{X}_j = k]. \tag{16}$$

Um einen Schätzer S_k für den bedingten Erwartungswert zu definieren, sei

- n_k = Anzahl der Beobachtungen mit der Kategorie k in $\mathbf{x}_j$
- M_k = Menge alle Beobachtungen mit der Kategorie k in $\mathbf{x}_j$

Für S_k könnte der Mittelwert $\bar{y}_k$ der Zielwerte innerhalb der Kategorie k angesetzt werden:

$$S_k = \frac{\sum_{i \in M_k} y_i}{n_k} = \bar{y}_k. \tag{17}$$

Ist n_k jedoch klein, führt dieser Ansatz zu Overfitting, da S_k in diesem Fall auf nur wenigen Beobachtungen basiert. Aus diesem Grund wird eine Regularisierung vorgenommen und für den Schätzer S_k neben $\bar{y}_k$ auch der globale Mittelwert der Zielwerte $\bar{y}$ herangezogen:

$$S_k = \lambda(n_k) \cdot \frac{\sum_{i \in M_k} y_i}{n_k} + (1 - \lambda(n_k)) \cdot \frac{\sum_{i=1}^{n} y_i}{n} = \lambda(n_k) \cdot \bar{y}_k + (1 - \lambda(n_k)) \cdot \bar{y}. \tag{18}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Die Gewichtungsfunktion λ ist monoton steigend und sorgt dafür, dass bei großem n_k der kategoriale Mittelwert dominiert, während bei kleinem n_k der globale Mittelwert stärker berücksichtigt wird.
Micci-Barreca schlägt in [17] unter anderem

$$\lambda(n) = \frac{n}{m+n} \tag{19}$$

als mögliche, durch den Glättungsparameter m gesteuerte, Gewichtungsfunktion vor. Mit steigendem m nimmt die Glättung zu, wodurch der globale Mittelwert stärker ins Gewicht fällt. Der Parameter m wird individuell für jede Kategorie datengetrieben als das Verhältnis der Varianz der Zielwerte innerhalb der Kategorie zur globalen Varianz der Zielwerte

$$m_k = \frac{s_{y_k}}{s_y} \tag{20}$$

angesetzt. Dadurch fließt bei hoher Variabilität innerhalb einer Kategorie der globale Mittelwert stärker in den Schätzer S_k ein.
Target-Encoding ermöglicht es, eine Variable mit s Kategorien - statt wie beim One-Hot-Encoding in s Dummy-Variablen - durch eine einzige numerische Variable zu ersetzen. Abbildung 6 stellt das soeben beschriebene Target-Encoding für den Fall $s = 2$ auf Basis von 6 Beobachtungen dar. Neben der einfachen Implementierung berücksichtigt Target-Encoding auch Informationen der Zielvariable in der kodierten Darstellung. Pargent et al. zeigen in [18], dass Target-Encoding anderen Methoden überlegen ist, was die Wahl als Kodierungsmethode bei einer hohen Anzahl an Kategorien rechtfertigt.
In der Modellpipeline werden die kategorialen Variablen folgendermaßen zu numerischen Merkmalen transformiert:

- Target Coding: Prozessspezifikation, Materialspezifikation, Zulieferer, Triebwerksprogramm, Schmelzmethode
- One-Hot-Encoding: Lage, Material Name, Wärmebehandlungsmethode

Bei Prozessspezifikation, Materialspezifikation und Zulieferer wird aufgrund der hohen Anzahl der Kategorien (jeweils über dem Schwellenwert von 15) Target-Encoding angewendet, während bei Triebwerksprogramm und Schmelzmethode diese Methode aufgrund von vorkommenden fehlenden Werten zum Einsatz kommt.

Fließgrenze	Schmelzm.
980	Triple Melt
1100	Triple Melt
960	Double Melt
890	Triple Melt
1200	Double Melt

→

Target-Encoding
990
990
1080
990
1080

Lage
2
3
2
2
3

→

Lage_2	Lage_3
1	0
0	1
1	0
1	0
0	1

Abbildung 6: Target-Encoding und One-Hot-Encoding bei kategorialen Variablen

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die Variablen Temperatur, Jahr und ⌀Bolzen werden nicht kodiert, da sie eine natürliche Ordnung besitzen (z.B. niedrigere/höhere Temperaturen, frühere/spätere Jahreszahlen, kleinere/größere Durchmesser). Eine Kodierung würde die zugrunde liegende Struktur dieser Variablen verfälschen und potenziell relevante Informationen verlieren lassen.

Skalierung: Die Notwendigkeit einer Skalierung der Eingabedaten hängt vom eingesetzten Vorhersagemodell ab. Bei Regularisierungsmodellen wie Ridge Regression und Lasso Regression wird in den meisten Implementierungen eine Skalierung automatisch vorgenommen. Während k-Nearest-Neighbor (KNN) empfindlich auf unterschiedlich skalierte Variablen reagiert [1], bringt eine Skalierung für Modelle wie die Lineare Regression oder baumbasierte Methoden keine Vorteile. Da eine Skalierung bei Modellen, für die sie nicht zwingend erforderlich ist, jedoch keinen Nachteil verursacht, wird in dieser Arbeit ein einheitliches Vorgehen gewählt und die Eingabedaten im Rahmen der Datenaufbereitung vor allen Vorhersagemodellen skaliert.
Da sowohl die Notwendigkeit als auch die Qualität der Skalierung modellabhängig ist - und es folglich grundlegend keine „beste“ Methode gibt (vergleiche dazu [19] von Ahsan et al., wo verschiedene Skalierungsmethoden verglichen werden) - wird in diesem Algorithmus mit der Standardisierung die gängigste Methode zur Skalierung eingesetzt.
Bei der Standardisierung einer Variable $\mathbf{X}_j$ werden die Werte x_{ij} durch den bereits in (8) eingeführten Z-Score ersetzt (da zuvor Ausreißer entfernt wurden, ist die Verwendung des Z-Scores vertretbar):

$$\tilde{x}_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j}. \tag{21}$$

Das daraus resultierende skalierte Merkmal $\tilde{\mathbf{x}}_j$ weist einen Mittelwert von 0 und eine Varianz von 1 auf.
Kategoriale Variablen, die durch One-Hot-Encoding in binäre Merkmale überführt wurden, bleiben von der Standardisierung ausgenommen, da diese dort nicht erforderlich ist. Die Standardisierung erfolgt in Python mit dem Paket **scikit-learn** [20] und ist in Abbildung 7 visualisiert.

Unskaliert
980
1100
960
890
1200

→

Skaliert
-0,417
0,672
-0,599
-1,234
1,579

Abbildung 7: Skalierung mittels Standardisierung

Die grau hinterlegten Felder markieren den optionalen Teil des Algorithmus. Wie bei der Skalierung lassen sich auch diese Optionen per Input steuern.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Imputationstest: Wie bereits beschrieben, ist das Ziel herauszufinden, inwieweit experimentelle Versuche reduziert werden können, ohne dass die Vorhersagegüte signifikant leidet. Darüber hinaus soll die Performance der eingesetzten Imputationsmethoden verglichen werden. Da bei allen Beobachtungen eines konkreten Bauteils dieselben Werte für die Testergebnisse am Testring vorliegen, müssen im Sinne der Praxisnähe - sobald für eine dieser Beobachtungen künstlich fehlende Werte eingefügt werden - die entsprechenden Werte bei sämtlichen Beobachtungen dieses Bauteils entfernt werden. Die untenstehende Abbildung 8 soll dieses Vorgehen verdeutlichen. Blau markiert sind jene Zellen, in denen bereits initial oder durch auftretende Ausreißer die Werte fehlen. Grau hinterlegt ist eine Beobachtung mit einer einfach vorkommenden CutUpID, während die gelb markierten Zellen Beobachtungen mit gemeinsamer CutUpID darstellen. Da es sich um dasselbe Bauteil handelt und der Versuch am Testring identisch ist, müssen entweder bei beiden Beobachtungen die Einträge entfernt werden oder bei keiner.

Fließgrenze	CutUpID	Lage	Fließgr. (1)	A4 (1)	A5 (1)	Z (1)	Zugfest.
1100	700	2	1100	55	45	19.3	1400
1004	700	3	1100	55	45	19.3	1400
953	300	2	900	49		22.4	1359
890	800	3	910	43	43	19.9	1500
1200	900	5	1110		53	20.0	900

Fließgrenze	CutUpID	Lage	Fließgr. (1)	A4 (1)	A5 (1)	Z (1)	Zugfest.
	700	2					
	700	3					
953	300	2	900	49		22.4	1359
890	800	3	910	43	43	19.9	1500
	900	5					

Abbildung 8: Fehlende Werte für einen Imputationstest einstreuen

Weil bei ausgelassenem Testversuch am Testring auch ein Testergebnis an anderen Lagen ausgeschlossen werden kann, muss auch der beobachtete Wert in der Zielvariable entfernt werden. Für diese Fälle sind nur die allgemeinen Testgegebenheiten und die Zuliefererroute bekannt. Auf die Imputation in diesem Fall wird im weiteren Verlauf näher eingegangen.

Imputation: Da initial und durch die vorangegangenen Datenaufbereitungsschritte fehlende Werte im Datensatz vorkommen, muss im Rahmen der Datenaufbereitung eine passende Methode zur Imputation fehlender Werte bestimmt werden. Dieser Schritt erhält im Rahmen dieser Arbeit besondere Aufmerksamkeit.
Verglichen wird eine univariate Methode (Einsetzen des Mittelwerts bei numerischen bzw. der häufigsten Kategorie bei kategorialen Variablen, siehe Abbildung 9) mit einer Imputation auf Basis des k-Nearest-Neighbor-Algorithmus sowie dem MissForest-Verfahren. Beide Imputationsmethoden lassen sich mit dem **scikit-learn**-Paket umsetzen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Num. Variable
1000
953
890
1200

Num. Variable
1000
1010,75
953
890
1200

Kat. Variable
Triple Melt
Double Melt
Triple Melt
Triple Melt

Kat. Variable
Triple Melt
Double Melt
Triple Melt
Triple Melt
Triple Melt

Abbildung 9: Imputation mittels Mittelwert bzw. Häufigste Kategorie. Nach erfolgter Kodierung ist nur der numerische Fall zu beachten

Gewichtung: Die Gewichtung einzelner Beobachtungen für den Fit eines Vorhersagemodells ist ein weiterer optionaler Teil der Pipeline. Da im Zuge eines Imputationstests in einigen Beobachtungen eine hohe Anzahl an fehlenden Werten auftreten kann, während andere vollständig sind, liegt es nahe, die Beobachtungen unterschiedlich zu gewichten. In der Literatur werden Gewichte in verschiedenen Bereichen der Statistik zu unterschiedlichen Zwecken eingesetzt. So findet beispielsweise das Verfahren der gewichteten kleinsten Quadrate (WLS: Weighted least squares) Anwendung, bei dem Beobachtungen mit der inversen Varianz gewichtet werden, um heteroskedastischen Fehlern entgegenzuwirken [62, 63].
Im Zusammenhang mit imputierten Daten und anschließenden Vorhersagen ist der gezielte Einsatz von Gewichten bislang jedoch nur wenig verbreitet. Die vorliegende Arbeit greift daher die Idee der gewichteten kleinsten Quadrate auf, berücksichtigt jedoch nicht die zu schätzende Varianz, sondern die bekannte Anzahl fehlender Werte. Darüber hinaus wird anstelle des Kehrwerts der Anzahl fehlender Werte eine Exponentialfunktion verwendet:

$$w_i = \exp\left(-\alpha \cdot \#\mathrm{NA}_i\right). \tag{22}$$

Dadurch wird verhindert, dass Beobachtungen mit bereits zwei fehlenden Werten nur noch halb so stark einfließen wie eine vollständige Beobachtung. Mithilfe der Exponentialfunktion erfolgt eine weniger drastische Abwertung von Beobachtungen mit fehlenden Werten.

Fließgrenze	Fließgr. (1)	A4 (1)	A5 (1)	Z (1)	Zugfest.	#NA	Gewicht
1100	1100		45	19.3	1400	1	0,905
						6	0,549
953	900	49			1359	2	0,819
	910			19.9	1500	3	0,741
1200	1110	49	53	20.0	900	0	1

Abbildung 10: Gewichte für Beobachtungen auf Basis der Anzahl fehlender Werte

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Dabei ist α der Gewichtungsfaktor, der bestimmt, wie stark fehlende Werte bestraft werden sollen. Für den Modellvergleich wird dieser mit 0,1 angesetzt. Ein höherer Wert von α führt zu geringeren Gewichten und damit zu einem schwächeren Einfluss entsprechender Beobachtungen im Modelltraining. Im weiteren Verlauf kann dafür ein Parametertuning durchgeführt werden.

Die in Abbildung 2 in blau dargestellten Felder betreffen die aus dem Modellvergleich resultierende optimale Modellkonfiguration, also jene Kombination aus Imputationsmethode, Gewichtung und Vorhersagemodell, die für einen bestimmten Imputationstest die besten Ergebnisse liefert.

Parameter: Um die Komplexität und damit auch die Rechenzeit im Modellvergleich gering zu halten, werden zunächst alle Modelle mit ihren Standardparametern (Default-Werten) evaluiert. Erst nachdem eine geeignete Konfiguration für das Problem bestimmt ist, erfolgt eine gezielte Optimierung der Hyperparameter. Damit wird bewusst ein Kompromiss eingegangen: Die geringere Komplexität wird höher gewichtet als die Option, dass ein Modell mit optimalen Parametern bessere Ergebnisse liefert als ein anderes, obwohl es mit den Standardparametern unterlegen ist.

Die pink eingefärbten Felder stellen die Auswahl des Vorhersagemodells dar. Nach erfolgter Datenaufbereitung soll die Zielvariable auf Basis von nun $p = 39$ Merkmalen vorhergesagt werden.

Modelltraining: Für die Vorhersage der Fließgrenze an verschiedenen Lagen werden in diesem Schritt unterschiedliche Modelle trainiert. Folgende sechs Modelle werden berücksichtigt:

Lineare Modelle	**Nichtlineare Modelle**
Lineare Regression	k-Nearest-Neighbor [21]
Ridge Regression [22]	Random Forest [23]
Lasso Regression [24]	XGBoost [25]

Tabelle 3: Vorhersagemodelle

Die Lineare Regression gilt als weniger flexibel und setzt strengere Annahmen voraus, liefert aber dafür Vorteile im Bereich der Komplexität und Interpretierbarkeit. Sie stellt den theoretischen Grundbaustein vieler Verfahren dar und liefert robuste Schätzungen, wenn die Beziehung zwischen Prädiktoren und Zielvariable annähernd linear ist.
Eine Erweiterung zum linearen Regressionsmodell sind die beiden Regularisierungsverfahren Ridge Regression und Lasso Regression. Diese bieten den Vorteil, dass wichtigere Prädiktoren bevorzugt und weniger relevante Variablen abgeschwächt werden. Insbesondere bei hochdimensionalen Daten ist das vorteilhaft.
Der k-Nearest-Neighbor (KNN) Algorithmus ist ein flexibles Verfahren, bei dem für die Vorhersage eines Datenpunkts nach ähnlichen Beobachtungen in den Trainingsdaten gesucht wird, um den Zusammenhang zwischen den erklärenden Variablen und der Zielvariable zu beschreiben.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die beiden Ensemble-Methoden Random Forest und XGBoost kombinieren einzelne Entscheidungsbäume und ermöglichen eine nichtlineare Modellierung. Zwar liefern sie in der Regel eine hohe Prognosegenauigkeit, dies wird jedoch auf Kosten der Interpretierbarkeit realisiert.
Bei der Wahl des Modells sollten neben der Vorhersagegüte auch die Interpretierbarkeit und der Rechenaufwand einfließen. In Python lassen sich diese Modelle mit den Paketen **scikit-learn** und **xgboost** [25] realisieren.

Vorhersage: Nach dem Training des Modells wird dieses zur Vorhersage der Fließgrenze herangezogen. Basis dafür sind die erklärenden Variablen, gesammelt in den Testdaten. Wie der Datensatz in Trainings- und Testdaten aufgeteilt wird, wird im nächsten Unterpunkt beschrieben.

Validierung: Zur Abschätzung der Modellgüte sowie zur Optimierung der Modellparameter wird die weit verbreitete k-Fold Cross-Validation eingesetzt. Grundsätzlich dient Cross-Validation zur Schätzung des unbekannten Testfehlers. Der Testfehler ist der durchschnittliche Fehler, der beim Einsatz des trainierten Modells auf zuvor unbeobachtete Daten entsteht. Dieser liegt in der Regel über dem Trainingsfehler (jener Fehler bei Anwendung des Modells auf das Trainingsset).
In verschiedenen Lehrbüchern [9, 10, 26] werden drei Ansätze zur Cross-Validation unterschieden. Grundlage bietet der Validierungsset-Ansatz, bei dem der Datensatz in zwei Teilmengen unterteilt wird. Auf der einen Teilmenge, dem Trainingsset, wird das Modell gefittet und mithilfe des Testsets wird ein Schätzer für den Testfehler bestimmt. Dieser Ansatz ist zwar leicht umzusetzen, weist jedoch zwei Nachteile auf: Erstens hat der Schätzer für den Testfehler eine hohe Varianz, da er stark von der konkreten Auswahl der Teilmengen abhängt. Zweitens wird der Testfehler tendenziell überschätzt, da das Modell nur auf einem Teil der verfügbaren Daten trainiert wird.
Entsprechend empfehlen sowohl Molinaro et al. in [27] als auch Hawkins et al. in [28], vom Validierungsset-Ansatz abzusehen. Eine Erweiterung stellt die k-Fold Cross-Validation dar, die darauf abzielt, die genannten Nachteile des Validierungsset-Ansatz zu vermeiden. Hierzu wird der Datensatz in k annähernd gleich große Teilmengen, sogenannte Folds, unterteilt. Einer der k Folds wird als Testset definiert, während auf den restlichen k-1 Folds das Modell trainiert wird. Dieses Vorgehen wird k-mal wiederholt, sodass jede Teilmenge genau einmal als Testset fungiert. Dadurch wird bei jeder Wiederholung eine andere Gruppe von Beobachtungen zur Evaluation herangezogen und eine Fehlermaßzahl bestimmt. Übliche Fehlermetriken in diesem Zusammenhang sind:

Der Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{23}$$

und der Mean Absolute Error

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|\,. \tag{24}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Der geschätzte Testfehler ergibt sich schließlich als Durchschnitt der k ermittelten Fehler (MSE oder MAE):

$$\mathrm{CV}^{(k)} = \frac{1}{k} \sum_{i=1}^{k} \mathrm{Error}_i. \tag{25}$$

Ein Spezialfall der k-Fold Cross-Validation ist die Leave-One-Out Cross-Validation (LOOCV) mit $k = n$. Dabei bildet jede einzelne Beobachtung einen Fold, wodurch n Modelle entstehen. Aufgrund dessen ist die k-Fold Cross-Validation mit $k < n$ rechnerisch effizienter. Zudem unterscheiden sich die k Trainingssets stärker voneinander als bei LOOCV, was bei kleineren k zu einer geringeren Varianz der Schätzung führt. Grundsätzlich gilt: Je größer das Trainingsset ist, desto geringer der Bias. Umgekehrt ist die Varianz des Schätzers gering, wenn sich die zugrunde liegenden Trainingssets stark voneinander unterscheiden. Die k-Fold Cross-Validation weist somit einen geringeren Bias als der Validierungsset-Ansatz und eine geringere Varianz im Vergleich zur Leave-One-Out Cross-Validation auf. Diese Überlegungen spielen auch bei der Wahl der Anzahl der Folds k eine Rolle. Es besteht ein „Bias-Varianz-Kompromiss“, der von Kohavi in [29] ausführlich analysiert wird.

In der Literatur gibt es grundsätzlich keine Regelung für die Wahl von k. Molinaro zeigt in [27], dass LOOCV und 10-Fold Cross-Validation ähnliche Resultate liefern. Aufgrund des geringeren Rechenaufwands ist die 10-Fold Cross-Validation in der Regel meist die bevorzugte Methode. Typischerweise wird in der Praxis $k = 5$ oder $k = 10$ angesetzt. Bei der Wahl von $k = 5$ wird der geringere Rechenaufwand gegenüber dem im Vergleich zu $k = 10$ höheren Bias in Kauf genommen. Gemäß [27] gleichen sich die Unterschiede der Verfahren bei größeren Stichproben weitgehend aus, was die Wahl von $k = 5$ rechtfertigt.

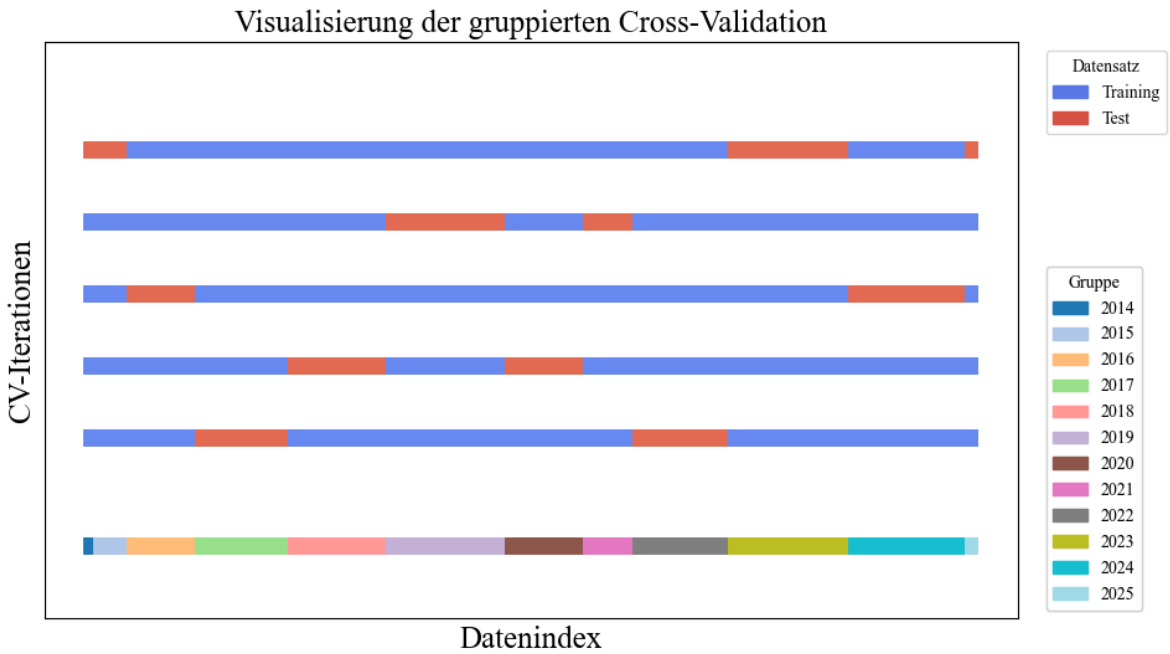

Abbildung 11: Nach dem Jahr gruppierte 5-Fold Cross-Validation

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Ein wichtiger Aspekt bei der korrekten Umsetzung der k-Fold Cross-Validation ist das Vermeiden von Informationsleakage, also dem Einbezug eigentlich unbekannter Informationen. Nun ist es in der Praxis üblich, Vorhersagen für neue Bauteile treffen zu müssen. In diesem Kontext würde das bedeuten, dass die Fließgrenze von einer Beobachtung mit bisher unbekannter Prozessspezifikation vorhergesagt werden soll.
In Abbildung 11 ist die Aufteilung in die 5 Folds grafisch dargestellt. Da eine Prozessspezifikation stets nur innerhalb eines Jahres auftritt, erfolgt die Aufteilung der Daten gruppiert gemäß der Variable Jahr. Man erkennt, dass alle Beobachtungen eines Jahres stets entweder vollständig im Trainingsset oder vollständig im Testset enthalten sind. Wie bereits erläutert, ist bei der Vorhersage der Fließgrenze keine zeitliche Abhängigkeit zu erwarten, sodass dieses Vorgehen als unproblematisch betrachtet werden kann.

Bei den zuvor beschriebenen Datenaufbereitungsschritten sind einige methodische Aspekte zu beachten, um Informationsleakage zu vermeiden:
Jeder Schritt der Datenaufbereitung muss zunächst auf den Trainingsdaten trainiert und anschließend auf die Testdaten transformiert werden. Dabei dürfen Informationen aus dem Testset nicht in das Training einfließen. So wird bei der Ausreißeranalyse sowohl beim Trainingsset als auch im Testset der modifizierte Z-Score mit dem Median und dem MAD auf Basis der Trainingsdaten bestimmt. Die Verwendung des Medians des gesamten Datensatzes würde hingegen Informationen einbeziehen, die eigentlich nicht verfügbar sind.
Beim Kodieren der kategorialen Variablen muss analog vorgegangen werden. Beim Target-Encoding werden die Mittelwerte auf Basis der Zielwerte in den Trainingsdaten ermittelt. Da die Variable Prozessspezifikation in den Testdaten neue, im Training nicht beobachtete Kategorien enthält, gibt es zwei Möglichkeiten: Entweder wird der globale Mittelwert der Trainingsdaten für alle neuen Kategorien angesetzt, oder die entsprechenden Zellen werden als fehlend behandelt und später imputiert. Bei Anwendung der simplen Imputationsmethode, bei der lediglich der Mittelwert der bekannten Einträge der Variable für die fehlenden Werte verwendet wird, sind beide Optionen äquivalent. Aus diesem Grund wird die Option mit den fehlenden Werten anstelle des globalen Mittelwerts vorgezogen.
Die Skalierung der Features wird ebenfalls auf den Trainingsdaten trainiert und auf Basis dessen auf den Testdaten umgesetzt.
Der bereits erwähnte Imputationstest, bei dem ein beliebiger Anteil der Daten entfernt wird, wird sowohl auf dem Trainingsset als auch auf dem Testset durchgeführt. Das praxisbedingt notwendige Entfernen der Zielwerte im Trainingsset ist grundsätzlich unüblich und würde eigentlich bedeuten, diese Beobachtungen vom Modelltraining auszuschließen. Da jedoch die verbleibenden Informationen für das Modelltraining genutzt werden sollen und gleichzeitig die Imputationsmethoden evaluiert werden, werden diese Beobachtungen nicht entfernt. Daraus folgt aber, dass Überlegungen angestellt werden müssen, wie man die fehlenden Werte der Zielvariable imputiert.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Algorithmus 1: Imputation fehlender Zielwerte mittels Gruppierung

Input: Zielvariable $\mathbf{y} \in \mathbb{R}^n$ mit fehlenden Werten, Regressormatrix $\mathbf{X}$,
Schwellenwert $q \in \mathbb{N}$ (Default: $q = 5$)
Output: Vektor $\hat{\mathbf{y}}$ mit imputierten Zielwerten

```
foreach i ∈ {1, ..., n} mit y_i = NA do
    Bestimme Gruppe G_1(i) mit gleicher (Material, Temperatur, Zulieferer)
    if |G_1(i)| ≥ q then
        Setze ŷ_i := ȳ_{G_1(i)}
    else
        Bestimme Gruppe G_2(i) mit gleicher (Material, Temperatur)
        if |G_2(i)| ≥ q then
            Setze ŷ_i := ȳ_{G_2(i)}
        else
            Bestimme Gruppe G_3(i) mit gleichem Material
            if |G_3(i)| ≥ q then
                Setze ŷ_i := ȳ_{G_3(i)}
            else
                Setze ŷ_i := ȳ
            end
        end
    end
end
foreach i mit y_i ≠ NA do
    Setze ŷ_i := y_i
end
return ŷ
```

Für die Imputation fehlender Zielwerte wird das Trainingsset nach den Variablen Material, Temperatur und Zulieferer gruppiert, da angenommen wird, dass die Zuliefererroute in Kombination mit dem Material und der Versuchstemperatur am stärksten mit der Fließgrenze korreliert. Innerhalb dieser Gruppen wird der Mittelwert gebildet und als Imputationswert verwendet. Da innerhalb einzelner Gruppen nur wenige Beobachtungen auftreten können, wird ein Schwellenwert von $q = 5$ eingeführt. Liegt die Anzahl an Beobachtungen in einer Gruppe unterhalb des Schwellenwerts, erfolgt eine gröbere Gruppierung: Es wird der Mittelwert der Gruppe Material und Temperatur verwendet. Ist auch diese Gruppe zu klein, wird der Mittelwert innerhalb des jeweiligen Materials herangezogen. Das in Algorithmus 1 dargestellte Vorgehen bezieht erklärende Variablen in die Imputation der Zielwerte ein und stellt mithilfe des Schwellenwerts sicher, dass keine einzelnen Beobachtungen einen Imputationswert bestimmen, wodurch Overfitting vermieden wird. Analog zu den vorherigen Datenaufbereitungsschritten wird diese Prozedur auf den Trainingsdaten trainiert und anschließend auf den Testdaten angewendet. Dieses Training-Test-Vorgehen wird ebenso bei der Imputation der erklärenden Variablen angewendet.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Importance: Um Vermutungen zur Relevanz einzelner Variablen zu überprüfen, wird nach erfolgtem Modellvergleich und der Auswahl des besten Modells untersucht, wie sich die Wichtigkeit der Features bestimmen lässt. Für Random Forest und XGBoost liefern die Implementierungen in **scikit-learn** und **xgboost** bereits standardmäßig Ausgaben für die Feature Importance. Bei linearen Modellen erfolgt die Einschätzung hingegen über die Koeffizienten. Für das k-Nearest-Neighbors-Modell besteht hingegen keine Möglichkeit, die Bedeutung einzelner erklärender Variablen aufzuzeigen.

Intervalle: Ziel der gesamten Modellpipeline ist es nicht nur, den Umgang mit fehlenden Werten zu analysieren und Zusammenhänge zwischen Zielvariable und beobachteten Versuchsergebnissen zu modellieren, sondern diese datenbasierten Erkenntnisse auch in einen Entscheidungsprozess zu überführen. Zu diesem Zweck werden mithilfe des Bootstrap-Verfahrens Intervalle um die Vorhersagen gebildet. Da eine Fließgrenze oberhalb der oberen Intervallgrenze in der Praxis unkritisch ist, liegt der Fokus auf dem Anteil der Beobachtungen, deren wahrer Zielwert unterhalb der unteren Intervallgrenze liegt.
Für jede Kombination aus Material und Temperatur gibt es in der Praxis einen minimalen Wert für die Fließgrenze, der bisher zur Bestimmung von akzeptablen Bauteilen herangezogen wurde. Anhand der Vorhersage und des zugehörigen Intervalls kann überprüft werden, ob die Fließgrenze des Bauteils an der betrachteten Testlage den Mindestanforderungen genügt oder ob eine nähere Untersuchung bzw. eine Ablehnung des Bauteils sinnvoll wäre.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

4 Imputation fehlender Werte

Little und Rubin definieren fehlende Werte in [30] als nicht beobachtete Werte, die für die Analyse bedeutsam wären, wenn sie vorlägen; mit anderen Worten: Ein fehlender Wert verbirgt einen sinnvollen Wert.

Das Auftreten fehlender Werte ist ein weit verbreitetes Problem, das in nahezu allen wissenschaftlichen Bereichen auftritt. Beispiele hierfür sind die medizinische Forschung, etwa bei Labordaten [31], Patientendiagnosen und der zugehörigen Behandlungshistorie [32] aber auch elektronischen Gesundheitsakten [33]. Auch innerhalb der Sozialforschung [34], beispielsweise bei gesellschaftlichen Erhebungen im Bildungsbereich [35, 36], sind Nichtantworten keine Ausnahme. Darüber hinaus treten fehlende Werte auch in der biomedizinischen Forschung [37] und den Materialwissenschaften, beispielsweise bei dem für diese Arbeit vorliegenden Datensatz, regelmäßig auf. Insbesondere bei großen Datensätzen ist die Behandlung fehlender Werte meist unumgänglich.

Die Gründe für fehlende Werte sind vielseitig und umfassen unter anderem menschliche Fehler bei der Datenverarbeitung, das Ablehnen von Antworten oder technische Defekte, etwa durch fehlerhafte Messgeräte [33]. Fehlende Werte können jedoch auch gezielt entstehen, so etwa im vorliegenden Datensatz, bei dem in vielen Fällen entweder die Bruchdehnung A4 oder die Bruchdehnung A5 erhoben wurde. Darüber hinaus werden fehlende Werte auch bewusst bei der Behandlung von Ausreißern und kategorialen Variablen oder im Zuge des Imputationstests zur Validierung der Imputationsmethoden in Kauf genommen. In anderen Fällen beruhen fehlende Daten im vorliegenden Datensatz auf menschlichen Versäumnissen, beispielsweise wenn im Rahmen des Zugversuchs das Testdatum nicht eingetragen wurde, wodurch diese Angabe infolgedessen fehlt.

Nachdem das Ignorieren von fehlenden Werten in den meisten Fällen aufgrund der Anforderungen von Algorithmen des maschinellen Lernens keine Option für die weitere statistische Analyse darstellt, müssten Beobachtungen mit fehlenden Werten entfernt werden, sofern man diese im Vorfeld nicht behandeln möchte. Dieser Prozess wird in der Literatur häufig als „Listwise Deletion“ (Listenlöschung) bezeichnet und stellt nur selten eine optimale Strategie dar. Durch das Entfernen unvollständiger Beobachtungen gehen potenziell wertvolle Informationen verloren, weshalb die Listwise Deletion zum Verlust statistischer Aussagekraft führt [38]. Darüber hinaus wird für die Listwise Deletion vorausgesetzt, dass das Fehlen von Werten im Datensatz zufällig ist. Ist diese Annahme verletzt, repräsentieren die verbleibenden Beobachtungen nicht mehr die Grundgesamtheit, was infolgedessen zu verzerrten Schätzungen und fehlerhaften Schlussfolgerungen führt [30]. Dementsprechend sind Methoden, die zur Löschung einzelner Beobachtungen führen, häufig im Zuge der Datenaufbereitung nicht empfohlen. Aus diesem Grund wird die Verwendung von Imputationsverfahren empfohlen, bei denen gezielt versucht wird, fehlende Werte auf Basis der im Datensatz vorkommenden Zusammenhänge zu schätzen und adäquat zu ersetzen. Ein gutes Imputationsverfahren soll dabei nicht nur die bestmöglichen Vorhersagen liefern, sondern auch Verzerrung vermeiden, wie sie bei einer statistischen Analyse ausschließlich vollständiger Beobachtungen entstehen würden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Die theoretischen Grundlagen zur Behandlung fehlender Werte wurden maßgeblich von Rubin in [39, 40, 41] sowie gemeinsam mit Little in [30] gelegt. In diesen Arbeiten wird dem Muster sowie dem Mechanismus fehlender Daten zentrale Bedeutung für die Aussagekraft der Datenanalyse eingeräumt. Das Muster beschreibt die strukturelle Verteilung fehlender Werte im Datensatz und soll Auskunft darüber geben, welche Werte beobachtet wurden und welche fehlen. Der Mechanismus von fehlenden Werten hingegen bezieht sich auf den Grund des Fehlens und beschreibt, in welcher Weise das Fehlen von Daten mit beobachteten oder unbeobachteten Datenwerten zusammenhängt. Sowohl das zugrunde liegende Muster als auch der Mechanismus fehlender Werte sind entscheidend für die Behandlung fehlender Werte und die dafür verwendete Imputationsmethode. Insofern besteht die initiale Herausforderung darin, die Struktur und die Ursachen für das Fehlen von Daten zu identifizieren [34].

4.1 Muster und Mechanismen fehlender Werte

Für die Definition der unterschiedlichen Muster und Mechanismen ist es zunächst notwendig, die für die theoretische Beschreibung fehlender Werte notwendige Notation einzuführen. Dabei folgt diese Arbeit im Allgemeinen jener Beschreibung von Rubin und Little aus [30]. Da jedoch in dieser und in weiteren Arbeiten die (vollständige) Datenmatrix mit $\mathbf{Y}$ notiert wird, womit bereits in Abschnitt 2.2 in (1) die Zielvariable eingeführt wurde, wird die Notation geringfügig adaptiert, um Verwechslungen zu vermeiden. Stattdessen wird für die im Kontext der Theorie fehlender Werte betrachtete Datenmatrix die Bezeichnung $\mathbf{X}$ verwendet. Nachdem sich die nachfolgenden Imputationsmethoden ausschließlich auf die, zum Teil unvollständigen, erklärenden Variablen beziehen, führt dies zu keiner Kollision mit der bereits eingeführten Notation. Lediglich der Begriff „Regressormatrix“ für $\mathbf{X}$ wird im Rahmen der Analyse von fehlenden Werten nicht weiter verwendet und im Folgenden durch „Datenmatrix“ ersetzt. Während Rubin in [42] sowie Schafer und Graham in [43] zwischen vollständig beobachteten und unvollständigen Variablen ($\mathbf{X}_j$ vs. $\mathbf{Y}_j$) unterscheiden, folgt die Vorgehensweise hier jener von Stef van Buuren [44], der diese in einer gemeinsamen Datenmatrix $\mathbf{X}$ zusammenfasst . Dadurch enthält $\mathbf{X}$ sowohl unvollständige als auch vollständig beobachtete Variablen.
Seien x_{ij} für $i = 1, \ldots, n$ und $j = 1, \ldots, p$ die Elemente der Datenmatrix $\mathbf{X}$, dann ist die Indikatormatrix für fehlende Daten durch

$$\mathbf{M} = (m_{ij}) \quad \text{mit } i = 1, \ldots, n;\ j = 1, \ldots, p \tag{26}$$

definiert. Wobei

$$m_{ij} := \begin{cases} 0, & \text{wenn } x_{ij} \text{ beobachtet ist} \\ 1, & \text{wenn } x_{ij} \text{ fehlt} \end{cases} \tag{27}$$

gilt.

Rubin und Litte führen in [30] sechs unterschiedliche Muster fehlender Werte ein. In weiterer Literatur, etwa bei Schafer und Graham in [43] oder bei van Buuren in [44] werden diese Muster zu drei beziehungsweise vier übergeordneten Kategorien - univariat, monoton sowie entweder nicht-monoton oder als File matching und allgemein - zusammengefasst. In Abbildung 12 sind jene Muster dargestellt, die für den vorliegenden Datensatz in Frage kommen, ergänzt mit zwei häufig vorkommenden Mustern.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

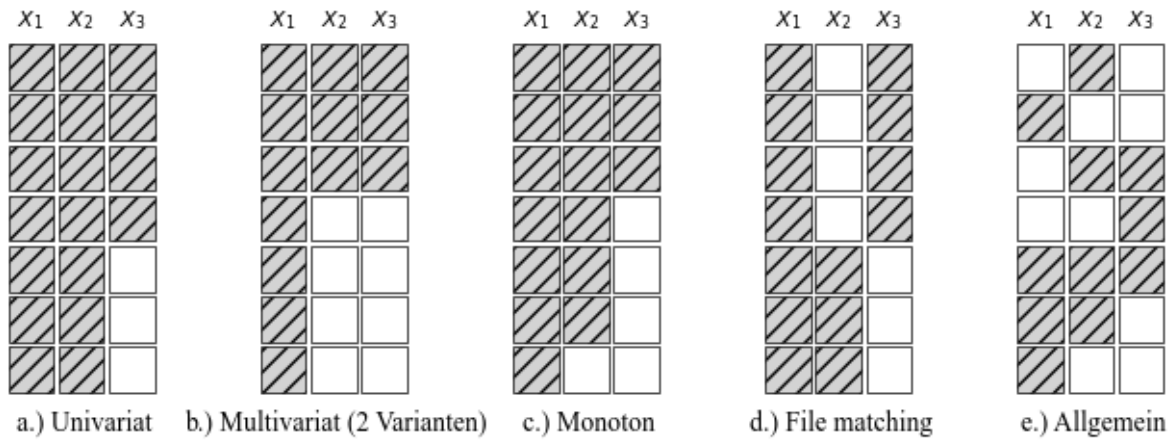

Abbildung 12: Mögliche Muster fehlender Daten, die in einem praxisnahen Datensatz vorkommen können.
Angelehnt an Little und Rubin, *Statistical Analysis with Missing Data* [30]

Univariat: Ein univariates Muster liegt vor, wenn fehlende Werte, wie in Abbildung 12a dargestellt, in nur einer einzigen Variable vorkommen. Es ist das einfachste Muster, in dem fehlende Werte vorkommen können. Also für alle $i \in \{1, \ldots, n\}$ und alle $j \in \{1, \ldots, p\} \setminus \{j^*\}$ gilt:

$$m_{ij} = 0, \quad \text{aber es gibt } i \text{ sodass } m_{ij^*} = 1 \text{ gilt} \tag{28}$$

Multivariat (2 Varianten): Dieser Fall tritt auf, wenn mehrere Variablen gleichzeitig entweder alle beobachtet oder alle fehlend sind. In Abbildung 12b ist $\mathbf{x}_1$ vollständig beobachtet, während $\mathbf{x}_2$ und $\mathbf{x}_3$ in denselben Beobachtungen fehlende Werte haben. Es existiert also eine Partition der Beobachtungen:

$$\{1, \ldots, n\} = A \cup B \quad \text{mit} \quad A \cap B = \emptyset, \tag{29}$$

sodass für alle $i \in A$

$$m_{ij} = 0 \quad \forall j \in \{1, \ldots, p\} \tag{30}$$

gilt. Und für alle $i \in B$

$$m_{ij} = 1 \quad \forall j \in J \subseteq \{1, \ldots, p\}, \quad \text{mit } |J| \geq 1 \tag{31}$$

gilt.

Monton: Man spricht von einem monotonen Muster (Abbildung 12c), wenn die Variablen so angeordnet werden können, dass bei einem fehlenden Wert einer Variable für diese Beobachtung die Werte der darauffolgenden Variablen ebenfalls fehlen. Also, wenn für alle $i \in \{1, \ldots, n\}$

$$\text{wenn } m_{ij} = 1 \text{ für ein } j \in \{1, \ldots, p-1\}, \text{ dann gilt: } m_{i,j'} = 1 \quad \forall j' > j. \tag{32}$$

gilt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

File matching: Das in Abbildung 12d dargestellte File-matching-Muster liegt vor, wenn bestimmte Variablenpaare niemals gemeinsam für dieselbe Beobachtung vorhanden sind. In diesem Fall existieren zwei disjunkte Mengen von Beobachtungen $A, B \subseteq \{1, \ldots, n\}$ mit $A \cap B = \emptyset$, sodass:

$$\begin{aligned} \text{Für alle } i \in A: \quad m_{ij} = 1 \quad \forall j \in J_2 \\ \text{Für alle } i \in B: \quad m_{ij} = 1 \quad \forall j \in J_1 \end{aligned} \tag{33}$$

mit $J_1, J_2 \subseteq \{1, \ldots, p\}$, $J_1 \cap J_2 = \emptyset$.

Allgemein: Das allgemeine Muster ist das häufigste und zugleich komplexeste Muster in Datensätzen. Das Fehlen von Werten einer Variable impliziert dabei nicht notwendigerweise das Fehlen von Werten anderer Variablen, wodurch ein willkürliches Muster, wie in Abbildung 12e visualisiert, entsteht.

Die vorliegende Struktur der fehlenden Werte beeinflusst maßgeblich die Effektivität der verwendeten Imputationsmethode. Generell gilt, dass je komplexer und unregelmäßiger das Fehlmuster ist, desto flexibler sollte die Methode sein [30]. Im vorliegenden Datensatz kommen grundsätzlich zwei der fünf dargestellten Fehlmuster vor. Wie bereits beschrieben, fehlen die Werte einer Beobachtung regelmäßig entweder für die Variable Bruchdehnung A4 oder Bruchdehnung A5, da in der Praxis oft nur einer der beiden Kennwerte erfasst wird. Insofern kann man von einem File-matching-Muster sprechen. Die im Rahmen des Imputationstests eingestreuten fehlenden Werte folgen gemäß der im vorherigen Kapitel vorgestellten Konstruktion einem multivariaten Muster mit zwei Varianten. Die Beobachtungen, in denen der Imputationstest durchgeführt wird, sind gemäß obiger Notation aus der Menge B und die übrigen Beobachtungen aus der Menge A. Die Kombination der beiden Muster mit den initial fehlenden Werten sowie jenen, die im Zuge der Ausreißeranalyse und der Kodierung kategorialer Variablen entstanden sind, ergibt insgesamt ein allgemeines Muster. Daher ist anzunehmen, dass flexiblere Methoden bessere Imputationsergebnisse liefern.

Neben dem Muster der fehlenden Werte hängt die Performance einer Imputationsmethode auch stark vom Mechanismus der fehlenden Werte ab [45, 34].
Rubin hat in [39] drei unterschiedliche Mechanismen für fehlende Daten eingeführt. Wie in [44] wird mit $\mathbf{X}^{(\text{OBS})}$ die beobachteten und mit $\mathbf{X}^{(\text{MIS})}$ die fehlenden Daten bezeichnet. Dabei wird einer Notation in Matrixform gefolgt, die ebenso von Schafer und Graham in [43] verwendet wird, da diese eine bessere Übersichtlichkeit bietet. Insgesamt ergibt sich die Datenmatrix als $\mathbf{X} = (\mathbf{X}^{(\text{OBS})}, \mathbf{X}^{(\text{MIS})})$.

Angenommen ϕ enthält unbekannte Parameter, so lässt sich der Mechanismus fehlender Daten durch die bedingte Wahrscheinlichkeit $\mathbb{P}(\mathbf{M} \,|\, \mathbf{X}^{(\text{OBS})}, \mathbf{X}^{(\text{MIS})}, \phi)$ charakterisieren.

Missing completely at random (MCAR): Daten sind MCAR, wenn die Wahrscheinlichkeit, dass ein Wert fehlt, für alle Beobachtungen gleich ist. Die Wahrscheinlichkeit des Fehlens ist damit unabhängig von den Daten, fehlend oder beobachtet, und hängt nur von ϕ ab:

$$\mathbb{P}(\mathbf{M} = 1 \,|\, \mathbf{X}^{(\text{OBS})}, \mathbf{X}^{(\text{MIS})}, \phi) = \mathbb{P}(\mathbf{M} = 1 \,|\, \phi). \tag{34}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

In dem vorliegenden Datensatz wäre das beispielsweise der Fall, wenn die Werte für die Variable Fließgrenze Lage 1 aufgrund eines Ausfalls der Zugprüfmaschine - völlig unabhängig von der Testumgebung (Temperatur, CutUpID, etc.) - fehlen. Die Wahrscheinlichkeit, dass der Wert für die Fließgrenze am Testring fehlt, ist dann für alle Beobachtungen gleich.

Missing at random (MAR): Eine weniger restriktive Annahme als MCAR besteht darin, dass das Fehlen eines Wertes von den beobachteten Daten abhängt:

$$\mathbb{P}(\mathbf{M} = 1 \,|\, \mathbf{X}^{(\mathrm{OBS})}, \mathbf{X}^{(\mathrm{MIS})}, \phi) = \mathbb{P}(\mathbf{M} = 1 \,|\, \mathbf{X}^{(\mathrm{OBS})}, \phi). \tag{35}$$

In dem Kontext der Fließgrenzenbestimmung wäre das der Fall, wenn Messausfälle der Fließgrenze abhängig von anderen beobachteten Werten sind. Beispielsweise, wenn höhere Versuchstemperaturen zu einer höheren Wahrscheinlichkeit von Messausfällen führen. Gemäß der Konstruktion des Imputationstests, bei dem fehlende Werte gemäß der CutUpID (also die Kombination aus Serialnummer und Temperatur) zufällig eingestreut werden, hängt die Wahrscheinlichkeit des Fehlens von anderen (beobachteten) Daten ab.

Missing not at random (MAR): Wenn die Wahrscheinlichkeit für das Fehlen von Daten abhängig von unbeobachteten Daten selbst ist, wird von Missing not at random (MNAR) gesprochen. Das ist der Fall, wenn die obige Annahme verletzt ist:

$$\mathbb{P}(\mathbf{M} = 1 \,|\, \mathbf{X}^{(\mathrm{OBS})}, \mathbf{X}^{(\mathrm{MIS})}, \phi) \neq \mathbb{P}(\mathbf{M} = 1 \,|\, \mathbf{X}^{(\mathrm{OBS})}, \phi). \tag{36}$$

Dieser Fall liegt vor, wenn niedrige Fließgrenzenwerte aus Qualitätsgründen nicht dokumentiert werden. Also ist die Wahrscheinlichkeit hoch, dass ein Wert fehlt, wenn der tatsächliche (aber im Datensatz fehlende) Wert niedrig ist.

Der zugrunde liegende Mechanismus fehlender Werte kann die Performance der Imputationsverfahren entscheidend beeinflussen. Tang und Ishwaran zeigen in einer Analyse baumbasierter Imputationsmethoden [46], dass die Genauigkeit beim Übergang von MCAR auf MAR und MNAR systematisch sinkt. Während bei MCAR die Verteilung der beobachteten Daten und der fehlenden Daten nahezu identisch ist, können bei MAR-Daten Unterschiede in den Verteilungen auftreten. Diese Differenz ist aber im Gegensatz zu MNAR-Daten zum einen geringer und zum anderen wird diese bei MAR-Daten durch die beobachteten Daten erklärt. Infolgedessen sagt man, dass der Mechanismus „ignorierbar“ ist, wenn zumindest MAR gilt [44, 43]. Fehlende Werte, die nicht zufällig fehlen, werden auch als „nicht ignorierbar“ bezeichnet. Das spiegelt sich in der Tatsache wider, dass MNAR den komplexesten Mechanismus darstellt. Dadurch, dass das Fehlen der Daten mit dem fehlenden Wert selbst zusammenhängt, kann eine systematische Verzerrung der Verteilungen selbst nach einer Imputation vorliegen [47], was sich in weiterer Folge negativ auf Vorhersagen auswirkt. Die einfachsten Imputationsmethoden setzen einen MCAR-Mechanismus voraus [44], während der Großteil der Methoden auf MAR-Daten ausgelegt ist und vielversprechende Ergebnisse erzielt [45].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

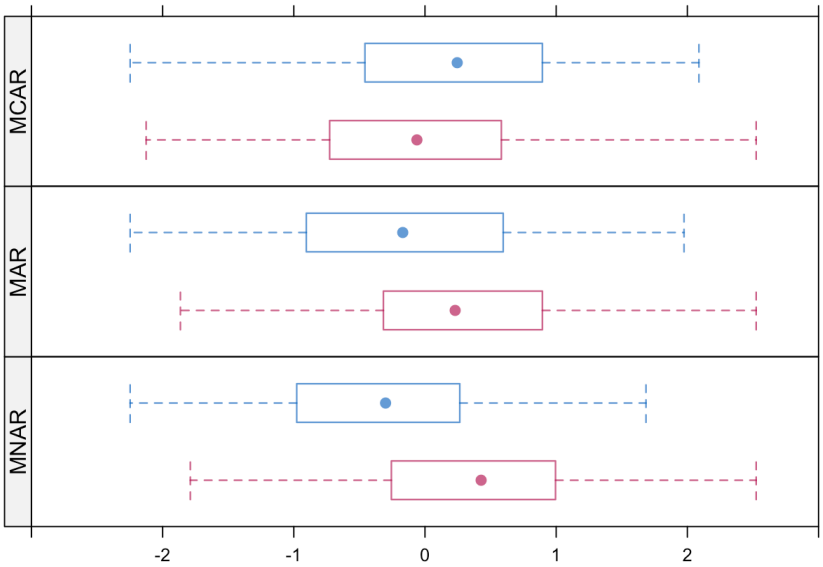

Abbildung 13: Unterschiede in den Verteilungen beobachteter (blau) und fehlender (rot) Daten bei unterschiedlichen Mechanismen fehlender Werte.
Quelle: van Buuren, *Flexible Imputation of Missing Data* [44]

Van Buuren hat diese Unterschiede in Abbildung 13 visualisiert. Dabei ist die Verteilung von $\mathbf{X}^{(\mathrm{OBS})}$ in Blau und die Verteilung von $\mathbf{X}^{(\mathrm{MIS})}$ in Rot dargestellt. Wie beschrieben wird der Unterschied zwischen den Verteilungen von MCAR zu MAR und MNAR immer signifikanter, was die sinkende Performance der Imputationsmethoden erklärt.
Eine Bestimmung, welcher Mechanismus vorliegt, ist allerdings nur schwer durchführbar. Für die Unterscheidung zwischen MCAR und MAR hat Little in [48] einen Test, der heutzutage weitläufig als „Little's MCAR Test“ bekannt ist, vorgestellt. Dieser und weitere Tests werden aber aufgrund niedriger praktischer Relevanz in der empirischen Anwendung kaum eingesetzt [44]. Ein Test, der bestimmen soll, ob ein MAR oder ein MNAR Mechanismus vorliegt, ist aufgrund von fehlenden Informationen, die für so einen Test notwendig wären, nicht möglich [49].

Aufgrund dessen wird, auch ohne statistischen Nachweis, versucht, die Annahme zu stützen, dass der vorliegende Datensatz zumindest dem MAR-Mechanismus unterliegt und somit die Voraussetzung der meisten Imputationsmethoden erfüllt:
Die initial fehlenden Werte in den Variablen Bruchdehnung A4 und Bruchdehnung A5 folgen einem MAR-Mechanismus, da das Fehlen in einer Variable vom Wert der jeweils anderen Variable abhängt. Die anfänglich fehlenden Werte in den weiteren Variablen (Jahr, Triebwerksprogramm und Schmelzmethode - siehe Tabelle 2) können sogar als MCAR angenommen werden. Diese basieren auf menschlichen Fehlern (unvollständige Dokumentation) und nicht auf den eigentlichen Werten, wie es bei MNAR der Fall wäre.
Die im Zuge der Ausreißerbehandlung und der Kodierung kategorialer Variablen entstandenen fehlenden Werte könnten jedoch die Annahme eines MAR-Mechanismus verletzen, da das Fehlen, insbesondere bei der Ausreißerbehandlung, direkt auf den jeweiligen Werten basiert. Zwei Faktoren können dieses Problem abschwächen. Wenn es sich beim entfernten Wert tatsächlich um einen Ausreißer handelt, dann unterscheiden sich zwar die Verteilung der fehlenden Werte und der beobachteten Werte, doch in diesem Fall ist es so-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

wieso das Ziel, diese Differenz aus dem Datensatz zu nehmen und die ausreißenden Werte gemäß der Verteilung beobachteter Daten zu ersetzen. Weiters liegt eine hohe Korrelation zwischen den Variablen innerhalb des Datensatzes vor, wodurch das Fehlen auch durch andere Variablen beschrieben werden kann. Insbesondere baumbasierte Imputationsmethoden profitieren von einer hohen Korrelation [46].
Beim Imputationstest wird ein bestimmter Anteil fehlender Werte zufällig in den Variablen, die Informationen über den Testring enthalten, entfernt. Innerhalb einer CutUpID werden alle Daten entfernt, da sonst - wie bereits im vorangegangenen Abschnitt beschrieben - Informationen über einen Zugversuch am Testring für eine Beobachtung entfernt, aber für eine andere beobachtet werden. Der Grund des Fehlens hängt also von beobachteten Informationen wie der Serialnummer und der Temperatur ab. Durch die durchgeführte Ausreißeranalyse und die anschließende Entfernung der identifizierten Ausreißer können in der Variable Temperatur fehlende Werte entstehen. In der Konstruktion für den Imputationstest werden diese als eigene Kategorie gewertet, weshalb man bei den fehlenden Werten im Zuge des Imputationstests einen MAR-Mechanismus annehmen kann.
Insgesamt lässt sich die Annahme, dass der Datensatz einem MAR-Mechanismus unterliegt, wohl rechtfertigen, was für die Anwendung der Imputationsmethoden relevant ist.

Neben der Art und Weise, wie fehlende Daten auftreten (Muster) und der Gründe für das Fehlen (Mechanismus) spielt auch der Anteil der fehlenden Daten eine große Rolle bei der Performance von Imputationsmethoden, wie von Hu et al. [32] und Saunders et al. [34] gezeigt wird. In [32] wird darüber hinaus nachgewiesen, dass auch der Datentyp, also ob quantitativ oder qualitativ, entscheidend für die Performance einer Imputationsmethode sein kann.
Zusammenfassend lässt sich sagen, dass die Performance einer Imputationsmethode stark vom zugrunde liegenden Datensatz abhängt [36, 50].

4.2 Imputationsmethoden

Imputationsmethoden lassen sich auf unterschiedliche Weisen kategorisieren. Während von Tiwaskar et al. in [33] nach der Art der zugrunde liegenden Modelle (statistisches Modell oder Machine-Learning-Modell) unterschieden wird, kann man Imputationsmethoden auch nach der Anzahl der einfließenden Variablen (univariat oder multivariat) differenzieren. Eine weitere Möglichkeit ist die Unterscheidung der Methoden basierend auf dem Mechanismus (MCAR, MAR oder MNAR), den sie voraussetzen. Zudem stellt die Anzahl der Imputationen (single oder multiple) eine sinnvolle Unterscheidung dar. In Tabelle 4 werden vier häufig angewendete Imputationsmethoden nach den eben beschriebenen Möglichkeiten kategorisiert. Die dabei von Van Buuren und Oudshoorn entwickelte vorkommende Multivariate Imputation by Chained Equation (MICE) [51, 52] ist ein iteratives Verfahren, bei dem für jede Variable mit fehlenden Werten ein eigenes Regressionsmodell auf Basis der übrigen Variablen trainiert wird, um fehlende Werte zu schätzen. Man kann es als hybrid bezeichnen, da es grundsätzlich ein statistisches Framework ist, aber für die Schätzung sowohl Machine-Learning-Modelle als auch statistische Modelle verwenden kann. MICE ist eine Methode der Multiple Imputation (MI), die von Rubin in [42] vorgestellt wurde.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Methode	**Anzahl d. Imputationen**	**Univariat/ Multivariat**	**Modellart**	**Mechanismus**
Mittelwertimputation	Single	Univariat	Statistisch	MCAR
KNN-Imputation [53]	Single	Multivariat	Machine Learning	MCAR, MAR
MissForest [54]	Single	Multivariat	Machine Learning	MCAR, MAR
MICE [51]	Multiple	Multivariat	Hybrid	MCAR, MAR

Tabelle 4: Kategorisierung ausgewählter Imputationsmethoden

Multiple Imputation ist ein Überbegriff für Methoden, die jeden fehlenden Wert nicht nur durch einen einzigen Wert imputieren. Stattdessen werden mehrere plausible Werte in Betracht gezogen und anschließend zu einem finalen Imputationswert kombiniert. Dadurch werden Unsicherheiten der fehlenden Werte bei der Imputation miteinbezogen.
Obwohl MICE eine häufig verwendete Methode ist und bei den multiplen Imputationsverfahren hervorsticht [55], zeigt MICE Schwächen bei großen Datensätzen [56, 37].

Aus Gründen der Berechnungszeit beschränkt sich diese Arbeit auf Methoden mit nur einer einzigen Imputation. Im Folgenden werden die drei weiteren Verfahren aus Tabelle 4 vorgestellt und für die Imputation der fehlenden Werte des vorliegenden Datensatzes verwendet und verglichen.

4.2.1 Mittelwertimputation

Die Mittelwertimputation zählt zu den einfachsten Verfahren, um einen Ersatz für fehlende Werte zu finden [31]. Dabei werden sämtliche fehlenden Werte einer numerischen Variable durch den Mittelwert der beobachteten Werte imputiert. Handelt es sich hingegen um eine kategoriale Variable, so werden fehlende Werte durch den Modus, also die am häufigsten vorkommende Kategorie, ersetzt. Algorithmus 2 stellt diese einfache Vorgehensweise dar.

Algorithmus 2: Mittelwertimputation

Input: Realisierung $\mathbf{x}_j$ mit fehlenden Werten
Output: Vektor $\hat{\mathbf{x}}_j$ mit imputierten Werten
Berechne den Mittelwert $\bar{x}_j$ auf Basis der beobachteten Daten
foreach $i \in \{1, \ldots, n\}$ **do**
  if $x_{ij} = NA$ **then**
    Setze $\hat{x}_{ij} := \bar{x}_j$
  else
    Setze $\hat{x}_{ij} := x_{ij}$
  end
end
return $\hat{\mathbf{x}}_j$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Der Vorteil der Einfachheit dieser Methode wird jedoch durch einige Nachteile relativiert. So unterschätzt die Mittelwertimputation die Varianz der Variable, und Korrelationen innerhalb des Datensatzes werden als zusätzliche, möglicherweise relevante Informationen ignoriert, da es sich um eine univariate Methode handelt [31, 56]. Außerdem wird für die Mittelwertimputation ein MCAR-Mechanismus angenommen [34], der in der Praxis nur selten vorkommt. Auch im vorliegenden Datensatz ist diese Voraussetzung nicht erfüllt, weshalb die Mittelwertimputation in erster Linie als Referenz für den Vergleich mit den beiden komplexeren Methoden herangezogen wird.

4.2.2 KNN-Imputation

Die von Troyanskaya et al. in [53] eingeführte Imputationsmethode „KNN-Imputation“ basiert auf dem k-Nearest-Neighbor-Algorithmus. Dabei werden fehlende Werte einer Beobachtung durch eine Kombination der beobachteten Variablenwerte der k ähnlichsten Beobachtungen ersetzt.
Die zu berechnende Ähnlichkeit beruht dabei auf den vorhandenen Werten und wird standardmäßig durch eine, speziell für fehlende Werte angepasste, euklidische Metrik bestimmt. Diese, im Python-Paket **scikit-learn** für die KNN-Imputation eingesetzte Metrik beruht auf der von Dixon in [57] vorgestellten Idee, für die Distanzberechnung lediglich Paare bekannter Werte heranzuziehen. Auf Basis dessen werden die k Nachbarn identifiziert, deren Werte für die im Rahmen der Imputation betrachtete Variable bekannt sind.

Algorithmus 3: KNN-Imputation

Input: Datenmatrix $\mathbf{X}$ mit fehlenden Werten, Anzahl der Nachbarn k
Output: Imputierte Matrix $\hat{\mathbf{X}}$
foreach *Realisierung* $\mathbf{x}_j$ *mit fehlenden Werten* **do**
  foreach *Beobachtung* i *mit fehlendem* x_{ij} **do**
    foreach *Beobachtung* $m \neq i$ *mit beobachtetem* x_{mj} **do**
      Bestimme die Menge $S_{im} \subseteq \{1, \ldots, p\} \setminus \{j\}$, mit gemeinsam beobachteten Variablen
      Berechne $d(i,m) := \sqrt{\frac{p}{|S_{im}|} \sum_{l \in S_{im}} (x_{il} - x_{ml})^2}$ [57]
    end
    Wähle die k ähnlichsten Beobachtungen $\mathcal{N}(x_{ij})$ auf Basis von $d(i,m)$
    Setze $\hat{x}_{ij} := \frac{1}{k} \sum_{m \in \mathcal{N}(x_{ij})} x_{mj}$
  end
end
foreach $x_{ij} \neq NA$ **do**
  Setze $\hat{x}_{ij} := x_{ij}$
end
return $\hat{\mathbf{X}}$

Die Auswahl der Nachbarn für eine Beobachtung ist nur dann eindeutig, wenn die Beobachtung bei nur einer Variable einen fehlenden Wert ausweist. Andernfalls wird für jede nicht beobachtete Variable eine eigene Menge von Nachbarn identifiziert. Der finale Imputationswert wird anschließend als das arithmetische Mittel der Werte dieser Variable

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

bei den identifizierten Nachbarn bestimmt. Optional kann dieser auch durch ein distanzgewichtetes Mittel ermittelt werden, bei dem näherliegende (geringere Distanz) Nachbarn einen stärkeren Einfluss auf den Imputationswert haben.

Das in Algorithmus 3 dargestellte Verfahren ist in dieser Form nur für numerische Datensätze einsetzbar. Dies entspricht der Tatsache, dass die KNN-Imputation in Python standardmäßig nur für numerische Datensätze implementiert ist. Um das Verfahren auf Daten mit gemischten Merkmalstypen anwenden zu können, benötigt es sowohl eine alternative Distanzmetrik, wie beispielsweise die Gower-Metrik [58], als auch eine angepasste Aggregationsregel (Modus für kategoriale Variablen). Im weiteren Verlauf wird von numerischen Daten ausgegangen, da kategoriale Variablen gegebenenfalls bereits entsprechend kodiert wurden.

Aufgrund der Einbeziehung weiterer Variablen bei der Imputation eines Wertes und der dadurch berücksichtigten Korrelation kann bei der KNN-Imputation im Allgemeinen von einer höheren Performance im Vergleich zur Mittelwertimputation ausgegangen werden. In [53] wurde dies bei einem Anteil fehlender Werte im Bereich von $1 - 20\%$ gezeigt. Dennoch wird in [32] gezeigt, dass das Verfahren im Gegensatz zur Mittelwertimputation empfindlich auf unterschiedliche Anteile fehlender Werte reagieren kann.
Die KNN-Imputation gilt als leistungsstarke Methode zur Behandlung fehlender Werte, die in der Literatur, wie beispielsweise in [37], im Laufe der Zeit durch verschiedene Varianten weiterentwickelt wurde. Die Methode erfordert mit der Anzahl der Nachbarn nur die Festlegung eines einzigen Parameters.

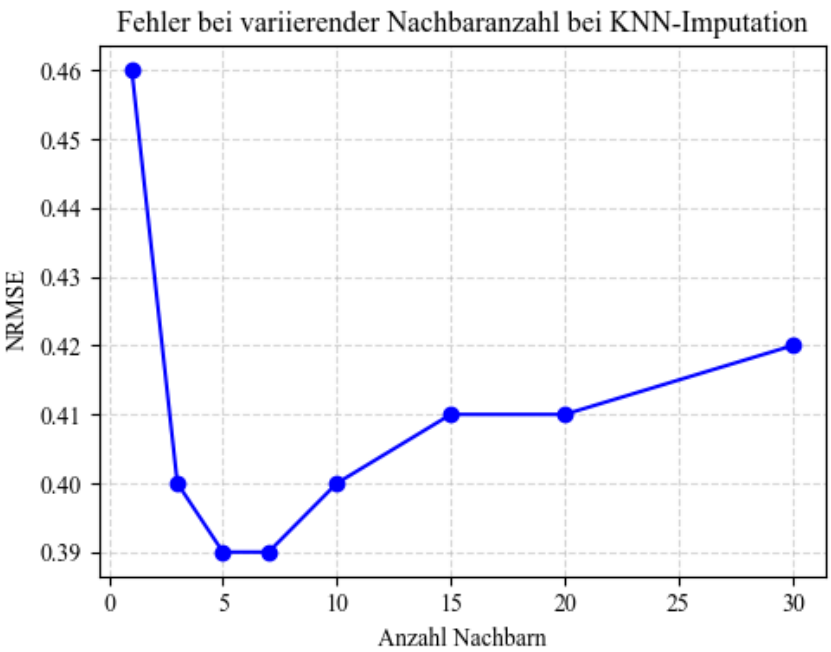

Abbildung 14: Resultierender Fehler bei Anwendung verschiedener Anzahlen von Nachbarn k bei einem einheitlichen Anteil fehlender Daten von 20%. Auf das Fehlermaß NRMSE wird später näher eingegangen

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Sowohl in [46], unter Berücksichtigung aller Mechanismen und verschiedener Anteile fehlender Daten, als auch in [31] für einen Anteil von $10 - 30\%$ fehlender Daten unter einem MCAR-Mechanismus, wurde im Rahmen der Experimente gezeigt, dass die Methode robust gegenüber der Wahl des Parameters k ist.
Wie in Abbildung 14 dargestellt, gilt dies auch für den Datensatz der vorliegenden Arbeit. Ab einer Anzahl von drei Nachbarn erzielt die KNN-Imputation die niedrigsten Imputationsfehler. Insbesondere bei $k = 5$ und $k = 7$ erreicht die Methode die besten Ergebnisse, bevor die Ersatzwerte bei einer höheren Anzahl von Nachbarn zu stark geglättet werden. Dieser Effekt fällt jedoch nur geringfügig ins Gewicht, sodass der NRMSE - auf den später genauer eingegangen wird - lediglich unwesentlich steigt. Aus diesem Grund wird die Performance der KNN-Imputation im weiteren Verlauf mit der Standardeinstellung $k = 5$ überprüft.

4.2.3 MissForest

MissForest ist ein leistungsstarkes und in der Literatur oft zitiertes Verfahren zur Ersetzung fehlender Werte. Die von Stekhoven und Bühlmann in [54] erstmals vorgestellte Methode basiert auf dem Random-Forest-Algorithmus [23] - auf den im folgenden Abschnitt näher eingegangen wird - und interpretiert das Problem fehlender Werte als Vorhersageproblem. Folgend aus der Verwendung des Random-Forest-Algorithmus kann die Methode sowohl mit numerischen als auch mit kategorialen Variablen umgehen.

Sei für eine beliebige Variable $\mathbf{X}_j$ die Indexmenge aller Beobachtungen mit fehlenden Werten in $\mathbf{x}_j$ durch $I_j^{(\mathrm{MIS})} \subseteq \{1, \ldots, n\}$ gegeben und bezeichne $\mathbf{X}_{-j}$ die Datenmatrix ohne $\mathbf{x}_j$. Während der Prozedur wird der Datensatz $\mathbf{X}$ für jede Variable folgendermaßen in vier Teile unterteilt:

- $\mathbf{y}_j^{(\mathrm{OBS})}$ als die beobachteten Werte von $\mathbf{x}_j$
- $\mathbf{y}_j^{(\mathrm{MIS})}$ als die fehlenden Werte von $\mathbf{x}_j$
- $\mathbf{X}_j^{(\mathrm{OBS})}$ als die Variablen aus $\mathbf{X}_{-j}$ mit Beobachtungen $I_j^{(\mathrm{OBS})} = \{1, \ldots, n\} \setminus I_j^{(\mathrm{MIS})}$
- $\mathbf{X}_j^{(\mathrm{MIS})}$ als die Variablen aus $\mathbf{X}_{-j}$ mit Beobachtungen $I_j^{(\mathrm{MIS})}$

Zu Beginn des Algorithmus werden die Variablen aufsteigend nach der Anzahl fehlender Werte sortiert. Durch die zuvor beschriebene Aufteilung in vier Teile resultiert für jede Variable $\mathbf{X}_j$ eine vollständig beobachtete Zielvariable in Form von $\mathbf{y}_j^{(\mathrm{OBS})}$, wie sie für den Random-Forest-Algorithmus erforderlich ist. Da jedoch die in diesem Kontext als erklärende Variablen fungierenden $\mathbf{X}_j^{(\mathrm{OBS})}$ fehlende Werte enthalten, werden diese initial mittels Mittelwertimputation geschätzt.
Nach erfolgter Durchführung der beiden anfänglichen Schritte - der Sortierung und der initialen Schätzung - folgt der iterative Teil des Algorithmus. Dabei wird für jede Variable ein Random-Forest-Modell auf der Zielvariable $\mathbf{y}_j^{(\mathrm{OBS})}$ und den zugehörigen erklärenden Variablen $\mathbf{X}_j^{(\mathrm{OBS})}$ gefittet. Anschließend werden die fehlenden Werte in $\mathbf{y}_j^{(\mathrm{MIS})}$ durch Anwendung des gefitteten Modells auf $\mathbf{X}_j^{(\mathrm{MIS})}$ vorhergesagt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Algorithmus 4: MissForest Imputation

Input: Datenatrix $\mathbf{X}$ mit fehlenden Werten
Output: Imputierte Matrix $\hat{\mathbf{X}}$
Sortiere Spalten der Datenmatrix $\mathbf{x}_j$ aufsteigend nach Anzahl fehlender Werte und speichere Indexreihenfolge in I
foreach $j \in I$ **do**
  Bestimme Indexmengen für beobachtete und fehlende Werte
    $I_j^{(\mathrm{MIS})} \subseteq \{1, \ldots, n\}$: Indizes mit fehlendem x_{ij}
    $I_j^{(\mathrm{OBS})} = \{1, \ldots, n\} \setminus I_j^{(\mathrm{MIS})}$
  Zielvariable (beobachtet/fehlend)
    $\mathbf{y}_j^{(\mathrm{OBS})} := \mathbf{x}_j[I_j^{(\mathrm{OBS})}]$
    $\mathbf{y}_j^{(\mathrm{MIS})} := \mathbf{x}_j[I_j^{(\mathrm{MIS})}]$
  Erklärende Variablen (beobachtet/fehlend)
    $\mathbf{X}_j^{(\mathrm{OBS})} := \mathbf{X}_{-j}[I_j^{(\mathrm{OBS})}]$
    $\mathbf{X}_j^{(\mathrm{MIS})} := \mathbf{X}_{-j}[I_j^{(\mathrm{MIS})}]$
end
Initialisiere fehlende Werte in $\mathbf{X}$ durch Mittelwert in $\hat{\mathbf{X}}$
repeat
  Speichere bisherige Imputation: $\mathbf{X}^{\mathrm{old}} \leftarrow \hat{\mathbf{X}}$
  foreach $j \in I$ **do**
    Trainiere Random Forest: $\mathbf{y}_j^{(\mathrm{OBS})} \sim \mathbf{X}_j^{(\mathrm{OBS})}$
    Sage fehlende Werte voraus: $\hat{\mathbf{y}}_j^{(\mathrm{MIS})} := \mathrm{Predict}(\mathbf{X}_j^{(\mathrm{MIS})})$
    Aktualisiere Matrix: Setze $\hat{\mathbf{x}}_j[I_j^{(\mathrm{MIS})}] := \hat{\mathbf{y}}_j^{(\mathrm{MIS})}$
  end
  Abbruchkriterium aktualisieren Vergleiche $\hat{\mathbf{X}}$ mit $\mathbf{X}^{\mathrm{old}}$ bzgl. γ
until *Abbruchkriterium γ erfüllt*
return $\hat{\mathbf{X}}$

Wie in Algorithmus 4 erkennbar ist, entstehen im Verlauf der Prozedur mit jedem Iterationsschritt neue Imputationswerte. Dieses iterative Vorgehen wird solange wiederholt, bis ein vordefiniertes Abbruchkriterium erfüllt ist.

Stekhoven und Bühlmann schlagen als mögliches Abbruchkriterium vor, den Imputationsprozess zu beenden, sobald sich der Unterschied zwischen der neu imputierten und der zuvor imputierten Datenmatrix erstmals vergrößert. Dieser Unterschied kann für die Menge der numerischen Variablen N folgendermaßen ausgedrückt werden:

$$\Delta_N = \frac{\sum_{j \in N} \left(\mathbf{x}_j^{\mathrm{new}} - \mathbf{x}_j^{\mathrm{old}}\right)^2}{\sum_{j \in N} \left(\mathbf{x}_j^{\mathrm{new}}\right)^2}. \tag{37}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Dieses Kriterium kann - entsprechend adaptiert - auch auf die Menge F der kategorialen Variablen umgesetzt werden:

$$\Delta_F = \frac{\sum_{j \in F} \sum_{i=1}^{n} \mathbb{1}_{(\mathbf{x}_{ij}^{\text{new}} \neq \mathbf{x}_{ij}^{\text{old}})}}{\#\text{NA}}. \tag{38}$$

MissForest liefert eine Reihe von Vorteilen, die einerseits aus der Konstruktion des Verfahrens hervorgehen und andererseits von Stekhoven und Bühlmann anhand verschiedener Datensätze aus biologischen und medizinischen Forschungsbereichen empirisch aufgezeigt wurden. Ein wesentlicher Vorteil besteht darin, dass es sich um eine Imputationsmethode handelt, die weder Annahmen über die Verteilungen der Variablen voraussetzt noch ein aufwendiges Parametertuning erfordert. Darüber hinaus wird in [54] festgehalten, dass MissForest insbesondere bei komplexen Interaktionen und nichtlinearen Beziehungen innerhalb der Daten eine hohe Imputationsgenauigkeit erzielt. Weiters ist im Gegensatz zur KNN-Imputation eine Skalierung der Daten im Vorfeld nicht notwendig. Selbst bei hochdimensionalen Datensätzen liefert MissForest laut Stekhoven und Bühlmann exzellente Imputationsergebnisse.

Die aufgezählten Vorteile spiegeln sich in den durchaus beachtlichen Ergebnissen diverser Vergleichsstudien aus der Literatur wider:
In [54] selbst erwies sich MissForest, angewendet auf unterschiedliche Datensätze mit einem Anteil fehlender Werte zwischen 10% und 30%, der KNN-Imputation und MICE als überlegen. Bei gleichem Anteil fehlender Werte, aber unter einem MCAR-Mechanismus, zeigte MissForest in [31] im Vergleich zu MICE, KNN-Imputation und der Mittelwertimputation den geringsten Imputationsfehler sowie im weiteren Verlauf die geringste Auswirkung auf die Vorhersagegenauigkeit. Bei einem Vergleich von MissForest und KNN-Imputation in [50] weisen beide Imputationsmethoden bei der Behandlung fehlender Daten im Bereich von $5-20\%$ eine gute Performance vor. Auch für einen MAR-Mechanismus ist MissForest gemäß [33] bei einem Anteil fehlender Werte zwischen 5 und 30% der KNN-Imputation und MICE überlegen. Auf analoge Ergebnisse stoßen auch [35] und [59].
Wie bei der KNN-Imputation existieren auch für den MissForest-Algorithmus Weiterentwicklungen, die dessen ohnehin schon hohe Genauigkeit weiter verbessern sollen. So wird beispielsweise in [32] eine Adaption vorgestellt, bei der der Einfluss irrelevanter Features reduziert wird. Diese Erweiterung übertrifft unter einem MCAR-Mechanismus und einem Fehldatenanteil in Höhe von $10-50\%$ die Mittelwertimputation, die KNN-Imputation, MICE und MissForest selbst.

Dennoch kann nicht daraus geschlossen werden, dass MissForest grundsätzlich die überlegene Imputationsmethode darstellt. Etwa erzielt MissForest in [60] schwache Imputationsergebnisse bei der Anwendung auf ausschließlich numerische Daten. Darüber hinaus argumentieren Hong und Lynn in [45], dass MissForest bei weitem kein „Allheilmittel“ zur Behandlung fehlender Werte sei und zeigen, dass das Verfahren Schwächen bei stark schiefen Daten in Kombination mit einem MAR-Mechanismus aufweist.
In [46] wird gezeigt, dass speziell die Korrelation innerhalb der Daten einen starken Einfluss auf die Performance baumbasierter Imputationsmethoden hat.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Zusammenfassend lässt sich festhalten, dass keine universell beste Methode existiert. Diese Erkenntnis wird von Liao et al. in [37] durch verschiedene Simulationen gestützt, in denen sich keine Methode konstant als überlegen erwiesen hat. Wie bereits im Verlauf des Abschnitts beschrieben, hängt die Performance einer Imputationsmethode maßgeblich vom zugrunde liegenden Datensatz ab - unter anderem vom Muster, Mechanismus und Anteil fehlender Werte, der Dimensionalität, der Korrelation zwischen den Variablen sowie dem Variablentyp betroffener Variablen.
Insofern ist es sinnvoll, sich im Zuge des Imputationstests mit verschiedenen Methoden auseinanderzusetzen und deren Ergebnisse zu vergleichen.

4.3 Imputationstest

In diesem Abschnitt werden die zuvor beschriebenen Imputationsverfahren anhand des vorliegenden Datensatzes verglichen. Für die Durchführung des Vergleichs werden, gemäß der Beschreibung der Modellpipeline, zufällig fehlende Werte künstlich erzeugt. Dabei werden die in Abbildung 15 dargestellten vier verschiedenen Anteile fehlender Daten angesetzt.

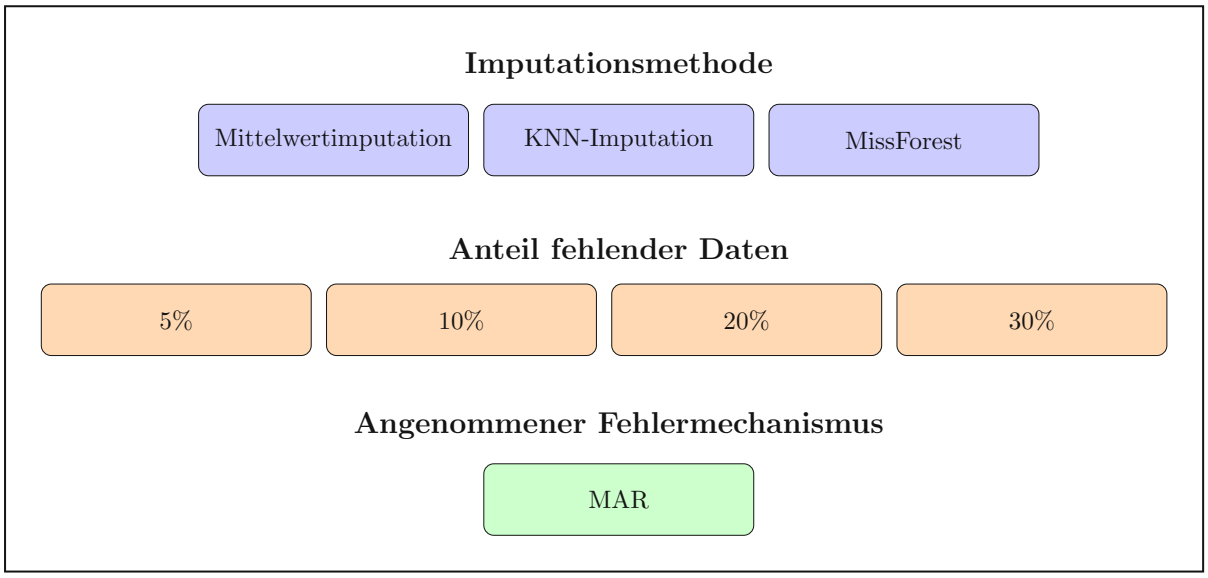

Abbildung 15: Übersicht Imputationstest

Um die Performance der Imputation beurteilen zu können, wird der von Oba et al. in [61] eingeführte „Normalized Root Mean Squared Error“ bestimmt.
Dieser ist für numerische Daten als

$$\text{NRMSE} = \sqrt{\frac{\text{mean}\left((\mathbf{X}^{\text{true}} - \mathbf{X}^{\text{imp}})^2\right)}{\text{var}(\mathbf{X}^{\text{true}})}}, \tag{39}$$

definiert. Dabei bezeichnet $\mathbf{X}^{\text{true}}$ die vollständige „wahre“ Datenmatrix und $\mathbf{X}^{\text{imp}}$ die imputierte Datenmatrix. Mittelwert und Varianz werden auf Basis jener Zellen der Datenmatrix berechnet, an denen fehlende Werte eingestreut wurden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Bei der Behandlung eines Datensatzes mit gemischten Datentypen kann analog zu [54] der Anteil falsch klassifizierter Einträge (PFC)

$$\text{PFC} = \frac{\text{Anzahl falsch imputierter kategorialer Werte}}{\text{Anzahl aller imputierter kategorialer Werte}} \tag{40}$$

als zusätzliches Gütemaß angesetzt werden.
Beide Fehlermetriken erlauben damit einen direkten Vergleich der Imputationsmethoden hinsichtlich ihrer Genauigkeit. Sowohl für den NRMSE als auch für den PFC gilt: Je näher der Wert bei 0 liegt, desto genauer ist die Imputation. Werte nahe 1 deuten hingegen auf eine schlechte Performance der Imputationsmethode hin. Da im Rahmen des Modellvergleichs ein numerischer Datensatz vorliegt, erfolgt die Beurteilung der Imputationsmethoden im Folgenden nur auf Basis des NRMSE.

Die nachfolgende Tabelle 5 fasst die Resultate des Imputationstests zusammen.

Fehlend	**Mittelwertimputation**	**KNN-Imputation**	**MissForest**
5%	1,02	0,38	0,60
10%	1,02	0,39	0,57
20%	1,02	0,39	0,59
30%	1,02	0,39	0,58

Tabelle 5: NRMSE-Vergleich der Imputationsmethoden

Die schwache Performance der Mittelwertimputation ist wenig überraschend. Wie im vorangegangenen Abschnitt beschrieben, liefert diese Methode selbst bei einem MCAR-Mechanismus häufig ungenaue Imputationswerte, da sie bei der Bestimmung keine Korrelationen zwischen den Variablen berücksichtigt. Als Referenzmethode zeigt sie jedoch deutlich, dass komplexere Methoden weitaus bessere Ergebnisse liefern, auch wenn dies mit einer höheren Berechnungszeit einhergeht.

Obwohl der MissForest-Algorithmus in vielen Arbeiten als überlegene Imputationsmethode hervorsticht, liefert die KNN-Imputation für diesen expliziten Datensatz konstant den niedrigsten NRMSE. Trotz bereits bekannter Ergebnisse in der Literatur, die zeigen, dass MissForest nicht in jedem Fall anderen Imputationsmethoden überlegen ist, kommt dieses Ergebnis dennoch überraschend und bedarf einer Erklärung.

Im vorliegenden Anwendungsfall profitiert die KNN-Imputation von der Struktur des Datensatzes. Zugversuche, die aufgrund gleicher Testumgebungen oder Materialien Cluster bilden, führen zu stark korrelierten Beobachtungen. Diese Ähnlichkeiten werden bei der Imputation mittels k Nachbarn optimal genutzt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

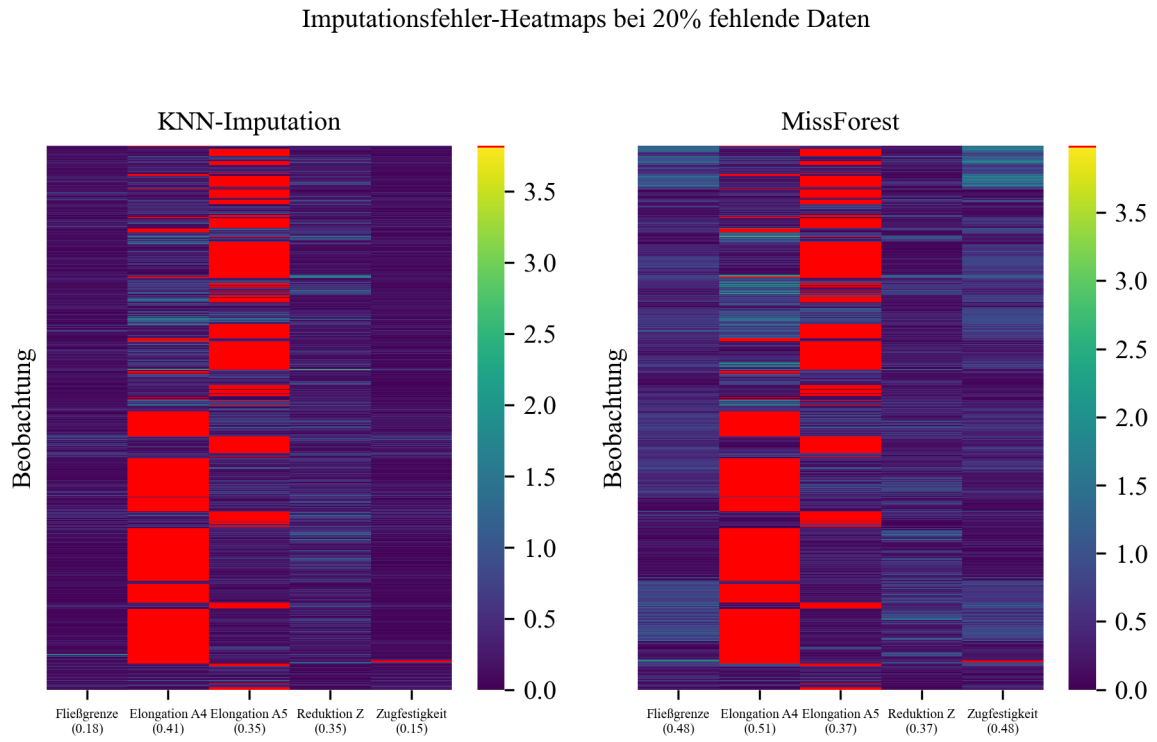

Abbildung 16: Heatmap des Imputationsfehlers zur Visualisierung auftretender Strukturen. In Rot sind jene Zellen dargestellt, in denen initial oder als Folge der Datenaufbereitung fehlende Werte vorkommen. Da der NRMSE ein Durchschnittsfehler ist, wird hier der mit der Standardabweichung normalisierte absolute Fehler angesetzt.

In Abbildung 16, in der Beobachtungen analog zu den Überlegungen bei der Imputation der Zielvariable nach Material, Temperatur und Zulieferer sortiert sind, wird diese Struktur anhand einer Heatmap des Imputationsfehlers bei den künstlich entfernten Werten dargestellt. Insbesondere in den Variablen Fließgrenze Lage 1 und Zugfestigkeit Lage 1 nutzt die KNN-Imputation die Clusterbildung, innerhalb derer die Fließgrenze eine geringe Varianz aufweist. Die ausgeprägte Clusterbildung im Datensatz könnte somit die bessere Performance der KNN-Imputation erklären.

Selbst die in [32] erwähnte Empfindlichkeit gegenüber unterschiedlichen Anteilen fehlender Daten ist in diesem Test nicht zu erkennen. Alle Imputationsmethoden zeigen eine gute Robustheit gegenüber zunehmendem Datenverlust, sodass meist nur eine geringe Zunahme des NRMSE mit wachsendem Anteil fehlender Daten erkennbar ist. Im Fall der KNN-Imputation liegt der NRMSE dabei stets im Bereich von 0,38 und 0,39.

Auch dieses Ergebnis ist unerwartet, kann jedoch durch die Konstruktion des Imputationstests erklärt werden. Wie bereits bei der Einführung der Fehlermechanismen erläutert, kann von einem MAR-Mechanismus ausgegangen werden, wodurch die entfernten Daten annähernd die gleiche Verteilung wie die beobachteten Daten besitzen. Dies wird in Abbildung 17 für alle Fehldatenanteile dargestellt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

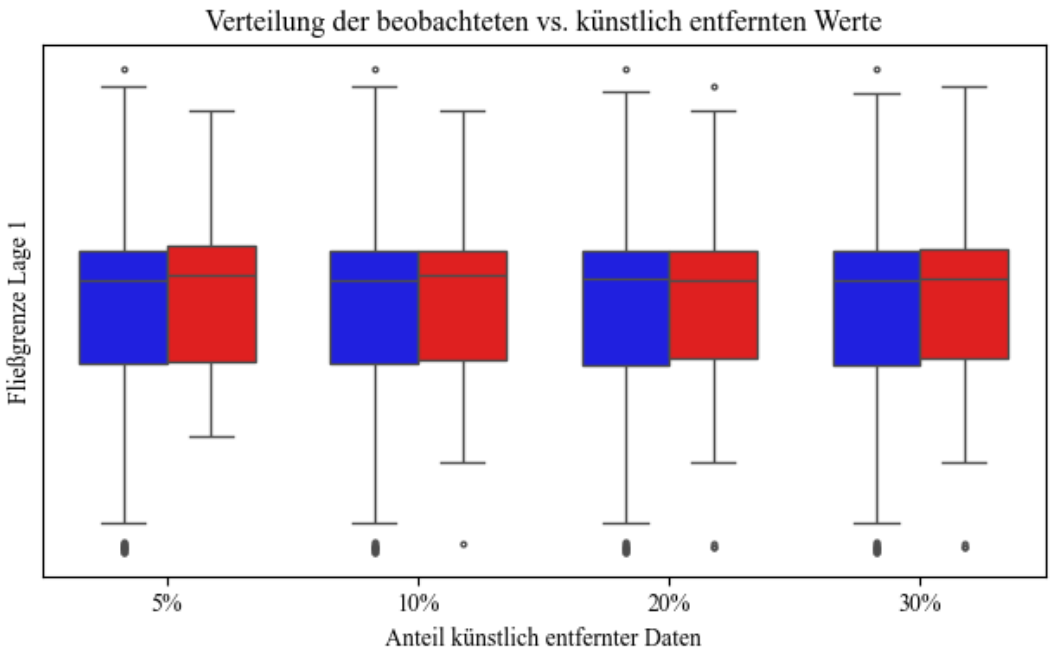

Abbildung 17: Vergleich der Verteilungen beobachteter (blau) und fehlender (rot) Daten für variierende Anteile fehlender Daten

Selbst bei 30% entfernter Daten bleiben die Verteilungen nahezu deckungsgleich, sodass die Imputationsmethoden präzise Ersatzwerte liefern können.

Neben dem Fehlermechanismus kann auch die Korrelation innerhalb des Datensatzes den annähernd konstanten NRMSE bei variierenden Fehldatenanteilen erklären. Auf Grundlage ähnlicher Beobachtungen und relevanter vollständig beobachteter Variablen lässt sich der Einfluss des Anteils fehlender Werte minimieren.

Dieses Ergebnis unterstreicht ein weiteres Mal die Aussage, dass die Performance einer Imputationsmethode stark vom zugrunde liegenden Datensatz und den fehlenden Daten (Muster und Mechanismus) abhängt. In diesem Fall könnte MissForest möglicherweise aufgrund des Datentyps schwächer als die KNN-Imputation performen.
Als Fazit des durchgeführten Imputationstests in Kombination mit den Erkenntnissen aus der weitreichenden Literatur lässt sich festhalten, dass vor Anwendung einer Imputationsmethode sowohl die Struktur des Datensatzes als auch die Art der auftretenden fehlenden Werte sorgfältig untersucht werden sollte. Gegebenenfalls ist es sinnvoll, verschiedene Methoden zu testen, bevor im Anschluss die für den Datensatz am besten geeignete Methode angesetzt wird, um möglichst präzise Imputationswerte zu erzielen.

Vor Abschluss dieses Abschnitts sei darauf hingewiesen, dass der Imputationstest auch die Auswirkungen auf die Vorhersagegenauigkeit berücksichtigt. Damit wird das nächste Kapitel eingeleitet, in dem Vorhersagen für die drei Imputationsmethoden bei den vier unterschiedlichen Anteilen fehlender Werte bestimmt werden.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5 Vorhersagemodelle

In diesem Abschnitt wird ein kompakter Überblick über die angesetzten Vorhersagemodelle gegeben. Ein besonderes Augenmerk liegt dabei auf einem Ansatz, bei dem Beobachtungen gewichtet werden, sodass ihr Einfluss in Abhängigkeit von der Anzahl fehlender Werte gesteuert wird.
Im Rahmen der Modellvalidierung wird dieser Gewichtungsansatz mit einem Standardmodell verglichen, in dem jeder Beobachtung die gleiche Wichtigkeit zugesprochen wird. Ziel dieses Vergleichs verschiedener Vorhersagemodelle bei unterschiedlichen Anteilen fehlender Werte ist es, das Modell zu bestimmen, das am besten mit Datenunsicherheiten umgehen kann und den geringsten Vorhersagefehler aufweist. Dabei ist zu beachten, dass eine niedrige Abweichung der imputierten Werte nicht zwangsläufig zu einer hohen Vorhersagegüte führt [45]. Wie Waljee et al. in [31] aufzeigen, können Variablen mit geringem Imputationsfehler wenig Einfluss auf das Vorhersagemodell haben, während Variablen mit einem großen Imputationsfehler entscheidend zur Vorhersagegüte beitragen. In diesem Fall kann es sinnvoll sein, alternative Imputationsmethoden zu präferieren, da diese zwar nicht die realistischsten Ersatzwerte erzeugen, aber im Zusammenspiel mit dem Vorhersagemodell bessere Ergebnisse erzielen. Diese Überlegung unterstreicht die Notwendigkeit, Imputation und Vorhersage gemeinsam zu bewerten.
Für die Beschreibung der theoretischen Hintergründe der Vorhersagemodelle werden im Folgenden die Lehrbücher „An Introduction to Statistical Learning“ von James et al. [9] und „The Elements of Statistical Learning“ von Hastie et al. [26] herangezogen.

5.1 Lineares Regressionsmodell

Das lineare Regressionsmodell ist ein sehr einfacher und dennoch häufig verwendeter Ansatz zur Vorhersage einer quantitativen Zielvariable. Es basiert auf der Annahme, dass die Beziehung zwischen den erklärenden Variablen und der Zielvariable durch eine lineare Funktion beschrieben werden kann. Zudem wird vorausgesetzt, dass die Beobachtungen unabhängig sind und die Regressormatrix vollen Spaltenrang aufweist. Darüber hinaus sichern weitere Annahmen wie homoskedastische und normalverteilte Fehler die statistische Aussagekraft hinsichtlich der Genauigkeit und Unsicherheit der Schätzungen. Im Vergleich zu komplexeren Modellen ist das lineare Regressionsmodell relativ unflexibel und besitzt bei nichtlinearen Zusammenhängen einen hohen Bias.

Die in Abschnitt 2.2 für die Formulierung des multiplen Regressionsmodells eingeführte Funktion f wird in der Form

$$f(\mathbf{X}) = \beta_0 + \sum_{j=1}^{p} \beta_j \mathbf{X}_j \tag{41}$$

angenommen. Die Parameter β_j für $j = 0, \ldots, p$ werden Koeffizienten genannt. β_0 wird als Intercept bezeichnet und ist der vorhergesagte Wert der Zielvariable, wenn alle erklärenden Variablen den Wert 0 annehmen. Die weiteren Koeffizienten β_j geben Auskunft über den durchschnittlichen Effekt auf die Zielvariable $\mathbf{Y}$ bei einer Erhöhung von $\mathbf{X}_j$ um eine Einheit, während alle anderen erklärenden Variablen fixiert bleiben.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die lineare Beziehung zwischen den erklärenden Variablen und der abhängigen Variable ergibt sich gemäß (6) und (41) zu

$$\mathbf{Y} = \beta_0 + \sum_{j=1}^{p} \beta_j \mathbf{X}_j + \boldsymbol{\varepsilon}. \tag{42}$$

Um für eine Beobachtung $x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ den Wert der Zielvariable abhängig von den beobachteten Features vorherzusagen, müssen die unbekannten Koeffizienten β_j für $j = 0, \ldots, p$ geschätzt werden:

$$\hat{y}_i = \hat{\beta}_0 + \sum_{j=1}^{p} \hat{\beta}_j x_{ij}. \tag{43}$$

Angenommen $(x_1, y_1), \ldots, (x_n, y_n)$ seien die beobachteten Trainingsdaten. Zur Bestimmung der Schätzer für die unbekannten Koeffizienten eignet sich die Methode der kleinsten Quadrate (Least squares). Bei dieser wird die Summe der quadrierten Abweichungen zwischen den beobachteten Zielwerten und den vorhergesagten Werten minimiert. Diese Summe wird als Residuenquadratsumme (RSS: Residual sum of squares) bezeichnet und durch

$$\mathrm{RSS}(\boldsymbol{\beta}) = \sum_{i=1}^{n} (y_i - f(x_i))^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \tag{44}$$

mit $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_p)^\top$ definiert.
Zur Vereinfachung der Darstellung wird für die Minimierung der Residuenquadratsumme auf die Matrixschreibweise zurückgegriffen. Dafür sei

$$\tilde{\mathbf{X}} = \begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix} \tag{45}$$

die für den Intercept erweiterte Regressormatrix. Daraus ergibt sich

$$\mathrm{RSS}(\boldsymbol{\beta}) = (\mathbf{y} - \tilde{\mathbf{X}}\boldsymbol{\beta})^\top (\mathbf{y} - \tilde{\mathbf{X}}\boldsymbol{\beta}) = \|\mathbf{y} - \tilde{\mathbf{X}}\boldsymbol{\beta}\|_2^2 \tag{46}$$

für die Summe der quadrierten Residuen. Ableiten nach $\boldsymbol{\beta}$ liefert

$$\frac{\partial \mathrm{RSS}}{\partial \boldsymbol{\beta}} = -2\tilde{\mathbf{X}}^\top (\mathbf{y} - \tilde{\mathbf{X}}\boldsymbol{\beta}) \tag{47}$$

$$\frac{\partial^2 \mathrm{RSS}}{\partial \boldsymbol{\beta}\, \partial \boldsymbol{\beta}^\top} = 2\tilde{\mathbf{X}}^\top \tilde{\mathbf{X}}. \tag{48}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Mit der Annahme, dass $\tilde{\mathbf{X}}$ vollen Spaltenrang hat, folgt die positive Definitheit von $\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}}$. Setzt man (47) gleich 0, so erhält man als eindeutiges Minimum der Residuenquadratsumme RSS:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \mathrm{RSS}(\boldsymbol{\beta}) = (\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^\top\mathbf{y}. \tag{49}$$

Schlussendlich erhält man die gefitteten Werte der Trainingsdaten aus:

$$\hat{\mathbf{y}} = \tilde{\mathbf{X}}\hat{\boldsymbol{\beta}} = \tilde{\mathbf{X}}(\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^\top\mathbf{y} \tag{50}$$

und die Vorhersage eines Inputvektors x_0 ist gegeben durch

$$\hat{f}(x_0) = \begin{pmatrix} 1 & x_0 \end{pmatrix}^\top \hat{\boldsymbol{\beta}}. \tag{51}$$

5.1.1 Gewichtung

Der im **scikit-learn**-Paket implementierte Gewichtungsansatz für Beobachtungen im Rahmen des linearen Regressionsmodells basiert im Wesentlichen auf der Theorie der gewichteten kleinsten Quadrate (WLS). Wie in [62, 63] beschrieben, eignet sich dieses Verfahren insbesondere dann, wenn die Fehler keine konstante Varianz aufweisen. In diesem Fall spricht man von heteroskedastischen Fehlern. Aufgrund der unterschiedlichen Anzahl fehlender Werte je nach Beobachtung ist die Annahme heteroskedastischer Fehler durchaus begründet. Wobei eine höhere Varianz mit einer höheren Anzahl fehlender Werte einhergeht.
Während die klassische WLS-Theorie von bekannten individuellen Fehlervarianzen ausgeht, sind diese in der Praxis häufig unbekannt [62]. Aus diesem Grund wird im Folgenden vereinfacht die Implementierung von Gewichten abhängig von der Anzahl fehlender Werte einer Beobachtung herangezogen und überprüft, ob dieser Gewichtungsansatz eine Verbesserung der Modellgüte erzielen kann.
Seien w_i für $i = 1, \ldots, n$ die in (22) definierten Gewichte. Für die Bestimmung der Parameter wird die um die Gewichte ergänzte RSS aus (44)

$$\mathrm{RSS}^{\mathbf{W}}(\boldsymbol{\beta}) = \sum_{i=1}^{n} w_i(y_i - f(x_i))^2 \tag{52}$$

minimiert. Sei

$$\mathbf{W} = \mathrm{diag}(w_1, w_2, \ldots, w_n) = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_n \end{pmatrix} \tag{53}$$

eine Diagonalmatrix mit den Gewichten in der Diagonale. Dann ergibt sich $\mathrm{RSS}^{\mathbf{W}}$ in der Matrixform zu:

$$\mathrm{RSS}^{\mathbf{W}}(\boldsymbol{\beta}) = (\mathbf{y} - \tilde{\mathbf{X}}\boldsymbol{\beta})^\top\mathbf{W}(\mathbf{y} - \tilde{\mathbf{X}}\boldsymbol{\beta}). \tag{54}$$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Analog zur Bestimmung des Schätzers für den ungewichteten Ansatz ($w_i = 1$ für alle $i \in \{1, \ldots, n\}$) ist der optimale Schätzer $\hat{\boldsymbol{\beta}}^{\mathbf{W}}$ ableitbar. Dieser ist durch

$$\hat{\boldsymbol{\beta}}^{\mathbf{W}} = (\tilde{\mathbf{X}}^\top \mathbf{W} \tilde{\mathbf{X}})^{-1} \tilde{\mathbf{X}}^\top \mathbf{W} \mathbf{y} \tag{55}$$

gegeben.
Mit diesem Vorgehen soll der Einfluss potenziell verzerrter Beobachtungen reduziert werden.

5.1.2 Ergebnisse

	Mittelwertimputation		**KNN-Imputation**		**MissForest**	
Fehlend	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**
0%	23,5	23,6	23,3	23,3	23,4	23,4
5%	28,8	28,1	23,5	23,5	24,4	24,3
10%	31,9	31,2	23,6	23,6	24,9	24,7
20%	34,6	34,3	24,4	24,4	27,8	27,5
30%	36,5	36,3	25,4	25,4	27,6	27,4

Tabelle 6: MAE-Vergleich bei Linearer Regression angewendet nach der Imputation durch die vorgestellten Imputationsmethoden bei unterschiedlichem Anteil fehlender Werte. Dabei wird zwischen dem Standardmodell und dem gewichteten Ansatz differenziert. Der dargestellte MAE-Wert beruht auf einer durchgeführten 5-Fold Cross-Validation.

Tabelle 6 liefert einige interessante Schlussfolgerungen. Zunächst zeigt sich, dass die KNN-Imputation im Vergleich zu den anderen Imputationsmethoden nicht nur im Hinblick auf den Imputationsfehler überlegen ist, sondern auch präzisere Vorhersagen ermöglicht. Für den Fall, in dem lediglich die initial vorkommenden fehlenden Werte imputiert werden (0% künstlich entfernt) liegen die MAE-Werte für alle drei Methoden im selben Bereich. Insbesondere die Mittelwertimputation reagiert erwartungsgemäß sehr sensibel auf einen steigenden Anteil fehlender Werte. Der MAE steigt um rund 55%, wenn 30% der Daten künstlich entfernt werden. Am robustesten gegenüber fehlenden Daten erweist sich die KNN-Imputation, bei der lediglich ein geringfügiger Anstieg des MAE beobachtbar ist.
Für sämtliche Methoden zeigt der gewichtete Ansatz seine Wirkung erst bei höheren Fehlanteilen. Die durch die Gewichtung unsicherer Daten erzielte Verbesserung fällt allerdings insgesamt sehr gering aus. Für die KNN-Imputation liegt der Unterschied beispielsweise nur im Komma-Bereich. Insgesamt liefert die Lineare Regression bemerkenswert gute Ergebnisse mit einem mittleren absoluten Fehler im Bereich von $20 - 40$ MPa und komplexere Imputationsmethoden erweisen sich erst bei höherem Anteil fehlender Daten als wirklich notwendig.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

5.2 Ridge Regression

Die von Hoerl und Kennard in [22] vorgestellte Ridge Regression ist die erste der beiden in dieser Arbeit im Hinblick auf die Vorhersagegenauigkeit bei imputierten Werten untersuchten Shrinkage-Methoden. Bei diesen wird der Schätzer für die Regressionskoeffizienten regularisiert, indem er in einen Bereich um den Nullpunkt eingeschränkt wird. Ziel dahinter ist eine Reduzierung der Varianz des Schätzers.

Anstelle der Minimierung von RSS, wie es bei der Kleinste-Quadrate-Methode der Fall ist, wird für die Ridge Regression ein von der Größe der Koeffizienten abhängiger Strafterm berücksichtigt:

$$\hat{\boldsymbol{\beta}}^{ridge} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\}. \tag{56}$$

Dabei ist im Gegensatz zum Linearen Regressionsmodell $\boldsymbol{\beta}$ durch $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$ ohne β_0 definiert. Mit $\lambda \geq 0$ wird im Strafterm ein Tuning-Parameter eingeführt. Wie bereits in Abschnitt 3 beschrieben, werden für den Vergleich der Imputation-Vorhersage-Gewichtung Kombinationen Default-Parameter angesetzt. In Übereinstimmung mit der Modellpipeline wird der Parameter dementsprechend mit $\lambda = 1$ gesetzt.
Allgemein gilt: je größer λ, desto stärker fließt der Strafterm ein. In diesem Fall werden die Koeffizienten stärker in Richtung 0 geschrumpft. Für $\lambda = 0$ entspricht der Ridge-Schätzer $\hat{\boldsymbol{\beta}}^{ridge}$ dem Kleinste-Quadrate-Schätzer aus (49). Analog zu (46) lässt sich auch das Optimierungsproblem der Ridge Regression in Matrixform darstellen. Dafür gilt ohne Beschränkung der Allgemeinheit, dass die Variablen bereits standardisiert wurden. Dies wurde im Zuge der vorgestellten Modellpipeline bereits durch die Datenaufbereitung sichergestellt.
Nachdem die Regularisierung den Koeffizienten des Intercepts nicht berücksichtigt, wird dieser mit $\hat{\beta}_0^{ridge} = \bar{y}$ geschätzt. Für die weiteren Komponenten des Koeffizientenvektors sei $\mathbf{X}$ die Matrix ohne zusätzliche Eins-Spalte. So lässt sich (56) als

$$\mathrm{RSS}_\lambda(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta} = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \tag{57}$$

in Matrixform ausdrücken und die Lösung der Minimierung ist gegeben durch

$$\hat{\boldsymbol{\beta}}^{ridge} = \arg\min_{\boldsymbol{\beta}} \mathrm{RSS}_\lambda(\boldsymbol{\beta}) = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{y}. \tag{58}$$

5.2.1 Gewichtung

Der gewichtete Ansatz erfolgt analog zu jenem der linearen Regression ohne Strafterm. Gemäß vorangegangener Überlegungen seien die Daten bereits standardisiert. Der separat behandelte Koeffizient für den Intercept wird durch

$$\hat{\beta}_0^{ridge,\mathbf{W}} = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} \tag{59}$$

geschätzt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die Gewichtung für die Bestimmung der weiteren Regressionskoeffizienten erfolgt im ersten Term der Zielfunktion gemäß (56)

$$\mathrm{RSS}_\lambda^{\mathbf{W}}(\boldsymbol{\beta}) = \sum_{i=1}^{n} w_i(y_i - f(x_i))^2 + \lambda \sum_{j=1}^{p} {\beta_j}^2. \tag{60}$$

Sei $\mathbf{W}$ die Matrix aus (53), dann lässt sich $\mathrm{RSS}_\lambda^{\mathbf{W}}$ in Matrixform als

$$\mathrm{RSS}_\lambda^{\mathbf{W}}(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top \mathbf{W}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \boldsymbol{\beta}^\top \boldsymbol{\beta} \tag{61}$$

schreiben. Der Koeffizientenschätzer für den gewichteten Ansatz ist anschließend durch

$$\hat{\boldsymbol{\beta}}^{ridge,\mathbf{W}} = (\mathbf{X}^\top \mathbf{W}\mathbf{X} + \lambda \mathbf{I}_p)^{-1} \mathbf{X}^\top \mathbf{W}\mathbf{y} \tag{62}$$

gegeben.

5.2.2 Ergebnisse

	Mittelwertimputation		**KNN-Imputation**		**MissForest**	
Fehlend	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**
0%	23,5	23,6	23,3	23,3	23,4	23,4
5%	29,0	28,2	23,5	23,5	24,4	24,3
10%	32,1	31,4	23,6	23,6	24,9	24,7
20%	34,9	34,7	24,4	24,4	27,8	27,6
30%	36,8	36,7	25,4	25,3	27,7	27,6

Tabelle 7: MAE-Vergleich bei Ridge Regression

Bei den Ergebnissen der Ridge Regression in Kombination mit den drei Imputationsmethoden lassen sich analoge Rückschlüsse wie bei der Linearen Regression ohne Strafterm ziehen. Sowohl hinsichtlich der Gewichtung als auch bezüglich der dominierenden Imputationsmethode lassen sich vergleichbare Aussagen für Ridge treffen. Im Vergleich zur Linearen Regression erzielt die Ridge Regression keine signifikante Verbesserung. Die MAE-Werte sind nahezu ident. Das könnte dem Umstand geschuldet sein, dass kein Hyperparametertuning vorgenommen wurde und die in Tabelle 7 dargestellten Ergebnisse für festes $\lambda = 1$ bestimmt wurden.

5.3 Lasso Regression

Die von Tibshirani in [24] vorgestellte Lasso Regression bietet gegenüber der Ridge Regression den Vorteil, dass einzelne Koeffizienten exakt auf 0 gesetzt werden können, wodurch Lasso eine automatische Variablenselektion durchführt. Dieser Effekt ergibt sich daraus, dass Lasso anstelle der L^2-Norm (wie bei Ridge) die L^1-Norm des Koeffizientenvektors als Strafterm verwendet.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Wie in Abbildung 18 zu erkennen ist, ermöglicht die Form der Einschränkung bei Lasso (links) eine Nullsetzung einzelner Koeffizienten, während diese bei Ridge selbst bei großem λ lediglich gegen 0 schrumpfen und weiterhin positiv bleiben.

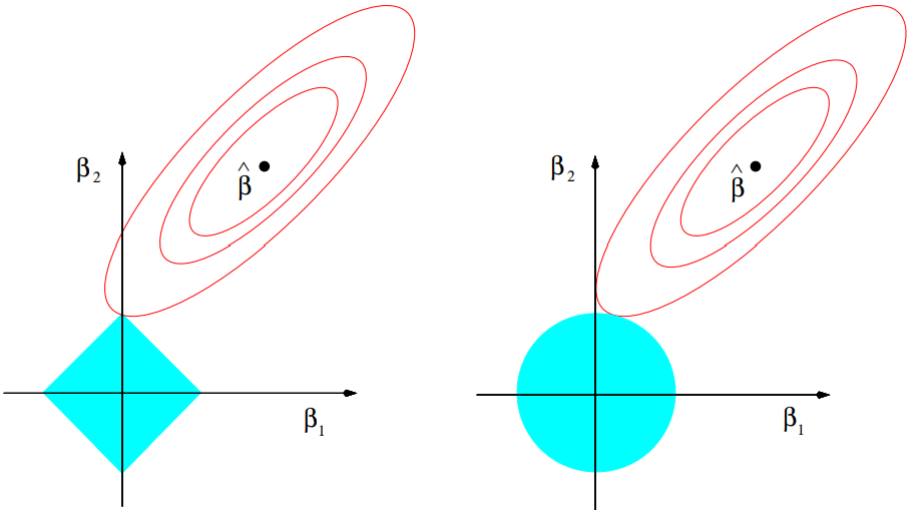

Abbildung 18: Vergleich von Lasso und Ridge: Für beide Methoden stellt $\hat{\boldsymbol{\beta}}$ den Kleinste-Quadrate-Schätzer dar. Entlang einer Ellipse ist die RSS konstant. Die Regularisierungsmethoden Lasso (links) und Ridge (rechts) lassen nur Koeffizienten zu, deren Werte innerhalb des türkisen Bereichs liegen.
Quelle: Trevor Hastie et al., *The Elements of Statistical Learning* [26]

Analog zur Ridge Regression reduziert auch Lasso mithilfe des Strafterms die Varianz des Schätzers. Der Schätzer ergibt sich dabei aus:

$$\hat{\boldsymbol{\beta}}^{lasso} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}. \tag{63}$$

Wie bereits bei der Ridge Regression ist $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^\top$, und der Tuning-Parameter ist gemäß der Modellpipeline für den Vergleich auf $\lambda = 1$ festgelegt.
Der Schätzer für den Koeffizienten des Intercepts $\hat{\beta}_0^{lasso}$ wird analog zur Ridge Regression auf den Mittelwert $\bar{y}$ gesetzt. Folgend lässt sich die Minimierung der Zielfunktion in Matrixform durch

$$\mathrm{RSS}_\lambda^{lasso}(\boldsymbol{\beta}) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1 = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_1 \tag{64}$$

ausdrücken.
Im Gegensatz zu Ridge gibt es für Lasso jedoch keine geschlossene Lösung. Lasso erzielt in der Regel bessere Ergebnisse als die Ridge Regression, wenn nur wenige Prädiktoren die Varianz der Zielvariable beschreiben. Ridge hingegen eignet sich insbesondere dann, wenn alle Prädiktoren in ähnlichem Maße zur Erklärung der Zielgröße beitragen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

5.3.1 Gewichtung

Der Gewichtungsansatz für Lasso erfolgt analog zur Ridge Regression, weshalb an dieser Stelle darauf nicht näher eingegangen wird.

5.3.2 Ergebnisse

	Mittelwertimputation		KNN-Imputation		MissForest	
Fehlend	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**
0%	24,9	25,0	24,7	24,8	24,8	24,8
5%	30,9	30,2	25,0	25,0	25,9	25,9
10%	35,2	34,5	25,3	25,2	26,5	26,4
20%	39,6	39,5	26,2	26,2	30,8	31,0
30%	41,7	42,2	27,3	27,2	30,7	30,6

Tabelle 8: MAE-Vergleich bei Lasso Regression

Auch bei der Lasso Regression erweist sich die KNN-Imputation den anderen Imputationsmethoden gegenüber als überlegen. Der gewichtete Ansatz zeigt, analog zu den vorherigen Modellen, nur bedingt Verbesserungen verglichen mit dem Standardmodell. Wie bereits bei den vorangegangenen Methoden zeigen sich die Effekte der Gewichtung erst bei einem hohen Anteil fehlender Werte.
Im Vergleich zur Linearen Regression ohne Strafterm sowie zur Ridge Regression liefert die Lasso Regression jedoch weniger präzise Vorhersagen: Für alle betrachteten Kombinationen liegt der MAE-Wert oberhalb der Werte der beiden anderen Methoden.

5.4 k-Nearest-Neighbor

Eine Variante des k-Nearest-Neighbor-Algorithmus wurde bereits für die Imputation fehlender Werte vorgestellt. Das von Cover und Hart in [21] vorgestellte Vorhersagemodell bildet die Grundlage der Imputationsmethode und funktioniert analog. Im Gegensatz zur Linearen Regression handelt es sich bei KNN um ein nichtparametrisches und dadurch flexibleres Verfahren. Die Vorhersage für einen Testdatenpunkt basiert direkt auf den Zielwerten der k nächsten Nachbarn im Merkmalsraum (die Menge aller möglichen Werte für die erklärenden Variablen $\mathbf{X}_1, \ldots, \mathbf{X}_p$) aus den Trainingsdaten. Während bei linearen Modellen Annahmen über die Beziehung zwischen Prädiktoren und Zielvariable getroffen werden, kommt der k-Nearest-Neighbor-Ansatz ohne solche Annahmen aus. Ein Nachteil der Methode besteht jedoch in der geringen Interpretierbarkeit.
Für gegebenes k sei x_0 eine Beobachtung aus den Testdaten. Auf Basis der euklidischen Metrik wird die Menge $\mathcal{N}(x_0)$ der k Beobachtungen mit dem geringsten Abstand aus den Trainingsdaten identifiziert.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Die Vorhersage für den Zielwert der Beobachtung x_0 wird durch den Durchschnitt der Zielwerte dieser k Nachbarn

$$\hat{y}_0 = \frac{1}{k} \sum_{x_i \in \mathcal{N}(x_0)} y_i \tag{65}$$

bestimmt. Aufgrund der Empfindlichkeit gegenüber unterschiedlichen Skalen der Distanzberechnung ist eine Standardisierung der Prädiktoren im Vorfeld des k-Nearest-Neighbor-Algorithmus erforderlich. Diese Skalierung wurde bereits im Rahmen der Datenaufbereitung gemäß der Modellpipeline durchgeführt.
KNN verfügt mit der Anzahl der Nachbarn k über einen Tuning-Parameter. Ist k zu klein gewählt, besteht die Gefahr des Overfittings. In diesem Fall orientiert sich die Vorhersage einer Testbeobachtung nur an einer kleinen Anzahl von Trainingsdaten. Ist k hingegen zu groß, führt dies zu einer Glättung der Vorhersage. Für den Modellvergleich wird die Anzahl mit $k = 5$ standardmäßig festgelegt.

5.4.1 Gewichtung

KNN ist die einzige der in dieser Arbeit betrachteten Methoden, für die es in Python keine Möglichkeit für eine Gewichtung einzelner Trainingsbeobachtungen gibt. Dennoch kann mit einer Modifikation der Aggregation bei der Vorhersage ein gewichteter Ansatz implementiert werden.
Anstelle von (65) wird für die Vorhersage von y_0 folgende gewichtete Mittelung angesetzt:

$$\hat{y}_0 = \frac{1}{k} \sum_{x_i \in \mathcal{N}(x_0)} w_i y_i. \tag{66}$$

Dadurch werden die Beobachtungen zwar bei der Bestimmung der k Nachbarn gleich gewertet, ihr Einfluss auf die Vorhersage wird jedoch gemäß ihrer Gewichte w_i reduziert.

5.4.2 Ergebnisse

	Mittelwertimputation		**KNN-Imputation**		**MissForest**	
Fehlend	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**
0%	20,4	20,4	20,3	20,3	20,4	20,4
5%	22,3	22,2	20,8	20,8	21,0	21,0
10%	23,5	23,5	21,0	20,9	21,4	21,4
20%	24,6	24,5	21,5	21,4	22,3	22,4
30%	25,7	25,5	22,5	22,3	23,8	23,7

Tabelle 9: MAE-Vergleich bei k-Nearest-Neighbor

Aus den Ergebnissen des k-Nearest-Neighbor-Modells (KNN) lässt sich ableiten, dass diese Methode für den vorliegenden Datensatz den drei linearen Modellen überlegen ist. Selbst bei einem Anteil von 30% künstlich erzeugter fehlender Daten liegt der mittlere absolute

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Fehler unterhalb jenem der linearen Modelle, bei denen lediglich die initial fehlenden Werte imputiert wurden (0% künstlich erzeugte fehlende Daten). Besonders hervorzuheben ist die vergleichsweise hohe Stabilität der Mittelwertimputation bei Vorhersagen mit dem k-Nearest-Neighbor-Verfahren. Während der MAE bei den linearen Modellen bei hohem Fehlanteil teilweise sogar über 40 MPa liegt, ist der MAE für die Kombination aus Mittelwertimputation und KNN-Vorhersage mit 25,7 MPa einigermaßen niedrig.
Auch bei KNN führt die KNN-Imputation zu präziseren Vorhersagen als die anderen Imputationsmethoden. Dennoch kann die Mittelwertimputation aufgrund ihrer geringen Komplexität als ernstzunehmende Alternative, selbst bei einem hohen Anteil fehlender Werte, in Betracht gezogen werden.
Analog zu den linearen Modellen zeigt der gewichtete Ansatz insbesondere bei einem hohen Fehlanteil eine geringfügige Verbesserung.

5.5 Random Forest

Random Forest ist eine der beiden baumbasierten Methoden, die in dieser Arbeit untersucht werden. Zur Einführung folgt zunächst eine kompakte Darstellung von Entscheidungsbäumen. Dabei folgt die Arbeit den Formulierungen von [9].
Entscheidungsbäume verfolgen das Ziel, den Merkmalsraum in T disjunkte Regionen bzw. Blätter $R_1, \ldots, R_T$ zu unterteilen. Auf Basis dessen sollen Beobachtungen aus den Testdaten anschließend über Aufteilungskriterien (splitting rules) einem dieser T Blätter zugeteilt werden. Die Vorhersage für eine Beobachtung entspricht dem Wert des Blattes v_j, typischerweise dem Mittelwert $\hat{y}_j$ der Zielwerte der Trainingsbeobachtungen in Region R_j.

In Abbildung 19 ist ein beispielhafter Entscheidungsbaum für den vorliegenden Datensatz dargestellt.

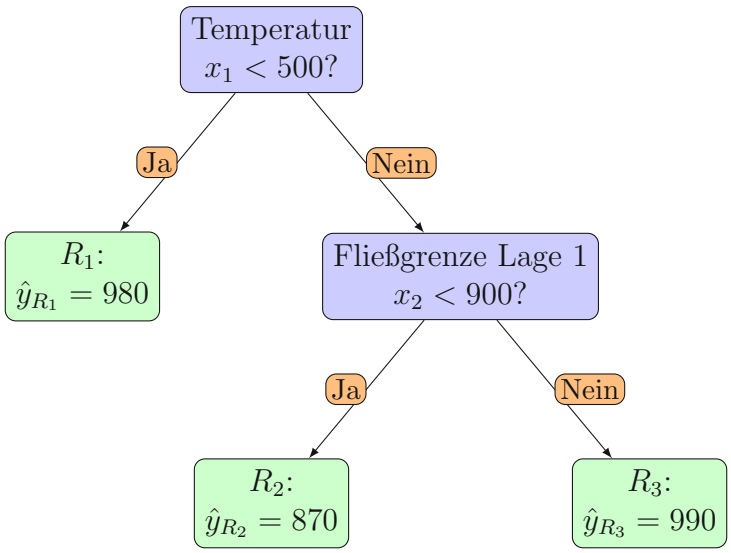

Abbildung 19: Darstellung eines simplen Entscheidungsbaums: Bei der Wurzel (oben) startend, wird eine Testbeobachtung über Aufteilungskriterien einer der drei Regionen (Blätter) zugeteilt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Der Aufbau eines Entscheidungsbaums verfolgt das Ziel, Regionen zu finden, die

$$\sum_{j=1}^{T} \sum_{i\,:\,x_i \in R_j} (y_i - v_j)^2 \tag{67}$$

minimieren. Nachdem die Berücksichtigung aller Partitionen des Featureraums einen hohen Rechenaufwand mit sich zieht, wird ein Baum für gewöhnlich mittels „Recursive Binary Splitting“ gebildet. Der Aufbau des Entscheidungsbaums beginnt dabei mit einer Teilung des Featureraums in zwei Teile (in Abbildung 19 wäre das die Aufteilung über die Temperatur bei einem Wert von 500 Kelvin). Dazu wird jener Split, bestehend aus Prädiktor $\mathbf{X}_j$ und Grenzpunkt s, gesucht, der (67) minimiert: Es werden zwei Regionen

$$R_1(j,s) = \{\mathbf{X} \,|\, \mathbf{X}_j < s\} \quad R_2(j,s) = \{\mathbf{X} \,|\, \mathbf{X}_j \geq s\} \tag{68}$$

gesucht, sodass

$$\sum_{i\,:\,x_i \in R_1(j,s)} (y_i - v_1)^2 + \sum_{i\,:\,x_i \in R_2(j,s)} (y_i - v_2)^2 \tag{69}$$

minimal ist. Resultierend daraus entstehen zwei neue Knoten, die zu diesem Zeitpunkt Terminalknoten (Blätter) sind. Anschließend wird eine der beiden Regionen analog aufgeteilt (In Abbildung 19 bleibt der linke Knoten ein Blatt, während der rechte Knoten weiter aufgeteilt wird). Diese Splittings erfolgen solange, bis ein Abbruchkriterium erfüllt ist. Beispielsweise, wenn jedes Blatt weniger als eine festgesetzte Minimalanzahl an Beobachtungen enthält.

Ein daraus entstehender Entscheidungsbaum bzw. Regressionsbaum ist im Allgemeinen zu komplex, in dem Sinne, dass er für das Trainingsset gute Vorhersagen liefert, aber eine schlechte Testperformance aufweist. Ein kleinerer (Teil-)baum, den man mittels „Tree-Pruning“ bestimmen kann, ist in der Lage, Overfitting zu reduzieren und bringt eine niedrigere Varianz, jedoch einen höheren Bias mit sich.

Sei

$$q : \mathbb{R}^p \to \{1, \ldots, T\} \tag{70}$$

eine Funktion, die einer Beobachtung x das Blatt R_j eines Baums f zuordnet. Der Wert v_j des Blattes, dem eine Beobachtung x zugeordnet wird, wird anschließend als $v_{q(x)}$ geschrieben. Somit lassen sich Entscheidungsbäume für eine Beobachtung x_i durch

$$f(x_i) = v_{q(x_i)} \tag{71}$$

notieren. Entscheidungsbäume sind insgesamt leicht erklärbar und graphisch darstellbar. Darüber hinaus stellen sogar qualitative Features für diese Art von Regressionsmethode kein Problem dar. Nachdem ein einzelner Baum in der Regel eine vergleichsweise geringe Vorhersagegenauigkeit aufweist und darüber hinaus kleine Änderungen in den Daten große Auswirkungen auf den Baum mit sich bringen, sind Ensemble-Methoden ein interessanter Ansatz, diesen Nachteilen entgegenzuwirken.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Bei einer Ensemble-Methode werden mehrere schwache Modelle (weak learners) zu einem starken kombiniert. Im Kontext der Ensemble-Methode Random Forest handelt es sich bei den schwachen Modellen um einzelne Entscheidungsbäume.

Die von Breimann in [23] vorgestellte Methode Random Forest ist eine spezielle Form des sogenannten Bagging (kurz für Bootstrap Aggregation). Im Rahmen des von Efron in [64] vorgestellten Bootstrap-Verfahren werden aus den Trainingsdaten wiederholt Stichproben mit Zurücklegen gezogen. Daraus erhält man B unterschiedliche, jedoch gleich große Datensätze. Auf diesen werden anschließend Entscheidungsbäume gefittet. Ziel ist es, die hohe Varianz eines einzelnen, nicht durch Tree-Pruning verkleinerten Baumes durch eine Aggregation von B Bäumen zu verringern.
Sei x_i eine Testbeobachtung und

$$f^1(x_i), \ldots, f^B(x_i) \tag{72}$$

die Vorhersagen für x_i der B auf unterschiedlichen Bootstrap-Datensätzen gefitteten Bäume, dann ist die resultierende Vorhersage durch

$$f^{\mathrm{RF}}(x_i) = \frac{1}{B} \sum_{b=1}^{B} f^b(x_i) \tag{73}$$

gegeben.
Die bisherige Vorgehensweise beschreibt die Verallgemeinerung Bagging. Die zentrale Eigenschaft von Random Forest besteht darin, die Korrelation zwischen den B unabhängigen Bäumen zu reduzieren. Bei jedem Split wird pro Baum nicht über alle p Merkmale entschieden, sondern über eine zufällige Teilmenge von m Prädiktoren.
Das soll verhindern, dass die Bäume für alle Datensätze die selbe Form annehmen. Daraus würde folgen, dass die zugehörigen Vorhersagen stark korrelieren und nur wenig variabel sind. Die Varianzreduktion durch die Kombination mehrerer Entscheidungsbäume hätte in diesem Fall nur bedingten Erfolg.

Algorithmus 5: Random Forest Regression gemäß [26]

Input: Trainingsdaten $\mathbf{X}$, Anzahl Bäume B, Anzahl zufällig gewählter Variablen m, minimale Blattgröße $n_{\min}$
Output: Random-Forest-Modell f^{RF}
for $b \in \{1, \ldots, B\}$ **do**
  Ziehe eine Bootstrap-Stichprobe $\mathbf{X}^b$ aus den Trainingsdaten
  Wachse einen Baum f^b auf $\mathbf{X}^b$ bis zur minimalen Blattgröße $n_{\min}$, indem rekursiv wiederholt wird:
  foreach *Blatt* **do**
    Ziehe zufällig m Variablen aus $\{\mathbf{X}_1, \ldots, \mathbf{X}_p\}$
    Wähle unter diesen m die beste Variable sowie den optimalen Splitpunkt
    Teile den Knoten anhand dieses Splits in zwei resultierende Knoten
  end
end
return $f^{RF} := \frac{1}{B} \sum_{b=1}^{B} f^{(b)}$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Gemäß des Modellvergleichs basierend auf den Standardparametern, werden für das Random-Forest-Modell folgende Parameter angesetzt:

- $B = 100$
- $m = \sqrt{p}$
- $n_{\min} = 1$

Obwohl bei Random Forest große Bäume gebildet werden, besteht im Gegensatz zu einzelnen Regressionsbäumen, keine Gefahr des Overfittings. Selbst bei einer hohen Baumanzahl B ist Overfitting im Fall von Random Forest auszuschließen.

5.5.1 Gewichtung

In **scikit-learn** nehmen individuelle Gewichte für einzelne Beobachtungen an zwei Stellen Einfluss auf das Modellfitting. Zum einen werden Unsicherheiten in den Daten in Form von Gewichten beim optimalen Aufteilen in zwei Regionen (Splitting) berücksichtigt:

$$\sum_{i\,:\,x_i \in R_k(j,s)} w_i (y_i - \hat{y}_{R_k})^2 + \sum_{i\,:\,x_i \in R_l(j,s)} w_i (y_i - \hat{y}_{R_l})^2. \tag{74}$$

Weiters wird anstelle des Mittelwerts der Zielwerte eines Blattes der gewichtete Mittelwert angesetzt:

$$\hat{y}_{R_k} = \frac{\sum_{i\,:\,x_i \in R_k} w_i\, y_i}{\sum_{i\,:\,x_i \in R_k} w_i}. \tag{75}$$

Eine mögliche Gewichtung der Beobachtung für den Boostrap-Vorgang ist standardmäßig nicht berücksichtigt.

5.5.2 Ergebnisse

	Mittelwertimputation		**KNN-Imputation**		**MissForest**	
Fehlend	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**
0%	18,2	18,2	17,7	17,6	17,8	17,8
5%	18,8	18,8	18,1	18,0	18,1	18,1
10%	19,4	19,3	18,3	18,2	18,2	18,2
20%	30,1	30,2	18,9	19,0	19,2	19,2
30%	31,3	31,0	19,2	19,4	19,9	19,9

Tabelle 10: MAE-Vergleich bei Random Forest

Wie bereits beim k-Nearest-Neighbor-Modell übertrifft Random Forest die linearen Modelle hinsichtlich der Vorhersagegenauigkeit. Random Forest erzielt sogar noch bessere Ergebnisse als KNN, wodurch der beste MAE-Wert je Anteil fehlender Daten meist um rund 3 MPa niedriger ist.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Analog zu den vorangegangenen Modellen liefert eine im Vorfeld durchgeführte KNN-Imputation anschließend die Vorhersagen mit der höchsten Genauigkeit. Im Gegensatz zu den bisherigen Methoden ist jedoch der Abstand zur Imputation mittels MissForest nicht mehr ganz so signifikant. Bei geringem Anteil fehlender Werte liefert die Mittelwertimputation unter Random Forest zwar beachtliche Ergebnisse, reagiert jedoch sehr sensibel auf zunehmenden Fehldatenanteil. So steigt der MAE-Wert bei einem Anteil von 10% fehlender Werte auf beinahe das Doppelte, wenn der Anteil auf 20% steigt. Hingegen für die anderen beiden Imputationsmethoden steigt der mittlere absolute Fehler nur langsam mit der Anzahl fehlender Daten.
Überraschenderweise führt der gewichtete Ansatz bei einem niedrigeren Anteil fehlender Werte zu besseren Ergebnissen als das Standardmodell, was im Widerspruch zu bisherigen Ergebnissen und der naheliegenden Intuition, dass die Gewichtung wirksamer für einen hohen Fehldatenanteil wird, steht.

5.6 XGBoost

Der XGBoost-Algorithmus [25] (kurz für eXtreme Gradient Boosting) ist eine Erweiterung des klassischen Gradient Boosting [65]. Im Gegensatz zu Bagging-Verfahren wie Random Forest, bei denen die Bäume unabhängig voneinander trainiert werden, erfolgt beim Boosting der Aufbau der Bäume sequentiell. Jeder Baum nutzt Informationen der zuvor trainierten Bäume und wird explizit auf den Fehlern seiner Vorgänger aufgebaut. Ziel ist eine additive Modellstruktur bestehend aus B Entscheidungsbäumen:

$$f^B(x) = \sum_{b=1}^{B} h^b(x). \tag{76}$$

Im Gegensatz zu Bagging werden die Bäume dabei nicht auf Bootstrap-Datensätzen, sondern auf dem vollständigen Trainingsdatensatz trainiert.

Im Folgenden wird ein kompakter Überblick über das Gradient-Boosting-Verfahren gemäß der Beschreibung in [26] gegeben. Ziel des Verfahrens ist die Minimierung einer Verlustfunktion bezüglich f. Analog zu den vorherigen Methoden ist diese standardmäßig durch den quadratischen Fehler gegeben:

$$f^{\mathrm{GB}} = \arg\min_f \sum_{i=1}^{n} L(y_i, f(x_i)) = \arg\min_f \sum_{i=1}^{n} \frac{1}{2}(y_i - f(x_i))^2. \tag{77}$$

Die Konstante $\frac{1}{2}$ wird für die Vereinfachung folgender Ableitungsberechnungen hinzugefügt. Wird die additive Struktur von f zunächst ignoriert, so kann (77) als numerisches Optimierungsproblem in $\mathbb{R}^n$ interpretiert werden. Dabei kann f als Vektor der Vorhersagen an x_i aufgefasst werden:

$$f = (f(x_1), \ldots, f(x_n))^\top. \tag{78}$$

Die Grundidee von Gradient Boosting besteht darin, ein komplexes Vorhersagemodell nicht direkt zu optimieren, sondern sich der Lösung iterativ im Funktionenraum anzunähern.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

Ausgehend von einer initialen Vorhersage durch einen Startbaum f^0 werden iterativ Bäume auf die aktuellen Fehler trainiert. Diese Fehler werden als Pseudoresiduen bezeichnet und entsprechen den negativen Gradienten der Verlustfunktion bezüglich $f(x_i)$:

$$r_i^b = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f^{b-1}}. \tag{79}$$

Dabei ist der Begriff Pseudoresiduen darauf zurückzuführen, dass bei quadratischer Fehlerfunktion der negative Gradient den Residuen entspricht:

$$r_i^b = y_i - f^{b-1}(x_i). \tag{80}$$

Aufgrund der Auffassung als numerisches Optimierungsproblem wird der Gradient der Verlustfunktion mit Bezug auf die aktuellen Vorhersagen berechnet, um darauffolgend auf den Pseudoresiduen einen Entscheidungsbaum h^b zu fitten. Damit folgt Gradient Boosting der Idee des Gradientenverfahrens, sich zur Optimierung einer Zielfunktion entlang des Gradienten zu bewegen. Da die Zielfunktion minimiert werden soll, handelt es sich in diesem Kontext um den negativen Gradienten. Dabei beschreibt

$$f^b(x) = f^{b-1}(x) + \lambda h^b(x) \tag{81}$$

den Schritt in Richtung des negativen Gradienten mit einer Schrittlänge, die durch eine Lernrate λ gedämpft wird. f^b ist eine Aktualisierung von f^{b-1}, die durch den Baum h^b verbessert wurde.

Im Gegensatz zu Random Forest, bei dem die Vorhersage einer Region dem Mittelwert der Zielwerte entspricht, wird im Rahmen des Boosting-Verfahrens der Vorhersagewert (Blattwert) eines Baums h^b gezielt durch Minimierung der jeweiligen Verlustfunktion bestimmt. Handelt es sich dabei um die quadratische Verlustfunktion, so ergibt sich der optimale Blattwert als Mittelwert der aktuellen Residuen im jeweiligen Blatt.

Sei q^b die Zuteilungsfunktion aus (70) für den Baum h^b und $v_{q^b(x)}$ die zugehörigen Blattwerte. Dann werden die schwachen Lerner durch

$$h^b(x) = v_{q^b(x)} \tag{82}$$

beschrieben.

Im Vergleich zur Darstellung in der Ursprungsversion von Friedmann [65] wird in Algorithmus 6 keine optimale Schrittweite entlang des Gradienten bestimmt. Stattdessen wird für jeden Entscheidungsbaum h^b eine konstante Lernrate λ angesetzt.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Algorithmus 6: Gradient Boosting gemäß [26]

Input: Trainingsdaten $\mathbf{X}$, Anzahl Iterationen B, Verlustfunktion $L(y, f(x))$
Output: Gradient-Boosting-Modell f^{GB}
Ohne Beschränkung der Allgemeinheit gilt: $L(y, f(x)) = \frac{1}{2}(y - f(x))^2$
Initialisiere $f^0(x) := \arg\min_\gamma \sum_{i=1}^n L(y_i, \gamma) = \arg\min_\gamma \sum_{i=1}^n \frac{1}{2}(y_i - \gamma)^2 = \bar{y}$
for $b \in \{1, \ldots, B\}$ **do**
  for $i \in \{1, \ldots, n\}$ **do**
    Berechne Pseudoresiduen:
    $r_i^b := -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f^{b-1}} = y_i - f^{b-1}(x_i)$
  end
  Fitte einen Regressionsbaum h^b auf r_i^b mit Regionen R_j^b für $j = 1, \ldots, T_b$
  for $j \in \{1, \ldots, T_b\}$ **do**
    Berechne optimalen Blattwert:
    $v_j^b := \arg\min_\gamma \sum_{x_i \in R_j^b} L(y_i, f^{b-1}(x_i) + \lambda\gamma)$
  end
  Aktualisiere das Modell:
  $h^b(x) := v^b_{q^b(x)}$
  $f^b(x) := f^{b-1}(x) + \lambda h^b(x)$
end
return $f^{GB} := f^B$

Der XGBoost-Algorithmus ist wie Random Forest eine Ensemble-Methode und basiert auf demselben Grundgerüst wie das klassische Gradient Boosting, in dem sequentiell schwache Lerner h^b zu einer aktuellen Vorhersage addiert werden sollen. Die Erweiterung zum Gradient Boosting ist ein Strafterm in der Zielfunktion, der die Komplexität von h^b einschränken soll. Das Optimierungsproblem aus (77) wird demnach zu:

$$f^{\mathrm{XG}} = \arg\min_f \left\{ \sum_{i=1}^n L(y_i, f(x_i)) + \sum_{b=1}^B \Omega(h^b) \right\} \tag{83}$$

mit

$$\Omega(h^b) = \alpha T + \frac{1}{2}\nu \sum_{j=1}^J {v_j}^2. \tag{84}$$

Im Gegensatz zum klassischen Gradient Boosting wird für die Optimierung der Zielfunktion nicht die Taylorentwicklung erster Ordnung (Gradient) herangezogen, um dem Gradientenverfahren zu folgen, sondern eine Taylorentwicklung zweiter Ordnung, wodurch das Verfahren dem Newton-Raphson-Verfahren ähnelt. Dadurch wird im Zuge der Optimierung zusätzliche Information über die Verlustfunktion gewonnen, wodurch die Konvergenz zur Lösung beschleunigt wird.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Somit wird ein Iterationsschritt mit der Annahme einer quadratischen Fehlerfunktion

$$\sum_{i=1}^{n} \frac{1}{2} \left(y_i - f^{b-1}(x_i) - h^b(x_i)\right)^2 + \Omega(h^b) \tag{85}$$

zu

$$\sum_{i=1}^{n} \left(g_i^b \; h^b(x_i) + \frac{1}{2} \; d_i^b \; \left(h^b(x_i)\right)^2 \right) + \Omega(h^b) \tag{86}$$

vereinfacht. Dabei fällt $(L(y_i, f^{b-1}(x_i))$ für die Optimierung als konstanter Term weg und

$$g_i^b := \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f^{b-1}} = f^{b-1}(x_i) - y_i \tag{87}$$

ist der Gradient sowie

$$d_i^b := \left[\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2} \right]_{f=f^{b-1}} = 1 \tag{88}$$

die zweiten Ableitungen der Verlustfunktion, also die Diagonalelemente der Hessematrix.

Algorithmus 7: XGBoost gemäß [25]

Input: Trainingsdaten $\mathbf{X}$, Anzahl Iterationen B, Verlustfunktion $L(y, f(x))$
Output: XGBoost-Modell f^{XG}
Ohne Beschränkung der Allgemeinheit gilt: $L(y, f(x)) = \frac{1}{2}(y - f(x))^2$
Initialisiere $f^0(x) := \arg\min_\gamma \sum_{i=1}^n L(y_i, \gamma)$
for $b \in \{1, \ldots, B\}$ **do**
  for $i \in \{1, \ldots, n\}$ **do**
    Berechne Gradienten und Hesse-Diagonale:
    $g_i^b := \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f^{b-1}} = f^{b-1}(x_i) - y_i$
    $d_i^b := \left[\frac{\partial^2 L(y_i, f(x_i))}{\partial f(x_i)^2} \right]_{f=f^{b-1}} = 1$
  end
  Fitte einen Regressionsbaum h^b mit Regionen R_j^b für $j = 1, \ldots, T_b$
  for $j \in \{1, \ldots, T_b\}$ **do**
    Berechne optimalen Blattwert:
    $v_j^b := -\frac{\sum_{i\,:\,x_i \in R_j^b} g_i^b}{\sum_{i\,:\,x_i \in R_j^b} d_i^b + \nu} = -\frac{\sum_{i\,:\,x_i \in R_j^b} \left(f^{b-1}(x_i) - y_i\right)}{|R_j| + \nu}$
  end
  Setze $h^b(x) := v_{q^b(x)}^b$
  Aktualisiere das Modell:
  $f^b(x) := f^{b-1}(x) + h^b(x)$
end
return $f^{XG} := f^B$

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Anschließend wird der Baum h^b auf den Pseudoresiduen g_i^b gefittet, wobei die Beobachtungen individuell mit den Diagonalelementen der Hessematrix d_i^b gewichtet werden (im Fall der quadratischen Fehlerfunktion sind alle Gewichte konstant 1). Für eine gegebene Struktur des Baums h^b, der die Funktion in (86) minimieren soll, lassen sich die optimalen Blattwerte v_j^b analytisch bestimmen:

$$v_j^b = -\frac{\sum_{i\,:\,x_i \in R_j^b} g_i^b}{\sum_{i\,:\,x_i \in R_j^b} d_i^b + \nu} = -\frac{\sum_{i\,:\,x_i \in R_j^b} \left(f^{b-1}(x_i) - y_i\right)}{|R_j| + \nu}. \tag{89}$$

Die Möglichkeit, die optimalen Blattwerte v_j^b direkt analytisch zu bestimmen, stellt eine wesentliche Verbesserung gegenüber dem klassischen Gradient Boosting dar.

Gemäß des Modellvergleichs basierend auf den Standardparametern werden für das XGBoost-Modell folgende Parameter angesetzt:

- $B = 100$
- $\lambda = 0,1$
- $\alpha = 0$
- $\nu = 1$

5.6.1 Gewichtung

Der gewichtete Ansatz im XGBoost-Algorithmus wird über die Verlustfunktion gesteuert:

$$\sum_{i=1}^{n} w_i \cdot L(y_i, f(x_i)) + \sum_{b=1}^{B} \Omega(h^b). \tag{90}$$

Wodurch zum einen die Bestimmung der Baumstruktur und zum anderen die explizite Darstellung der optimalen Blattwerte beeinflusst werden:

$$v_j^b = -\frac{\sum_{i\,:\,x_i \in R_j^b} w_i\, g_i^b}{\sum_{i\,:\,x_i \in R_j^b} w_i\, d_i^b + \nu} = -\frac{\sum_{i\,:\,x_i \in R_j^b} w_i \left(f^{b-1}(x_i) - y_i\right)}{\sum_{i\,:\,x_i \in R_j^b} w_i + \nu}. \tag{91}$$

Auch beim XGBoost-Algorithmus erhofft man sich dadurch eine Verbesserung des Modellfits, wenn unsichere Beobachtungen weniger Einfluss zugesprochen bekommen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

5.6.2 Ergebnisse

	Mittelwertimputation		KNN-Imputation		MissForest	
Fehlend	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**	**Standard**	**Gewichtet**
0%	18,5	18,3	18,0	17,8	18,0	18,1
5%	19,3	19,1	18,2	18,2	18,4	18,2
10%	19,6	19,6	18,5	18,5	18,1	18,3
20%	27,0	26,8	18,7	19,1	19,1	18,8
30%	28,0	27,6	19,6	19,5	19,9	19,7

Tabelle 11: MAE-Vergleich bei XGBoost

Das XG-Boost-Modell liefert vergleichbare Ergebnisse wie Random Forest, wobei die Bagging-Methode minimal präzisere Vorhersagen erzielt.
Auffallend ist, dass für einen Anteil von 10% fehlender Daten MissForest bessere Ergebnisse mit sich zieht als KNN-Imputation. Bei Random Forest war Ähnliches beobachtbar, jedoch waren dort die Ergebnisse der beiden Imputationsmethoden exakt gleich.
Ansonsten kann man ähnliche Schlussfolgerungen bezüglich der Gewichtung und der Steigerung des MAE wie bei den vorherigen Modellen ziehen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

6 Resultate

In diesem Abschnitt werden die Einzelergebnisse aus dem vorherigen Kapitel zusammengefasst und interpretiert. Darauffolgend soll ein kompakter Überblick über weitere Optimierungsmöglichkeiten und die Verwendung der Vorhersagen im Kontext der Qualitätssicherung gegeben werden.

In Abbildung 20 ist für jeden prozentualen Anteil fehlender Werte die beste Kombination aus Imputationsmethode und Gewichtungsansatz je Vorhersagemodell dargestellt. Wie zu erwarten, liefern komplexere Vorhersagemethoden die besten Resultate. Dabei ist jedoch anzumerken, dass auch die linearen Modelle durchaus präzise Vorhersagen erzeugen, weshalb eine Empfehlung für den praktischen Einsatz aufgrund ihrer hohen Interpretierbarkeit und des geringen Rechenaufwands gerechtfertigt ist. Selbst bei einem hohen Anteil fehlender Werte bleiben die Vorhersagen mit einem MAE im Bereich von 23 MPa bis 27 MPa auf einem zufriedenstellenden Niveau. Die baumbasierten Ensemble-Methoden Random Forest und XGBoost erweisen sich als überlegen in diesem Modellvergleich. Für jeden Anteil fehlender Daten erhält man durch eine der beiden Methoden den niedrigsten Testfehler. Dass Ensemble-Methoden den simplen Modellen überlegen sind, war zu erwarten, weniger jedoch, dass Random Forest die Boosting-Methode XGBoost in mehreren Fällen übertrifft.

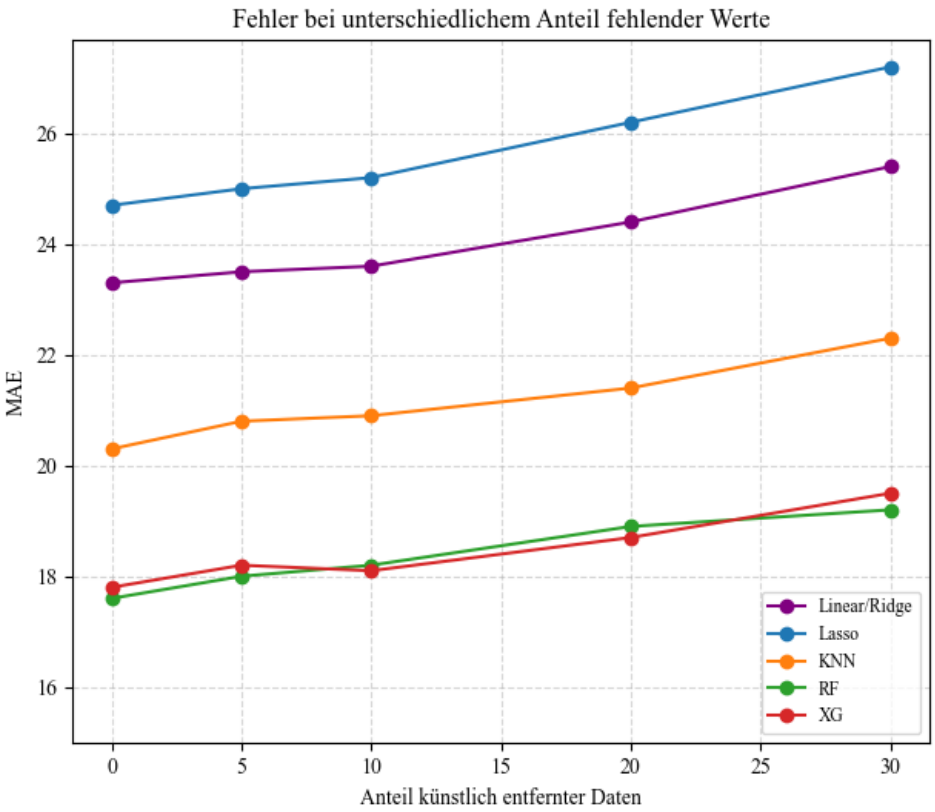

Abbildung 20: Visualisierung des MAE der jeweils besten Kombination aus Imputation, Vorhersage und Gewichtung basierend auf 5-Fold-Cross Validation.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Diverse Arbeiten, die auf einen Vergleich von baumbasierten Methoden abzielen, resümieren Boosting-Verfahren als Methoden für präzisere Vorhersagen. Beispielsweise zeigen Joharestani et al. in [66], dass XGBoost für die Vorhersage von Luftschmutzung besser abschneidet als die Bagging-Methode Random Forest. Ein niedrigerer Testfehler bei XGBoost ist ebenso für die Vorhersage von Hauspreisen in [67] beobachtbar.
Ein möglicher Grund dafür, dass XGBoost für den vorliegenden Datensatz keine Verbesserung zu Random Forest darstellt, liegt im Overfitting. Huang et al. zeigen dies im Zuge eines Vergleichs von fünf Vorhersagemodellen in [68]: Während XGBoost unter allen Modellen den niedrigsten Trainingsfehler aufweist, ist die Methode allen anderen Modellen bezüglich des Testfehlers unterlegen. Im Gegensatz zu XGBoost, wo Overfitting mithilfe der Parameter gesteuert werden muss, kommt es bei Random Forest selbst bei den Standardparametern zu keinen vergleichbaren Problemen. Folgend erzeugt XGBoost in dieser Arbeit für einige Fehldatenanteile ungenauere Vorhersagen als Random Forest. Ähnliche Schlussfolgerungen sind in [69] beobachtet worden: Hier zeigen sich keine signifikanten Unterschiede zwischen Random Forest und XGBoost bei Anwendung der Standardparameter.

In Kombination mit einer anschließenden Vorhersage ist die KNN-Imputation für den vorliegenden Datensatz die beste Wahl. Für alle betrachteten Vorhersagemodelle erweist sich die KNN-Imputation sowohl hinsichtlich des Imputationsfehlers als auch des Testfehlers den anderen Methoden gegenüber als überlegen. Darüber hinaus zeigt sich die KNN-Imputation im Gegensatz zur simplen Mittelwertimputation als robust gegenüber einem hohen Anteil fehlender Werte. Die Mittelwertimputation stellt insbesondere für einen niedrigen Fehldatenanteil eine mögliche Alternative dar, wenn der Fokus auf einem geringen Rechenaufwand liegt.
MissForest erzeugt für den vorliegenden Datensatz einen hohen Imputationsfehler, der nur durch komplexe Vorhersagemodelle ausgeglichen werden kann. Aufgrund der hohen Komplexität dieser Imputationsmethode und der wenig zufriedenstellenden Ergebnisse stellt die baumbasierte Methode keine empfehlenswerte Option für diesen Anwendungsfall dar.

Der vorgestellte Ansatz, den Einfluss unsicherer Daten für das Training des Vorhersagemodells zu reduzieren, liefert keine signifikante Verbesserung des Vorhersagefehlers. Für die linearen Modelle verringert die Gewichtung den MAE für einen hohen Anteil fehlender Werte minimal. Hier lässt sich ein erwartbares Muster erkennen, denn die individuelle Beobachtungsgewichtung wirkt sich insbesondere bei einem höheren Anteil fehlender Werte aus. Bei den nichtlinearen Modellen hingegen ist so ein Muster nicht vorliegend, weshalb zusammenfassend einzuräumen ist, dass der Gewichtungsansatz für den vorliegenden Datensatz vernachlässigbar ist.

Insgesamt unterstreichen die Ergebnisse die Notwendigkeit eines Modellvergleichs zur Bestimmung jener Kombination mit dem geringsten Vorhersagefehler, da sowohl die Wahl der Imputationsmethode als auch die anschließende Vorhersage stark vom zugrunde liegenden Datensatz abhängen. Auch die eingangs formulierte Frage nach der Möglichkeit, einzelne Zugversuche zu reduzieren, lässt sich positiv beantworten. Unter Einsatz einer geeigneten Imputationsmethode zeigen sich die Vorhersagen robust gegenüber imputierten Daten und liefern selbst bei hohem Anteil fehlender Daten beachtliche Ergebnisse.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Im weiteren Verlauf der Arbeit wird für die Optimierung der Tuningparameter und die Verwendung der Ergebnisse die beste Kombination aus Imputation, Vorhersage und Gewichtung bei 0% künstlich erzeugten fehlenden Werten herangezogen.

Für diesen Fall erweist sich die Kombination aus **KNN-Imputation** mit **Random Forest** und einem **Gewichtungsansatz** mit einem mittleren absoluten Fehler von **17,6 MPa** als die beste Wahl. Dabei sei anzumerken, dass der Gewichtungsansatz keinen signifikanten Unterschied zum Standardmodell erzeugt, was den Einsatz des Modells ohne Gewichtung mit einem MAE in Höhe von 17,7 MPa ebenfalls rechtfertigt. Speziell in diesem Fall, bei dem fehlende Daten nicht künstlich erzeugt werden, erzeugt der gewichtete Ansatz nur deshalb marginal präzisere Vorhersagen, weil möglicherweise die betroffenen Beobachtungen grundsätzlich weniger Aussagekraft für die Vorhersage der Zielvariable haben.

Abbildung 21 soll ein weiteres Mal die beachtlichen Vorhersageergebnisse verdeutlichen:

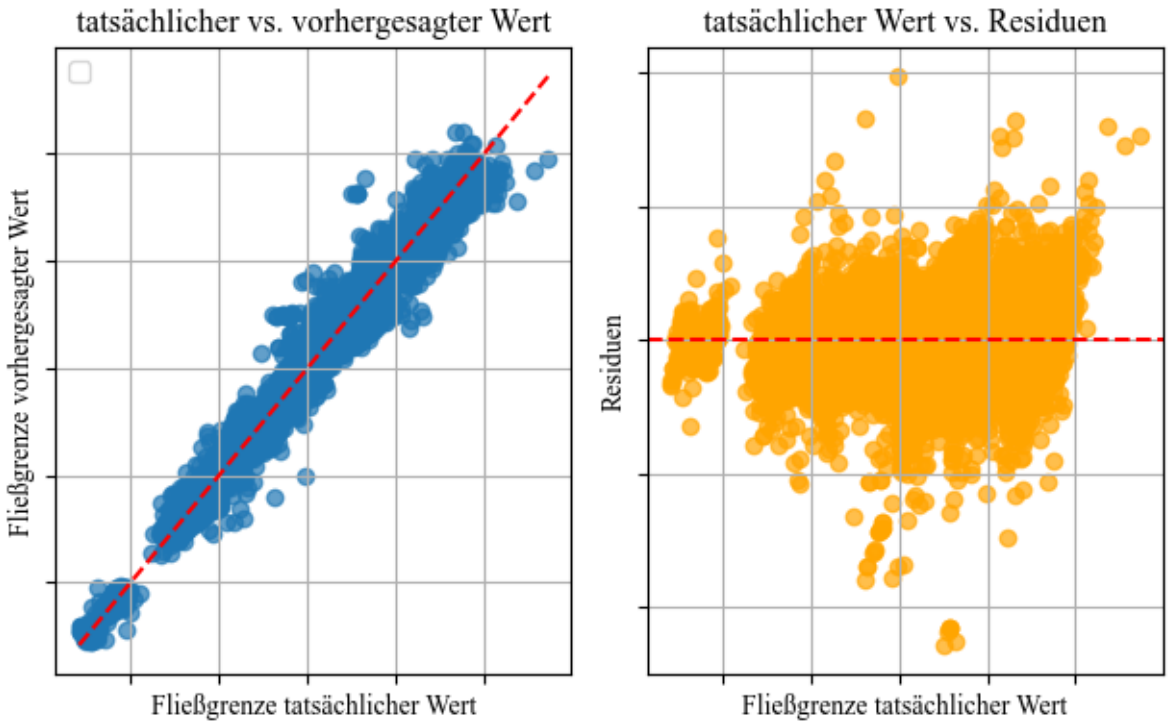

Abbildung 21: Links: Gegenüberstellung der Vorhersagen und der wahren Werte. Rechts: Gegenüberstellung der Vorhersagen und der zugehörigen Residuen.

6.1 Parametertuning

Zur möglichen Verringerung des Vorhersagefehlers kann im Rahmen der Cross-Validation eine Optimierung der Parameter durchgeführt werden. Hierfür werden mit einem Grid-Search verschiedene Werte in Erwägung gezogen, um jene Kombination zu identifizieren, die den niedrigsten MAE erzielt. In dieser Arbeit wird der Einfluss von drei Parametern

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

auf die Kombination aus KNN-Imputation, Random Forest und Gewichtung untersucht. Zur Analyse eines einzelnen Parameters werden unterschiedliche Werte getestet, unter der Annahme, dass die weiteren Parameter mit ihren Standardwerten fixiert bleiben.

Die in (22) eingeführte exponentielle Gewichtung in Abhängigkeit von der Anzahl der fehlenden Werte einer Beobachtung basiert auf dem Parameter α, der im Zuge des Modellvergleichs für alle Kombinationen mit $\alpha = 0{,}1$ festgelegt wurde. Für die Analyse des Gewichtungsparamaters werden vier unterschiedliche Werte für α verglichen:

$\boldsymbol{\alpha}$	$\mathbf{0,05}$	$\mathbf{0,10}$	$\mathbf{0,20}$	$\mathbf{0,30}$
MAE	17,66	17,63	17,66	17,62

Tabelle 12: Parametertuning für den Parameter der Gewichtung α. In grün der minimale Wert für die optimale Wahl des Parameters unter der Annahme, dass die anderen Parameter fix sind. In grau der Standardparameter.

Aus Tabelle 12 geht hervor, dass der Parameter für die Gewichtung einen geringen Einfluss auf die Modellgüte hat. Der Einsatz verschiedener Werte für α führt lediglich zu einer Variation des MAE um etwa 0,1 MPa. Insofern kann der Parameter, zumindest für den vorliegenden Datensatz, als weitgehend unwesentlich eingestuft werden.

Ein wichtiger Parameter beim Einsatz des Random-Forest-Modells ist die Anzahl der Bäume B, die zur Aggregation herangezogen werden. Da eine hohe Anzahl an Bäumen in der Regel nicht zu Overfitting führt, ist in diesem Fall eher die Mindestanzahl an Bäumen von Interesse, um eine ausreichende Modellgenauigkeit zu gewährleisten. Standardmäßig wird dieser Parameter mit $B = 100$ festgesetzt. In der Regel ist davon auszugehen, dass der Vorhersagefehler mit steigender Baumanzahl sinkt. Das ist durch die beschriebene Verringerung der Varianz infolge der Aggregierung vieler einzelner schwacher Lerner begründet. Für den vorliegenden Datensatz bestätigt sich diese Annahme:

B	**20**	**30**	**50**	**100**	**200**
MAE	17,85	17,83	17,72	17,63	17,57

Tabelle 13: Parametertuning für die Anzahl der Bäume B für Random Forest

Je mehr schwache Lerner, also einzelne Regressionsbäume, zur Bestimmung des Vorhersagewerts eingesetzt werden, desto niedriger fällt der resultierende Vorhersagefehler aus. Im Vergleich zum Gewichtungsparameter α lässt sich eine minimal stärkere Auswirkung auf den MAE beobachten. Besonders hervorzuheben sind die guten Ergebnisse bei einer geringen Anzahl von Bäumen, etwa bei $B = 20$. Folgend daraus kann eine Ensemble-Methode bereits mit wenigen Bäumen vielversprechende Vorhersagen liefern.

Der dritte Parameter steuert die Tiefe n_t der Entscheidungsbäume. Diese beschreibt die maximale Anzahl an Knoten entlang des längsten Pfades von der Wurzel bis zu einem Blatt. Standardmäßig wird die Tiefe nicht beschränkt ($n_t = \texttt{None}$). Der Einsatz einer maximalen Tiefe kann einerseits die Modellkomplexität reduzieren, andererseits jedoch den

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Bias erhöhen. Vor der Analyse dieses Parameters ist anzumerken, dass eine bestimmte Variable entlang eines Pfades mehrfach vorkommen kann. Die Tiefe eines Baums ist somit nicht durch die Anzahl der Variablen beschränkt, weshalb im vorliegenden Fall auch Werte über 30 berücksichtigt werden können.

$\mathbf{n_t}$	**None**	**5**	**10**	**20**	**30**	**50**
MAE	17,63	21,56	18,24	17,65	17,65	17,63

Tabelle 14: Parametertuning für die maximale Tiefe eines Baums bei Random Forest

Aus Tabelle 14 lässt sich schlussfolgern, dass eine strikte Einschränkung der Baumtiefe n_t die Vorhersagegenauigkeit negativ beeinflussen kann.

Ab einer Tiefe von 30 Knoten stabilisiert sich der MAE im Bereich zwischen 17,63 MPa und 17,65 MPa. In diesem Fall liefert der Standardparameter `None` die besten Ergebnisse. Sofern die Modellkomplexität nicht vorrangig ist, ist dieser Parameter bereits optimal gewählt, und ein weiteres Tuning erübrigt sich.

Testet man jede Kombination der drei Parameter, so liefert folgende Zusammensetzung den niedrigsten Vorhersagefehler:

$\boldsymbol{\alpha}$	**B**	$\mathbf{n_t}$
0,3	200	`None`

Tabelle 15: Optimale Kombination der untersuchten Parameter

Die Kombination stimmt somit mit jenen Werten der Parameter überein, die auch in der Einzelanalyse als optimal identifiziert wurden. Der daraus resultierende Vorhersagefehler beträgt **17,56 MPa**. Im Vergleich zum Modell mit den Standardparametern zeigt sich keine signifikante Verbesserung der Vorhersagegenauigkeit durch das Parametertuning. Dennoch liefert die Einzelanalyse wichtige Erkenntnisse über die Wirkung der einzelnen Parameter.

6.2 Feature Importance

Eine bedeutsame Eigenschaft von Random Forest ist die Auswertung der Feature Importance. Ohne diese wäre die Interpretation des Modells stark limitiert, da man anders als bei einem einzelnen Entscheidungsbaum das Modell nicht grafisch darstellen kann. Die Auswertung der Variablen mit dem größten Einfluss auf die Beschreibung der Zielgröße ist dementsprechend von Bedeutung und ermöglicht Annahmen aus der Praxis zu überprüfen.

Die Wichtigkeit einer Variable ergibt sich pro Baum aus der Reduktion der RSS, die durch eine Aufteilung basierend auf dieser Variable entsteht. Anschließend wird der Mittelwert dieser Fehlerreduktionen über alle B Bäume genommen. Ein hoher Wert deutet auf eine hohe Wichtigkeit des entsprechenden Prädiktors hin. In der Python-Implementierung

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub

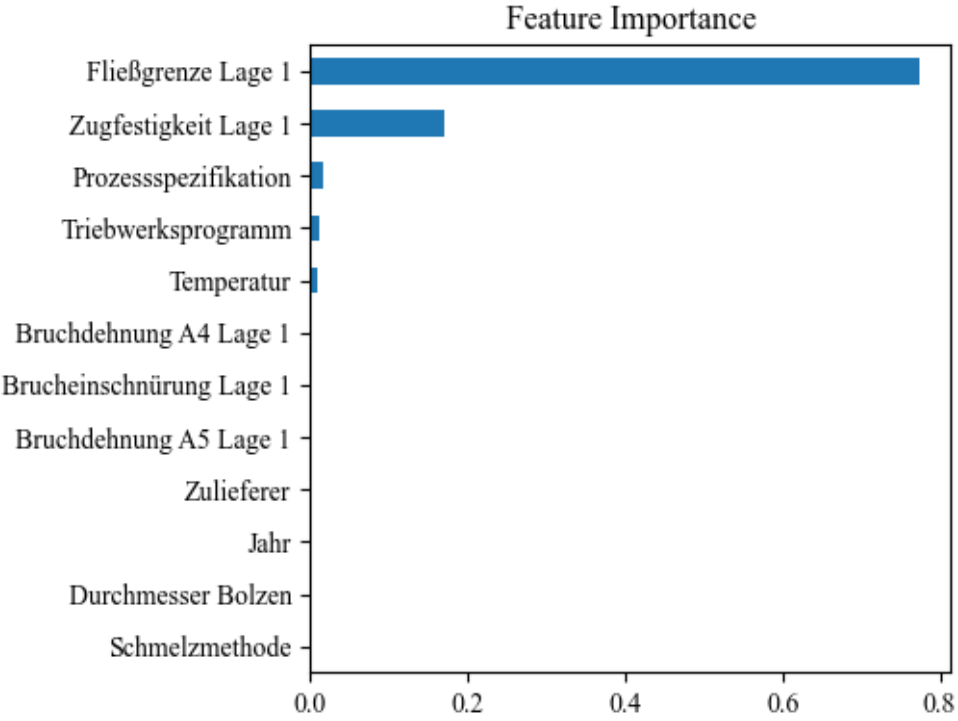

Abbildung 22: Feature Importance bei Random Forest als normalisierter Wert. Ein hoher Wert spricht für einen großen Einfluss bei der Erklärung der Zielgröße.

durch **scikit-learn** werden diese Mittelwerte anschließend normalisiert, sodass die relative Bedeutung jeder erklärenden Variable angegeben ist und die Summe aller Feature Importances 1 ergibt.

Gemäß Abbildung 22 bestätigt sich die Annahme aus Abschnitt 2 bezüglich der Wichtigkeit der Ergebnisse am Testring. So beruhen die guten Vorhersagen zum Großteil auf den Informationen, die im Rahmen eines Zugversuchs an der Testringlage gesammelt wurden. Erwartungsgemäß ist die Feature Importance bei der Variable Fließgrenze Lage 1 am höchsten. Diese starke Korrelation zwischen den Fließgrenzen unterschiedlicher Lagen ist wünschenswert und ein abweichendes Ergebnis müsste hinterfragt werden. Um den seltenen Fall aufzufangen, bei dem die Fließgrenze im Inneren des Bauteils im Vergleich zu der am Testring ungewöhnlich niedrig ist, sollen die weiteren Variablen unterstützend dienen. Zu diesen Variablen zählen unter anderem die Testtemperatur sowie Informationen aus der CutUpID wie die Prozessspezifikation und das Triebwerksprogramm.
Die weiteren Variablen tragen nur wenig zur Beschreibung der Zielgröße bei. Das ist vermutlich dadurch zu begründen, dass die Informationen dieser Variablen bereits durch wichtigere Variablen wiedergegeben wurden. Etwa ist die Fließgrenze an der Testringlage stark mit der Dehnung korreliert oder wird die Zuliefererroute durch die Prozessspezifikation beschrieben.
Aus Gründen der Übersicht wurden jene Variablen aus Abbildung 22 entfernt, die mittels One-Hot-Encoding in eine numerische Form überführt wurden. Diese zeigten lediglich einen geringen Einfluss im Modell. Dadurch fällt mit dem Material eine erklärende

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Variable weg, der man hohe Bedeutung in der Praxis zuschreibt. Auch in diesem Fall könnte man die Korrelation zu anderen Variablen oder die durchgeführte Kodierung als mögliche Gründe für die niedrige Wichtigkeit deuten. Dazu sei erwähnt, dass im Setting von Entscheidungsbäumen eine Kodierung im Allgemeinen nicht notwendig ist. Wie bereits erläutert, können baumbasierte Methoden gut mit kategorialen Variablen umgehen. Dennoch war es im Rahmen des Modellvergleichs insbesondere im Hinblick auf die linearen Modelle und die Imputationsmethoden notwendig, die Daten im Vorfeld entsprechend einheitlich aufzubereiten.

6.3 Entscheidungsprozess

Als abschließender Unterpunkt soll dargestellt werden, wie sich die Vorhersagen in einen datengetriebenen Entscheidungsprozess integrieren lassen. Um diese als Entscheidungsgrundlage heranziehen zu können, müssen die damit verbundenen Unsicherheiten quantifiziert werden. Dabei sind zum einen die Streuung der Modellvorhersagen selbst und zum anderen die Abweichung der Vorhersage vom wahren Wert von Bedeutung. Letzteres wird bereits in aggregierter Form durch den durchschnittlichen Vorhersagefehler in Höhe von 17,6 MPa beschrieben. Für die Beschreibung der Streuung der Vorhersagen eignen sich Vorhersageintervalle. Vorhersageintervalle haben das Ziel, für zukünftige Beobachtungen einen Bereich anzugeben, in dem die Vorhersage mit einer bestimmten Wahrscheinlichkeit enthalten ist. Es wird also nicht nur der wahrscheinlichste Wert, sondern auch die damit verbundene Unsicherheit berücksichtigt. Um die Abweichung einer Vorhersage vom unbekannten tatsächlichen Wert zu beschreiben, können Konfidenzintervalle hilfreiche Informationen liefern. Ein Konfidenzintervall stellt einen Bereich dar, in dem der tatsächliche Wert mit einer bestimmten Wahrscheinlichkeit liegt. Sowohl Vorhersage- als auch Konfidenzintervalle sind bei einem unparametrischen Modell wie Random Forest, im Gegensatz zur Linearen Regression, nicht analytisch ableitbar, sondern müssen geschätzt werden.

Für Random Forest gibt es eine Vielzahl an Möglichkeiten zur Approximation von Vorhersageintervallen. Beispielsweise können die sogenannten Out-of-bag-Beobachtungen herangezogen werden. Das sind jene Beobachtungen, die beim Ziehen einer Bootstrap-Stichprobe für einzelne Entscheidungsbäume nicht berücksichtigt werden. Da diese Beobachtungen nicht zum Training des jeweiligen Baums verwendet werden, können sie für eine Vorhersage durch genau diesen Baum herangezogen werden. Gemäß [9] liefert dieser Ansatz für jede der n Trainingsbeobachtungen eine Vorhersage. Die von Zhang et al. in [70] vorgestellte Konstruktion von Vorhersageintervallen basiert auf diesen Vorhersagen. Für jede der n Vorhersagen wird das Residuum bestimmt, woraus eine Verteilung der Fehler resultiert. Durch diese Verteilung lassen sich Quantile ableiten, auf deren Basis die Konstruktion der Intervalle erfolgt.

Eine weitere Option, die ebenfalls auf einem einzelnen Random-Forest-Modell basiert, sind die von Meinshausen in [71] vorgestellten Quantile Regression Forests. Dabei enthält jedes Blatt, neben der Information über den Mittelwert der Zielwerte, zusätzlich die Zielwerte der blattzugehörigen Beobachtungen. Somit wird eine Menge an Zielwerten von Beobachtungen gesammelt, die ähnlich zu einer Testbeobachtung sind, für die ein Intervall bestimmt werden soll.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

In diesem Fall werden die Quantile auf Basis der Verteilung der Zielwerte im zugeordneten Blatt bestimmt und zur Konstruktion der Vorhersageintervalle verwendet.

Da im Rahmen dieser Arbeit die Konstruktion der Vorhersageintervalle gemäß der Modellpipeline unabhängig von der Modellwahl erfolgen soll, wird ein verallgemeinerter Ansatz zur Erstellung dieser Intervalle gewählt. Wie für die Vorhersage bei Random Forest kommt dafür das Bootstrap-Verfahren zum Einsatz.
Auf Basis dessen werden 100 verschiedene Datensätze aus den Trainingsdaten erstellt und damit eine gleiche Anzahl an Vorhersagen für eine Beobachtung generiert. Das ermöglicht die Bestimmung der Quantile zur Festsetzung einer oberen und einer unteren Intervallsgrenze. Beim Random-Forest-Modell ergibt sich dadurch eine gewisse Redundanz, da bereits für das Training des Modells unterschiedliche Datensätze generiert werden. Insofern wären die eben vorgestellten Ansätze vorzuziehen. Da in diesem Kapitel aber nur ein kompakter Überblick über die Möglichkeiten zur Weiterverarbeitung der Vorhersagen gegeben werden soll, werden diese im Rahmen dieser Arbeit nicht weiter betrachtet.

In der Qualitätssicherung von Rotorscheiben ist insbesondere die untere Intervallgrenze von Bedeutung. Sowohl für den tatsächlichen Zielwert als auch den Vorhersagewert ist die obere Intervallgrenze in diesem Zusammenhang vernachlässigbar. Aus diesem Grund besteht das verfolgte Ziel darin, die Intervalle zu vergrößern, sodass nicht nur die Vorhersage, sondern auch der tatsächliche Wert mit einer Wahrscheinlichkeit von 95% oberhalb der unteren Intervallgrenze liegt. Dafür wird ein simpler Ansatz verfolgt, bei dem die Intervallgrenzen für jede Beobachtung gleichermaßen skaliert werden, sodass für 95% der Trainingsdaten der Zielwert oberhalb der unteren Grenze liegt. Dieser trainierte Skalierungswert kann anschließend für Testbeobachtungen angesetzt werden. Die große Anzahl an Trainingsbeobachtungen rechtfertigt die Wahl dieses einfachen Vorgehens.

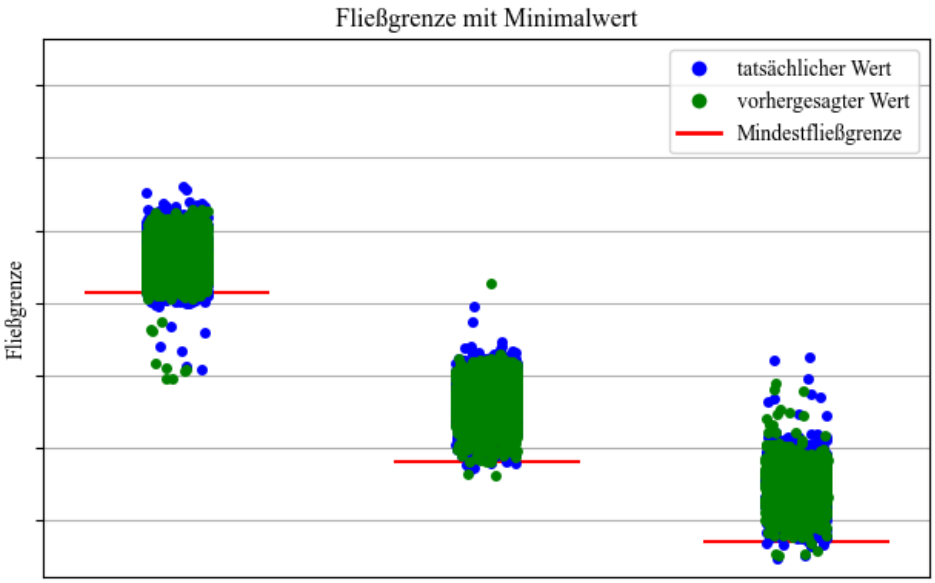

Abbildung 23: Darstellung des Minimalwerts eines Materials bei drei unterschiedlichen Temperaturen. Für die Verwendung eines Bauteils muss die Fließgrenze oberhalb dieses Werts liegen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Mithilfe dieser Intervalle können nun für einzelne Vorhersagen Entscheidungen getroffen werden. Wie bereits angeschnitten, gibt es für jede Kombination aus Temperatur und Material einen Minimalwert der Fließgrenze. Damit ein Bauteil als ordnungsgemäß eingestuft werden kann, muss die Fließgrenze diesen Minimalwert für jede Testlage übersteigen. Abbildung 23 stellt diesen Minimalwert in Kombination mit den Vorhersagen und den wahren Werten dar.

Nachdem die Fließgrenze bei den Testlagen im Inneren eines Bauteils nicht messbar ist, werden stattdessen die Vorhersagen herangezogen. Um die Unsicherheiten ebenfalls zu berücksichtigen, werden die konstruierten Intervalle angesetzt. Damit kann ein Bauteil als ordnungsgemäß angesehen werden, wenn die untere Intervallgrenze höher als der Minimalwert ist. In diesem Fall sind sowohl die Vorhersage als auch der tatsächliche Wert mit einer bestimmten Wahrscheinlichkeit oberhalb des Minimalwerts. Auf Basis der Trainingsdaten und der Kenntnis über die tatsächlichen Fließgrenzen können Beobachtungen in vier Kategorien unterteilt werden, um dieses Vorgehen zu überprüfen:

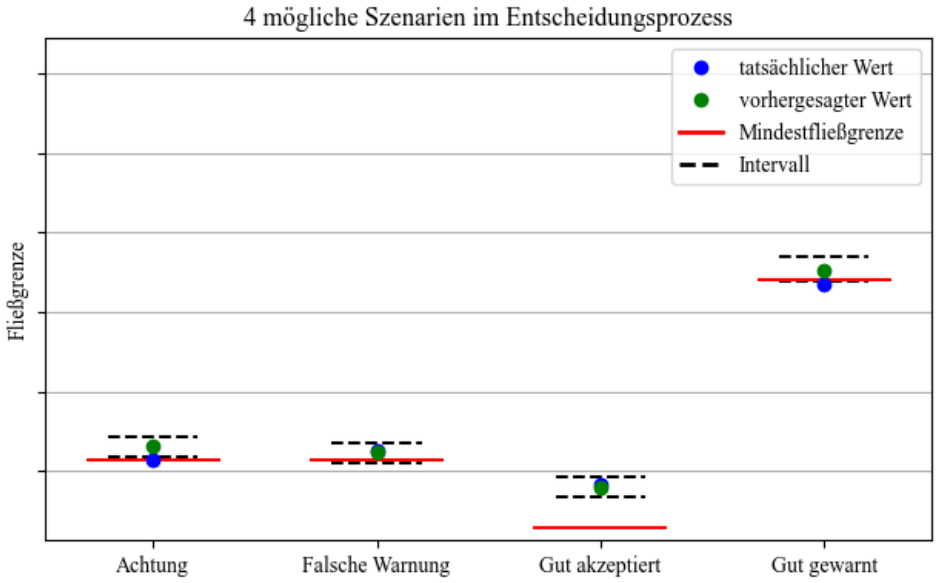

Abbildung 24: Darstellung der vier möglichen Szenarien bei Kenntnis über den tatsächlichen Wert der Fließgrenze.

Abbildung 24 stellt die vier möglichen Szenarien dar. Mit dem Fall „Achtung“ tritt der ungünstigste Fall ein. Dabei fällt ein Bauteil auf Basis der Vorhersage nicht als ungewöhnlich auf, aber der tatsächliche Wert liegt unterhalb des Minimalwerts. Dieses Szenario soll möglichst ausgeschlossen werden. Sollte der tatsächliche Wert der Fließgrenze unterhalb des Minimalwerts liegen, so erhofft man sich eine entsprechende Warnung. Dieser Fall geht mit „Gut gewarnt“ einher. „Falsche Warnung“ kommt vor, wenn die untere Intervallgrenze unterhalb des Minimalwerts liegt, aber die Fließgrenze ausreichend groß ist. Das bestmögliche Szenario ist „Gut akzeptiert“, bei dem sowohl die untere Intervallgrenze als auch der tatsächliche Wert oberhalb des Minimalwerts der Fließgrenze liegen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Diese Integration der Vorhersagen in den Entscheidungsprozess zeigt sich im Grunde als vielversprechend in dem Sinne, dass der erstgenannte Fall in wenigen Fällen eintritt. Allerdings führt dieser Ansatz zu einer hohen Anzahl falscher Warnungen. Das liegt daran, dass die Intervalle durch die Skalierung zur Erfassung der tatsächlichen Werte zu groß konstruiert sind. Insofern muss ein Kompromiss eingegangen werden. Denn je größer die Intervalle, desto wahrscheinlicher werden tatsächliche Werte eingefangen. Allerdings führen große Intervalle zu vielen falschen Warnungen, da die untere Intervallgrenze häufig unterhalb des Minimalwerts liegt. Umgekehrt führen kleine Intervalle zu weniger falschen Warnungen, sind jedoch als Approximation von Konfidenzintervallen tendenziell unbrauchbar.

Resultierend daraus lässt sich zusammenfassen, dass die Integration von Vorhersagen basierend auf skalierten Intervallen ein interessanter Ansatz ist, jedoch noch Verbesserungspotenzial im Hinblick auf den Konflikt zwischen Sicherheit und Fehlwarnungen aufweist.

Eine alternative Entscheidungsgrundlage bietet die Verteilung der Absolutwerte der Residuen. Daraus lässt sich jene Abweichung der Vorhersage vom tatsächlichen Zielwert ableiten, bei der 95% der Trainingsdaten einen geringeren Vorhersagefehler aufweisen. Für das Random-Forest-Modell beträgt dieser Schwellenwert 45 MPa. Im Gegensatz zur Intervallbildung auf Basis des Bootstrap-Verfahrens mit anschließender Skalierung ist der Unsicherheitsbereich in diesem Fall konstant und beträgt für alle Beobachtungen jeweils ± 45 MPa um die Vorhersage.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

7 Conclusio

In dieser Arbeit konnte erfolgreich eine vollständige Modellpipeline vorgestellt werden, die neben der Erzeugung von Vorhersagen und deren Integration in Entscheidungsprozesse auch die oftmals wenig beleuchtete Datenaufbereitung umfassend und detailliert beschreibt. Neben der Bereinigung von Ausreißern und der Kodierung kategorialer Variablen stand insbesondere der Umgang mit fehlenden Werten im Vordergrund.
Um in diesem Zusammenhang die Reduktion experimenteller Zugversuche realistisch zu simulieren, wurden verschiedene Anteile fehlender Werte nach einer definierten Struktur künstlich erzeugt. Die Analyse basierte auf der Beschreibung des zugrunde liegenden Fehlermechanismus sowie der Fehlerstruktur des vorliegenden Datensatzes. Anschließend wurden drei Imputationsmethoden (Mittelwertimputation, KNN-Imputation und MissForest) miteinander verglichen. Dabei erzielte die KNN-Imputation den geringsten Imputationsfehler und generierte die realistischsten Ersatzwerte.
Im Anschluss wurde die Auswirkung der imputierten Daten auf die Vorhersagequalität untersucht. Dabei wurden sechs Regressionsmodelle (Lineare Regression, Ridge Regression, Lasso Regression, k-Nearest-Neighbor, Random Forest und XGBoost) sowie ein Ansatz zur individuellen Gewichtung von Vorhersagen abhängig von der Anzahl fehlender Daten in Betracht gezogen. Der Gewichtungsansatz brachte jedoch keine signifikante Verbesserung gegenüber den Standardmodellen.
Für fast alle Kombinationen aus Fehldatenanteil und Vorhersagemodell erwies sich die KNN-Imputation als robusteste Methode. Bei geringem Anteil fehlender Werte erzielte die Mittelwertimputation vergleichbare Ergebnisse bei geringerem Rechenaufwand. Die besten Vorhersagen wurden mit dem Random-Forest-Modell erzeugt. Ohne künstlich entfernte Daten lag der mittlere absolute Fehler (MAE) bei $17,6$ MPa. Durch Identifikation der optimalen Tuningparameter konnte dieser Fehler leicht auf $17,56$ MPa gesenkt werden.
Eine anschließende Analyse der Feature Importance bestätigte Praxisannahmen hinsichtlich der Relevanz bestimmter Kennwerte. Abschließend wurde ein kompakter Überblick über die Einbindung der Vorhersagen in den Entscheidungsprozess gegeben.
Insgesamt liefert diese Arbeit einen praxisnahen Beitrag zu einer datenbasierten Werkstoffbewertung in der Qualitätssicherung. Insbesondere wurde gezeigt, dass selbst bei eingeschränkter Datenverfügbarkeit robuste Vorhersagen möglich sind, wodurch eine Reduktion experimenteller Zugversuche realisiert werden kann.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Literaturverzeichnis

[1] Alice Zheng und Amanda Casari. *Feature Engineering for Machine Learning*. O'Reilly Media, 2018. ISBN: 978-1-491-95323-5.

[2] Uttam Bhandari, Md Rumman Rafi, Congyan Zhang und Shizhong Yang. „Yield strength prediction of high-entropy alloys using machine learning". In: *Materials Today Communications* 26 (2021).

[3] Christian Hinte. *Hochentropielegierungen: Zwischen Chaos und Ordnung*. Zugriff am 6. August 2025. Institut für Werkstoffkunde, Leibniz Universität Hannover, 2021. URL: `https://phi-hannover.de/hochentropielegierungen-zwischen-chaos-und-ordnung/`.

[4] C.V. Haden, Peter C. Collins und Gary Harlow. „Yield Strength Prediction of Titanium Alloys". In: *JOM* 67.6 (2015).

[5] Aditi Mahajan, Simran Bajoliya, Shanu Khandelwal, Ruchika Guntewar und Navneet Arora. „Comparison of ML algorithms for prediction of tensile strength of polymer matrix composites". In: *Materials Today: Proceedings* (2022).

[6] Weiguo Li, Xianhe Zhang, Haibo Kou, Ruzhuan Wang und Daining Fang. „Theoretical prediction of temperature dependent yield strength for metallic materials". In: *International Journal of Mechanical Sciences* 105 (2016), S. 273–278.

[7] Gideon A. Lyngdoh, Mohd Zaki, N. M. Anoop Krishnan und Sumanta Das. „Prediction of concrete strengths enabled by missing data imputation and interpretable machine learning". In: *Cement and Concrete Composites* 128 (2022).

[8] Earl Yin. „Evaluation of Imputation Methods for Handling Missing Data in Mechanical Materials Datasets". In: (2024).

[9] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani und Jonathan Taylor. *An Introduction to Statistical Learning: with Applications in Python*. 3. Aufl. Springer, 2023. ISBN: 978-3-031-38746-3.

[10] Max Kuhn und Kjell Johnson. *Applied Predictive Modeling*. Springer, 2013. ISBN: 978-1-4614-6848-6.

[11] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke und Travis E. Oliphant. „Array programming with NumPy". In: *Nature* 585.7825 (2020), S. 357–362.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

[12] Wes McKinney. „Data Structures for Statistical Computing in Python“. In: *Proceedings of the 9th Python in Science Conference*. 2010.

[13] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt und SciPy 1.0 Contributors. „SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python“. In: *Nature Methods* 17.3 (2020), S. 261–272.

[14] John D. Hunter. „Matplotlib: A 2D Graphics Environment“. In: *Computing in Science & Engineering* 9.3 (2007), S. 90–95.

[15] Vic Barnett und Toby Lewis. *Outliers in Statistical Data*. 3rd. Wiley, 1994. ISBN: 978-0-471-93094-5.

[16] Boris Iglewicz und David C. Hoaglin. *How to Detect and Handle Outliers*. ASQC Quality Press, 1993. ISBN: 978-0-873-89247-6.

[17] Daniele Micci-Barreca. „A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems“. In: *ACM SIGKDD Explorations Newsletter* 3.1 (2001), S. 27–32.

[18] Florian Pargent, Florian Pfisterer und Bernd Bischl. „Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features“. In: *Computational Statistics* 37 (2022), S. 2671–2692.

[19] Md Manjurul Ahsan, M. A. Parvez Mahmud, Pritom Kumar Saha, Kishor Datta Gupta und Zahed Siddique. „Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance“. In: *Technologies* 9.3 (2021), S. 52.

[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot und Édouard Duchesnay. „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12 (2011), S. 2825–2830.

[21] Thomas M. Cover und Peter E. Hart. *Nearest Neighbor Pattern Classification*. Bd. 13. 1. IEEE Transactions on Information Theory, 1967, S. 21–27.

[22] Arthur E. Hoerl und Robert W. Kennard. „Ridge Regression: Biased Estimation for Nonorthogonal Problems“. In: *Technometrics* 12.1 (1970), S. 55–67.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

[23] Leo Breiman. „Random Forests“. In: *Machine Learning* 45.1 (2001), S. 5–32.

[24] Robert Tibshirani. „Regression Shrinkage and Selection via the Lasso“. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58.1 (1996), S. 267–288.

[25] Tianqi Chen und Carlos Guestrin. „XGBoost: A Scalable Tree Boosting System“. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016, S. 785–794.

[26] Trevor Hastie, Robert Tibshirani und Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd. Springer, 2009. ISBN: 978-0-387-84857-0.

[27] Annette M. Molinaro, Richard Simon und Ruth M. Pfeiffer. „Prediction Error Estimation: A Comparison of Resampling Methods“. In: *Bioinformatics* 21.15 (2005), S. 3301–3307.

[28] Douglas M. Hawkins, Subhash C. Basak und Denise Mills. „Assessing Model Fit by Cross-Validation“. In: *Journal of Chemical Information and Computer Sciences* 43.2 (2003), S. 579–586.

[29] Ron Kohavi. „A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection“. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)*. Bd. 2. Aug. 1995, S. 1137–1143.

[30] Roderick Little und Donald Rubin. *Statistical Analysis with Missing Data*. 3. Aufl. Wiley Series in Probability and Statistics. Wiley, 2019. ISBN: 978-0-470-52679-8.

[31] Akbar K. Waljee, Ashin Mukherjee, Amit G. Singal, Yiwei Zhang, Jeffrey Warren, Ulysses Balis, Jorge Marrero, Ji Zhu und Peter D. Higgins. „Comparison of imputation methods for missing laboratory data in medicine“. In: *BMJ Open* 3.8 (2013).

[32] Ya-Han Hu, Ruei-Yan Wu, Yen-Cheng Lin und Ting-Yin Lin. „A novel MissForest-based missing values imputation approach with recursive feature elimination in medical applications“. In: *BMC Medical Research Methodology* 24 (2024).

[33] Shweta Tiwaskar, Sandip Thite und Mamoon Rashid. „A Comparative Analysis of Machine Learning Imputation Techniques for MAR Missingness“. In: *Advances in Nonlinear Variational Inequalities* 28.3 (2025).

[34] Jeanne A. Saunders, Nancy Morrow-Howell, Edward Spitznagel, Peter Dore, Enola K. Proctor und Richard Pescarino. „Imputing Missing Data: A Comparison of Methods for Social Work Researchers“. In: *Social Work Research* 30.1 (2006), S. 19–31.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

[35] Intan Nirmala, Hari Wijayanto und Khairil Anwar Notodiputro. „Prediction of Undergraduate Student's Study Completion Status Using MissForest Imputation in Random Forest and XGBoost Models". In: *ComTech Computer Mathematics and Engineering Applications* 13.1 (2022), S. 53–62.

[36] Li Li und Darran P. Goshawk. „Comparison of Random Forest and Multiple Imputation for Imputing Missing Data: a Case Study of the Education Panel Survey of the City of China". In: ().

[37] Serena G. Liao, Yan Lin, Dongwan D. Kang, Divay Chandra, Jessica Bon, Naftali Kaminski, Frank C. Sciurba und George C. Tseng. „Missing value imputation in high-dimensional phenomic data: imputable or not, and how?" In: *BMC Bioinformatics* 15 (2014).

[38] Craig K. Enders. *Applied Missing Data Analysis*. Guilford Press, 2010. ISBN: 978-1-60623-639-0.

[39] Donald B. Rubin. „Inference and Missing Data". In: *Biometrika* 63.3 (1976), S. 581–592.

[40] Donald B. Rubin. *Multiple Imputations in Sample Surveys – A Phenomenological Bayesian Approach to Nonresponse*. Techn. Ber. 1978.

[41] Donald B. Rubin. „Multiple Imputation after 18+ Years". In: *Journal of the American Statistical Association* 91.434 (1996), S. 473–489.

[42] Donald B. Rubin. *Multiple Imputation for Nonresponse in Surveys*. Wiley Series in Probability and Statistics. Wiley, 1987. ISBN: 978-0-47108-705-2.

[43] Joseph L. Schafer und John W. Graham. „Missing data: our view of the state of the art". In: *Psychological Methods* 7.2 (2002), S. 147–177.

[44] Stef van Buuren. *Flexible Imputation of Missing Data*. 2nd. Chapman und Hall/CRC, 2018. ISBN: 978-0-36725-318-9.

[45] Shangzhi Hong und Henry S. Lynn. „Accuracy of random-forest-based imputation of missing data in the presence of non-normality, non-linearity, and interaction". In: *BMC Medical Research Methodology* 20 (2020).

[46] Fei Tang und Hemant Ishwaran. „Random Forest Missing Data Algorithms". In: *Statistical Analysis and Data Mining: The ASA Data Science Journal* 10.6 (2017), S. 363–377.

[47] John W. Graham. „Missing data analysis: making it work in the real world". In: *Annual Review of Psychology* 60 (2009), S. 549–576.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

[48] Roderick J. A. Little. „A Test of Missing Completely at Random for Multivariate Data with Missing Values“. In: *Journal of the American Statistical Association* 83.404 (1988), S. 1198–1202.

[49] Krishnan Bhaskaran und Liam Smeeth. „What is the difference between missing completely at random and missing at random?“ In: *International Journal of Epidemiology* 43.4 (2014), S. 1336–1339.

[50] Tlamelo Emmanuel, Thabiso Maupong, Dimane Mpoeleng, Thabo Semong, Banyatsang Mphago und Oteng Tabona. „A survey on missing data in machine learning“. In: *Journal of Big Data* 8 (2021).

[51] Stef van Buuren und Karin Groothuis-Oudshoorn. *Flexible multivariate imputation by MICE*. Techn. Ber. 1999.

[52] Stef van Buuren und Karin Groothuis-Oudshoorn. „mice: Multivariate Imputation by Chained Equations in R“. In: *Journal of Statistical Software* 45.3 (2011).

[53] Olga G. Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein und Russ B. Altman. „Missing Value Estimation Methods for DNA Microarrays“. In: *Bioinformatics* 17.6 (2001), S. 520–525.

[54] Daniel J. Stekhoven und Peter Bühlmann. „Missforest—non-parametric missing value imputation for mixed-type data“. In: *Bioinformatics* 28.1 (2011), S. 112–118.

[55] Gareth Ambler, Rumana Z Omar und Patrick Royston. „A comparison of imputation techniques for handling missing predictor values in a risk model with a binary outcome“. In: *Statistical Methods in Medical Research* 16.3 (2007), S. 277–298.

[56] Peter Schmitt, Jonas Mandel und Mickael Guedj. „A Comparison of Six Methods for Missing Data Imputation“. In: (2015).

[57] John K. Dixon. „Pattern Recognition with Partly Missing Data“. In: (1978).

[58] John C. Gower. „A General Coefficient of Similarity and Some of Its Properties“. In: *Biometrics* 27.4 (1971), S. 857–871.

[59] Caterina Penone, Ana D. Davidson, Kevin T. Shoemaker, Moreno Di Marco, Carlo Rondinini, Thomas M. Brooks, Bruce E. Young, Catherine H. Graham und Gabriel C. Costa. „Imputation of missing data in life-history trait datasets: which approach performs the best?“ In: *Methods in Ecology and Evolution* 5.9 (2014), S. 961–970.

[60] Nadia Solaro, Alessandro Barbiero, Giancarlo Manzi und Pier Alda Ferrari. „A simulation comparison of imputation methods for quantitative data in the presence of multiple data patterns“. In: *Journal of Statistical Computation and Simulation* 88.18 (2018), S. 3588–3619.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

[61] Shigeyuki Oba, Masa-aki Sato, Ichiro Takemasa, Morito Monden, Ken-ichi Matsubara und Shin Ishii. „A Bayesian missing value estimation method for gene expression profile data“. In: *Bioinformatics* 19.16 (2003), S. 2088–2096.

[62] Yadolah Dodge. *The Concise Encyclopedia of Statistics*. Springer, 2008. ISBN: 978-0-387-31742-7.

[63] Michael H. Kutner, Christopher J. Nachtsheim, John Neter und William Li. *Applied Linear Statistical Models*. 5. Aufl. McGraw-Hill Education, 2005. ISBN: 0-07-238688-6.

[64] Bradley Efron. „Bootstrap Methods: Another Look at the Jackknife“. In: *Annals of Statistics* 7.1 (1979), S. 1–26.

[65] Jerome H. Friedman. „Greedy function approximation: a gradient boosting machine“. In: *Annals of statistics* 29.5 (2001), S. 1189–1232.

[66] Mehdi Zamani Joharestani, Chunxiang Cao, Xiliang Ni, Barjeece Bashir und Somayeh Talebiesfandarani. „$PM_{2.5}$ Prediction Based on Random Forest, XGBoost, and Deep Learning Using Multisource Remote Sensing Data“. In: *Atmosphere* 10.7 (2019).

[67] Erik Henriksson und Kristopher Werlinder. „Housing Price Prediction over Countrywide Data: A comparison of XGBoost and Random Forest regressor models“. Magisterarb. 2021.

[68] Jiun-Chi Huang, Yi-Chun Tsai, Pei-Yu Wu, Yu-Hui Lien, Chih-Yi Chien, Chih-Feng Kuo, Jeng-Fung Hung, Szu-Chia Chen und Chao-Hung Kuo. „Predictive modeling of blood pressure during hemodialysis: a comparison of linear model, random forest, support vector regression, XGBoost, LASSO regression and ensemble method“. In: *Computer Methods and Programs in Biomedicine* (2020).

[69] Candice Bentéjac, Anna Csörgő und Gonzalo Martínez-Muñoz. „A Comparative Analysis of XGBoost“. In: (2019).

[70] Haozhe Zhang, Joshua Zimmerman, Dan Nettleton und Daniel J. Nordman. „Random Forest Prediction Intervals“. In: *The American Statistican* 74.4 (2020).

[71] Nicolai Meinshausen. „Quantile Regression Forests“. In: *The Journal of Machine Learning Research* 7 (2006), S. 983–999.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abbildungsverzeichnis

1 Übersicht: Skizze einer Rotorscheibe mit Testlagen 9
2 Übersicht: Aufbau der Modellpipeline 16
3 Ausreißer: Daten vor Entfernen der Ausreißer 17
4 Ausreißer: Behandlung 19
5 Ausreißer: Daten nach Entfernen der Ausreißer 19
6 Encoding: Behandlung kategorialer Variablen 21
7 Skalierung: Standardisierung 22
8 Imputation: Künstliche Erzeugung fehlender Werte 23
9 Imputation: Behandlung fehlender Werte 24
10 Gewichtung: Gewichte abhängig von der Anzahl fehlender Werte 24
11 Validierung: 5-Fold Cross-Validation gruppiert nach dem Jahr 27
12 Imputation: Muster fehlender Werte 33
13 Imputation: Verteilungen bei Mechanismen fehlender Werte 36
14 Imputation: Vergleich der Anzahl der Nachbarn 40
15 Imputation: Übersich Imputationstest 44
16 Imputation: Heatmap des Imputationsfehlers 46
17 Imputation: Verteilung der beobachteten vs. künstlich entfernten Daten . . 47
18 Vorhersagemodelle: Lasso vs. Ridge 54
19 Vorhersagemodelle: Entscheidungsbaum 57
20 Resultate: Ergebnisse des Modellvergleichs 67
21 Resultate: Vorhersage vs. wahre Werte 69
22 Resultate: Feature Importance 72
23 Resultate: Minimalwert für drei unterschiedliche Temperaturen 74
24 Resultate: Minimalwert und Intervallgrenzen 75

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Tabellenverzeichnis

1 Datenübersicht: Beispielbeobachtungen 10
2 Datenübersicht: Ausreißer, Fehlende Werte, Datentypen 13
3 Vorhersagemodelle 25
4 Kategorisierung ausgewählter Imputationsmethoden 38
5 Ergebnisse: NRMSE-Vergleich der Imputationsmethoden 45
6 Ergebnisse: Lineare Regression 51
7 Ergebnisse: Ridge Regression 53
8 Ergebnisse: Lasso Regression 55
9 Ergebnisse: k-Nearest-Neighbor 56
10 Ergebnisse: Random Forest 60
11 Ergebnisse: XGBoost 66
12 Parametertuning: Individuelle Gewichtung von Beobachtungen 70
13 Parametertuning: Anzahl der Bäume bei Random Forest 70
14 Parametertuning: Maximale Tiefe eines Baums bei Random Forest 71
15 Parametertuning: Optimale Wahl der Parameter 71

Algorithmusverzeichnis

1 Zielvariablenimputation 29
2 Mittelwertimputation 38
3 KNN-Imputation 39
4 MissForest Imputation 42
5 Random Forest 59
6 Gradient Boosting 63
7 XGBoost 64

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.