Schlussbericht

Forschungsthema

„Untersuchung der Pozentiale und Risiken des Einsatzes von künstlicher Intelligenz am Beispiel der Vorhersage der 28d-Festigkeit“

gefördert durch die „Dres. Edith und Klaus Dyckerhoff-Stiftung“

Bearbeiter:
Dr. Viktoria Erfurt, Kevin Treiber, Prof. Dr. Philipp Fleiger
Tel.: 0049 211 4578 406
E-Mail: philipp.fleiger@vdz-online.de

Zuwendungsempfänger:
VDZ gGmbH
Tannenstraße 2
40476 Düsseldorf

Ansprechpartner:
Dr. Martin Schneider
Tel.: 0049 211 4578 200
E-Mail: Martin.Schneider@vdz-online.de

Dr. Jörg Rickert
Tel.: 0049 211 4578 233
E-Mail: Joerg.Rickert@vdz-online.de

Inhaltsverzeichnis

Vorwort und Dank

Das Forschungsvorhaben „Untersuchung der Potentiale und Risiken des Einsatzes von künstlicher Intelligenz am Beispiel der Vorhersage der 28d-Festikgeit“ stellt einen Baustein der systematischen Auseinandersetzung mit den Methoden der Digitalisierung an konkreten Beispielen entlang der Wertschöpfungskette von Zement und Beton dar. Durch besseres Verständnis von Methoden wie dem maschinellen Lernen soll die Transparenz und damit langfristig auch die Akzeptanz für die digitale Transformation gesteigert werden. Die Ergebnisse der vorliegenden Studie werden dazu nicht nur in weitere Forschungsprojekte sondern vor allem auch in die Arbeit des VDZ-Weiterbildungswerks einfließen.

Das Forschungsvorhaben wurde über die Dres. Edith und Klaus Dyckerhoff-Stiftung finanziell gefördert. Für die Förderung der Forschungsvorhaben danken wir der Dres. Edith und Klaus Dyckerhoff-Stiftung ganz herzlich.

1 Problemstellung und Zielsetzung

Unter den weit gefassten Überbegriffen der „Digitalisierung“ und „Industrie 4.0“ werden aktuell an vielen Stellen in Industrie und Gesellschaft Projekte vorangetrieben, die die Generierung und den Umgang mit digitalen Daten sowie insbesondere deren Auswertung und Nutzung betrachten. Die rasante Zunahme der Datenmenge in Verbindung mit der Verfügbarkeit von Rechenleistung und ausgereiften Methoden zur Auswertung haben dabei Möglichkeiten geschaffen, die in vielen Bereichen der Industrie grundlegende Veränderungen herbeiführen können. Zentrale Bedeutung kommt dabei insbesondere dem Thema der Künstlichen Intelligenz (KI) und den Methoden des maschinellen Lernens zu. Diese Algorithmen bieten auch Lösungsansätze für viele Fragestellungen der Zementindustrie, darunter z.B. bei der Prozesssteuerung, der vorbeugenden Instandhaltung (Predictive Maintenance) oder der Überwachung der Produktqualität.

So werden bereits heute Mahlanlagen mittels KI-unterstützter Expertensysteme gesteuert und die Eindüsung von Harnstoff im Rahmen des SNCR-Verfahrens wird durch KI optimiert. Algorithmen berechnen die Lebensdauer von Bauteilen und virtuelle Sensoren („Softsensoren“) können schwer erfassbare Messdaten durch Modelle aus besser verfügbaren Messdaten generieren. Diese Lösungen sind bisher jedoch nur vereinzelt im Einsatz.

Der Umgang mit maschinellem Lernen, also dem Trainieren eines (Black-Box) Vorhersagemodells aus gegebenen Daten, unterscheidet sich jedoch wesentlich von der klassischen Modellbildung auf Basis chemischer oder physikalischer Zusammenhänge. Die entstehenden Modelle selbst sind intransparent und ihre Verlässlichkeit hängt wesentlich von ihrer Datenbasis ab. Dies erfordert eine völlig neue Denk- und Herangehensweise. Die komplexen mathematischen Methoden, die sehr abstrakten Beschreibungen und die oftmals sehr komplizierten Anwendungen, die daraus entstehen, führen jedoch dazu, dass KI-Lösungen in der Praxis an vielen Stellen noch abgelehnt werden. Die Akzeptanz dieser Lösungen und die Integration von KI-Werkzeugen in den Arbeitsalltag ist jedoch eine wesentliche Voraussetzung für eine erfolgreiche Digitalisierung der Zementindustrie.

Aus diesem Grund ist es erforderlich, die Arbeitsweise und Relevanz der Methoden der KI und des maschinellen Lernens systematisch zu untersuchen und transparent und nachvollziehbar darzustellen. Dies sollte idealerweise durch gut verständliche Praxisbeispiele geschehen. Ein Anwendungsfall, der bei den Methoden des maschinellen Lernens bereits erfolgreich angewendet werden konnte, ist die Vorhersage der Festigkeit von Zementen nach 28d auf Basis von Labordaten wie Glühverlust, Mahlfeinheit, Dichte oder Erstarrungsbeginn und den gemessenen 2d-Festigkeiten. Für diesen Anwendungsfall können Modelle durch maschinelles Lernen trainiert werden. Dies ist jedoch mit einer Reihe von Fragestellungen verbunden. Die verwendeten Daten müssen für die weitere Auswertung formatiert und bereinigt werden, relevante Daten müssen identifiziert und fehlende Daten gegebenenfalls ergänzt werden. Es müssen dann geeignete Methoden des maschinellen Lernens ausgewählt und parametriert werden. Es existiert dazu eine Vielzahl von Software-Anwendungen auf dem Markt, die dies Quellcode-basiert oder mit graphischen Arbeitsoberflächen wie RapidMiner oder KNIME realisieren. Die Ergebnisse sind abschließend in Hinblick auf ihre Vorhersagequalität, aber auch hinsichtlich der Robustheit des Modells, zu bewerten. Dabei ist zu beachten, dass die Qualität der Daten und mögliche Störgrößen einen erheblichen Einfluss haben können.

Im Forschungsvorhaben wurden auf Basis von Labordaten aus der Güteüberwachung von Zementen verschiedene Vorhersagemodelle entwickelt, mit denen auf Basis einer Grundcharakterisierung des Zementes und seiner gemessenen 2d-Festigkeit die 28d-Festigkeit vorhergesagt werden kann. Dieser Anwendungsfall diente als Untersuchungsgegenstand, um gezielt die Möglichkeiten und Grenzen verschiedener Methoden des maschinellen Lernens, aber auch den Einfluss einzelner Arbeitsschritte, aufzuzeigen. Dabei wurden nicht nur unterschiedliche Typen von Algorithmen, sondern auch deren Umsetzung in unterschiedlichen Softwarelösungen getestet. Zudem wurde untersucht, wie die verwendeten Daten (z.B. durch gezieltes Hinzunehmen oder Weglassen von Attributen) die Vorhersagequalität des Modells beeinflussen können. Um die generelle Leistungsfähigkeit der entstehenden Modelle aus maschinellem Lernen beurteilen zu können, wurde zusätzlich ein klassisches Regressionsmodell entwickelt und ebenfalls getestet. Dabei wurden grundsätzlich Labordatensätze mit hoher Qualität genutzt, um den störenden Einfluss von Mess- und Dokumentationsfehlern zu eliminieren.

Der vorliegende Bericht stellt die Arbeitsschritte eines typischen Projektes mit den Methoden des maschinellen Lernens dar und bietet auch einen Einblick in die dahinter stehenden Theorien. Da diese umfangreich und komplex sind, wird auf weiterführende Informationen durch entsprechende Literaturangaben verwiesen.

2 Theorie / Methoden

2.1 Gesamtüberblick

Der Begriff Machine Learning (ML) fasst allgemein alle Algorithmen und Methoden zusammen, die einem Computer ermöglichen, rein auf der Basis von Daten Modelle von realen Problemen zu erstellen. Der Anwender muss die physikalischen oder chemischen Phänomene dabei nicht explizit ausprogrammieren. In dieser Studie wird exemplarisch die Vorhersage der 28-Tage-Festigkeit nach DIN EN 196-1 untersucht.

Machine Learning ist wie folgt definiert:

> ‘Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed.’
>
> Arthur Samuel, 1959 @ IBM

> ‘A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E.’
>
> Tom Mitchell, 1998

Bei dem dort erwähnten abstrakten Begriff der „Erfahrung“ (experience E) handelt es sich in der Praxis in der Regel um Datensätze eines realen Prozesses. Verständlicherweise hängt die Zuverlässigkeit des Endmodells in erster Linie von der Menge und der Qualität dieser Daten ab. Deshalb wird im folgenden Kapitel zunächst auf die Datenanalyse und -aufarbeitung eingegangen. Danach werden einige Algorithmen des maschinellen Lernens kurz vorgestellt. Für weiterführende Informationen zur Datenanalyse und zum maschinellen Lernen wird an dieser Stelle an [Ger 2018; Bis 2006] verwiesen.

2.2 Datenaufbereitung

Bei der Anwendung von Machine Learning-Methoden gibt es zwei mögliche Fehlerquellen: Die Wahl eines ungeeigneten Algorithmus und die Verwendungen von ungeeigneten oder unzureichenden Daten. Deshalb sollte der Datensatz vor dem Trainieren des Modells sorgsam geprüft und bei Bedarf aufbereitet werden. Für das Trainieren eines datenbasierten Modells sind möglichst große Datenmengen erforderlich, d.h. bei der Datenaufbereitung sollten möglichst wenige Datenpunkte verloren gehen. Nicht nur die Quantität der Daten ist entscheidend sondern auch die Qualität. Oft sind Daten uneinheitlich, enthalten Fehlstellen oder ungleichmäßige Zeitstempel. Eine Datenanalyse und -aufbereitung ist daher zwingend nötig, um die Qualität der Daten zu beurteilen und zu verbessern.

Im konkreten Fall liegt ein Datensatz mit unterschiedlichen Zementsorten aus verschiedenen Werken vor. Dieser hat den Vorteil, dass die Einzelwerte unter hohen Qualitätsansprüchen in einem Prüflabor ermittelt wurden und in einer Datenbank gesammelt werden. Der Datensatz enthält nur wenig Ausreißer und ein geringes Rauschen. Deshalb war eine Bereinigung und Aufbereitung der Daten in dieser Hinsicht nicht notwendig. Bei einem Datensatz mit geringerer Qualität empfehlen sich eine Überprüfung der Daten durch Nutzung statistischer Methoden (Min-Max-Betrachtung, Standardabweichung) gefolgt von einer Bereinigung der Ausreißer.

Wie die meisten großen Datensätze weist auch der vorliegende Datensatz einige lückenhafte Merkmale auf. Während der Datenbereinigung werden diese Merkmale entweder komplett entfernt oder, falls nur wenige Lücken im Merkmal vorhanden sind, durch repräsentative Werte (z.B. Median) geschlossen. Da die meisten Machine Learning-Algorithmen nur mit numerischen Daten umgehen können, werden alle nicht numerischen Merkmale, zum Beispiel, Text-Daten, mit einer Kategorisierung in numerische Daten überführt. In der vorliegenden Studie wurden zwei Kategorisierungsverfahren ausprobiert:

- Die **Numerische Kodierung**, die jeder Klasse eine eindeutige dezimale Zahl zuordnet. Die Zementsorten CEM I, CEM II und CEM III wurden beispielsweise als Dezimalzahlen 1, 2 und 3 kodiert.
- Die **One-Hot-Kodierung**, die jeder Klasse eine binäre Zahl zuweist. Die Zementsorten CEM I, CEM II und CEM III wurden hier zum Beispiel als binäre Zahlen 001, 010 und 100 kodiert. Die One-Hot-Kodierung hat gegenüber der numerischen Kodierung den Vorteil, dass ein ML-Algorithmus keine Korrelation zwischen den einzelnen Klassen innerhalb des gleichen Merkmals vermuten wird.

Im letzten Schritt werden gezielt Merkmale aus dem Datensatz entfernt. Dieser Schritt basiert auf Erkenntnissen einer exploratorischen Datenanalyse (Visualisierung von statistischen Zusammenhängen) in Verbindung mit Domainwissen. So soll z.B. vermeiden werden, dass der Machine Learning-Algorithmus irreführende Korrelationen findet. Zum Trainieren und Testen von Modellen ist die Aufteilung der Daten in einen Trainings- und einen Testdatensatz üblich. Dieser Schritt wird üblicherweise nach der Aufbereitung der Daten beim Trainieren und Testen des Modells randomisiert durchgeführt.

2.3 Modelle des Maschinellen Lernens

Aktuell sind viele unterschiedliche Machine Learning-Algorithmen bekannt. Das „No-Free-Lunch-Theorem“ von David Wolpert [Wol 1996] sagt, dass es keinen Grund gibt, ein Modell gegenüber einem anderen zu bevorzugen, wenn das Problem nicht ausreichend bekannt ist. Eine geeignete Lösung kann also nur gewählt werden, wenn Zielgrößen und Daten analysiert wurden. Kein Modell funktioniert garantiert a priori besser als ein anderes. Idealerweise sollten auf einem Datensatz möglichst viele Modellalgorithmen angewandt und untereinander verglichen werden, um das beste Endmodell zu identifizieren. In der Praxis hat sich bewährt, die Auswertung auf einige sinnvoll ausgewählte Modelle zu beschränken. Erst falls sich herausstellt, dass die gewählten Modelle das Problem trotz Optimierungsschritten nicht hinreichend abbilden, werden weitere Modelle hinzugenommen. Die Verfahren des Machine Learning lassen sich grob nach folgenden Kriterien unterscheiden:

- Grad der menschlichen Überwachung
    - überwacht (supervised)
    - halbüberwacht
    - unüberwacht (unsupervised)
    - Reinforcement Learning

- Grad des inkrementellen Dazulernes
    - Online
    - Batch

Eine detaillierte Beschreibung dieser Unterscheidung kann in [Ger 2018] nachgeschlagen werden. Im weiteren Verlauf des Berichtes werden einige überwachte Verfahren im Batch-Modus sowie als Vergleich die lineare Regression näher betrachtet. Die folgenden Machine Learning-Algorithmen werden auf ihre Anwendbarkeit untersucht: Gradient Boosted Trees, Random Forests (beide basieren auf Entscheidungsbäumen) und künstliche neuronale Netze. Am Ende des Abschnitts werden ferner einige Optimierungsalgorithmen vorgestellt.

2.3.1 Lineare Regression

Lineare Regression ist ein einfaches klassisches mathematisches Verfahren. Dieses Verfahren bestimmt ein lineares Modell, das eine gewichtete Summe aus allen Eingabemerkmalen berechnet und einen konstanten Term, Bias oder Achsenabschnitt, hinzuaddiert:

$$\hat{y} = \theta_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \cdots + \theta_n \cdot x_n$$

$\hat{y}$: vorhergesagter Wert

n : Anzahl der Merkmale

x_i : Wert des i-ten Merkmals

θ_i : Modellparameter mit $j = [0,1, \dots, n]$

oder in Vektorschreibweise:

$$\hat{y} = h_\theta(x) = \theta^T \cdot x$$

θ : Parametervektor

x : Merkmalvektor

$h_\theta(x)$: Hypothesenfunktion

Während des Trainierens sucht das Verfahren den Parametervektor θ, für den die Kostenfunktion (hier: Der mittlere quadratische Fehler (MSE) auf dem Trainingsdatensatz X) am geringsten ist:

$$MSE(x, h_\theta) = \frac{1}{m} \cdot \sum_{i=1}^{m} \left(\theta^T \cdot x^i - y^i\right)^2 \rightarrow min$$

In der Praxis wird oft das Gradientenverfahren für die Optimierung der Parameter verwendet. Eine detaillierte Beschreibung des Verfahrens sowie weitere Verfahren können in [Noc; Wri 1999] nachgeschlagen werden.

Je mehr Freiheitsgrade eine Regression hat, desto eher neigt das Modell dazu, einen Datensatz einfach „auswendig zu lernen“ (Overfitting), und desto niedriger ist die Zuverlässigkeit bei der Nutzung eines anderen Datensatzes (Extrapolation). Um eine Überanpassung an den zu Grunde liegenden Datensatz zu vermeiden, kann eine Regression durch Einschränkungen der Gewichtungsparameter reguliert werden. Dabei werden gezielt Freiheitsgrade eingeschränkt.

Neben der linearen Regression existieren polynominelle, logistische, logarithmische und exponentielle Regressionen. Mehr Informationen dazu sind in [Ger 2018] [Ger 2018] und [Gru 2016] zu finden. Mittlerweile gehören diese Regressionen zu klassischen Auswertungsmethoden und können einfach z.B. in Excel oder Matlab angewandt werden.

2.3.2 Entscheidungsbäume und ihre Weiterentwicklungen

Entscheidungsbäume oder „Decision Trees“ sind baumartige, gerichtete Diagramme die zur Entscheidungsfindung eingesetzt werden. Entscheidungsbäume werden im Maschinellen Lernen, aber auch anderen Fachbereichen wie der Betriebswirtschaft, für Klassifizierungs- und Regressionsaufgaben genutzt. Entscheidungsbäume (Abbildung 1) bestehen aus Wurzel (Root-node), Knoten (Internal-node), Ästen (Edge) und Blättern (Leaf). In den Knoten werden die Entscheidungen getroffen, die Blätter geben das Ergebnis aus. Die Anzahl der Ebenen (tree depth) ist ein wichtiger Modellparameter

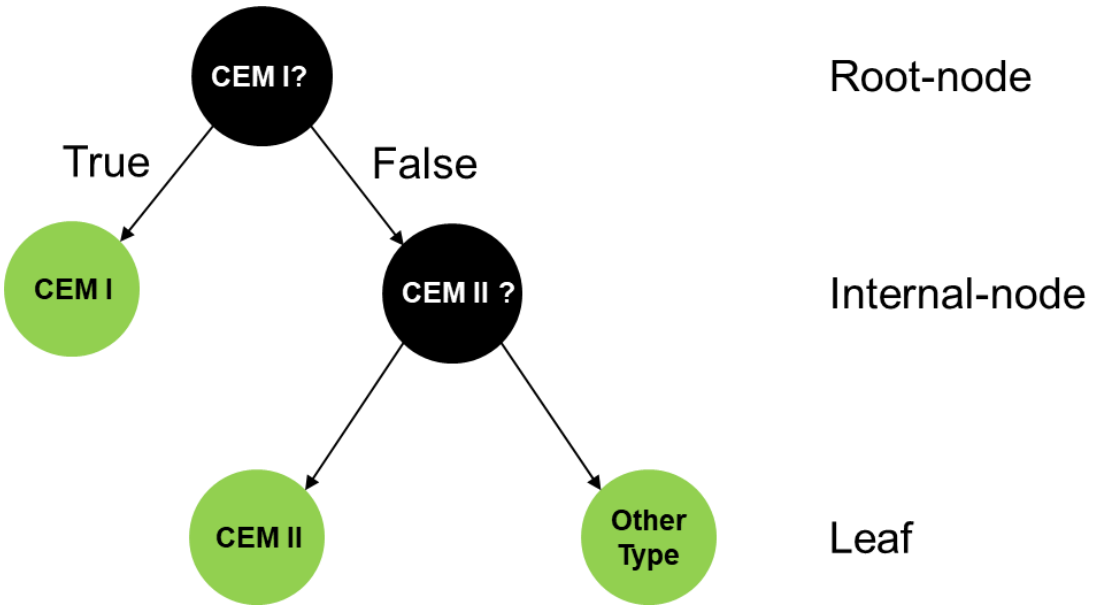


Abbildung 1 Veranschaulichung eines Decision Tree

„Entscheidungswälder“, die **Random Forests,** bestehen aus randomisiert parametrisierten Entscheidungsbäumen (Abbildung 2). Bei Random Forest-Modellen wird eine Gruppe von Entscheidungsbäumen trainiert. Dabei werden die Parameter der Bäume in einem vorgegebenen Rahmen zufällig variiert und zusätzlich in jedem Knoten für die Entscheidungsfindung eine zufällig ausgewählte Teilmenge der Merkmale betrachtet, um eine höhere Diversität unter den einzelnen Bäumen zu erreichen. Die Vorhersage wird aus den Einzel-Vorhersagen der Bäume im Ensemble bestimmt [Gru 2016].

Neben dem Random Forest ist das **Gradient Boosted Tree-**Verfahren (oder allgemein Gradient Boosting) ein weiteres Modell, das auf Basis der Entscheidungsbäume arbeitet. Während die Random Forests auf tiefe, unabhängige Bäume setzten, kommt hier eine Serie von weniger tiefen aufeinander aufbauenden Entscheidungsbäumen zum Einsatz. Die Parameter des vorherigen Baumes werden mithilfe des Gradientenverfahrens verfeinert und so der

Restfehler zwischen den Vorhersagen und den realen Daten verkleinert (Abbildung 3). Das Verfahren beansprucht im Vergleich zur Random Forest-Regression weniger Rechenaufwand und eignet sich auch besonders gut für das Online-Lernen. Mehr Informationen sind in [Ger 2018] zu finden.

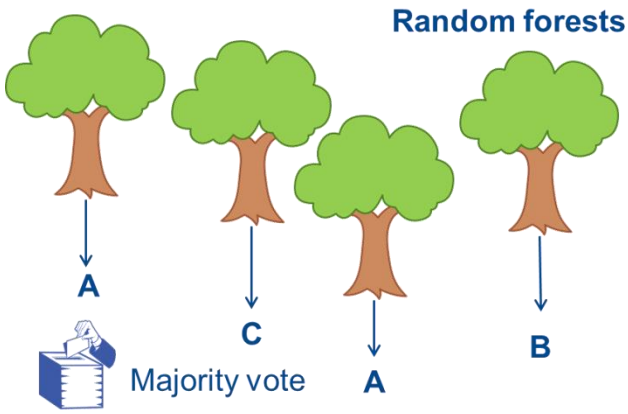


Abbildung 2 Random Forest

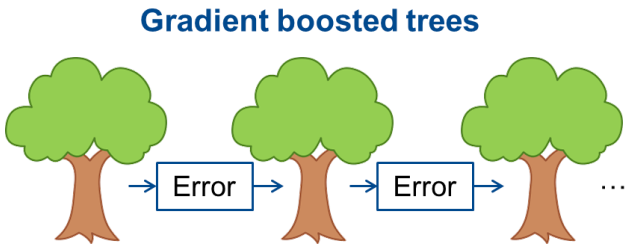


Abbildung 3 Gradient Boosted Trees

2.3.3 Künstliche Neuronale Netze

Das Verfahren der Künstlichen Neuronalen Netze (engl. artificial neural network, kurz ANN) ist von den biologischen Neuronen und dem Verhalten eines realen Gehirns inspiriert. Die künstlichen Neuronen sind miteinander verbunden und übertragen Informationen durch Übertragungsfunktionen, welche in der Regel gewichtete Summen der Ausgaben anderer Neuronen darstellen. Diese dienen als Eingang für Aktivierungsfunktionen in den Neuronen, die beim Erreichen eines Schwellwertes wiederum eine Ausgabefunktion auslösen [Kri 2007]. Durch das Training verändern sich die Parameter der Funktionen und die Gewichte der Verknüpfungen. Diese Werte repräsentieren schließlich das „Gelernte“ des Modells.

Ein mehrschichtiges Perzeptron (engl. multilayer perceptron, kurz MLP) ist das gängige Verfahren der künstlichen neuronalen Netze. Es setzt sich aus einer Eingabeschicht, einer oder mehreren verborgenen Schichten und einer Ausgabeschicht zusammen. Jede Schicht (außer der Ausgabeschicht) enthält ein sogenanntes Bias-Neuron mit einem konstanten Eingang und ist mit der nächsten Schicht vollständig verbunden, siehe Abbildung 5. Allgemein werden ANN mit zwei oder mehreren verborgenen Schichten als Deep-Learning-Netz (kurz DNN) bezeichnet [Kri 2007].

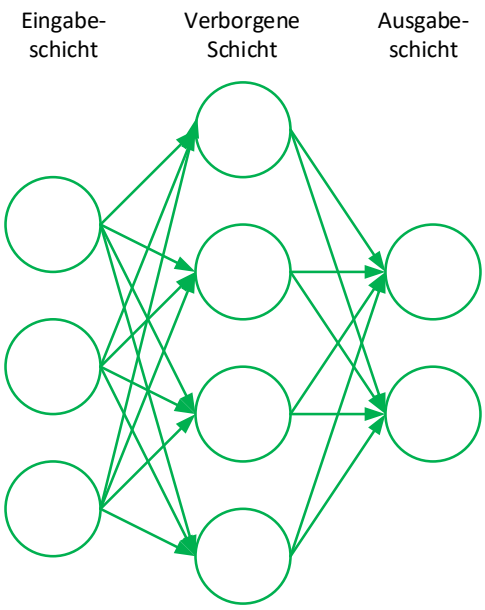


Abbildung 4: Skizze eines ANN mit einer verborgenen Schicht aus vier Knoten.

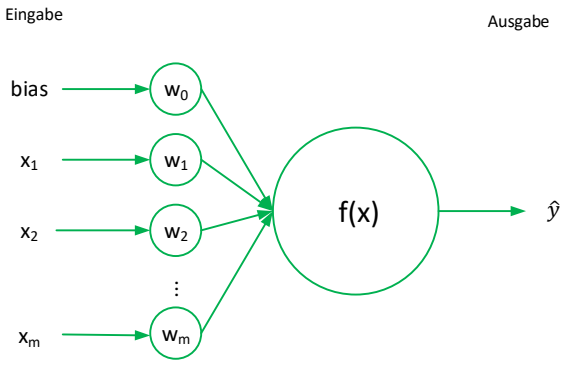


Abbildung 5: Skizze eines einzelne neuronalen Knoten

Jede Schicht besteht hierbei aus mehreren Neuronen, auch Knoten genannt. In jedem Knoten wird eine gewichtete Summe aus den vorhergehenden Ausgabeschichten und dem konstanten Bias-Wert berechnet. Der Bias-Wert als Modellparameter kann die Ausgangsfunktion des Neurons verschieben (verzerren). Anhand der gebildeten Summe (Ergebnis aus dem Neuron) wird oft eine sogenannte Aktivierungsfunktion genutzt, um den Knoten zu aktivieren (Fall ≥ 0) oder zu deaktivieren (Fall <0):

$$f(x) = \begin{cases} 1, if \sum_{i}^{m} (w_i \cdot x_i) + w_0 \cdot bias \geq 0 \\ 0, if \sum_{i}^{m} (w_i \cdot x_i) + w_0 \cdot bias < 0 \end{cases}$$

Das Verfahren sucht beim Trainieren ein Parameterset für diese Gewichte und Bias-Werte durch Optimieren einer Kostenfunktion. Zum Optimieren wird in der Regel der sogenannte Backpropagation-Algorithmus benutzt, der im Grunde ein Grandientenverfahren ist, das auch bei der linearen Regression bzw. Gradient Boosted Tree-Regression verwendet wird. In einfachen Worten ausgedrückt, wird der Fehler der Vorhersage durch das Netz zurückgeführt, um die Modell-Parameter anzupassen. Hierdurch verbessert sich das Modell mit jedem Durchlauf.

Eine Weiterentwicklung von DNN stellen die Verfahren des Convolutional Neural Networks (CNN) und Recurrent Neural Networks (RNN) dar. Eine detailliertere Beschreibung zu diesen Verfahren ist in [Ger 2018; War 2018; Ras 2017] zu finden.

2.3.4 Optimierungsalgorithmen

Die oben vorgestellten Verfahren können mit Optimierungsalgorithmen verfeinert werden. In dieser Studie wurden wahlweise zwei Optimierungsalgorithmen für Modellparameter ausprobiert: Grid Search und Random Search. Weiterhin spielt im Optimierungsprozess auch die Überprüfung und Bewertung der Modelle durch eine Kreuzvalidierung eine wichtige Rolle. Diese Methode kann ebenfalls dazu eingesetzt werden, optimale Modelle auszuwählen.

Bei der k-fachen Kreuzvalidierung wird der Datensatz zufällig auf eine Anzahl k unterschiedlicher Teilmengen aufgeteilt. Das Modell wird danach k-mal trainiert, wobei jedes Mal ein Datensatz aus bis zu k-1 unterschiedlichen Teilmengen zum Trainieren und der andere Teil zum Validieren benutzt werden. Als Endmodell wird ein Modell mit der besten Trefferquote empfohlen.

Grid Search- und Random Search-Verfahren optimieren ein Set von Parametern des untersuchten Modells, um ein besseres Endmodell zu bekommen. Bei der manuellen Grid Search-Methode gibt der Anwender vor, wie oft ein Modell trainiert wird, und welche Parameter in welchen Schritten angepasst werden sollen. Meist werden hier alle möglichen Kombinationen untersucht. Bei Random Search werden zufällig ausgewählte Parameter automatisiert angepasst. Beide Verfahren liefern ein Set von Modellen, wobei das Endmodell mit der besten Trefferquote empfohlen wird. Mehr Details können in [Ger 2018] nachgeschlagen werden.

3 Verfügbare Tools

Zur Bearbeitung von Daten mit den Methoden des Maschinellen Lernens sind verschiedene Softwarelösungen am Markt verfügbar. Abbildung 6 zeigt den Vergleich zweier Studien von Gartner [Gartner 2021], welcher verfügbare Lösungen kategorisiert und bewertet. Seit 2018 ist das weite Feld der Plattformen für Data Science deutlich zusammengewachsen. Viele Lösungen konnten sich nachhaltig etablieren, während andere deutlich zurückgefallen sind. Zu berücksichtigen ist, dass sich die Plattformen teils deutlich hinsichtlich ihrer Lizenzmodelle und Kosten unterscheiden.

Die meisten dieser Lösungen unterstützen verschiedene Methoden des Maschinellen Lernens, z.B. Regressionsverfahren, Klassier- und Clusteringverfahren, Dimensionsreduktion und Neuronale Netze. Es werden aber auch immer weitere Funktionen in Form von Datenaufbereitung, Schnittstellen und Datendarstellung mitgeliefert. Besonders bekannt sind dabei die Libraries und Frameworks, die für die Programmiersprachen Python oder R zur Verfügung stehen. Hierzu zählen besonders Tensorflow [Aba; al. 2015] und Scikit Learn [Pe 2011]. Auch bekannte Softwareumgebungen wie Matlab von Mathworks bieten Toolboxen zur Entwicklung von KI-Anwendungen.

Alternativ zu klassischen Programmiersprachen gibt es verschiedene Flow Sheet-Umgebungen, in denen einzelne Funktionen („Nodes“) zusammengeschaltet werden können. Hierzu zählt unter anderem die KNIME Analytics-Plattform, welche eine kostenlose Open-Source-Softwarelösung für die Arbeit an Data Science Projekten ist. RapidMiner bietet eine Softwarelösung, welche besonders Anfängern auf dem Gebiet des Maschinellen Lernens schnell vielversprechende Ergebnisse liefern kann. Der Anwender einer solchen Plattform muss dabei aber die spätere Nutzung der Applikation im Blick behalten und bewerten, ob und inwiefern die so entwickelten Modelle später auch praktisch anwendbar sind. Neben den genannten Softwareumgebungen sind noch viele weitere teils hochspezialisierte Lösungen auf dem Markt erhältlich.

Abbildung 6: Gartner Magic Quadrant für Data Science und Machine Learning Plattformen. Links 2018, Rechts 2021 [Gartner 2021]

4 Datenanalyse und Datenaufbereitung

Der gewählte Datensatz besteht im Rohzustand aus insgesamt 70.162 Datenpunkten mit jeweils 67 Merkmalen (engl. features), wobei jedes Merkmal unterschiedlich vollständig erfasst wurde. Die Daten wurden in Laufe von 24 Jahren (von 02.01.1996 bis 11.04.2018) an insgesamt 89 verschiedenen Werksstandorten von 43 Zementherstellern erfasst. Die Daten wurden im Laufe des Projektes nur in anonymisierter Version unter Angabe von Aliasnamen für die Werksstandorte verwendet. Folgende Merkmale sind im unbearbeiteten Datensatz erfasst:

Beschreibung der Merkmale	**Anzahl der Datensätze**
Zwei Zeitstempeln (Probeentnahme, Eingangsdatum)	Für Modellrechnungen irrelevante Datensätze
Interne Identifikationsnummer	
Probenehmer	
Alias für Hersteller und Werkstandort	70.149
Zementsorte	70.149
2-Tage-Festigkeit	56.177
7-Tage-Festigkeit [2)]	12.325
28-Tage-Festigkeit	70.149
Erstarrungsanfang	69.376
Erstarrungsende[2)]	10.262
Wasseranspruch	69.355
Le Chatelier-Messwert [2)]	
Rückstand 90µm Sieb [2)]	
Normklinkergehalt [2)]	
Eindringtiefe [2)]	
Luftporengehalt bei MC 5/MC 12,5 [2)]	
Wasserrückhaltevermögen [2)]	
Glühverlust	
Unl. Rückstand (unl) [2)]	
SO3-Gehalt	
Chlor-Gehalt	
Na2O-Gehalt	
Al2O3-Gehalt	
C3A-Gehalt [2)]	
Reindichte (zusammengefügt aus 3 Datensätzen) [1)]	67.997
Spezifische Oberfläche nach Blaine [1)]	60.805
Hydratationswärme [2)]	
Klinker-Gehalt	
Hüttensandgehalt [2)]	
Puzzolan-Gehalt [2)]	
Flugasche-Gehalt [2)]	
Schiefergehalt [2)]	
Kalkstein-Gehalt [2)]	
Nebenbestandteile	
CO_2–Gehalt	66.136
Chemische Zusammensetzung: SiO2, TiO2, P2O5, Fe2O3, Mn2O3, MgO, CaO, K2O, Na2O	

[1)] Fehlende Daten werden in jeder Merkmal-Spalte durch Medianwerte der entsprechenden Zementsorte ergänzt.
[2)] Aufgrund geringer Anzahl an Daten entfernt

Als Zielgröße für die Vorhersage der Modelle (engl. label) wurde die 28-Tage-Festigkeit gewählt. Die Werte sind für dieses Merkmal fast vollständig vorhanden. Insgesamt enthält die Spalte 70.149 gültige und 12 ungültige Einträge (NaN – engl. not a number). In der Abbildung 7 ist ein Histogramm der bereinigten Werte dargestellt. Das Histogramm zeigt trotz der drei möglichen Festigkeitsklassen einen bi-modalen Verlauf der Häufigkeitsverteilung für die CEM I- und den CEM II-Zemente. Die Normfestigkeit nach 28 Tagen der CEM I 32,5 Zemente im Datensatz beträgt im Mittel 48 MPa, dies stimmt auch mit dem ersten Peak im Histogramm überein. Ein zweiter Peak stellt sich bei etwa 58 MPa ein. Die Mittelwerte der Normdruckfestigkeit der Festigkeitsklassen 42,5 und 52,5 liegen mit 58 MPa und 67 MPa nahe beieinander, wobei viele der Zemente der Festigkeitsklasse 42,5 schon über dem Norm-Grenzwert der Festigkeitsklasse 52,5 liegen, der Übergang ist daher fließend. Weiterhin fallen die Häufigkeiten mit zunehmender Festigkeit über 60 MPa ab. Gleiches zeigt sich auch beim Vergleich der Mittelwerte der Normdruckfestigkeiten für die CEM II- und CEM III-Zemente. Die Häufigkeiten spiegeln nicht die Versandmengen der jeweiligen Sorten wieder.

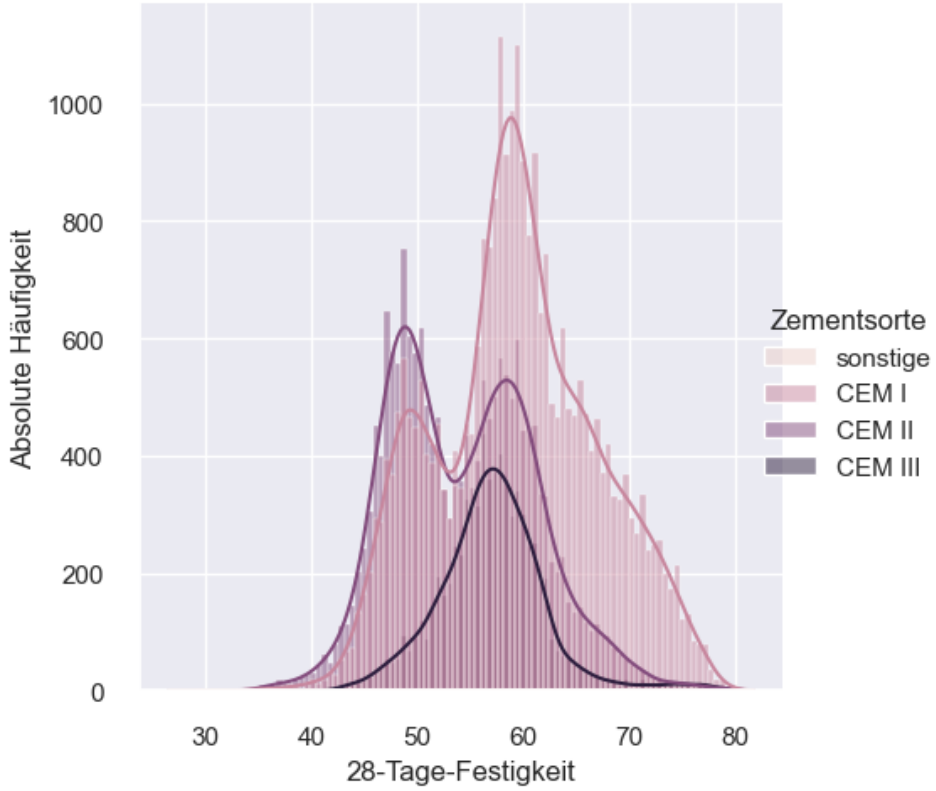


Abbildung 7: Histogramm der 28-Tage-Festigkeit aufgeteilt nach Zementsorte.

Tabelle 1 Häufigkeit der Zementfestigkeitsklassen in den Zementsorten in gesamten Datensatz

	Festigkeitsklasse		
	32,5	**42,5**	**52,5**
CEM I	8,7%	17,9%	15,5%
CEM II	12,6%	11,6%	4,5%
CEM III	13,3%	9,1%	1,2%
sonstige	5,5%		

Der Datensatz enthält im Labor gemessene Werte für 2-Tage-, 7-Tage- und 28-Tage-Festigkeit. Für jedes Prüfdatum wurden 6 Einzelmessungen an den 3 Einzelprismen durchgeführt, woraus ein Mittelwert gebildet wurde, welcher auch im Datensatz verwendet wird. Um Redundanz zu vermeiden, wurden von den Messergebnissen die Varianz und der Mittelwert berechnet und die ursprünglichen Messergebnisse aus dem Datensatz entfernt. Zur Beurteilung des Fehlers der Modelle sind auch statistische Informationen zum Prüfverfahren sinnvoll. Die Auswertung der Daten zeigt, dass die Einzelmessungen der Festigkeit nach 28 Tagen eine Abweichung zwischen dem Min-Max-Wert von etwa 2,3 MPa aufweisen.

Zeitstempel, Probenehmer, Eingangsdatum und interne Identifikationsnummer wurden, aufgrund fehlender nutzbarer Information und um falsche Korrelationen beim Lernen des Modells auszuschließen, bei der Datenaufbereitung entfernt. Die Merkmale Hersteller und Standort wurden unter Berücksichtigung der historischen Gegebenheiten zu einem Merkmal „Werk ID“ zusammengefasst und so anonymisiert. Aufgrund geringer Datenlage und Redundanz wurden drei Merkmale der „Reindichte“ zu einem Merkmal zusammengefasst.

Das Merkmal Sorte enthält insgesamt 283 unterschiedliche Sortenbezeichnungen nach DIN EN 197-1, sowie Sonderzemente. Die Datensätze der Sonderzemente wurden bei der Datenaufbereitung entfernt und die Bezeichnungen der Zementsorten nach DIN EN 197-1 entsprechend der Zusammensetzung und Charakteristik in maschinenlesbarer Form aufgeschlüsselt.

Zusätzlich enthält der Datensatz Informationen zur chemischen Zusammensetzung der Proben, d.h. Gehalt an SiO_2, TiO_2, P_2O_5, Fe_2O_3, Mn_2O_3, MgO, CaO, K_2O und Na_2O. Die Datendichte ist sehr dünn und enthält viele fehlende Einträge (Null). Um den Einfluss der Zusammensetzung auf die Modelle für die Vorhersage der 28-Tage-Festigkeit zu untersuchen, werden drei Datensätze aufbereitet.

- Im ersten Datensatz werden die Merkmal-Spalten der chemischen Zusammensetzung vollständig entfernt.
- Im zweiten Datensatz bleiben die Informationen zur Zementchemie unverändert enthalten.
- Im dritten Datensatz werden die fehlenden Einträge zur Zementchemie durch die Medianwerte abhängig von der jeweiligen Sorte ergänzt.

Teil der Datenanalyse ist die Betrachtung möglicher Korrelationen zwischen den Datensätzen. Informationen zur Korrelation können die Datenaufbereitung beeinflussen. Sind Datenpunkte besonders wichtig, können Lücken hier gezielt aufgefüllt oder Ausreißer gezielt eliminiert werden. Korrelieren Merkmale stark unterneinander, kann geprüft werden, ob einzelne Merkmale in der weiteren Betrachtung vernachlässigt werden können. Oft werden Korrelationsmatrizen genutzt, die neben dem alleinig quantitativen Vergleich der Korrelation zwischen zwei Datensätzen diese auch grafisch unterstützt darstellen können. Mit steigender Korrelation nähert sich die Kennzahl dem Wert 1 (bzw. -1) an. Die Korrelationsmatrizen für die Merkmale des Datensatzes sind in der Abbildung 8, Abbildung 9 und Abbildung 10 dargestellt.

Die Abbildungen zeigen, dass die Merkmale Druckfestigkeit nach 2 und nach 7 Tagen eine besonders hohe Korrelation aufweisen. Die Merkmale Druckfestigkeit nach 2 und nach

28 Tagen weisen eine etwas niedrigere Korrelation auf, was mit dem zeitlichen Abstand der Messungen und hohen materialbedingten Einflussfaktoren zu tun haben kann. Die Merkmale Glühverlust und CO_2 weisen eine hohe Korrelation auf, da sie chemisch eindeutig zusammenhängen. Es ist anzunehmen, dass die Berücksichtigung beider Merkmale keine Mehrinformation liefert. Daher kann ein Merkmal entfernt werden.

In der Abbildung 9 fällt eine starke lineare Abhängigkeit zwischen einzelnen Zusammensetzungen auf (helles Quadrat mittig rechts unten). Alle Korrelationseinträge liegen in diesem Quadrat zwischen 0,8 und 1,0. Dieser Zusammenhang ist auf die Tatsache zurückzuführen, dass bei der chemischen Analyse der Gehalt an SiO_2, TiO_2, P_2O_5, Fe_2O_3, Mn_2O_3, MgO, CaO, K_2O und Na_2O gleichzeitig gemessen wird. Wenn keine chemische Analyse durchgeführt wurde, sind diese Werte im Datensatz nicht enthalten (der Wert beträgt 0). Wenn also einer der Werte 0 ist, sind in der Regel auch alle anderen Werte der chemischen Analyse ebenfalls gleich 0. Die Korrelationsuntersuchung erkennt einen positiven linearen Zusammenhang, den es in diesem Fall chemisch-mineralogisch nicht gibt.

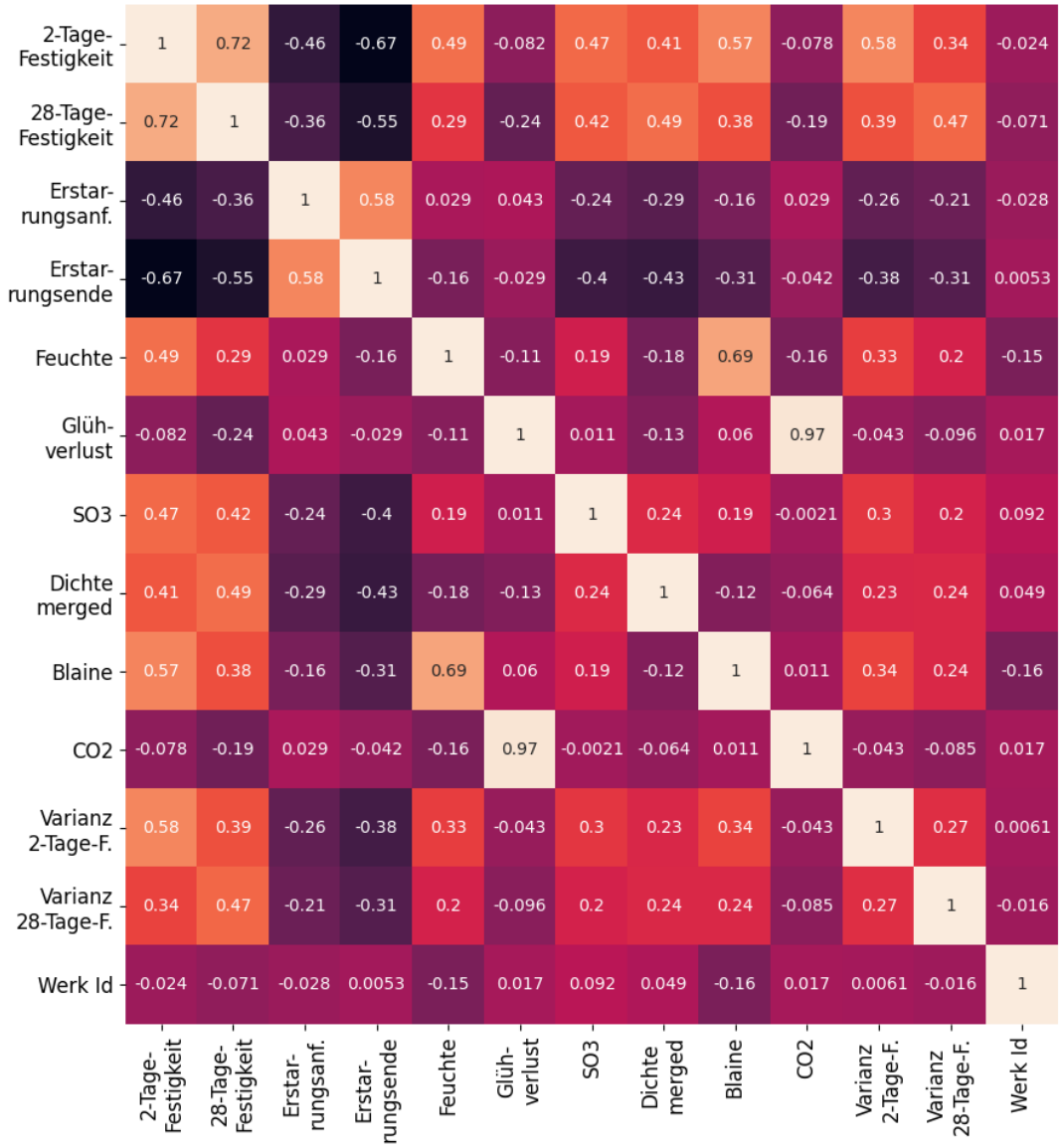


Abbildung 8: Datensatz 1: Korrelationsmatrix ohne chemische Zusammensetzung

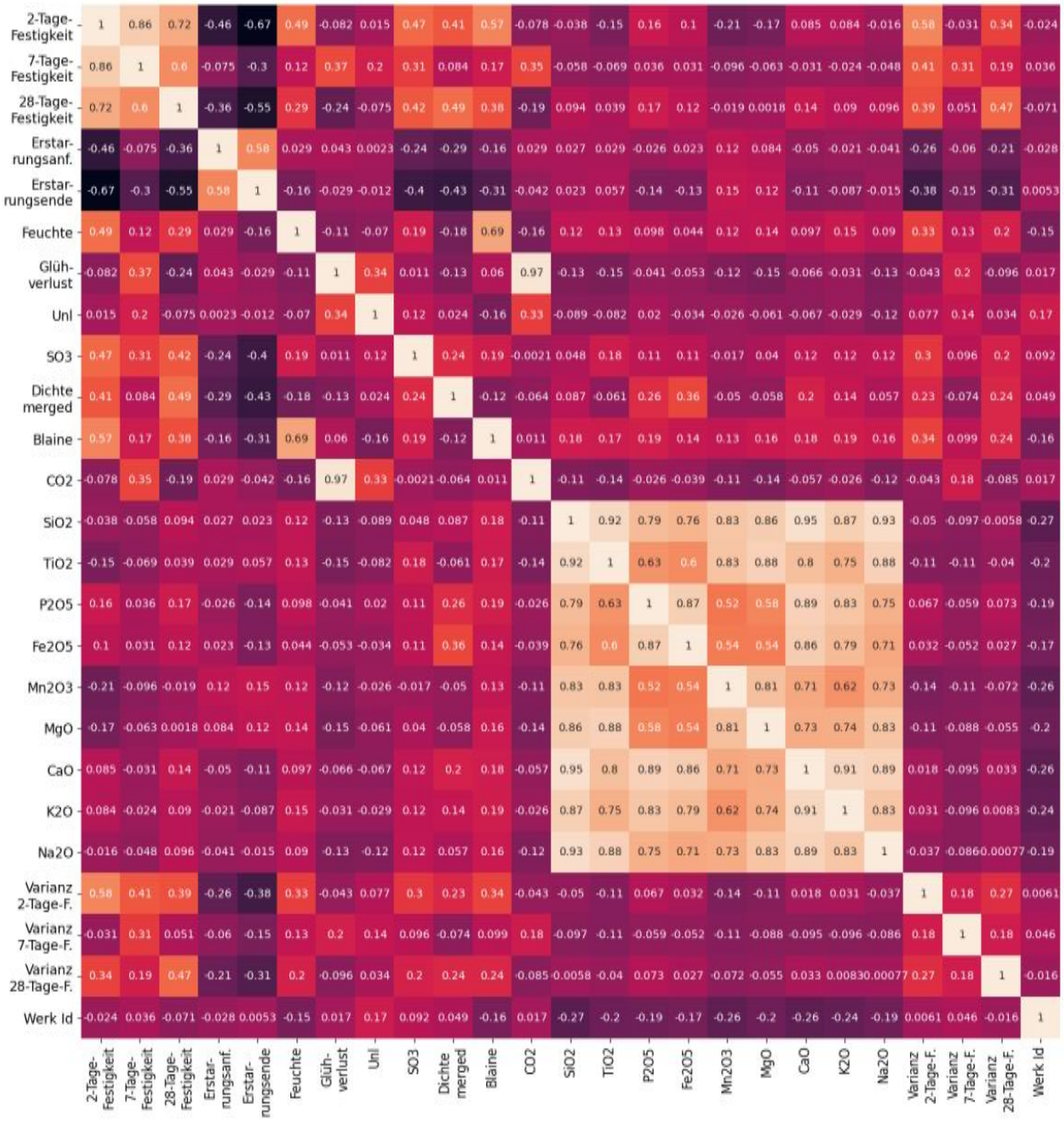


Abbildung 9: Datensatz 2: Korrelationsmatrix mit originaler chemischer Zusammensetzung, die auf Grund vieler Nulleinträge nicht vorhandene positive lineare Zusammenhänge suggeriert

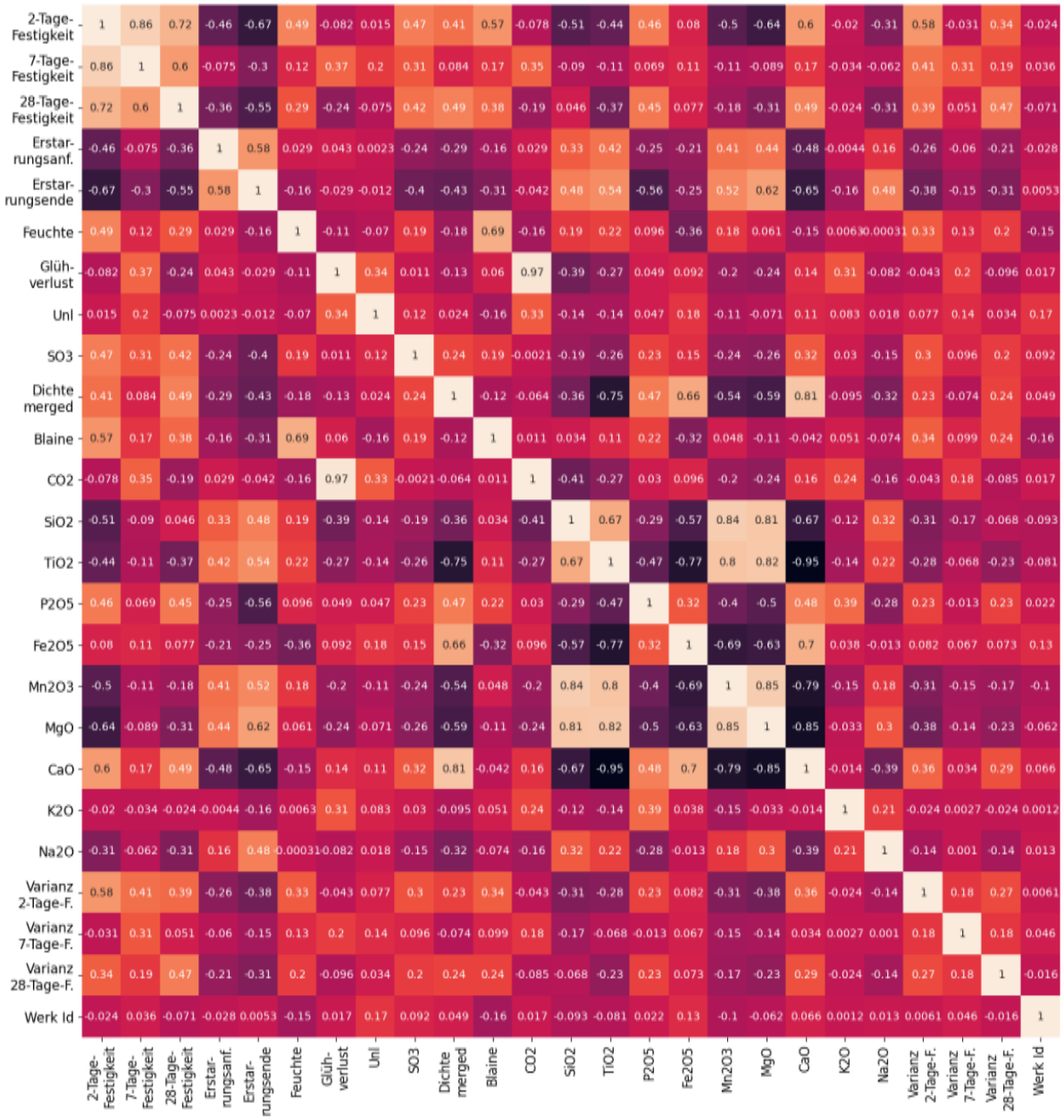


Abbildung 10: Datensatz 3: Korrelationsmatrix mit chemischer Zusammensetzung. Alle Null-Einträge sind durch Medianwerte entsprechend der Sorte und des Werkes ergänzt worden.

Eine weitere Möglichkeit, Merkmale des Datensatzes gegenüberzustellen, ist Visualisierung der paarweisen Vergleiche (Abbildung 11 zeigt dies exemplarisch). Hier werden die Datenpunkte je zweier Merkmale gegeneinander aufgetragen. Zudem werden Dichteverteilungen dargestellt, um die Verteilung der Merkmale zu beschreiben. Die gute Korrelation kann auch hier beim Vergleich von CO_2 und Glühverlust identifiziert werden. Die optische Bewertung liefert aber ein über die Bewertung eines möglichen linearen Zusammenhangs hinausgehendes Bild der Daten. Hinzu kommen weitere Darstellungsmöglichkeiten wie eine Farbkodierung oder die Verwendung unterschiedlicher Symbole.

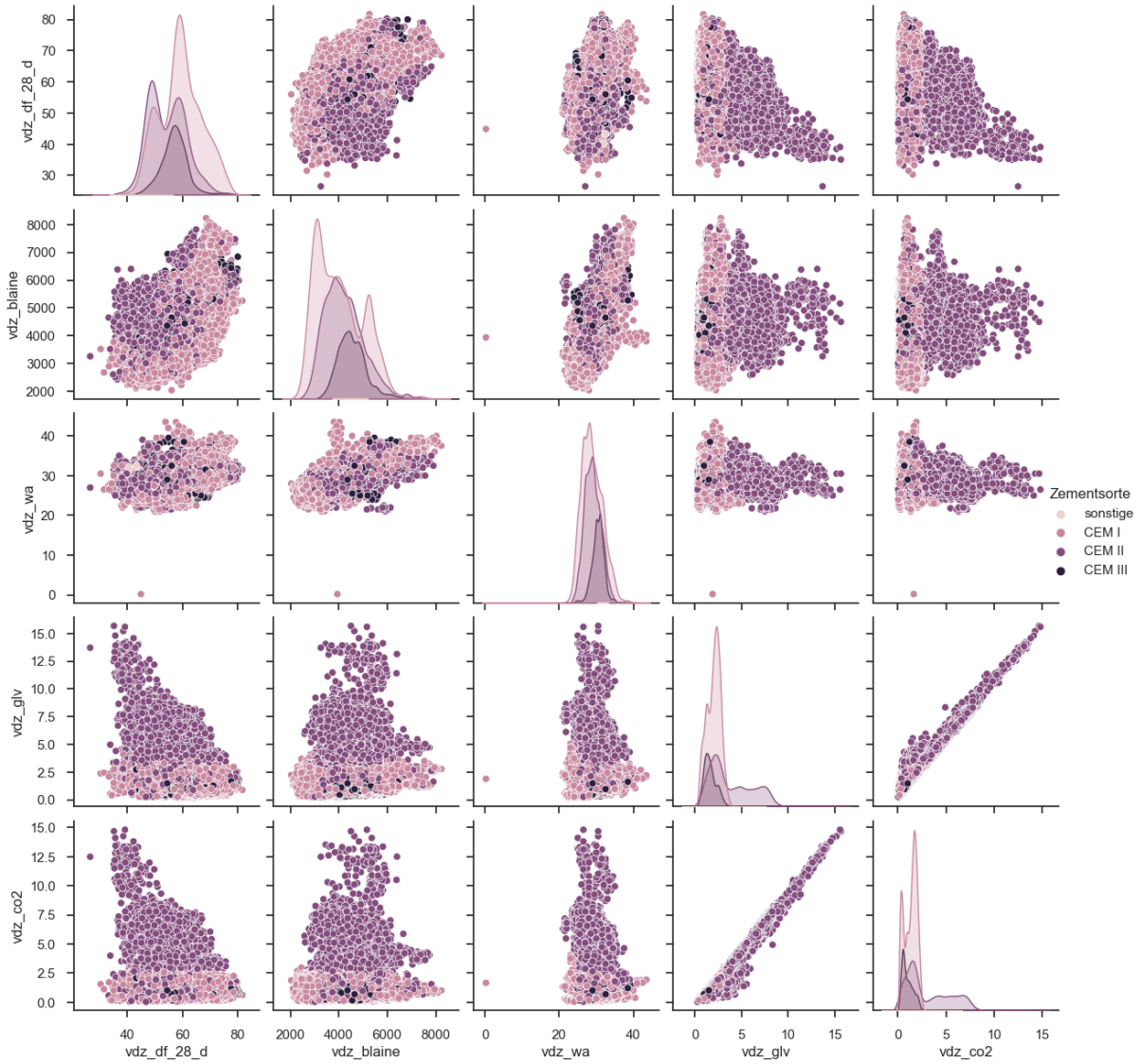


Abbildung 11: Paarweiser Vergleich der Merkmale aus Datensatz 1

5 Modellentwicklung und Ergebnisse

In den ausgewählten Softwarelösungen KNIME, SciKit Learn und Tensorflow wurde der Einfluss der 6 Datensätze (siehe Kapitel 4) auf die Vorhersage der Festigkeit nach 28 Tagen untersucht. Weiterhin wurde in Zusammenarbeit mit der Hochschule Düsseldorf auch die Software RapidMiner erprobt. In allen Fällen wurden verschiedene Modelle genutzt und vergleichend bewertet.

Die Bewertung der Daten erfolgt über den Root Mean Squared Error (RMSE) und die Trefferquote (H). Die Trefferquote ist in Formel 1 definiert. Die Trefferquote wird aus gemessener Druckfestigkeit nach 28 Tagen (S) und der Vorhersage des Modells (S_{Pred}) berechnet. Die Trefferquote zeigt, welcher prozentuale Anteil der Vorhersagen im Bereich von ±3 MPa um die tatsächlich gemessene Druckfestigkeit nach 28 Tagen liegt.

$$H(x) = \frac{1}{n}\sum_{1}^{n}\left[\left|\left(S_{Pred}(28) - S(28)\right)\right| < x\right] \qquad \text{Formel 1}$$

5.1 RapidMiner

RapidMiner stellt unter den verwendeten Softwarelösungen eine Besonderheit dar. Das Programm bietet die Möglichkeit, sich interaktiv durch die Erstellung einer einfachen Anwendung des Maschinellen Lernens führen zu lassen. Das Programm empfiehlt Schritte der Datenaufbereitung, unterstützt die Wahl von Attributen und testet abschließend verschiedene Algorithmen. Diese Funktion bietet einen sehr schnellen Einstieg, lässt aber an vielen Stellen die vollständige Transparenz vermissen. Alle Prozessschritte lassen sich aber später in graphische Modelle, ähnlich wie in der später näher vorgestellten Plattform KNIME, überführen und weiter bearbeiten (Abbildung 12). Ab diesem Punkt ähneln sich beide Softwarelösungen. Entsprechend wird hier auf eine detaillierte Vorstellung der Standardfunktionen von RapidMiner verzichtet.

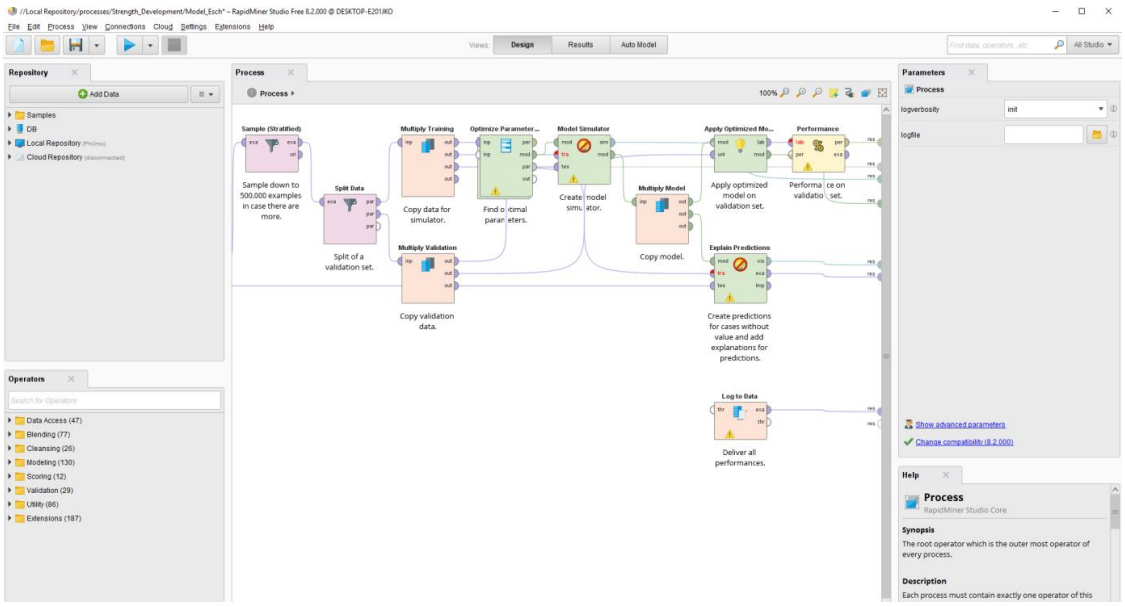
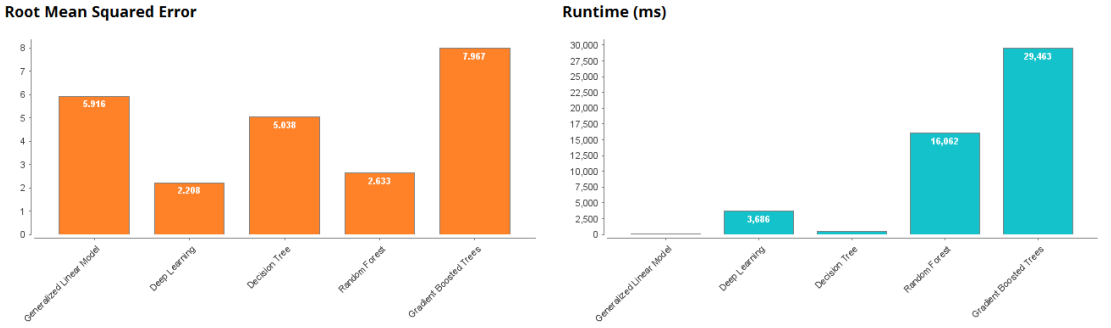


Abbildung 12 Graphische Modellierungsumgebung von RapidMiner

Ähnlich den detaillierten nachfolgen Untersuchungen wurden in RapidMiner ebenfalls mehrere Modelle, darunter Entscheidungsbäume, Random Forest, Gradient Boosted Trees und ein Neuronales Netz (Deep Learning), genutzt. Die verwendeten Attribute und die eingesetzte Datenmenge wurde schrittweise variiert, um so das Modell durch Versuch und Irrtum zu optimieren. Abbildung 13 zeigt zunächst die Modellergebnisse für den gesamten Trainingsdatensatz. Der minimale Fehler liegt hier bei 2,2 MPa. Die in Abbildung 14 dargestellten Ergebnisse stammen aus Modellen, die nur aus Daten einzelner Werksstandorte bestimmt wurden. Die Betrachtung von einzelnen Werksstandorten eliminiert den möglichen Einfluss unterschiedlicher Klinker aus der Vorhersage. Diese zeigen gegenüber dem Gesamtdatensatz eine leicht bessere Performance (minimaler RMSE 1,9 MPa). Es zeigt sich aber insgesamt eine Verbesserung aller Modelle und eine Vergleichmäßigung der Ergebnisse.

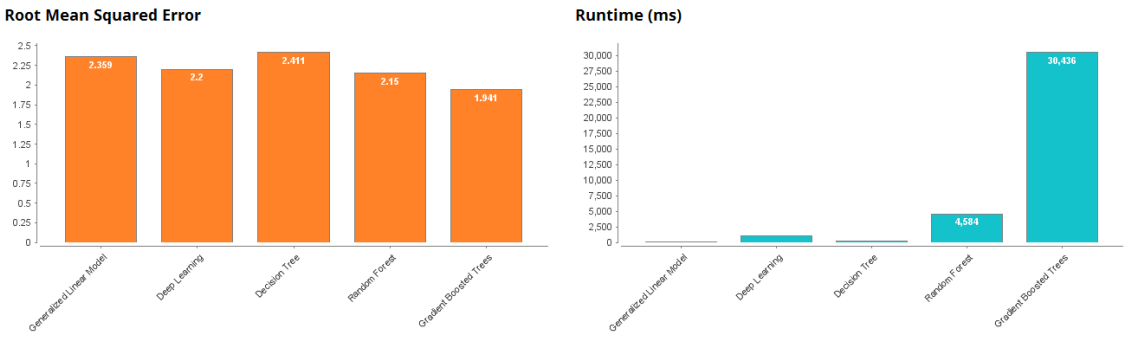
Die Ergebnisse der durchgeführten Studie werden im Rahmen der Vorlesung „Angewandte verfahrenstechnische Simulation“ im Master Simulations- und Experimentaltechnik in der Lehre genutzt.



Root Mean Squared Error

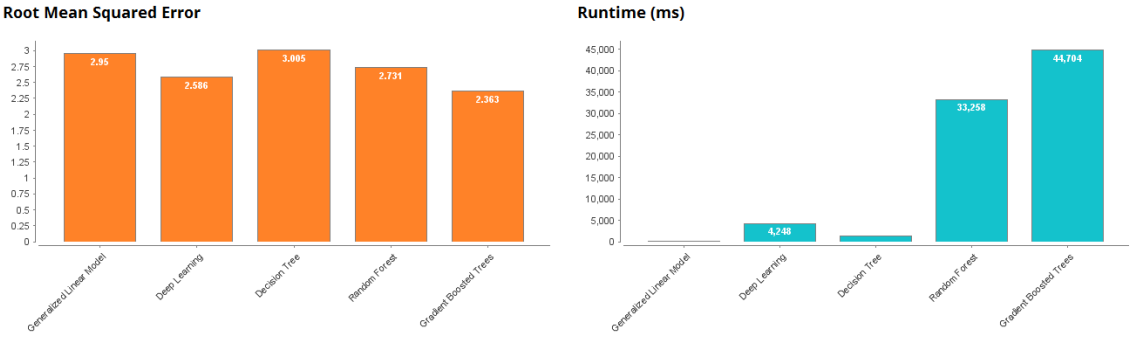
Model	Root Mean Squared Error	Run Time
Generalized Linear Model	5.916	69 ms
Deep Learning	2.208	4 s
Decision Tree	5.038	511 ms
Random Forest	2.633	16 s
Gradient Boosted Trees	7.967	29 s

Abbildung 13 RMSE verschiedener Modelle zur Vorhersage der Festigkeit nach 28 Tagen in RapidMiner für den Gesamtdatensatz (Quelle: Fleiger, HSD)



Root Mean Squared Error

Model	Root Mean Squared Error	Run Time
Generalized Linear Model	2.359	64 ms
Deep Learning	2.200	1 s
Decision Tree	2.411	170 ms
Random Forest	2.150	5 s
Gradient Boosted Trees	1.941	30 s



Root Mean Squared Error

Model	Root Mean Squared Error	Run Time
Generalized Linear Model	2.950	135 ms
Deep Learning	2.586	4 s
Decision Tree	3.005	1 s
Random Forest	2.731	33 s
Gradient Boosted Trees	2.363	45 s

Abbildung 14 RMSE verschiedener Modelle zur Vorhersage der Festigkeit nach 28 Tagen in RapidMiner für zwei zufällig ausgewählte Werksstandorte (Fleiger, HSD)

5.2 KNIME

In KNIME wurden die Methoden Gradient Boosted Trees, Random Forest und ein Neuronales Netz gegenübergestellt. Zunächst wurde eine Kategorisierung der Normbezeichnung durchgeführt (siehe Abbildung 15). Bei der Verwendung der One-Hot Datensätze wurden die Normbezeichnungen nach DIN EN 196-1/DIN 1045-2 bereits im Vorfeld kategorisiert.

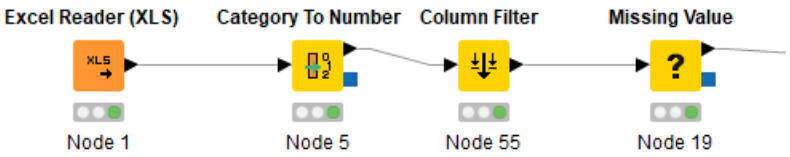


Abbildung 15 Datenimport und Kategorisierung der Daten in KNIME

Zur Nutzung der Methoden Gradient Boosted Trees und Random Forest wird nach dem Einlesen der Daten eine Partitionierung des Datensatzes in Trainingsdaten (80 %) und Validierungsdaten (20 %) vorgenommen. Zur Nutzung des Neuronalen Netzes werden die Daten im Vorfeld normalisiert und dann partitioniert.

In KNIME stehen für die Funktionen des Maschinellen Lernens je ein „Learner“ und ein „Predictor“ zur Verfügung. Im „Learner“ wird das Modell trainiert, dementsprechend wird hier mit 80 % ein Großteil der Daten verwendet. Der „Predictor“ erhält zum einen das gelernte Modell und zum anderen Daten, die vorhergesagt werden sollen. Der nachfolgende Vergleich der vorhergesagten mit den bekannten, gemessenen Daten zeigt die Qualität des gelernten Modells. Hierzu können zum Beispiel Bausteine wie der „Numeric Scorer“ und der „Scatter Plot“ verwendet werden.

5.2.1 Gradient Boosted Trees Regression

Für die Betrachtung der Gradient Boosted Trees wurde das in Abbildung 16 dargestellte Modell entwickelt. Im Funktionsblock wird zunächst der Zielwert (hier Festigkeit nach 28 Tagen) ausgewählt.

Für die gewählte Methode sollte eine „Tree depth“ von 4-10 gewählt werden. Größere Bäume neigen zu einer Überbestimmung (Overfitting). Die Anzahl der Modelle und die „Learning rate“ sind abhängig voneinander. Die Anzahl der Modelle hängt von der Größe des Datensatzes ab und kann von wenigen Modellen (z.B. 10) bis zu mehreren tausend Modellen für kleine Datensätze und wenige Kategorien schwanken. Um Overfitting zu vermeiden, muss mit steigender Anzahl der gerechneten Modelle die Learning Rate reduziert werden. Die Learning Rate gewichtet den Einfluss der einzelnen Ergebnisse der Bäume auf das Gesamtergebnis. [KNI 2021]

Ein geeignetes Verhältnis der Parameter kann durch „Trial and Error“ identifiziert werden. Dies führte im vorliegenden Fall zu einer hohen Anzahl an Modellen (500) und einer „Learning Rate“ von 0,05 bei einer Tree depth von 6. Es hat sich jedoch auch gezeigt, dass mit einer geringeren Anzahl von Modellen (z.B. 300) und einer etwas höheren Learning Rate (0,1) gute Ergebnisse erzielt werden können. Eine größere Anzahl der Modelle hat bei den vorliegenden Daten keine Verbesserung der Vorhersagequalität erzielt.

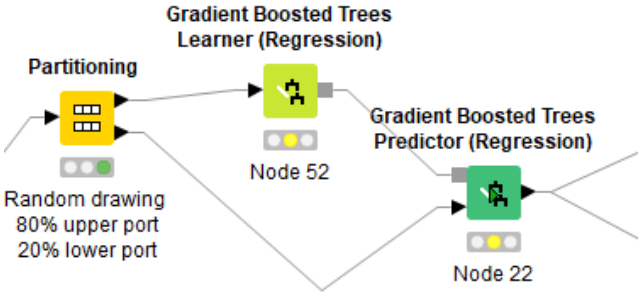


Abbildung 16 KNIME Gradient Boosted Trees

Die in den Abbildungen dargestellten Ergebnisse wurden durch Bildung des RMSE und der Trefferquote quantitativ in Tabelle 2 erfasst. Sowohl die Trefferquote (Formel 1) als auch der RMSE differieren kaum zwischen den unterschiedlichen Datensätzen. Beide liegen jedoch auf einem hohen Niveau und zeigen die gute Eignung des Gradient Boosted Trees-Modells zur Vorhersage. Die Datenaufbereitung nach dem One-Hot-Verfahren hatte keinen erkennbaren Vorteil. Ein kleiner Mehrwert ist durch die Datenaufbereitung mit Datensatz 2 erkennbar. Hier werden weniger Datensätze mit mehr Kategorien genutzt.

Tabelle 2 Vorhersagequalität Gradient Boosted Trees

	Trefferquote H(±3 MPa) in %	RMSE in MPa
Datensatz 1	83,7	2,25
Datensatz 1 (One-Hot)	81,4	2,38
Datensatz 2	86,2	2,09
Datensatz 2 (One-Hot)	86,5	2,09
Datensatz 3	83,5	2,25
Datensatz 3 (One-Hot)	83,5	2,24

Abbildung 17 bis Abbildung 19 zeigen die Ergebnisse bei Nutzung einer hohen Anzahl an Modellen (500) und einer Learning Rate von 0,05 bei einer Tree depth von 6. Dargestellt wird die Vorhersage gegenüber dem gemessenen Wert für die Test-Partition des Datensatzes. Eine Steigung von 1 wäre eine perfekte Vorhersage. Die Modelle zeigen zunächst eine gute Vorhersagequalität. Die Anzahl der Ausreißer ist sehr gering. Zudem zeigt die Betrachtung der Daten keinen offensichtlichen systematischen Fehler. Die Vorhersagequalität variiert zwischen den Datensätzen.

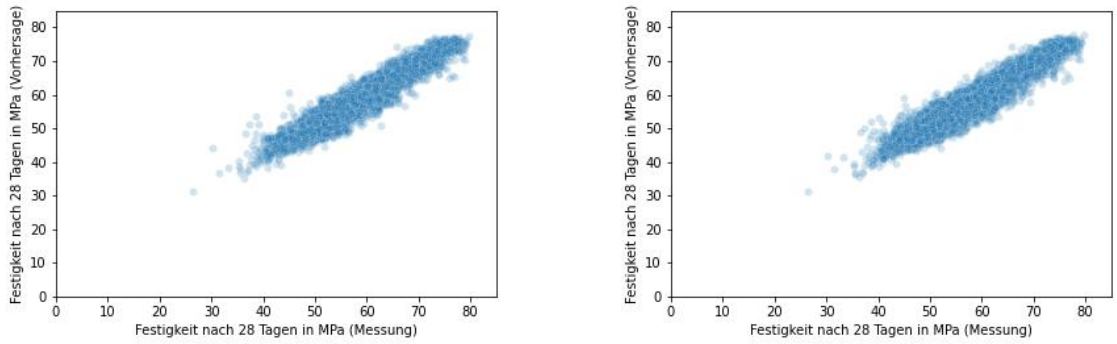


Abbildung 17 Gradient Boosted Trees – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 1, Rechts: Datensatz 1-One-Hot

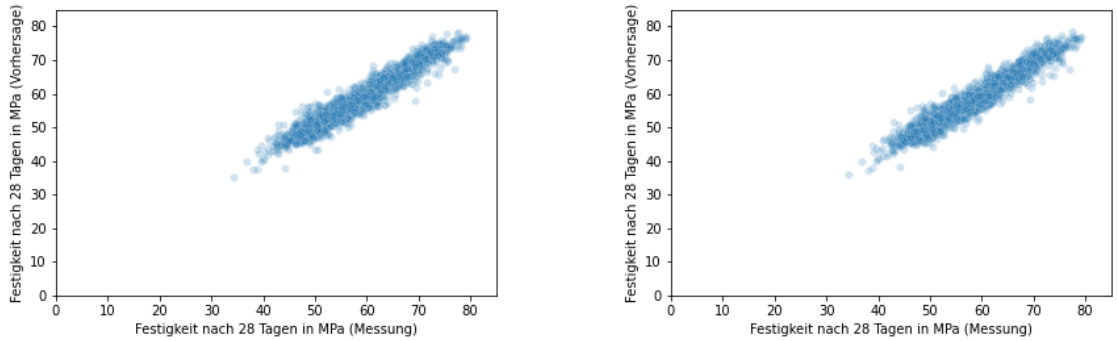


Abbildung 18 Gradient Boosted Trees – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 2, Rechts: Datensatz 2-One-Hot

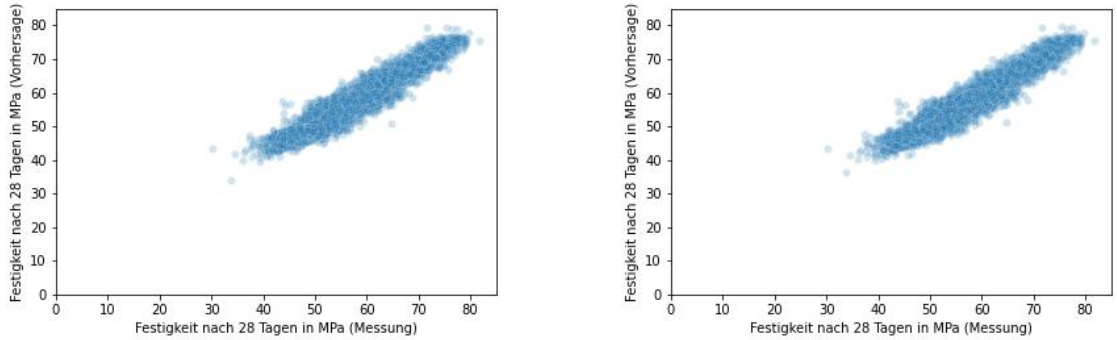


Abbildung 19 Gradient Boosted Trees – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 3, Rechts: Datensatz 3-One-Hot

5.2.2 Random Forest Regression

Im Modell Random Forest Regression (Abbildung 20) wurde ein Optimum für die Anzahl der Modelle bei 50 und für die Tree depth bei 15 ermittelt. Die Parameter unterscheiden sich damit deutlich vom Gradient Boosted Trees Regression-Modell, was im grundsätzlich anderen Modellansatz begründet liegt. Nur die Nutzung von Entscheidungsbäumen als Basis ist beiden Ansätzen gleich.

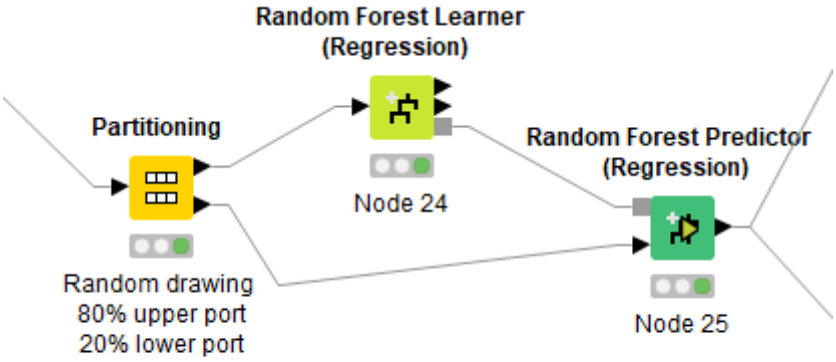


Abbildung 20 KNIME Random Forest

Die quantitative Betrachtung der Daten (Tabelle 3) zeigt jedoch eine niedrigere Trefferquote und einen höheren RMSE im direkten Vergleich zum Gradient Boosted Trees-Modell. Die Datenaufbereitung nach dem One-Hot-Verfahren wirkt sich auch hier nicht positiv auf das Ergebnis aus, wobei Datensatz 2 hier wieder die höchste Trefferquote und den niedrigsten RMSE liefert. Die Vorhersagequalität mit den Datensätzen, die durch das One-Hot-Verfahren kategorisierte wurden, ist beim Random Forest teils deutlich schlechter als bei den Vergleichsdatensätzen.

Tabelle 3 Vorhersagequalität Random Forest

	Trefferquote H(±3 MPa) in %	RMSE in MPa
Datensatz 1	80,8	2,42
Datensatz 1 (One-Hot)	66,9	3,27
Datensatz 2	81,5	2,39
Datensatz 2 (One-Hot)	76,2	2,72
Datensatz 3	79,4	2,50
Datensatz 3 (One-Hot)	76,1	2,69

Abbildung 21 bis Abbildung 23 zeigen die Ergebnisse des Test-Datensatzes für das Random Forest-Modell. Qualitativ unterscheiden sich die Ergebnisse des Random Forest nicht wesentlich vom Gradient Boosted Trees-Modell. In Abbildung 23 ist jedoch ein stufenförmiger Verlauf (anstelle eines stetigen Verlaufs) erkennbar, was auf ein systematisches Problem bei Datensatz 3 bei der Verwendung des Random Forest Modells schließen lässt.

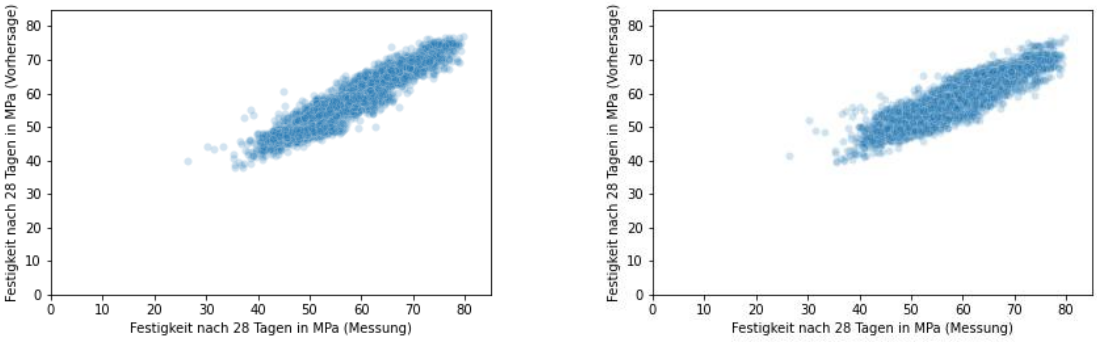


Abbildung 21 Random Forest – Vorhersagequalität der Festigkeit nach 28 Tagen.
Links: Datensatz 1, Rechts: Datensatz 1-One-Hot

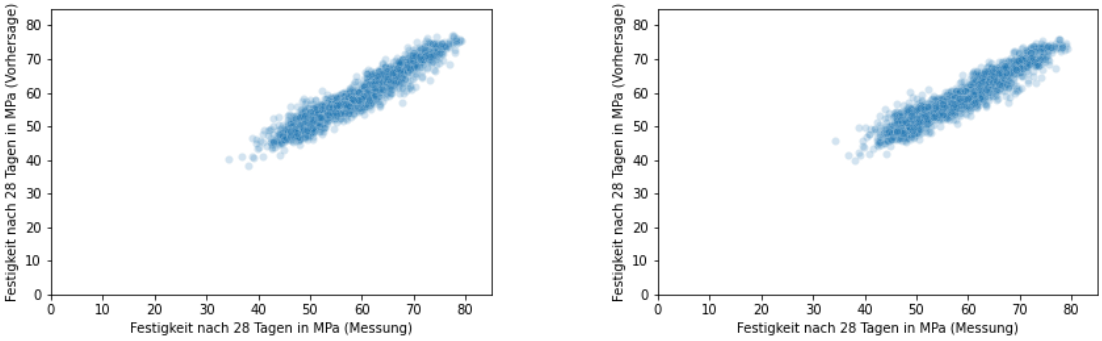


Abbildung 22 Random Forest – Vorhersagequalität der Festigkeit nach 28 Tagen.
Links: Datensatz 2, Rechts: Datensatz 2-One-Hot

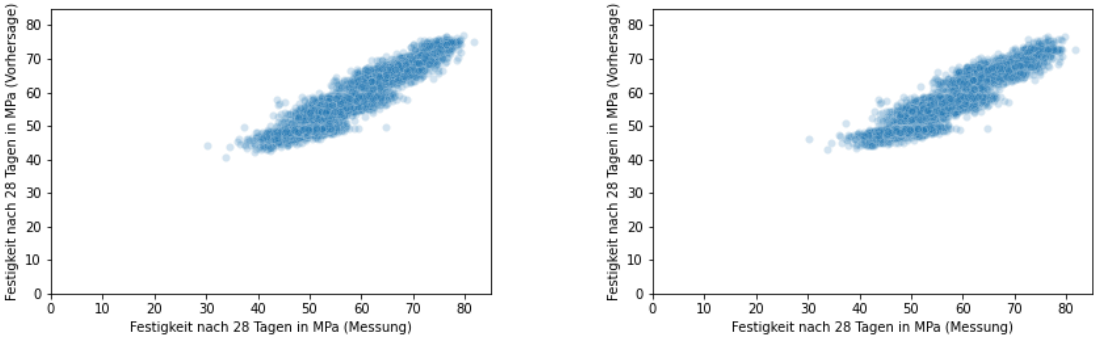


Abbildung 23 Random Forest – Vorhersagequalität der Festigkeit nach 28 Tagen.
Links: Datensatz 3, Rechts: Datensatz 3-One-Hot

5.2.3 Multilayer Perceptron (Neuronal Network)

Um die Eignung eines Neuronalen Netzes für die Vorhersage zu untersuchen, wurde beispielhaft ein Multilayer Perceptron verwendet. Die Vorhersagequalität hängt stark von den gewählten Parametern ab. In KNIME können die Anzahl der Schichten (Layer) und die Anzahl der Neuronen pro Schicht verändert werden. Darüber hinaus kann die Anzahl der Iterationen vorgegeben werden.

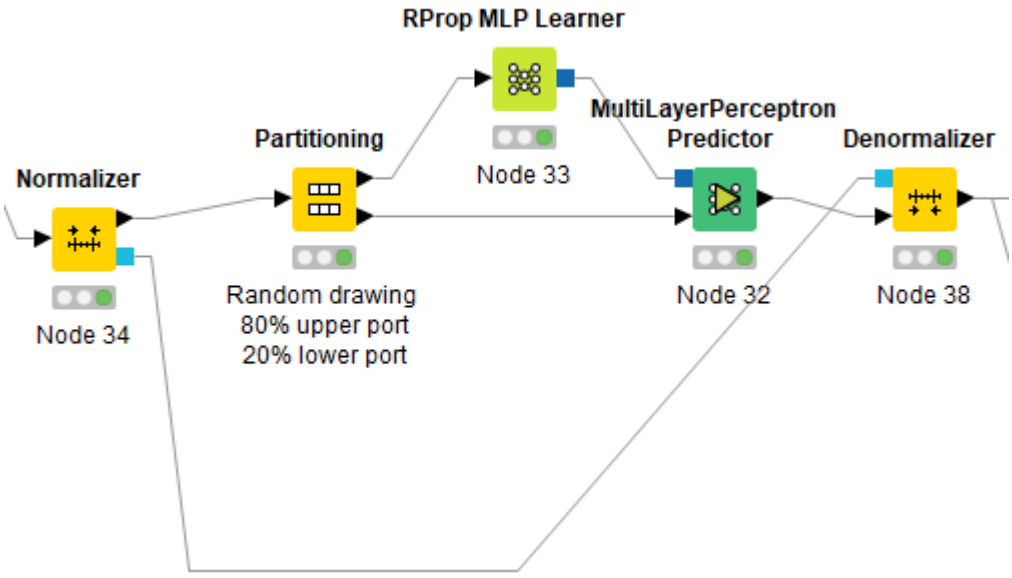


Abbildung 24 KNIME Multilayer Perceptron

Im Vergleich zu den beiden anderen untersuchten Methoden scheint das verwendete Neuronale Netz für die Vorhersage der Festigkeit nach 28 Tagen wenig geeignet. Voruntersuchungen haben gezeigt, dass sich hier deutlich höhere RMSE und niedrigere Trefferquoten einstellen, als bei den anderen Methoden.

Um eine mögliche Abhängigkeit von den Eingangsdaten festzustellen, wurden die Modellparameter an jeden Datensatz manuell angepasst (Tabelle 4). Es zeigt sich, dass bereits kleine Änderungen der Anzahl der „Hidden Layer“ und der Neuronen pro Layer deutlichen Einfluss auf das Ergebnis haben können und stark von den Eingangsdaten abhängen.

Tabelle 4 Modellparameter der Multilayer Perceptron-Modelle in KNIME

	Number	
	Hidden Layer	Hidden Neurons per layer
Datensatz 1	1	4
Datensatz 1 (One-Hot)		5
Datensatz 2		4
Datensatz 2 (One-Hot)		6
Datensatz 3		5
Datensatz 3 (One-Hot)		6

Die quantitative Betrachtung der Ergebnisse zeigt, dass Datensatz 1, vorbereitet durch das One-Hot-Verfahren, deutlich bessere Ergebnisse erzielt, als die restlichen Eingangsdatensätze (Tabelle 5). Mit Datensatz 3 wird die niedrigste Vorhersagequalität erreicht. Hier wurden die Lücken in den Merkmalen aufgefüllt (Kapitel 4).

Tabelle 5 Vorhersagequalität Multilayer Perceptron

	Trefferquote H(±3 MPa) in %	RMSE in MPa
Datensatz 1	31,4	6,87
Datensatz 1 (One-Hot)	33,8	6,28
Datensatz 2	31,0	6,87
Datensatz 2 (One-Hot)	28,9	6,95
Datensatz 3	21,5	8,63
Datensatz 3 (One-Hot)	24,3	8,38

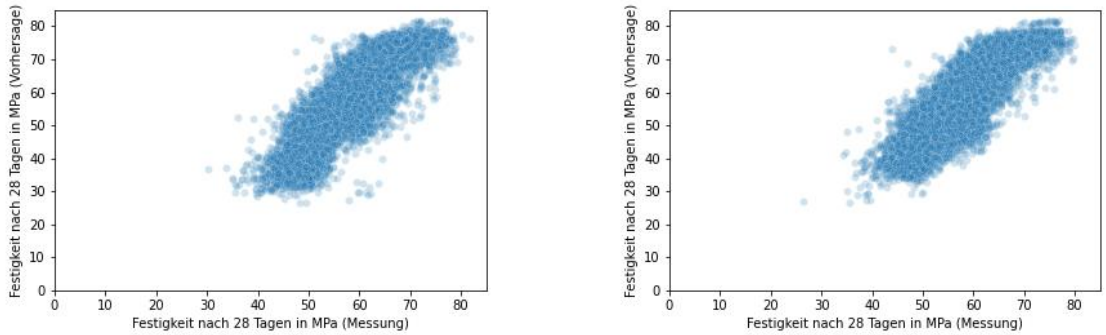


Abbildung 25 MLP – Vorhersagequalität der Festigkeit nach 28 Tagen.
Links: Datensatz 1, Rechts: Datensatz 1-One-Hot

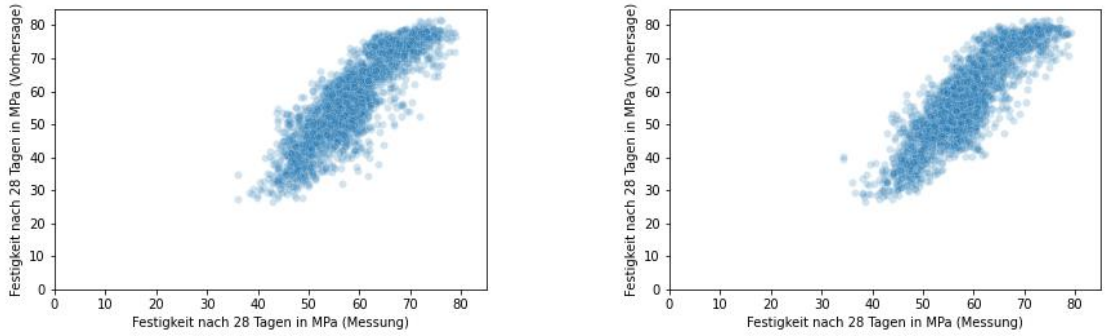


Abbildung 26 MLP – Vorhersagequalität der Festigkeit nach 28 Tagen.
Links: Datensatz 2, Rechts: Datensatz 2-One-Hot

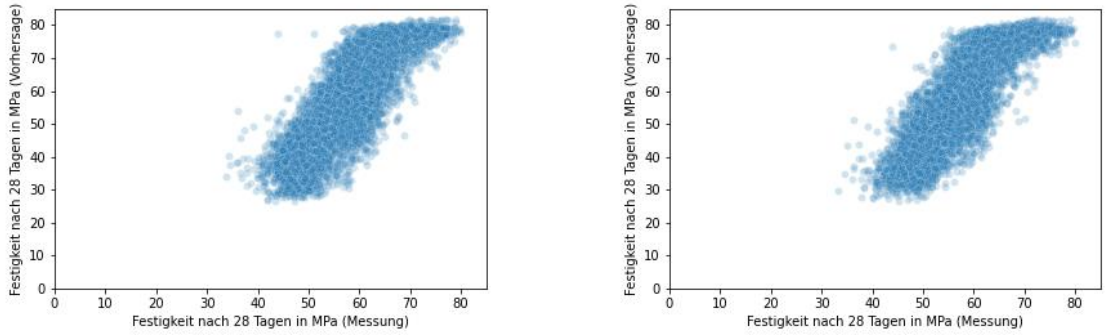


Abbildung 27 MLP – Vorhersagequalität der Festigkeit nach 28 Tagen.
Links: Datensatz 3, Rechts: Datensatz 3-One-Hot

Durch die Reduktion der Kategorien der Eingangsdaten konnte die Vorhersagequalität mit Datensatz 1 deutlich verbessert werden. Statt aller verfügbaren Kategorien wurden nur noch die Festigkeiten nach 2 und 28 Tagen (Zielwert), die spezifische Oberfläche nach Blaine, die Werks-Identifikationsnummer sowie ein Teil der Normbezeichnung (Zementtyp; A,B,C; S, L, LL, etc.) verwendet. Die Anzahl der Hidden Layer blieb bei 1 und die Anzahl der Neuronen wurde auf 8 erhöht. In Datensatz 1 konnte so der RMSE auf 5,78 MPa reduziert und die Trefferquote auf 39,5 % erhöht werden (Abbildung 28).

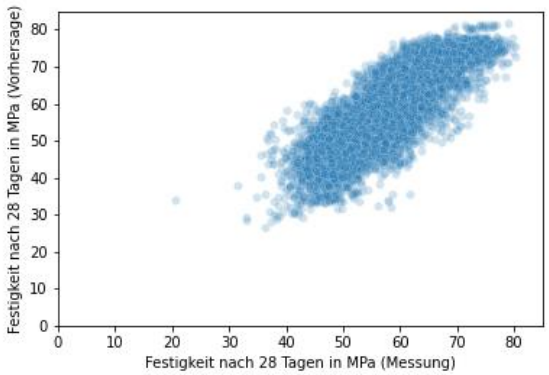


Abbildung 28 Optimierung des Multilayer Perceptron-Modells in KNIME

5.3 Scikit learn

Die Scikit Learn-Bibliothek für Python liefert eine Reihe von Machine Learning- und Optimierungsalgorithmen. In der Studie wurden folgende Algorithmen aus der Bibliothek eingesetzt: Lineare Regression, Decision Tree Regression und Random Forest Regression. Zusätzlich wurden an der Random Forest Regression zwei Optimierungsalgorithmen ausprobiert: Grid Search und Random Search. Die Bibliothek enthält keine eigene Implementierung für neuronale Netze, deshalb wurde zusätzlich ein neuronales Netz mit Tensorflow berechnet.

5.3.1 Lineare Regression

In den Abbildung 29 bis Abbildung 31 sind die Ergebnisse der Vorhersagequalität der linearen Regression dargestellt. In der Tabelle 6 sind die Trefferquoten und RMSE auf den sechs Datensätzen gegenübergestellt. Die lineare Regression zeigt bei Verwendung von Datensatz 1 und Datensatz 2 die besten Ergebnisse. Datensatz 1 mit One-Hot-Kodierung führt zu keinen nutzbaren Ergebnissen. Allgemein scheint diese Regression für die Vorhersage der 28-Tage-Festigkeit auf dem vorliegenden Datensatz wenig geeignet zu sein.

Tabelle 6 Vorhersagequalität Lineare Regression

	Trefferquote H(±3 MPa) in %	RMSE in MPa
Datensatz 1	58,15	3,77
Datensatz 1 (One-Hot)	0,00	$> 3 \times 10^{11}$
Datensatz 2	59,03	3,70
Datensatz 2 (One-Hot)	0,00	$> 1 \times 10^{11}$
Datensatz 3	61,03	3,55
Datensatz 3 (One-Hot)	0,00	$> 3 \times 10^{11}$

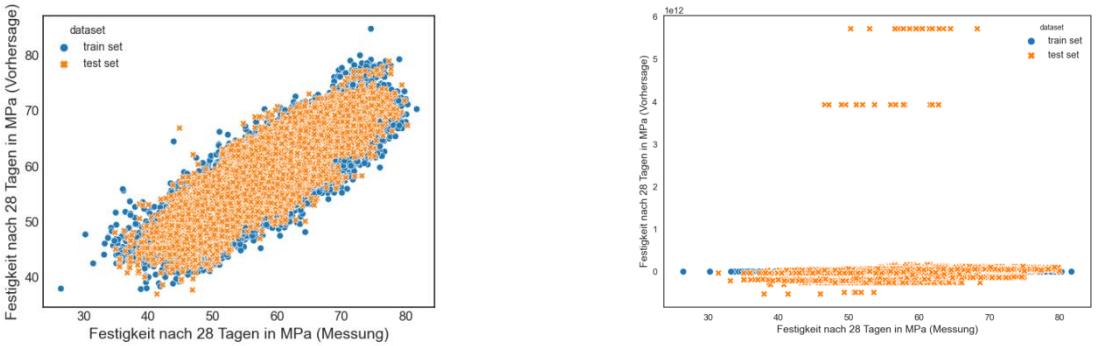


Abbildung 29 Lineare Regression – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 1, Rechts: Datensatz 1-One-Hot

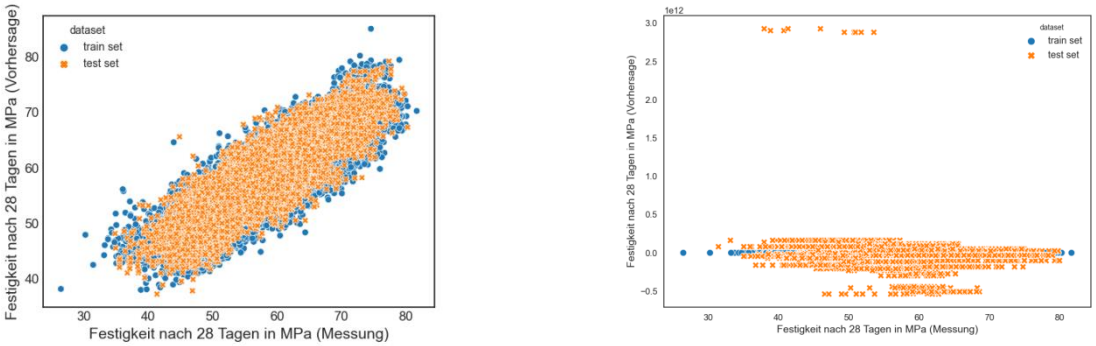


Abbildung 30 Lineare Regression – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 2, Rechts: Datensatz 2-One-Hot

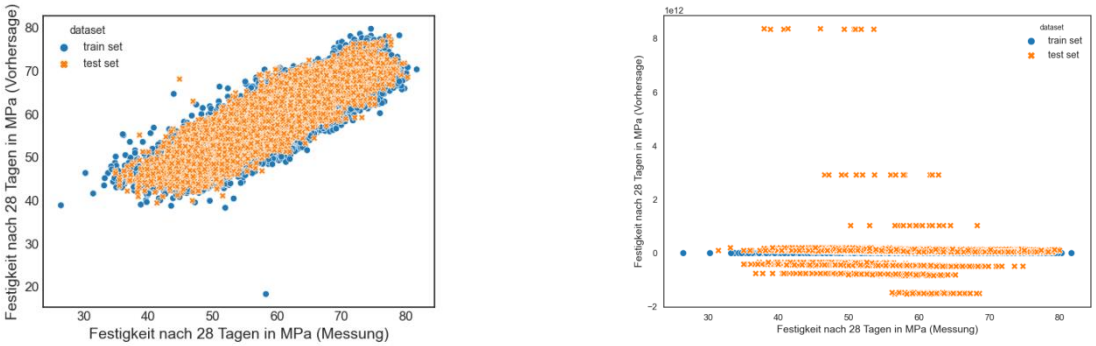


Abbildung 31 Lineare Regression – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 3, Rechts: Datensatz 3-One-Hot

5.3.2 Decision Tree Regression

Die Decision Tree Regression wurde mit vordefiniertem Standardparameter ausgewählt, hierbei wurde die Wahl der Tiefe, der Anzahl der Blätterknoten und Lernrate den internen Algorithmen überlassen. In den Abbildung 32 bis Abbildung 34 sind die Ergebnisse der Vorhersagequalität der Decision Tree Regression dargestellt. In Tabelle 7 sind die Trefferquoten und der RMSE der sechs Datensätze gegenübergestellt. Die Decision Tree Regression überfittet auf den Datensätzen 1 bis 3 mit numerischer Kodierung der Kategorien. Diese Erkenntnis kann aus der perfekten Vorhersage der Trainingsdaten (blau) und der wesentlich schlechteren Vorhersage des Testdatensatzes abgeleitet werden. Entsprechend ist an diesen Datensätzen die Trefferquote niedrig bei einem hohen RMSE von über 6 MPa. Bei den Datensätzen mit One-Hot-Kodierung sehen die Ergebnisse besser aus. Hier ist die Trefferquote höher als 69 % und RMSE liegt unter 3,5 MPa.

Tabelle 7 Vorhersagequalität Decision Tree Regression

	Trefferquote H(±3 MPa) in %	RMSE in MPa
Datensatz 1	34,43	7,23
Datensatz 1 (One-Hot)	69,67	3,20
Datensatz 2	33,84	7,18
Datensatz 2 (One-Hot)	69,01	3,29
Datensatz 3	35,90	6,85
Datensatz 3 (One-Hot)	69,52	3,21

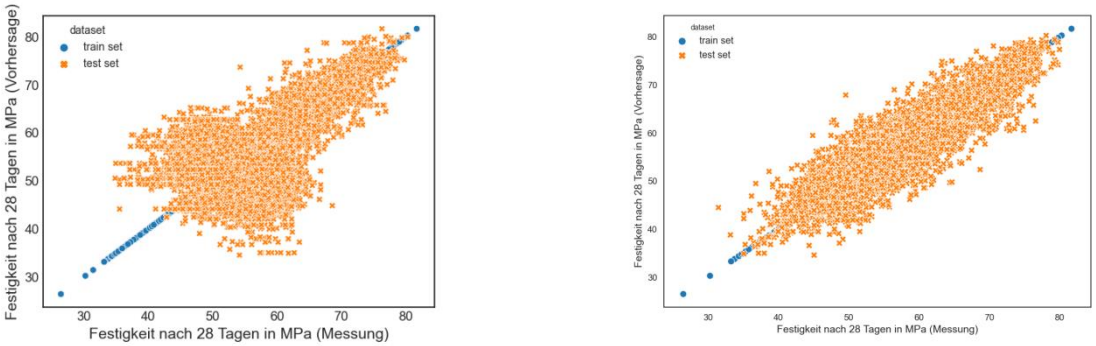


Abbildung 32 Decision Tree Regression – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 1, Rechts: Datensatz 1-One-Hot

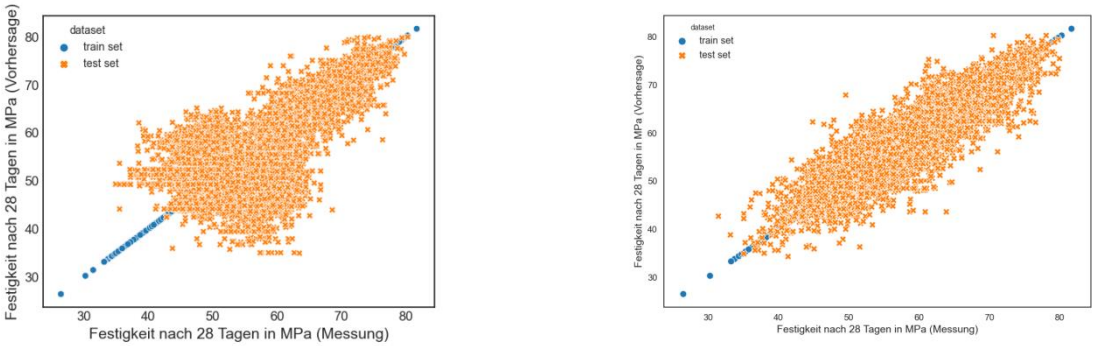


Abbildung 33 Decision Tree Regression – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 2, Rechts: Datensatz 2-One-Hot

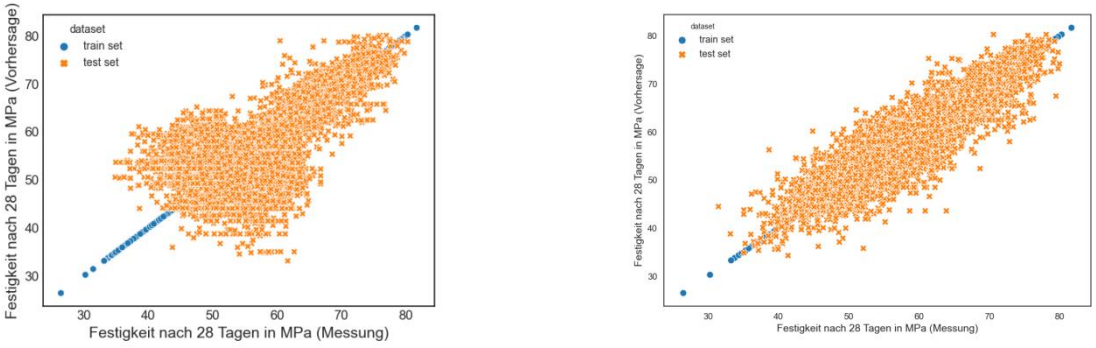


Abbildung 34 Decision Tree Regression – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 3, Rechts: Datensatz 3-One-Hot

5.3.3 Random Forest Regression

In den Abbildung 35 bis Abbildung 37 sind die Ergebnisse der Vorhersagequalität der Random Forest Regression dargestellt. In der Tabelle 8 sind die Trefferquoten und RMSE auf den sechs Datensätzen gegenübergestellt. Der Random Forest wurde aus insgesamt 10 Standard Decision Tree Regressionen aufgebaut. Auch hier zeigt sich ein deutlich besseres Ergebnis bei den One-Hot-Datensätzen.

Tabelle 8 Vorhersagequalität Random Forest Regression

	Trefferquote H(±3 MPa) in %	RMSE in MPa
Datensatz 1	30,59	6,93
Datensatz 1 (One-Hot)	81,66	2,43
Datensatz 2	30,92	6,86
Datensatz 2 (One-Hot)	82,32	2,39
Datensatz 3	32,55	6,54
Datensatz 3 (One-Hot)	81,88	2,38

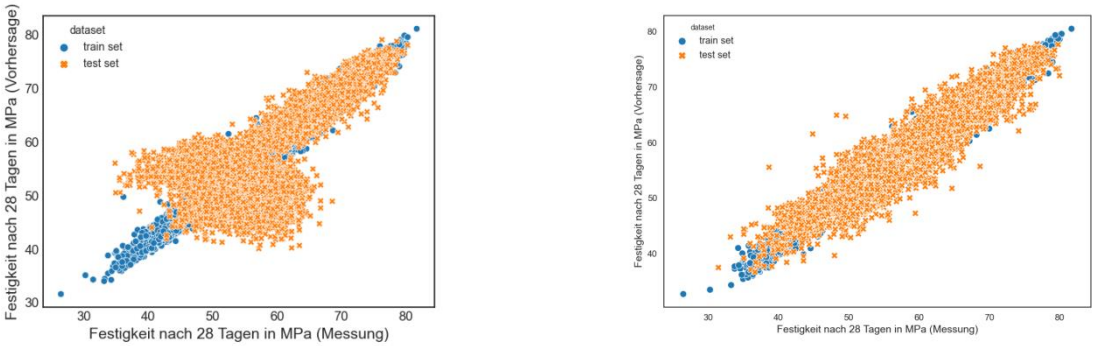


Abbildung 35 Random Forest Regression – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 1, Rechts: Datensatz 1-One-Hot

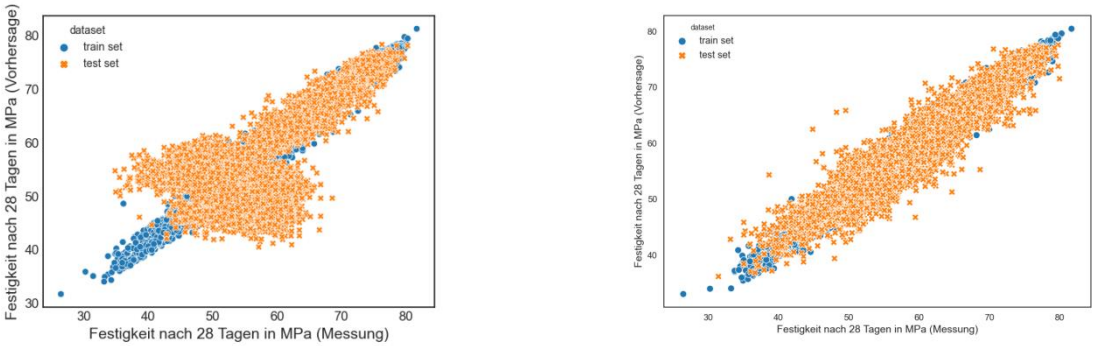


Abbildung 36 Random Forest Regression – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 2, Rechts: Datensatz 2-One-Hot

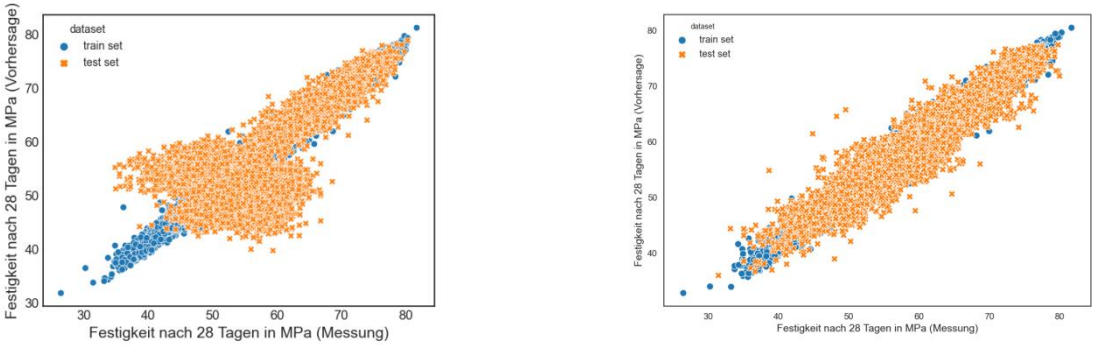


Abbildung 37 Random Forest Regression– Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 3, Rechts: Datensatz 3-One-Hot

5.3.4 Optimierung der Random Forest Regression mit Grid Search

In den Abbildung 38 bis Abbildung 40 sind die Ergebnisse der Vorhersagequalität nach Optimierung der Random Forest Regression mit Grid Search dargestellt. In der Tabelle 11 sind die Trefferquoten und RMSE-Fehler auf den sechs Datensätzen gegenübergestellt. Insbesondere auf den Datensätzen mit One-Hot-Kodierung bringt der Optimierungsalgorithmus eine kleine Verbesserung.

Tabelle 9 Vorhersagequalität Optimierung der Random Forest Regression mit Grid Search-Optimierung

	Trefferquote H(±3 MPa) in %	RMSE in MPa
Datensatz 1	34,70 (+0,8 %*)	5,90
Datensatz 1 (One-Hot)	82,15 (+17,9 %*)	2,39
Datensatz 2	41,95 (+24,0 %*)	5,08
Datensatz 2 (One-Hot)	82,07 (+18,9 %*)	2,38
Datensatz 3	48,32 (+34,6 %*)	4,35
Datensatz 3 (One-Hot)	82,12 (+18,1 %*)	2,36

* Relative Änderung im Vergleich zum Random Forest ohne Grid Search Optimierer

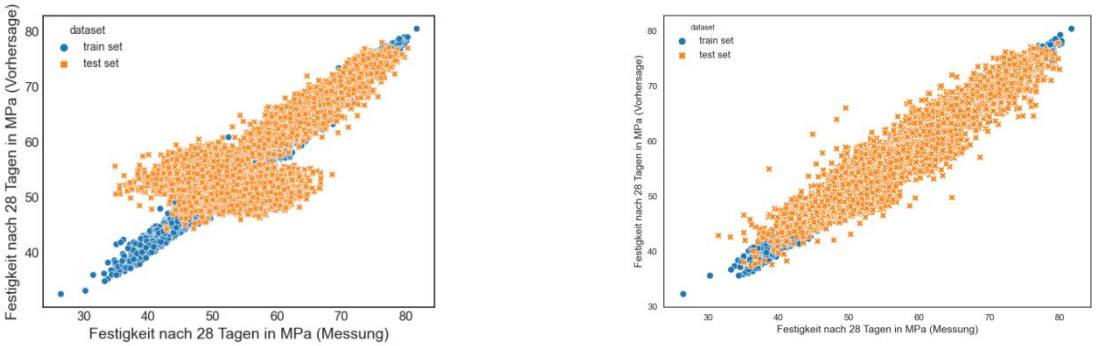


Abbildung 38 Optimierung der Random Forest Regression mit Grid Search – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 1, Rechts: Datensatz 1-One-Hot

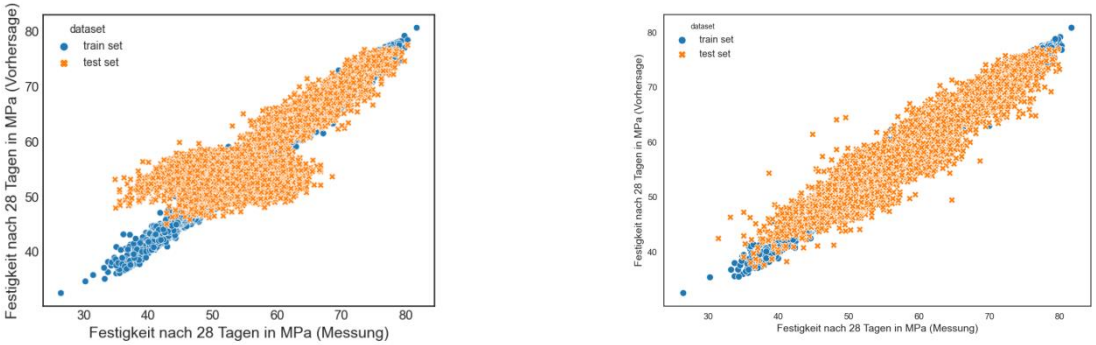


Abbildung 39 Optimierung der Random Forest Regression mit Grid Search – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 2, Rechts: Datensatz 2-One-Hot

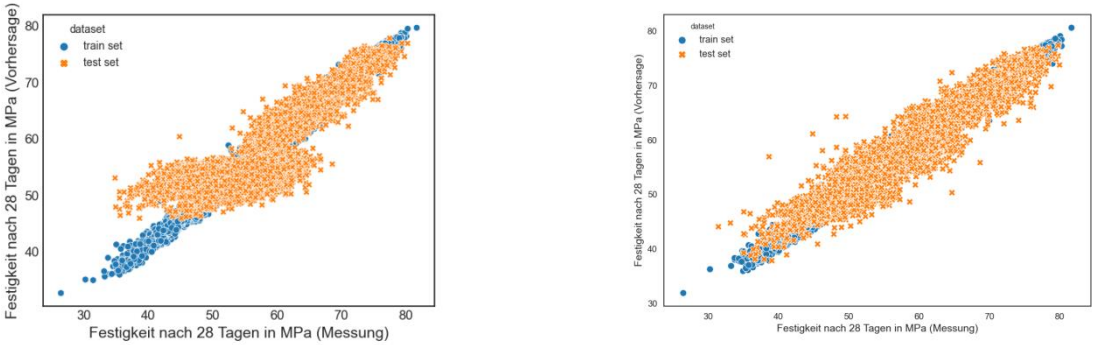


Abbildung 40 Optimierung der Random Forest Regression mit Grid Search – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 3, Rechts: Datensatz 3-One-Hot

5.3.5 Optimierung der Random Forest Regression mit Random Search

In Abbildung 41 bis Abbildung 43 sind die Ergebnisse der Vorhersagequalität der Optimierung der Random Forest Regression mit Grid Search dargestellt. In der Tabelle 11 sind die Trefferquoten und RMSE auf den sechs Datensätzen gegenübergestellt. Insbesondere auf den Datensätzen mit One-Hot-Kodierung bringt der Optimierungsalgorithmus eine kleine Verbesserung. Allgemein scheint Random Search-Algorithmus etwas bessere Ergebnisse zu liefern als der Grid Search-Algorithmus.

Tabelle 10 Vorhersagequalität Optimierung der Random Forest Regression mit Random Search-Optimierer

	Trefferquote H(±3 MPa) in %	RMSE in MPa
Datensatz 1	34,82 (+1,1%*)	5,79
Datensatz 1 (One-Hot)	82,42 (+18,3%*)	2,36
Datensatz 2	38,87 (+14,9%*)	5,17
Datensatz 2 (One-Hot)	82,30 (+19,3%*)	2,35
Datensatz 3	50,82 (+41,6%*)	4,08
Datensatz 3 (One-Hot)	82,31 (+18,4%*)	2,34

* Relative Änderung im Vergleich zum Random Forest ohne Random Search Optimierer

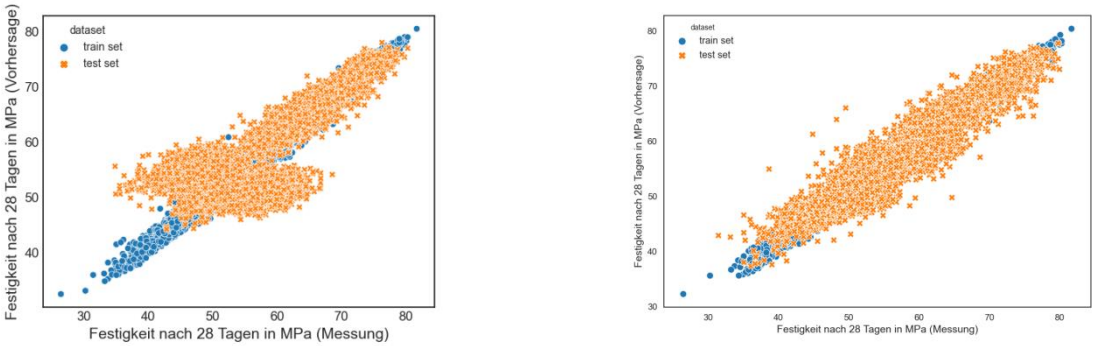


Abbildung 41 Optimierung der Random Forest Regression mit Random Search – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 1, Rechts: Datensatz 1-One-Hot

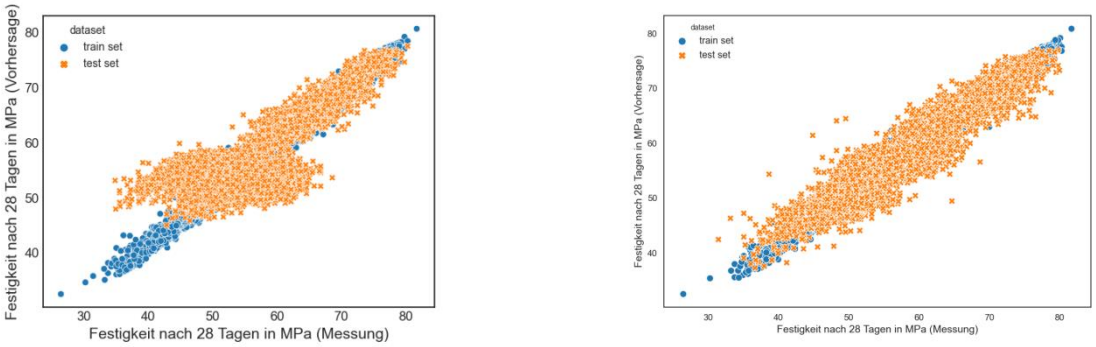


Abbildung 42 Optimierung der Random Forest Regression mit Random Search – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 2, Rechts: Datensatz 2-One-Hot

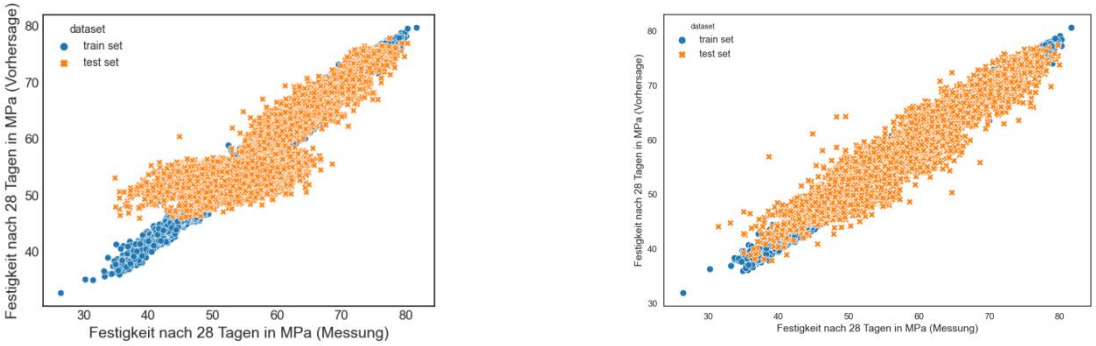


Abbildung 43 Optimierung der Random Forest Regression mit Random Search – Vorhersagequalität der Festigkeit nach 28 Tagen. Links: Datensatz 3, Rechts: Datensatz 3-One-Hot

5.4 Tensorflow – Neuronale Netzte

Für die Vorhersage wurde ein neuronales Netz mit Tensorflow aufgebaut, das aus einer einzigen verborgenen Schicht sowie jeweils einer Eingabe- und Ausgabeschicht besteht. In Abbildung 44 bis Abbildung 46 sind die Ergebnisse der Vorhersagequalität des neuronalen Netzes dargestellt. In der Tabelle 11 sind die Trefferquoten und RMSE auf den sechs Datensätzen gegenübergestellt.

Tabelle 11 Vorhersagequalität Neuronale Netzte in Tensorflow

	Trefferquote H(±3 MPa) in %	RMSE in MPa
Datensatz 1	37,95	5,51
Datensatz 1 (One-Hot)	75,27	2,70
Datensatz 2	53,67	4,03
Datensatz 2 (One-Hot)	75,90	2,69
Datensatz 3	46,88	4,58
Datensatz 3 (One-Hot)	77,30	2,67

Während des Gebrauchs von Tensorflow kam es zu größeren Schwierigkeiten nach einem Update von Tensorflow 1.x auf die Version 2.4. In der Version 2.4 kam es im Vergleich zu Version 1.x zur Änderungen der Namen- und Schnittstellendefinitionen einiger Methoden Die Fehler hätten nur mit großem Aufwand behoben werden können, sodass die älteren Quellcodes nicht mit der neuen Version lauffähig waren. Ein Wechsel zurück zur älteren Version war entsprechend nicht vermeidbar, um bereits entwickelte Programme weiterverwenden zu können. Die hohe Dynamik der Entwicklung neuer Methoden und Softwarelösungen ist bei der Entwicklung und beim Betrieb von Anwendungen des Maschinellen Lernens zu berücksichtigen.

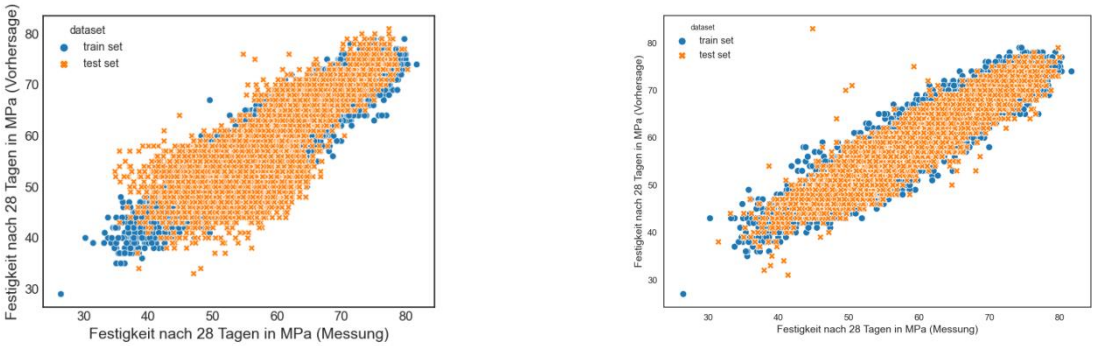


Abbildung 44 Neuronale Netzte – Vorhersagequalität der Festigkeit nach 28 Tagen.
Links: Datensatz 1, Rechts: Datensatz 1-One-Hot

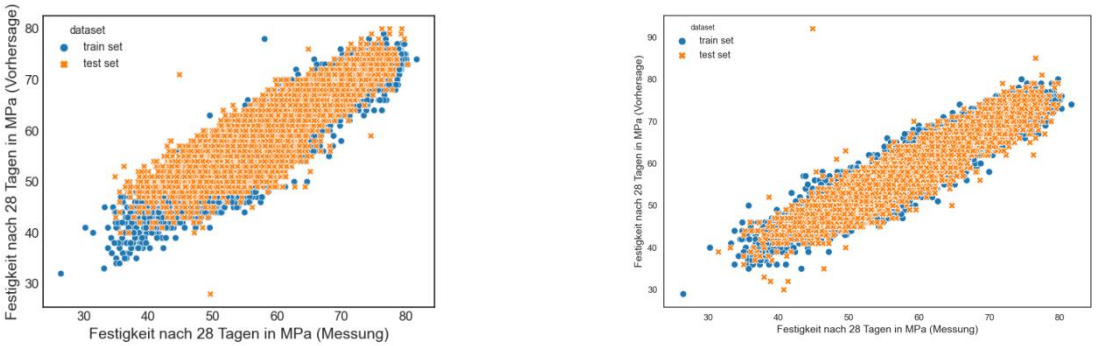


Abbildung 45 Neuronale Netzte – Vorhersagequalität der Festigkeit nach 28 Tagen.
Links: Datensatz 2, Rechts: Datensatz 2-One-Hot

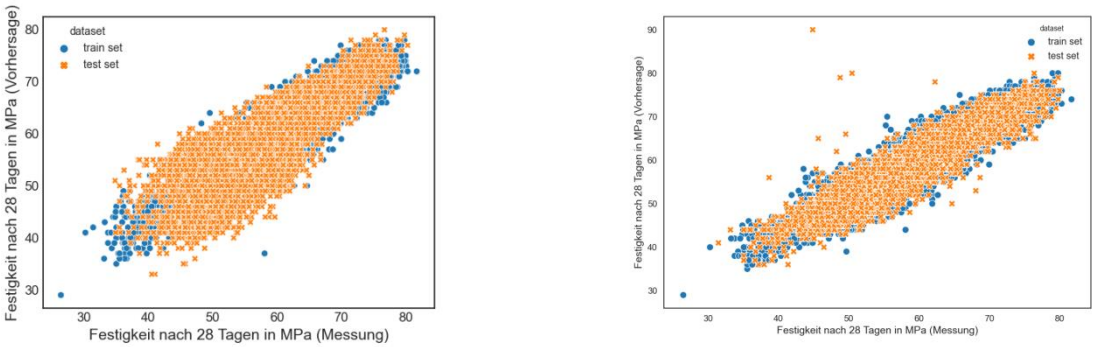


Abbildung 46 Neuronale Netzte – Vorhersagequalität der Festigkeit nach 28 Tagen.
Links: Datensatz 3, Rechts: Datensatz 3-One-Hot

6 Extrapolation von Ergebnissen

Die Modelle wurden allesamt auch für Zemente mit typischen Daten aus Laboruntersuchungen trainiert. Der Anteil an Ausreißern bzw. an Zementen, die die Normanforderungen nicht erfüllen, ist sehr klein. Innerhalb dieses Datenraums können, wie bereits gezeigt, teilweise recht gute Trefferquoten > 80 % bei der Vorhersage erzielt werden. Über die Vorhersagequalität bei wirklichen Ausreißern kann auf dieser Basis aber vermutlich keine fundierte Aussage getroffen werden. Um die Extrapolationsfähigkeit eines trainierten Modells dennoch beispielhaft betrachten zu können, wurde der Einfluss der Mahlfeinheit nach Blaine untersucht. Höhere Mahlfeinheiten bewirken in der Regel eine Zunahme der Festigkeit nach 28 Tagen.

Für eine systematische Untersuchung des Einflusses der Mahlfeinheit werden Datensätze des gleichen Zements bei unterschiedlicher Feinheit benötigt. Da die Mahlfeinheit aber auch direkt die Festigkeit nach 2 Tagen beeinflusst, können diese nicht durch einfache Veränderung der Mahlfeinheit erzeugt werden. Daher wurde zunächst ein Modell zur Vorhersage der 2d-Festigkeit trainiert. Dieses wurde genutzt, um die Festigkeit nach 2 Tagen für 3 Testdatensätze mit unterschiedlichen Mahlfeinheiten von 1800 bis 5500 cm²/g zu bestimmen. Für die Erstellung der 3 Datensätze wurden 3 CEM I-Zemente per Zufall aus der Datenbank ausgewählt, einer mit einer hohen, einer mit einer mittleren und einer mit einer geringen Ausgangsfeinheit. Die erzeugten synthetischen Daten wurden abschließend in KNIME einem Random Forest und einem Neuronalen Netz zugeführt. Die Ergebnisse sind in Abbildung 47 und Abbildung 48 dargestellt.

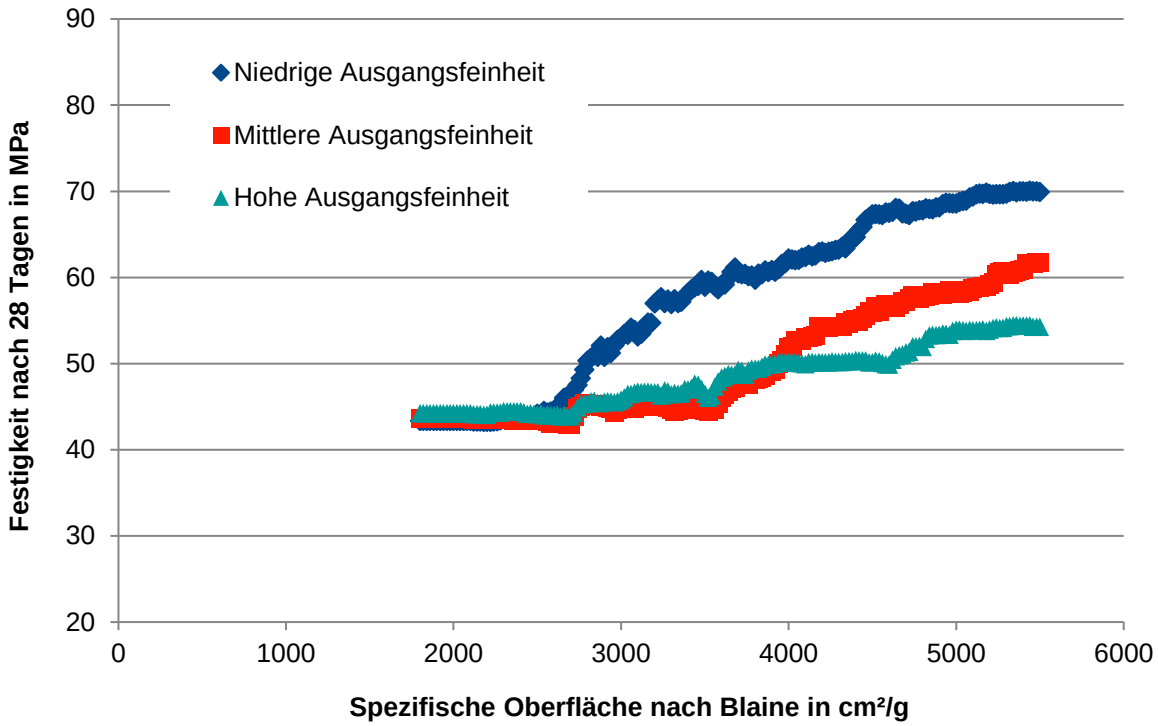


Abbildung 47 Extrapolation mit dem Random Forest-Modell

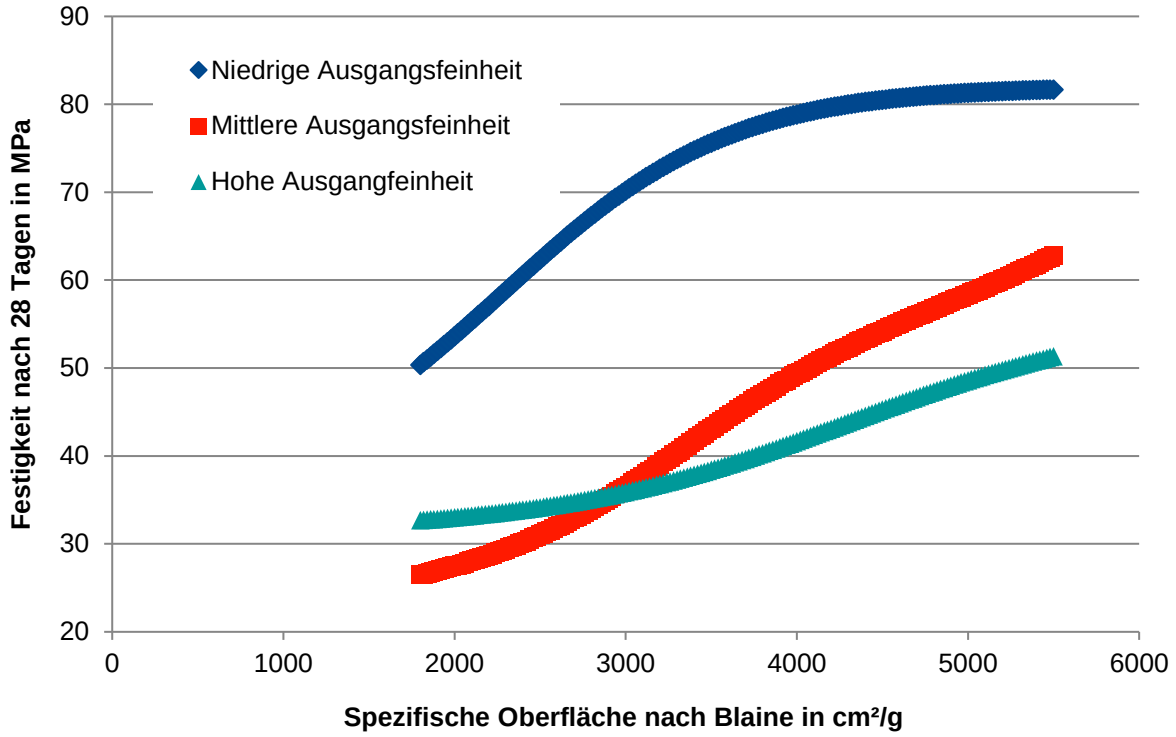


Abbildung 48 Extrapolation mit dem Neuronalen Netz

Die unterschiedlichen Verläufe der Kurven der 3 Ausgangszemente sind grundsätzlich durch die unterschiedliche Herkunft erklärbar. Die Plausibilität der quantitativen Unterschiede ist jedoch in Frage zu stellen. Insbesondere der Verlauf der durch das neuronale Netz (Abbildung 48) ermittelten blauen Kurve für die geringe Ausgangsfeinheit ist wenig realistisch. Bei 2000 cm²/g wird bereits eine Festigkeit vorhergesagt, die als zu hoch für die Festigkeitsklasse 32,5 anzusehen ist. Aus dieser stammte der zur Erzeugung der synthetischen Daten genutzte Ausgangsdatensatz. Davon abgesehen liefert das Neuronale Netzt einen stetigen und gleichmäßigen Verlauf der Festigkeiten, wie er grundsätzlich plausibel erscheint. Auch gelingt eine Extrapolation in Feinheitsbereiche (< 2500 cm²/g und > 5000 cm²/g), die in der Datenbank wenig bis gar nicht vertreten sind. Hier zeigt der Random Forest in Abbildung 47 deutlichere Schwächen. Bei sehr geringen Feinheiten wird dort kein weiterer Einfluss auf die Festigkeit mehr erkannt und das Modell liefert nahezu konstante Werte. Der Verlauf der Kurven ist zudem ungleichmäßiger und zeigt Sprünge.

Dieses ebenfalls datenbasierte Experiment ist mit Vorsicht zu behandeln. Da die synthetischen Daten mittels eines weiteren Modells für die Vorhersage der Festigkeit nach 2 Tagen erstellt wurden, unterliegen diese bereits einem Modellfehler, der nicht sichtbar in die Ergebnisse der hier visualisierten Vorhersage eingeht. Dennoch ist erkennbar, dass die Vorhersage von Daten außerhalb des Trainingsbereichs je nach Algorithmus problematisch sein kann. Entsprechend sind die geplante Nutzung von Modellen und die damit verbundenen Eingangsdaten immer vor dem Hintergrund der Trainingsdaten zu bewerten.

7 Zusammenfassung der Ergebnisse

In der vorliegenden Studie wurden verschiedene Softwarelösungen für das Maschinelle Lernen erprobt, um die Druckfestigkeit nach 28 Tagen vorherzusagen. Hierzu wurde eine Datenbank mit etwa 70.000 anonymen Datensätzen verschiedener Zementwerke verwendet. Die Daten enthalten wenige Ausreißer und sind gut strukturiert.

Am Markt ist eine Vielzahl von Softwarelösungen verfügbar, um Daten durch Maschinelles Lernen auszuwerten und Modelle zu trainieren. Diese Tools sind teils frei (z.B. Tensorflow, SciKit Learn) oder als kommerzielle Komplettlösung (z.B. RapidMiner, ProcessMiner) verfügbar. Auch kommerzielle Lösungen greifen teils auf OpenSource Libraries zurück, sind aber im Umgang mit den Daten überall vollständig transparent. Die Anzahl der Programme, die sowohl eine gute Anwendbarkeit als auch ein breites Methodenportfolio anbieten, ist über die letzten 3 Jahre stetig gestiegen. Erkennbar ist mittlerweile auch, dass etablierte Softwareanbieter, wie z.B. Mathworks für ihre Produkte, entsprechende Toolboxen anbieten. Insbesondere der Test von RapidMiner zeigt, dass sich in einigen Anwendungen sehr schnell Vorhersagemodelle durch maschinelles Lernen erzeugen lassen. Derartige Anwendungen eignen sich daher für einen schnellen Einstieg. Die Einfachheit verführt den Nutzer aber auch dazu, die erhaltenen Ergebnisse nicht weiter zu hinterfragen. Wie sensibel die Abstimmung von Daten, Aufbereitungsmethoden und eingesetzten Algorithmen sein kann, zeigen die in dieser Studie durchgeführten Untersuchungen.

Zur Modellbildung wurden hier Python mit Tensorflow und SciKit Learn sowie KNIME verwendet. Sämtliche Lösungen bieten mehr oder wenig umfangreiche Möglichkeiten der Datenanalyse und -aufbereitung. Für KNIME und die Python-basierten Libraries wurde die Datenaufbereitung in Python durchgeführt. KNIME bietet allerdings auch selbst umfangreiche Möglichkeiten der Datenaufbereitung. Der Einfluss der Datenaufbereitung wurde durch Verwendung von drei unterschiedlich aufbereiteten Datensätzen betrachtet. Im ersten Datensatz sind keine Informationen zur Zementchemie enthalten. Im zweiten Datensatz wurden diese Informationen zugelassen, obwohl die Datenlage relativ gering ist. Damit reduzierte sich die Größe des Datensatzes stark. Im dritten Datensatz wurden fehlende Informationen zur Chemie durch den Median der jeweiligen Zementsorte des Werkes ersetzt.

Die Ergebnisse der verschiedenen Datensätze in den unterschiedlichen Modellen zeigen, dass die Kombination von Daten und Modell einen wesentlichen Einfluss auf das Ergebnis haben kann. Dabei wurden einige Modelle mit Trefferquoten von 82 – 84 %, bezogen auf einen zulässigen Fehler von 3 MPa und einem RMSE von unter 2,5 MPa, erzeugt.

In KNIME wurden die Modelle „Gradient Boosted Trees“, „Random Forest“ und „Mulitlayer Perceptron“ gegenübergestellt. Das Modell „Gradient Boosted Trees“ erzielt die besten Ergebnisse mit einem RMSE von 2,09 MPa mit dem 2. Datensatz (Kapitel 5.2.1). Die „Random Forest“ Modelle waren etwas schlechter. Die Vorhersagequalität der beiden „Baum-Modelle“ profitiert teils von einer größeren Anzahl von Merkmalen, auch wenn dadurch weniger Datensätze verwendet werden können. Das Neuronale Netz – „Multilayer Perceptron“ wies den höchsten RSME (5,8 MPa) im Vergleich auf. Besonders das Neuronale Netz profitierte von weniger Merkmalen und einer größeren Anzahl von Datensätzen.

In RapidMiner und KNIME laufen im Hintergrund Optimierungsalgorithmen, die unsichtbar für den Anwender die Regressionsalgorithmen verbessern. Diese sind für den Nutzer auch nicht

manipulierbar. Dies kann unter anderem zu Problemen bei der Extrapolation des Modells auf andere Datensätze führen.

Die Methode der linearen Regression in SciKit Learn konnte nicht sinnvoll auf die Datensätze angewandt werden. Der RMSE war mit einem Wert von 3,55-5,77 MPa niedriger als bei dem Neuronalen Netz mit KNIME, aber dennoch sehr hoch. Bei der Nutzung der One-Hot-kodierten Datensätze versagte das Modell vollständig. Im direkten Vergleich der Random Forest Regression war in SciKit Learn eine höhere Abhängigkeit der Kategorisierung festzustellen, als in KNIME. Die Ergebnisse nach der One-Hot-Kategorisierung waren deutlich besser, als die der numerischen Kategorisierung, und damit mit den guten Ergebnissen aus KNIME vergleichbar. Dies zeigte sich auch in den Ergebnissen des Decision Trees.

Das Neuronale Netz mit TensorFlow liefert für den Trainings- und den Testdatensatz etwa gleich gute Ergebnisse. Auch hier zeigt sich eine Abhängigkeit der Kategorisierung. Die One-Hot-Kategorisierung liefert wesentlich bessere Ergebnisse. Auch im Vergleich zum Neuronalen Netz mit KNIME lieferte TensorFlow bessere Ergebnisse. Hier wurde ein RMSE von etwa 2,7 MPa bestimmt.

Stark geführte Werkzeuge, wie KNIME oder RapidMiner liefern schnell gute Ergebnisse. Eine Datenvorbereitung wird teilweise von der Software übernommen. Durch sinnvolle Datenauswertung und die Wahl geeigneter Modelle können aber auch mit Scikit Lern oder anderen Bibliotheken schnell gute Ergebnisse erzeugt werden. Hier wird jedoch ein Mindestmaß an Fachwissen benötigt und die Fehlerquellen sind höher, als bei den geführten Werkzeugen.

Insgesamt zeigt sich, dass allein durch die Wahl der Modelle und Methoden der Datenaufbereitung große Unterschiede bei den Ergebnissen erzeugt werden können. Dabei ist zu beachten, dass die Qualität der untersuchten Daten als sehr hoch anzusehen ist. Bei stärker verrauschten Betriebsdaten sind zusätzliche Probleme zu erwarten. Dies wurde durch Recherchen und Interviews mit weiteren Anwendern in der Zementindustrie bestätigt. Grundsätzlich ist es erforderlich, die Ergebnisse von Modellrechnungen sorgfältig zu prüfen und durch geeignete Prüfdaten zu validieren. Die Extrapolationsfähigkeit der Modelle ist dabei besonders zu beachten. Weiterhin ist beachten, dass in praktischen Anwendungen nicht allein die Trefferquote oder der RMSE ausschlaggebend ist. Faktoren wie die Robustheit der Anwendung gegenüber schwankenden Daten können dazu führen, dass Kompromisse bei der Genauigkeit in Kauf genommen werden müssen.

8 Ausblick

Die durchgeführten Arbeiten haben einen praktischen Einblick in die Anwendung der Methoden des Maschinellen Lernens ermöglicht und konnten gezielt den Einfluss unterschiedlicher Werkzeuge und Methoden darstellen. Die Erkenntnisse werden in weitere Forschungsprojekte, z.B. zur Robustheit von KI-Anwendungen, einfließen und zudem künftig im Weiterbildungsangebot des VDZ genutzt werden.

Die Ergebnisse sollen weiterhin dazu genutzt werden, um die Methoden des Maschinellen Lernens anhand des untersuchten Praxisbeispiels transparent und nachvollziehbar im Rahmen einer Grundlagenveröffentlichung aufzubereiten.

9 Literaturverzeichnis

[Pe 2011] Pedregosa et al.. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research 2011, **12**, S. 2825-2830

[Aba; al. 2015] Abadi, Martín; al., et. *TensorFlow: Large-scale machine learning on heterogeneous systems*. 2015, S. Software available from tensorflow.org

[Bis 2006] Bishop, Christopher M.. *Pattern Recognition and Mchine Learning*. Singapore.: Springer Science+Business Media 2006

[Gartner 2021] Gartner. *Gartner Magic Quadrant Data Science 2021, https://www.gartner.com/reviews/market/data-science-machine-learning-platforms*. abgerufen am 28.4.21 2021

[Ger 2018] Geron, Aurélien. *Praxiseinstieg Machine Learning mit Scikit-Learn & Tensorflow*. Heidelberg: dpunkt.verlag GmbH 2018

[Gru 2016] Grus, Joel. *Einführung in Data Science*. Heidelberg: dpunkt.verlag GmbH 2016

[Hac 2004] Hackley, Lum,Gintautas, Gerraris. *Particle Size Analysis by LaserDiffraction Spectrometry: Application to Cementitious Powders*. NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY 2004

[ISO 2009] ISO13320. *Particle size analysis — Laser diffraction methods, ISO 13320:2009*. 2009

[KNI 2021] KNIME. *Gradient Boosted Trees Learner, https://hub.knime.com/knime/extensions/org.knime.features.ensembles/latest/org.knime.base.node.mine.treeensemble2.node.gradientboosting.learner.classification.GradientBoostingClassificationLearnerNodeFactory2*. aberufen am 19.04.2021: 2021

[Kri 2007] Kriesel, D.. *A Brief Introduction to Neural Networks*. 2007

[Lag; Ree; Wri; Wri 1998] Lagarias, Jeffrey C.; Reeds, James A.; Wright, Margaret H.; Wright, Paul E.. *Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions*. SIAM Journal of Optimization 1998, **9**, S. 112–147

[Noc; Wri 1999] Nocedal, Jorge; Wright, Stephen J.. *Numerical Optimization*. New York: Springer-Verlag 1999

[Ras 2017] Rashid, Tariq. *Neuronale Netze selbst programmieren*. dpunkt.verlag GmbH: Heidelberg 2017

[Vol 2018] Volsund, M. et al.. *CEMCAP techno-economic and retrofitability analysis*. European Cement Research Academy, ECRA; Research Group CEMCAP; Research Group CLEANKER, Ed. Presentations and Posters of the ECRA/CEMCAP/CLEANKER Workshop 2018 on Carbon Capture Technologies in the Cement Industry 2018

[War 2018] Wartala, Ramon. *Praxiseinstieg Deep Learning*. Heidelberg: dpunkt.verlag GmbH 2018

[Wol 1996] Wolpert, David. *The Lack of A Priori Distinctions Between Learning Algorithms*. Neural computation 1996, **8.7**, S. 1341-1390