

Hochschule Fresenius
Fachbereich Wirtschaft & Medien
Studiengang: General Management
Studienort: Hamburg

Nutzenpotenziale von maschinellem Lernen für moderne Ansätze zur Unternehmensbewertung

Genehmigte MASTERARBEIT
zur Erlangung des akademischen Grades
eines Master of Business Administration (MBA)

Tim Stäcker
Matrikelnummer: 400214142

1. Prüfer: Prof. Dr. Michael Meser
2. Prüfer: Prof. Dr. Daniel R.M. Sonnet

Abgabedatum: 14.07.2025

Zusammenfassung

Die Unternehmensbewertung zählt zu den zentralen Aufgaben im Finanz- und Investitionsbereich. Während klassische Verfahren wie das Discounted-Cashflow-Modell (DCF) auf planungsbasierten Prognosen und finanzmathematischen Annahmen beruhen, eröffnen datengetriebene Ansätze auf Basis maschinellen Lernens neue Möglichkeiten der Bewertung. Diese Arbeit untersucht das Nutzenpotenzial ausgewählter Machine-Learning-Verfahren – insbesondere Random Forest und Gradient Boosting – im Vergleich zum DCF-Modell. Anhand realer Finanzdaten werden die Verfahren implementiert, evaluiert und hinsichtlich Prognosekraft, Transparenz und praktischer Anwendbarkeit analysiert. Die Ergebnisse zeigen, dass datenbasierte Modelle insbesondere bei komplexen oder datenreichen Unternehmenssituationen valide Ergänzungen zu klassischen Methoden darstellen können. Die Arbeit diskutiert darüber hinaus regulatorische Anforderungen und betont die Notwendigkeit eines integrativen Bewertungsverständnisses, das traditionelle und datenwissenschaftliche Perspektiven vereint.

Abstract

Business valuation is a fundamental task in finance and investment decision-making. While traditional methods such as the Discounted Cash Flow (DCF) model rely on planning-based forecasts and financial assumptions, data-driven approaches based on machine learning offer new perspectives. This thesis explores the potential of selected machine learning models—particularly Random Forest and Gradient Boosting—in comparison to the DCF approach. Using realistic financial data, the models are implemented, evaluated, and assessed with regard to predictive accuracy, transparency, and practical applicability. The results indicate that data-driven methods can serve as a valuable complement to traditional valuation models, especially in complex or data-rich contexts. The thesis also addresses regulatory requirements and emphasizes the importance of an integrated approach that combines conventional valuation logic with data science insights.

I. Inhaltsverzeichnis

II. Abbildungsverzeichnis	5
III. Tabellenverzeichnis.....	6
IV. Formelverzeichnis	7
1 Einleitung.....	8
2 Zielsetzung und Forschungsfrage.....	9
3 Traditionelle Unternehmensbewertungsmethoden.....	11
3.1 Grundlagen der Unternehmensbewertung.....	11
3.2 Regulatorische Anforderungen.....	12
3.3 Einzelbewertungsverfahren	15
3.3.1 Substanzwertverfahren	16
3.3.2 Liquidationswertverfahren	17
3.4 Mischbewertungsverfahren	17
3.5 Gesamtbewertungsverfahren	18
3.5.1 Kapitalwertmethode	19
3.5.2 Capital Asset Pricing Model (CAPM).....	20
3.5.3 Ertragswertverfahren	21
3.5.4 Discounted-Cashflow-Verfahren.....	23
3.5.5 ‚Multiple-basiertes‘ DCF-Modell.....	28
3.5.6 Effizientmarkthypothese.....	30
4 Künstliche Intelligenz und Maschinelles Lernen	31
4.1 Der Begriff Künstliche Intelligenz	32
4.2 Regelbasiertes Lernen	34
4.3 Maschinelles Lernen.....	35
4.4 Existierende Algorithmen.....	38
4.4.1 Random Forests	38
4.4.2 Gradient Boosting.....	41
4.4.3 Künstliche Neuronale Netze	43
4.4.4 Deep Learning	45
4.5 Explainable AI.....	45
4.5.1 Feature Importance	46

4.5.2 SHAP SHapley Additive exPlanations.....	47
4.5.3 LIME (Local Interpretable Model-agnostic Explanations)	48
4.6 Hyperparameter	48
4.7 Overfitting	49
4.8 Erfolgsmessung maschineller Lernmodelle	50
4.8.1 Mean Decrease Impurity (MDI)	50
4.8.2 Das Bestimmtheitsmaß R^2	51
4.8.3 Mean Absolute Error (MAE).....	52
4.8.4 Mean Absolute Percentage Error (MAPE).....	53
4.8.5 Weighted Mean Absolute Percentage Error (WMAPE)	53
5 Methodik.....	53
6 Design und Implementation selbst entwickelter Bewertungsalgorithmen	60
6.1 Entwicklung des Algorithmus für Random Forest Methode.....	60
6.2 Entwicklung des Algorithmus für Gradient Boosting Methode.....	64
6.3 Anwendung traditioneller Bewertung	66
7 Ergebnisse.....	68
8 Diskussion	71
9 Fazit.....	86
V. Literaturverzeichnis	91
VI. Anhang	99
VII. Glossar	135
VIII. Eidesstaatliche Erklärung	139

II. Abbildungsverzeichnis

Abbildung 1: WACC Zirkularitätsproblem	26
Abbildung 2: Unterscheidung von Regelbasiertem und Maschinellern in Abhängigkeit der vorliegenden Daten	33
Abbildung 3: Beispiel für Regelbasiertes Lernen.....	34
Abbildung 4: Differenzierung des Maschinellen Lernens in Unterarten.....	35
Abbildung 5: Hund vs. Wischmopp	36
Abbildung 6: Prinzip des Überwachten Lernens	37
Abbildung 7: Prinzip des Unüberwachten Lernens	37
Abbildung 8: Visualisierung des Random Forest Prinzips.....	39
Abbildung 9: Visualisierung der Gradient Boosting Methode	41
Abbildung 10: Prinzip künstlicher neuronaler Netze	44
Abbildung 11: Top Features Random Forests und berechnete Importance zur Bewertung der Merkmalsrelevanz mit relativer Bedeutung > 0,1%	62
Abbildung 12: Korrelationsmatrix Features Random Forest.....	63
Abbildung 13 Finale Feature Importance Random Forest.....	64
Abbildung 14: Feature und berechnete Importance zur Bewertung der Merkmalsrelevanz mit relativer Bedeutung > 0,1%	65
Abbildung 15: Feature Importance Gradient Boosting.....	66
Abbildung 16: Bestimmtheitsmaß R^2 pro Jahr pro Modell.....	69
Abbildung 17: <i>MAPE</i> pro Jahr pro Modell.....	69
Abbildung 18: <i>WMAPE</i> pro Jahr pro Modell	70
Abbildung 19: Verteilung der %-Abweichungen pro Modell	71
Abbildung 20: Verteilung der %-Abweichungen je Industrie	75

III. Tabellenverzeichnis

Tabelle 1: Vergleich der Methoden zur Unternehmensbewertung in Deutschland und International	15
Tabelle 2: Vergleich MDI vs. R^2	52
Tabelle 3: Vergleich der drei Modelle DCF, Random Forest und Gradient Boosting hinsichtlich der Prognosegüte für Unternehmensbewertungen	68

IV. Formelverzeichnis

Formel 1: Kapitalwert einer Investition.....	19
Formel 2: Risikoadjustierter Kapitalisierungszinssatz nach CAPM.....	20
Formel 3: Unternehmenswert Ertragswertverfahren	22
Formel 4: Unternehmenswert Discounted Cashflow Verfahren (Entity Ansatz)	25
Formel 5: Terminal Value (Entity Ansatz)	25
Formel 6: WACC mit Tax Shield.....	25
Formel 7: Unternehmenswert Discounted Cashflow Verfahren (Equity Ansatz)	27
Formel 8: Terminal Value (Equity Ansatz)	27
Formel 9: Unternehmenswert Discounted Cashflow Verfahren (multiple-basiert).....	30
Formel 10: Terminal Value im multiple-basiertem Ansatz	30
Formel 11: Bestimmtheitsmaß R^2	51
Formel 12: Mean Absolute Error (MAE)	52
Formel 13: Mean Absolute Percentage Error (MAPE)	53
Formel 14: Weighted Mean Absolute Percentage Error (WMAPE)	53

1 Einleitung

Als der Baseball-Manager Billy Beane Anfang der 2000er-Jahre begann, Spieler für die Oakland Athletics nicht mehr nach Bauchgefühl, sondern nach statistischen Kennzahlen zu bewerten, galt dies im Profibaseball als undenkbar - damit begann eine Revolution namens „Moneyball“.¹ Heute ist „Moneyball“ ein Synonym für datenbasierte Entscheidungsfindung in der Praxis – nicht nur im Sport, sondern zunehmend auch in der Wirtschaft. In ähnlicher Weise stellt sich auch in der Unternehmensbewertung die Frage: Was, wenn sich der wahre Wert eines Unternehmens nicht allein aus Planbilanzen und Cashflows ergibt – sondern aus Mustern, die in großen Mengen historischer Daten verborgen liegen?

Diese Frage gewinnt in einer zunehmend datengetriebenen Unternehmenslandschaft an Bedeutung. Während klassische Bewertungsverfahren wie das Discounted-Cashflow-Modell (DCF) oder das Ertragswertverfahren seit Jahrzehnten als methodische Grundlage für Kaufpreisverhandlungen, Finanzierungsentscheidungen oder steuerliche Bewertungen gelten, stoßen sie in volatilen Märkten, bei jungen Unternehmen oder datenarmen Situationen an ihre Grenzen. In solchen Fällen hängt das Bewertungsergebnis stark von subjektiven Annahmen und Prognosen ab – und damit von der Expertise und Erwartungshaltung des Bewertenden.

Gleichzeitig erleben datenbasierte Ansätze durch Fortschritte im Bereich des maschinellen Lernens eine rasante Entwicklung. Machine-Learning-Modelle (ML) wie Random Forests oder Gradient Boosting können große Mengen strukturierter Finanzdaten, Marktinformationen und sogar weicher Faktoren wie ESG-Ratings oder Medienberichterstattung analysieren und daraus bewertungsrelevante Muster ableiten. Die zentrale Frage ist jedoch nicht nur, ob das technisch möglich ist – sondern, ob es betriebswirtschaftlich sinnvoll ist.

„Wird AI die Bänker ersetzen“², titelte das Handelsblatt im Herbst 2023 – eine Schlagzeile, die nicht nur Aufmerksamkeit erregte, sondern auch eine tieferliegende Debatte anstößt: Sind Machine-Learning-Verfahren in der Lage, klassische Bewertungsmodelle zu übertreffen? Und wenn ja – unter welchen Bedingungen, mit welchen Daten und mit welchen Implikationen für Praxis und Regulierung?

¹ Vgl. Lewis [2004], o.S.

² Rothe et al. [2023], o.S.

Diese Arbeit widmet sich daher der Analyse des Nutzenpotenzials maschinellen Lernens für moderne Ansätze der Unternehmensbewertung. Ziel ist es nicht, klassische Modelle abzuwerten, sondern deren Stärken und Schwächen im Vergleich zu datenbasierten Verfahren herauszuarbeiten. Im Zentrum steht die Frage, inwiefern Machine-Learning-Modelle die Bewertungspraxis objektivieren, präzisieren oder beschleunigen können – und welche Anforderungen sie dabei an Daten, Modellverständnis und vor allem regulatorische Rahmenbedingungen stellen.

2 Zielsetzung und Forschungsfrage

Die vorliegende Arbeit ordnet sich in den Forschungsbereich zwischen etablierten bewertungstheoretischen Ansätzen und neueren empirischen Studien wie etwa von Grobys et. al oder Wei Chang und Tingting Zhang zu datenbasierten Bewertungsverfahren ein. Während Damodaran³ die Herausforderungen klassischer DCF-Modelle bei schwer prognostizierbaren Cashflows aufzeigt, deuten aktuelle Forschungsarbeiten von Grobys et. al⁴ oder Wei Chang und Tingting Zhang⁵ darauf hin, dass maschinelle Lernverfahren hier ergänzende Ansätze bieten können. Bisherige Studien konzentrierten sich jedoch primär auf die technische Leistungsfähigkeit dieser Algorithmen, ohne die betriebswirtschaftlichen Anwendungsvoraussetzungen systematisch zu untersuchen. Die vorliegende Arbeit trägt zu dieser Diskussion bei, indem sie einen analytischen Rahmen entwickelt, der technische ML-Kennzahlen (R^2 , MAPE, MAE) mit praktischen Implementierungskriterien in Beziehung setzt.

Vor dem Hintergrund der zunehmenden Digitalisierung betriebswirtschaftlicher Prozesse und dem Einsatz künstlicher Intelligenz im Finanzbereich ergibt sich die folgende zentrale Forschungsfrage dieser Arbeit:

Inwiefern bieten die Verfahren des maschinellen Lernens, Random Forest und Gradient Boosting, einen betriebswirtschaftlich relevanten Nutzen gegenüber der klassischen Methode der Unternehmensbewertung, dem Discounted-Cashflow-Modell?

In diesem Zusammenhang wird betriebswirtschaftlicher Nutzen mit den Faktoren Zeit und Prognosegüte der Unternehmensbewertung definiert. Diese Auswahl stützt sich auf die Überlegung, dass Unternehmensbewertungen vor allem dann als nützlich gelten,

³ Vgl. Damodaran [2012], o.S.

⁴ Vgl. Grobys et al. [2022], o.S.

⁵ Vgl. Chang & Zhang [2025], o.S.

wenn sie möglichst präzise sind und die Erstellung zeit- und kostenschonend sind, d. h. effizient.

Daraus ergeben sich die folgenden zwei Arbeitshypothesen:

1. Maschinelles Lernen erzielt eine geringere durchschnittliche Abweichung zum tatsächlichen Unternehmenswert bei limitierter Datenverfügbarkeit als das DCF-Modell.

Während DCF stark auf Annahmen basiert (z.B. zukünftige Cashflows, Abzinsungsfaktor), lernen Algorithmen des maschinellen Lernens aus historischen Daten direkt Zusammenhänge und können dadurch tatsächliche Werte oft besser antizipieren - insbesondere, wenn viele historische Transaktionen oder Bewertungen vorliegen.

2. ML-Verfahren ermöglichen eine schnellere und skalierbarere Unternehmensbewertung als das DCF-Modell, ohne signifikanten Verlust an Prognosequalität.

Sobald ein ML-Modell trainiert ist, lassen sich große Mengen an Unternehmen automatisch bewerten. DCF hingegen ist aufwendiger, da jede Bewertung individuell modelliert und validiert werden muss. Die Automatisierung macht ML besonders in datenintensiven Kontexten effizient.

Diese Forschungsfrage impliziert die Untersuchung quantitativer sowie qualitativer Unterschiede zwischen beiden Bewertungsansätzen, einschließlich ihrer Anwendbarkeit, Prognosegüte und praktischen Implikationen.

Zur Beantwortung der Forschungsfrage wird ein empirisch-analytischer Vergleichsansatz gewählt. Ziel ist es, den Nutzen von Machine-Learning-Verfahren für die Unternehmensbewertung systematisch mit der klassischen Bewertungsmethode Discounted Cashflow zu vergleichen. Zunächst werden in Kapitel 3 und 4 die theoretischen Grundlagen genereller Unternehmensbewertungsmethoden, sowie die Auswahl des Discounted-Cashflow-Modells als Benchmark klassischer Bewertungsmethoden ausführlich dargestellt. Dieses Modell dient als Referenzrahmen zur Einschätzung der Leistungsfähigkeit datengetriebener Ansätze. Im Anschluss erfolgt eine Einführung in zentrale Verfahren des maschinellen Lernens, wobei von diesen Verfahren in dieser Arbeit Random Forests und Gradient Boosting Machine Learning Algorithmen praktische Anwendung finden.

Die Entwicklung und das Training der genannten Machine Learning Algorithmen erfolgt auf Basis historischer Unternehmensdaten von 2012 bis 2020, ergänzt durch Markteinflussgrößen wie etwa Leitzinssatz, Inflationsrate und BIP-Veränderung. Die Daten werden entsprechend aufbereitet, normalisiert und in Trainings- und Testdatensätze unterteilt. Die exakte Darstellung der genutzten Daten erfolgt im Kapitel 5. In Kapitel 6 erfolgt die Entwicklung und das Training der Machine Learning Algorithmen, sowie die Ausführung des Discounted Cashflow Modell als traditionelle Bewertungsmethode auf Basis der Testdaten. In Kapitel 7 werden die Ergebnisse anhand geeigneter Metriken vorgestellt und evaluiert. Die Diskussion der Ergebnisse sowie Beantwortung der Forschungsfrage und Evaluierung der Forschungshypothese erfolgt in Kapitel 8.

3 Traditionelle Unternehmensbewertungsmethoden

In diesem Kapitel werden die Grundlagen der Unternehmensbewertung und ihren regulatorischen Anforderungen erarbeitet und auf die verschiedenen Ausprägungen der Bewertungsmethoden eingegangen. Die traditionellen Unternehmensbewertungsmethoden lassen sich in Einzelbewertungsverfahren, Mischbewertungsverfahren sowie Gesamtbewertungsverfahren unterteilen.⁶ Hierbei wird der regulatorische Rahmen – insbesondere die Vorgaben des IDW S1 Standards, des HGB sowie der IFRS – berücksichtigt, da er die Bewertungspraxis maßgeblich beeinflusst. Im Folgenden werden die Ansätze der Verfahrensarten sowie die Verfahrensmodelle und ihre Grundlagen dargestellt. Damit das abschließend genutzte Discounted-Cashflow-Modell im Rahmen der weiteren traditionellen Bewertungsmethoden eingeordnet werden kann, erfolgt auch eine Erläuterung dieser weiteren Verfahren.

3.1 Grundlagen der Unternehmensbewertung

Hinter einer Unternehmensbewertung stehen verschiedene Motive, die ihren Bedarf begründen können. Zu den wohl häufigsten Gründen zählt die Bewertung eines Unternehmens im Rahmen einer Akquisition oder der Wechsel der Eigentümer eines Unternehmens, sowie die bilanzielle Folgebewertung.⁷ Im Rahmen der Planung einer Unternehmensbewertung muss sich der Bewerter hierbei im Klaren über den Zweck, das implizierte Risiko und die individuelle Begrenzung des verwendeten Bewertungsverfahrens bewusst sein. Für unterschiedliche Zwecke können hierbei

⁶ Vgl. Peemöller [2023], S. 47

⁷ Vgl. Drukarczyk & Schüler [2021], S. 3

unterschiedliche Verfahren relevant sein, wie auch im folgenden Kapitel dargestellt wird. Der Autor Moxter ordnete dies wie folgt ein: „Es gibt nicht den schlechthin richtigen Unternehmenswert: Da Unternehmenswertermittlungen sehr unterschiedlichen Zwecken dienen können, ist der richtige Unternehmenswert jeweils der zweckadäquate.“⁸

Die Zwecke einer Unternehmensbewertung können je nach Situation und Anforderung variieren und beeinflussen maßgeblich die Auswahl der Bewertungsverfahren. Zu den wesentlichen Bewertungszwecken zählen Entscheidungsfindungen bei Kauf- und Verkaufsverhandlungen, steuerliche Zwecke, Bilanzierungszwecke, Schiedsverfahren oder gerichtliche Auseinandersetzungen.⁹

Ein fundamentales Prinzip bei der Unternehmensbewertung stellt das Going-Concern-Prinzip dar. Dieses Prinzip besagt, dass ein Unternehmen grundsätzlich in seiner bisherigen Form fortgeführt wird und nicht unter der Prämisse einer Liquidation bewertet wird. Das bedeutet, dass zukünftige Erträge oder Cashflows unter der Annahme prognostiziert werden, dass das Unternehmen seine Geschäftstätigkeit nachhaltig und auf unbestimmte Zeit weiterführen kann. Dieses Prinzip bildet die Basis für viele Bewertungsverfahren und beeinflusst sowohl die Wahl der Bewertungsmethode als auch die zugrunde liegenden Annahmen.¹⁰ Eine Ausnahme bilden Bewertungen, die speziell im Zuge einer Liquidation vorgenommen werden, wie das in Kapitel 3.3.2 Liquidationswertverfahren.

3.2 Regulatorische Anforderungen

Um eine konsistente, transparente und vergleichbare Bewertung sicherzustellen, unterliegt die Unternehmensbewertung sowohl nationalen als auch internationalen regulatorischen Anforderungen. Nachfolgend wird ein Überblick über die maßgeblichen gesetzlichen Grundlagen und Bewertungsstandards in Deutschland und auf internationaler Ebene dargestellt.

Das HGB bildet die Grundlage für die handelsrechtliche Rechnungslegung in Deutschland. Es enthält allgemeine Bewertungsgrundsätze, die auch für die Unternehmensbewertung relevant sind. Besonders hervorzuheben sind die Grundsätze

⁸ Moxter & Engel-Ciric [2019], S. 6

⁹ Vgl. Institut der Wirtschaftsprüfer in Deutschland [2017], Abschnitt 2.2 – 2.4

¹⁰ Vgl. Schmeisser et al. [2009], S. 34 f.

der Vorsicht,¹¹ der Einzelbewertung¹² und der Stetigkeit.¹³ Der Grundsatz der Vorsicht schreibt vor, Verluste bereits bei ihrer Entstehung und Gewinne erst bei Realisation zu berücksichtigen, um eine Überbewertung des Unternehmens zu vermeiden. Der Grundsatz der Stetigkeit verlangt, Bewertungsmethoden und Ansatzregeln periodengleich beizubehalten, um die Vergleichbarkeit der Jahresabschlüsse zu sichern. Der Grundsatz der Einzelbewertung schreibt vor, jeden Vermögensgegenstand und jede Schuld separat zu bewerten, statt pauschale Sammelbewertungen vorzunehmen. Diese Prinzipien beeinflussen somit die Bewertung von Vermögensgegenständen und Schulden und somit den ausgewiesenen Unternehmenswert.

Das Bewertungsgesetz regelt die steuerliche Bewertung von Vermögensgegenständen, insbesondere im Rahmen der Erbschaft- und Schenkungsteuer. Nach dem Bewertungsgesetz ist der gemeine Wert maßgeblich, der durch den Preis bestimmt wird, der im gewöhnlichen Geschäftsverkehr zu erzielen wäre.¹⁴ Für nicht börsennotierte Unternehmen sind dabei insbesondere das Ertragswertverfahren und das vereinfachte Ertragswertverfahren relevant.

Bei Umstrukturierungen wie Fusionen oder Spaltungen schreibt das Umwandlungsgesetz vor, dass ein angemessener Unternehmenswert ermittelt wird und verlangt die Erstellung eines Umwandlungsberichts, der auch eine Unternehmensbewertung enthalten muss.¹⁵

Das AktG stellt besondere Anforderungen an die Bewertung im Rahmen von Strukturmaßnahmen. So ist bei Squeeze-outs¹⁶ oder Beherrschungs- und Gewinnabführungsverträgen¹⁷ eine angemessene Abfindung zu gewähren, deren Höhe durch eine Unternehmensbewertung zu ermitteln ist. Die Bewertung muss der Prüfung durch einen sachverständigen Prüfer genügen.¹⁸

Neben den gesetzlichen Anforderungen ist für die praktische Umsetzung von Unternehmensbewertungen in Deutschland der Standard IDW S 1 des Instituts der Wirtschaftsprüfer (IDW) maßgeblich. Er legt die Grundsätze für die Durchführung von Unternehmensbewertungen fest und ist insbesondere bei Bewertungen im Rahmen von Transaktionen, steuerlichen Bewertungen und gerichtlichen Verfahren relevant. Der IDW

¹¹ Vgl. § 252 Abs. 1 Nr. 4 HGB [o.J.]

¹² Vgl. § 252 Abs. 1 Nr. 3 HGB [o.J.]

¹³ Vgl. § 252 Abs. 1 Nr. 6 HGB [o.J.]

¹⁴ Vgl. § 11 Abs. 2 BewG [o.J.]

¹⁵ Vgl. § 9 ff. UmwG [o.J.]

¹⁶ Vgl. § 327b AktG [o.J.]

¹⁷ Vgl. § 305 AktG [o.J.]

¹⁸ Vgl. § 293b AktG [o.J.]

S 1 fordert unter anderem die Berücksichtigung von Planungsrechnungen, Kapitalisierungszinssätzen und spezifischen Bewertungsanlässen.¹⁹

Im internationalen Kontext, insbesondere bei kapitalmarktorientierten Unternehmen, sind die International Financial Reporting Standards (IFRS) relevant. Hierbei kommt häufig das Discounted-Cash-Flow-Verfahren (DCF) zur Anwendung, das den Unternehmenswert auf Basis zukünftiger Zahlungsströme ermittelt, wie das Kapitel 3.5.4 zeigen wird.

Die IFRS, herausgegeben vom International Accounting Standards Board (IASB), sind ein weltweit anerkanntes Regelwerk für die Rechnungslegung. Sie legen besonderen Wert auf die Bewertung von Vermögenswerten und Schulden zum beizulegenden Zeitwert, den sogenannten Fair Value.

Der Standard IFRS 13 „Fair Value Measurement“ definiert den beizulegenden Zeitwert als den Preis, der in einer Transaktion zwischen Marktteilnehmern am Bewertungsstichtag für den Verkauf eines Vermögenswerts erzielt oder für die Übertragung einer Verbindlichkeit gezahlt werden würde. Der Standard stellt einheitliche Bewertungsgrundsätze bereit und verlangt umfangreiche Angaben zu den Bewertungsmethoden und -annahmen.²⁰

Die International Valuation Standards (IVS), herausgegeben vom International Valuation Standards Council (IVSC), bieten ein Rahmenwerk für die Durchführung von Bewertungen weltweit. Sie ergänzen den IFRS, indem sie detaillierte Anleitungen zur Anwendung verschiedener Bewertungsmethoden, wie dem Marktwert-, Ertragswert- und Kostenansatz, bereitstellen. Die IVS betonen die Bedeutung von Transparenz, Konsistenz und professionellem Urteilsvermögen bei der Bewertung.²¹ Die nachfolgende Tabelle zeigt die Unterschiede zwischen Deutschland und der Internationalen Bewertungsregulatorik im Vergleich.

¹⁹ Vgl. Institut der Wirtschaftsprüfer in Deutschland [2017], Abschnitt 2 bis 5

²⁰ Vgl. IFRS STANDARDS [2022], Abschnitt IFRS 13

²¹ Vgl. INTERNATIONAL VALUATION STANDARDS [2024], o.S.

Kriterium	Deutschland (IDW S 1)	International (IFRS 13 & IVS)
Bewertungsansatz	Primär Ertragswertverfahren; objektiver Unternehmenswert	Primär DCF; Fair Value als zentraler Bewertungsmaßstab
Regelungsdichte	Hohe Detailtiefe und spezifische Vorgaben für verschiedene Bewertungsanlässe	Prinzipienbasiert mit Fokus auf Transparenz und Marktbezug
Transparenz Anforderungen	Detaillierte Dokumentation und Offenlegung der Bewertungsannahmen	Umfangreiche Angaben zu Bewertungsmethoden und -annahmen gemäß IFRS 13
Anwendungsbereich	National, insb. für steuerliche Bewertungen, Transaktionen und gerichtliche Verfahren	Global, insbesondere für kapitalmarktorientierte Unternehmen und internationale Transaktionen
Verbindlichkeit	Hohe Verbindlichkeit durch gesetzliche Verankerung und Anwendung in der Praxis	Verbindlich für IFRS-anwendende Unternehmen; IVS als ergänzender Standard

Tabelle 1: Vergleich der Methoden zur Unternehmensbewertung in Deutschland und International (Quelle: Eigene Darstellung)

Generell werden Bewertungsverfahren in Einzel-, Misch- und Gesamtbewertungsverfahren unterteilt, die jeweils unterschiedliche methodische Schwerpunkte verfolgen. Nachfolgend werden zunächst Einzel- und Mischbewertungsverfahren erläutert und eingeordnet und abschließend detailliert auf die in der Praxis sehr gängigen Gesamtbewertungsverfahren eingegangen.

3.3 Einzelbewertungsverfahren

Einzelbewertungsverfahren ermitteln den sogenannten objektiven Unternehmenswert anhand der Summe der Einzelwerte seiner Vermögensgegenstände abzüglich Schulden. Hierbei wird die Substanz des Unternehmens, isoliert je Vermögenswert betrachtet, ohne direkt die zukünftige Ertragskraft des Gesamtunternehmens einzubeziehen.²²

In der Praxis wird hierbei zwischen dem Substanzwertverfahren und dem Liquidationswertverfahren unterschieden. Beide ermitteln den Unternehmenswert anhand eines Stichtags, indem sie alle bilanzierten Vermögenswerte sowie nicht-bilanzierte Werte (z. B. Inventar) berücksichtigen. Der Wert ergibt sich aus der Summe aller Vermögenswerte (betriebsnotwendig/nicht-notwendig) abzüglich der Verbindlichkeiten. Das Substanzwertverfahren (Reproduktionswert) unterstellt die Fortführung des Unternehmens. Das Liquidationswertverfahren unterstellt die Zerschlagung des Unternehmens. Entsprechend unter Annahme der Fortführung der

²² Vgl. Ballwieser & Hachmeister [2019], S. 11 f.

Reproduktionswert und dem Liquidationswert unterschieden.²³ Die Verfahren unterscheiden sich hierbei in der Ermittlung der einzelnen Werte, nicht aber in der Zusammensetzung.

3.3.1 Substanzwertverfahren

Das Substanzwertverfahren betrachtet den Reproduktionswert. Dieser spiegelt den Wert einer Neuerrichtung des Unternehmens im aktuellen Zustand zum Bewertungsstichtag wider. Das Substanzwertverfahren wird seinerseits noch einmal in den sogenannten Teilreproduktionswert und dem Vollreproduktionswert unterschieden.

Für den Teilreproduktionswert (TRW) werden aus der Bilanz des Unternehmens alle betriebsnotwendigen Vermögenswerte zu Marktpreisen bewertet und das zu Marktpreisen bewerte Fremdkapital abgezogen. Nicht betriebsnotwendige Vermögensgegenstände werden hinzuaddiert.²⁴

Der Vollreproduktionswert (VRW) leitet sich aus dem vollständigen Nachbau eines Unternehmens mit demselben Ertragspotenzial ab. Dies bedeutet, dass neben den oben genannten bilanzierungsfähigen Aspekten auch selbstgeschaffene Geschäfts- oder Firmenwerte erfasst werden. Dies beinhaltet bspw. selbst geschaffene Patente, insofern diese nicht als immaterielle Vermögenswerte aktiviert wurden, technisches Know-how, Markennamen und Kundenstamm.²⁵

In der Praxis dient der ermittelte Substanzwert zudem oft als Kontrollgröße, neben dem Ertragswertverfahren, das in Kapitel 3.5.3 näher erläutert wird. Der größte Vorteil des Substanzwertverfahrens ist, dass es nicht auf Prognosen zukünftiger Gewinne angewiesen ist, sondern sich auf die tatsächlich vorhandenen Vermögenswerte eines Unternehmens stützt.²⁶ Allerdings gibt es auch wichtige Nachteile: Beim Substanzwertverfahren werden nur die einzelnen Vermögenswerte zusammengerechnet, ohne zu berücksichtigen, wie sie im Unternehmen zusammenwirken. Synergieeffekte, die durch das Zusammenspiel der Unternehmensbereiche entstehen, bleiben also unberücksichtigt. Der Substanzwert verstößt daher gegen den Grundsatz der Bewertungseinheit, da nur einzelne Teile addiert werden.²⁷ Zudem ignoriert er immaterielle Werte wie Kundenstamm, Humankapital oder Marktposition, die nicht einzeln veräußerbar sind.²⁸ Deshalb kann das

²³ Vgl. Wöltje [2021], S. 49 f.

²⁴ Vgl. Wöltje [2021], S. 50 f.

²⁵ Vgl. Eayrs [2011], S. 286 f.

²⁶ Vgl. Ballwieser & Hachmeister [2019], S. 245

²⁷ Vgl. Wöltje [2021], S. 63

²⁸ Vgl. Schmeisser et al. [2009], S. 102

Substanzwertverfahren zu unrealistischen Ergebnissen führen – zum Beispiel bei Unternehmen, die zwar viele Vermögenswerte besitzen, aber kaum Gewinne erwirtschaften.²⁹

3.3.2 Liquidationswertverfahren

Das Liquidationswertverfahren bestimmt im Gegensatz zum Substanzwertverfahren den Unternehmenswert unter Annahme der sofortigen Liquidation des Unternehmens. Dafür werden die Vermögensgegenstände nicht im Verbund, sondern getrennt verkauft. Der Liquidationswert ergibt sich als Barwert der erwarteten Veräußerungserlöse aller Einzelwertgegenstände abzüglich Verbindlichkeiten, Liquidationskosten und Steuern. Man unterstellt einen begrenzten Zeithorizont für die Abwicklung anstelle der zeitlich unbegrenzten Fortführung.³⁰

Charakteristisch für das Verfahren ist, dass der erzielbare Wert stark von der Liquidationsgeschwindigkeit und der Zerschlagungsintensität, d.h. der Grad der Einzelveräußerungen, abhängt. Bei hoher Zerschlagungsgeschwindigkeit fallen in der Regel höhere Abschläge auf Veräußerungserlöse an, was den Liquidationswert weiter senkt. Das Verfahren ist insgesamt einfach anwendbar und liefert eine klare Untergrenze, eignet sich jedoch nur für Fälle tatsächlicher Insolvenzen, Sanierungen oder geplanter Auflösungen – da weder künftige Ertragskraft noch Synergie- oder immaterielle Werte berücksichtigt werden.³¹

Das Institut der Wirtschaftsprüfer konkretisiert im Standard für Grundsätze zur Durchführung von Unternehmensbewertungen, dass bei schlechter Ertragslage der Liquidationswert den Unternehmenswert begrenzt – “übersteigt der Barwert der finanziellen Überschüsse bei Liquidation den Fortführungswert, bildet der Liquidationswert die Untergrenze”³².

3.4 Mischbewertungsverfahren

Mischbewertungsverfahren kombinieren zusätzlich zum Substanzwert auch die zukünftige Ertragskraft in Form von Gewinnen.³³ Insgesamt gibt es drei verschiedene

²⁹ Vgl. Wöltje [2021], S. 63 f.

³⁰ Vgl. Wöltje [2021], S. 56 ff.

³¹ Vgl. Wöltje [2021], S. 58 f.

³² Institut der Wirtschaftsprüfer in Deutschland [2017], Tz 139

³³ Vgl. Ballwieser & Hachmeister [2019], S. 247

Mischbewertungsverfahren, das Mittelwert- und das Übergewinnverfahren sowie das Stuttgarter Verfahren.

Das Mittelwertverfahren ist, wie der Name bereits andeutet, eine Verknüpfung aus Ertrags- und Substanzwertverfahren. Entsprechend berechnet sich der Unternehmenswert aus dem Mittelwert des Ertrags- und des Substanzwertes, letzterer auf Basis des Teilreproduktionswertes gerechnet.³⁴

Das Übergewinnverfahren ergänzt den Substanzwert um den Barwert künftiger Übergewinne, die als Differenz zwischen dem nachhaltig erzielbaren Unternehmensergebnis und einer geforderten Verzinsung des eingesetzten Kapitals definiert sind. Über eine festgelegte Nachhaltigkeitsdauer werden die einzelnen Übergewinne diskontiert und zum Substanzwert addiert.³⁵

Als zusätzliches Mittelwertverfahren kommt das Stuttgarter Verfahren zum Einsatz. Ursprünglich für steuerliche Bewertungszwecke konzipiert, ermittelt das Stuttgarter Verfahren den Wert nicht börsennotierter Gesellschaften auf Basis des bilanziellen Vermögens im Verhältnis zum gezeichneten Kapital. Es greift auf Schätzvorschriften des Bewertungsgesetzes zurück und ist vor allem bei Erbschaft- und Schenkungssteuerfällen relevant.³⁶

Gesamthaft betrachtet findet sich eine kritische Würdigung durch den Autoren Seppelfricke, der hervorhebt, dass Mischverfahren weder die systematische Fundierung der Zukunftserfolgsverfahren noch die Transparenz der Einzelbewertungsverfahren bieten, weshalb sie auch in der Praxis nicht in der Breite vertreten sind.³⁷

3.5 Gesamtbewertungsverfahren

Gesamtbewertungsverfahren zielen darauf ab, Unternehmen als Ganzes zu bewerten, weshalb sie in der Praxis als auch in dieser Arbeit eine hohe Beachtung finden. Typischerweise finden sie Anwendung durch eine Barwertbetrachtung der gesamten zukünftigen Zahlungsüberschüsse. Die bekanntesten Gesamtbewertungsverfahren sind die Kapitalwertmethode, das Ertragswertverfahren und das Discounted-Cashflow-Verfahren (DCF). Nach dem IDW Standard S 1 stellen sowohl das Ertragswertverfahren als auch das DCF-Verfahren zentrale Instrumente zur Ermittlung objektiverter

³⁴ Vgl. Wöltje [2021], S.99

³⁵ Vgl. Schmeisser et al. [2009], S. 110 f.

³⁶ Vgl. Schmeisser et al. [2009], S. 110 f.

³⁷ Vgl. Seppelfricke [2020], S. 125 f.

Unternehmenswerte dar. Der IDW betont, dass der Unternehmenswert maßgeblich von den zukünftigen finanziellen Überschüssen und der adäquaten Berücksichtigung von Risiken abhängt.³⁸

Bevor nachfolgend auf das Ertragswertverfahren sowie die verschiedenen Discounted-Cashflow-Verfahren näher eingegangen wird, wird zunächst die Kapitalwertmethode sowie das Capital Asset Pricing Model (CAPM) näher erläutert, da diese eine fundamentale Basis für die Berechnungen im Ertragswertverfahren und dem Discounted-Cashflow-Verfahren darstellen.

3.5.1 Kapitalwertmethode

Die Kapitalwertmethode, im Englischen als Net Present Value (NPV) bezeichnet, ist ein dynamisches Verfahren der Investitionsrechnung. Charakteristisch für die Kapitalwertmethode ist, dass sämtliche zukünftig erwarteten Ein- und Auszahlungen mit einem bestimmten Kalkulationszinsfuß, dem sogenannten Diskontierungszinssatz auf den Zeitpunkt unmittelbar zu Beginn des Planungszeitraums abgezinst werden.³⁹ Dabei wird der sogenannte Zeitwert des Geldes berücksichtigt: Ein Euro heute ist mehr wert als ein Euro in der Zukunft, da dieser investiert werden könnte und zudem ein Risiko besteht, ob der zukünftige Betrag tatsächlich zufließt.⁴⁰

Der Kapitalwert stellt somit den Betrag dar, den eine Investition über eine vorgegebene Mindestverzinsung hinaus erwirtschaftet und gilt als Entscheidungsgrundlage für die Vorteilhaftigkeit von Investitionen.⁴¹ Mathematisch ergibt sich folgende Darstellung:

$C_0 = -A_0 + \sum_{t=1}^n \frac{Z_t}{(1+i)^t} + \frac{L_n}{(1+i)^n}$	
C_0	Kapitalwert
A_0	Anschaffungsauszahlung
Z_t	Periodenüberschuss
i	Risikoadjustierter Kapitalisierungszinssatz
n	Anzahl der Perioden
t	Periodenindex
L_n	Liquidationserlös oder Abbruchkosten am Ende der Laufzeit

Formel 1: Kapitalwert einer Investition

Die Kapitalwertmethode bildet die theoretische Grundlage für moderne Unternehmensbewertungsverfahren wie das Discounted-Cashflow-Verfahren und das

³⁸ Vgl. Institut der Wirtschaftsprüfer in Deutschland [2017], Tz 4.1-4.3

³⁹ Vgl. Wöltje [2021], S. 70

⁴⁰ Seppelfricke [2020], S. 35

⁴¹ Vgl. Geyer [2013], S. 358 f.

Ertragswertverfahren. Beide Verfahren operationalisieren die Kapitalwertmethode, indem sie die Zahlungsströme des Unternehmens (Cashflows oder Erträge) mit einem risikoadjustierten Zinssatz auf den Bewertungsstichtag abzinsen. Die Unternehmensbewertung nach IDW S 1 folgt ausdrücklich dieser Methodik.⁴²

Berücksichtigt werden sollte, vor allem für die nachfolgenden Bewertungsmodelle, dass die Aussagekraft des Kapitalwerts stark von der Qualität der zugrunde liegenden Plandaten sowie der Wahl des Kalkulationszinssatzes abhängt. Diese müssen unter wirtschaftlicher, markt- und unternehmensspezifischer Perspektive plausibilisiert werden.⁴³

3.5.2 Capital Asset Pricing Model (CAPM)

Das Capital Asset Pricing Model (CAPM) ist ein ebenso grundlegendes finanzwirtschaftliches Modell zur Herleitung der erwarteten Rendite eines Wertpapiers bzw. Bewertungsobjektes unter Berücksichtigung seines systematischen Risikos. Es bildet eine wichtige Grundlage für die Bestimmung der Eigenkapitalkosten im Rahmen von Unternehmensbewertungen und findet insbesondere im Ertragswertverfahren, sowie im Discounted-Cashflow-Verfahren Anwendung.

Konkreter ist der sogenannte risikoadjustierte Kapitalisierungszinssatz aus dem Capital Asset Pricing Model relevant und soll an dieser Stelle kurz näher erläutert werden. Er setzt sich zusammen aus dem risikofreien Basiszins r_f und einem Risikozuschlag, der durch das Produkt des Betafaktors β mit der Marktisikoprämie r_{zu} (häufig als Differenz zwischen markt- und risikofreiem Zins definiert) gebildet wird:

$i = r_f + (\beta * r_{zu})$	
i	Risikoadjustierter Kapitalisierungszinssatz
r_f	Basiszins einer risikofreien Kapitalanlage
β	Betafaktor
r_{zu}	Risikozuschlag

Formel 2: Risikoadjustierter Kapitalisierungszinssatz nach CAPM

Dabei misst β das systematische Risiko des Unternehmens relativ zum Gesamtmarkt ($\beta > 1$ = überdurchschnittliches Risiko, $\beta < 1$ = unterdurchschnittliches Risiko), während r_{zu} die zusätzliche Renditeanforderung für das Eingehen dieses Risikos darstellt.⁴⁴

⁴² Vgl. Institut der Wirtschaftsprüfer in Deutschland [2017], Abschnitt 4.1 & 7.3

⁴³ Vgl. Wöltje [2021], S. 76

⁴⁴ Vgl. Drukarczyk & Schüler [2021], S. 55

An dieser Stelle sei noch auf ein wesentliches Problem in der praktischen Anwendung des CAPM hingewiesen. Dieses Problem liegt in der Bestimmung eines geeigneten Beta-Faktors. Da viele Unternehmen – insbesondere im Deutschen Mittelstand – nicht börsennotiert sind, erfolgt die Herleitung des Betas in der Regel über vergleichbare, börsennotierte Peer-Group-Unternehmen, also Unternehmen, die in einer ähnlichen Vergleichsgruppe liegen. Diese Betawerte werden zunächst von ihrer Kapitalstruktur bereinigt, um anschließend auf das bewertete Unternehmen unter Berücksichtigung dessen Verschuldungsgrads übertragen zu werden. Dieser Prozess ist empirisch anspruchsvoll und mit zahlreichen Annahmen behaftet, die die Ergebnisqualität maßgeblich beeinflussen.⁴⁵

3.5.3 Ertragswertverfahren

Das Ertragswertverfahren ist ein traditionelles und insbesondere im deutschen Wirtschaftsraum etabliertes Verfahren zur Unternehmensbewertung. Es basiert auf dem Prinzip der Barwertberechnung zukünftiger finanzieller Überschüsse und ist damit konzeptionell vergleichbar zum Discounted-Cashflow-Modell, das im Folgekapitel näher erörtert wird. Im Gegensatz zu diesem konzentriert sich das Ertragswertverfahren auf dem den Anteilseignern zufließenden Ertrag, der unter Anwendung eines geeigneten Kapitalisierungszinssatzes auf den Bewertungsstichtag abgezinst wird.⁴⁶

Grundlage der Bewertung bildet der betriebswirtschaftlich begründete Ansatz, dass sich der Unternehmenswert aus dem Nutzen ergibt, den das Unternehmen seinen Eigentümern in Form zukünftiger finanzieller Überschüsse bieten kann. Diese Perspektive ist in der Rechtsprechung sowie in der Bewertungspraxis der Unternehmensnachfolge, der steuerlichen Bewertung oder in Schiedsverfahren weit verbreitet und auch vor allem, wie bereits eingangs erwähnt, im IDW S 1 Standard des Instituts der Wirtschaftsprüfer als Bewertungsansatz verankert.⁴⁷

Die Unternehmensbewertung erfolgt hierbei in zwei Phasen: In der sogenannten Detailplanungsphase (typischerweise 3–5 Jahre) werden die zu erwartenden finanziellen Ertragsüberschüsse explizit geplant. Anschließend wird ein konstantes Ertragsniveau unterstellt, das entweder dem letzten geplanten Wert oder einer Durchschnittsgröße entspricht und mit einer ewigen Rente kapitalisiert wird. Als alternativer Ansatz kann anstelle der unendlichen Lebensdauer auch der Liquidationserlös des Unternehmens

⁴⁵ Vgl. Kruschwitz [2019], S.164

⁴⁶ Vgl. Drukarczyk & Schüler [2021], S. 71 ff.

⁴⁷ Vgl. Institut der Wirtschaftsprüfer in Deutschland [2017], Tz 6 ff

berücksichtigt werden.⁴⁸ In der Praxis ist dieses Vorgehen allerdings unüblich, da , entsprechend des ‚Going Concern Prinzip‘, von einer unbegrenzten Lebensdauer des Unternehmens ausgegangen wird.⁴⁹

Für die benannte Detailplanungsphase der Ertragsüberschüsse wird eine Analyse der historischen Ergebnisse der letzten drei Jahre vorgenommen und um einmalige Sondereffekte korrigiert.⁵⁰ Zusätzlich werden Umsatzprognosen gebildet unter Annahmen des Bewerter über künftige Marktentwicklungen, sowie der geplanten Kapazität und Synergieeffekten, etwa aus Zusammenschlüssen. Auf Basis dieser Umsatzplanung erfolgt die Kostenplanung analog, um die Umsatzprognosen personell und materiell, sowie die erforderliche Produktionskapazität mittels einer Investitionsplanung zu gewährleisten. Dies wird ergänzt um Aufwendungen für Forschung & Entwicklung, Vertrieb & Verwaltung, sowie Zinserträge und Zinsaufwendungen, um mögliche Veränderungen in der Kapitalstruktur entsprechend zu reflektieren.⁵¹

Der risikoadjustierte Kapitalisierungszinssatz stellt für die Berechnung des Unternehmenswertes mittels der Ertragswertmethode eine sehr relevante Bedeutung dar. Wie im Kapitel 3.5.2 bereits näher erläutert, ergibt sich dieser gemäß IDW S 1 aus der risikofreien Rendite und einer Risikoprämie, die durch das individuelle Risiko des Unternehmens bestimmt wird.⁵² Das individuelle Risiko wird durch einen sogenannten Betafaktor, der das individuelle Risiko des zu bewertenden Unternehmens im Vergleich zum Markt, in welchem er agiert, mit einem Risikozuschlag multipliziert.

Es ergibt sich folgende Berechnung für den Unternehmenswert im Ertragswertverfahren:

$$UW_{EV} = \sum_{t=1}^n \frac{(E_t)}{(1+i)^t} + \frac{(E_{TV})}{(1+i)^n}$$

UW_{EV}	Unternehmenswert Ertragswertverfahren
E_t	geplanter Jahresüberschuss nach Steuern (bereinigt um außerordentliche Effekte)
E_{TV}	Ertrag der ewigen Rente ab der letzten Planungsperiode
i	risikoadjustierter Kapitalisierungszinssatz
t	Periode
n	Endperiode

Formel 3: Unternehmenswert Ertragswertverfahren

⁴⁸ Vgl. Wöltje [2021], S.78

⁴⁹ Vgl. Ballwieser & Hachmeister [2019], S. 72

⁵⁰ Vgl. Wöltje [2021], S.81

⁵¹ Vgl. Becker & Peppmeier [2022], S. 89 f.

⁵² Vgl. Institut der Wirtschaftsprüfer in Deutschland [2017], Tz 143 ff.

Ein wesentliches Merkmal des Ertragswertverfahrens ist die Substanzunabhängigkeit: Es unterstellt, dass die Fähigkeit zur Ertragserzielung den eigentlichen Unternehmenswert darstellt – nicht die Summe einzelner Vermögensgegenstände. Lediglich in Ausnahmefällen (z. B. bei ‚Asset Deals‘ oder Liquidationen) findet eine rein substanzwertorientierte Bewertung Anwendung.⁵³

Kritisch diskutiert wird das Ertragswertverfahren in der Literatur insbesondere aufgrund seiner Sensitivität gegenüber dem Kapitalisierungszinssatz und der Annahme eines konstanten, dauerhaft erzielbaren Ertragsniveaus. Gerade bei jungen, wachstumsorientierten oder zyklischen Unternehmen kann diese Annahme unrealistisch sein. Zudem ist die Ableitung der Ertragsprognose oft stark vergangenheitsorientiert, was bei dynamischen Geschäftsmodellen oder disruptiven Märkten problematisch ist.⁵⁴

Des Weiteren beruht die Ertragsprognose im klassischen Ertragswertansatz meist stark auf historischen Daten und subjektiven Planrechnungen. Die hierfür verwendeten Gewinnschätzungen der letzten drei Perioden sind häufig „zu kurz, um Erfolgsschwankungen zu berücksichtigen“⁵⁵, und basieren zudem auf Schätzungen und den oftmals am stärksten bilanzpolitisch beeinflussbaren Größen (Gewinn), was die Aussagekraft der Prognose weiter einschränkt.

Nichtsdestotrotz ist das Ertragswertverfahren insbesondere in steuerlichen Kontexten (z. B. im Rahmen der Erbschaftsteuer) sowie bei mittelständischen Bewertungen ohne Kapitalmarktrelevanz weit verbreitet und gesetzlich anerkannt. Auch in Schiedsverfahren oder bei Familienunternehmen wird es aufgrund seiner auf Kontinuität ausgerichteten Logik und seiner Bilanznähe regelmäßig herangezogen.⁵⁶

Insgesamt stellt das Ertragswertverfahren ein bewährtes, in vielen Bewertungskontexten akzeptiertes Verfahren dar, das insbesondere durch seine enge Verknüpfung mit handelsrechtlichen Gewinngrößen sowie seine einfache Kommunikationslogik überzeugt.

3.5.4 Discounted-Cashflow-Verfahren

Das Discounted-Cashflow-Verfahren (DCF) ist eines der zentralen Instrumente zur Bewertung von Unternehmen im Rahmen der internationalen als auch

⁵³ Vgl. Perridon et al. [2022], S. 308

⁵⁴ Vgl. Seppelfricke [2020], S. 52

⁵⁵ Schmeisser et al. [2009], S. 108

⁵⁶ Seppelfricke [2020], S. 27 ff.

investitionstheoretisch fundierten Unternehmenswertbestimmung und verdrängt in Deutschland zunehmend das Ertragswertverfahren.⁵⁷ Konkreter erkennt das Institut der Wirtschaftsprüfer die DCF-Verfahren als gleichwertig zum Ertragswertverfahren an.⁵⁸

Es basiert auf dem fundamentalen Prinzip, dass der Unternehmenswert dem heutigen Wert (Barwert) aller zukünftigen, dem Unternehmen zufließenden Zahlungsüberschüsse entspricht. Entsprechend bildet sich die in Kapitel 3.5.1 erläuterte Kapitalwertmethode als Basis für die DCF-Methode. Bewertungsrelevante Zahlungsgrößen im DCF-Modell sind die sogenannten ‚Free Cashflows‘ (*FCF*), die dem Unternehmen nach Finanzierungen wertsteigernder Investitionen zur freien Verfügung stehen. Sie werden entweder auf Basis der Kapitalflussrechnung direkt oder indirekt aus der Plan Gewinn- und Verlustrechnung abgeleitet.⁵⁹

Der Diskontierungszinssatz kann entweder der bereits dargestellte risikoadjustierte Zinssatz, auch Eigenkapitalkostensatz genannt, nach dem Capital Asset Price Model sein, oder ein gewogener durchschnittlicher Kapitalkostensatz (WACC – Weighted Average Cost of Capital), der sowohl Eigen- als auch Fremdkapitalkosten berücksichtigt.⁶⁰

Die DCF-Verfahren werden systematisch in Bruttokapitalisierungsverfahren (Entity-Ansätze) und Nettokapitalisierungsverfahren (Equity-Ansätze) unterschieden. Beim Entity-Verfahren wird der Unternehmenswert zweistufig ermittelt: Zunächst wird der Marktwert des Gesamtkapitals durch Diskontierung der Free Cashflows mit dem WACC (Weighted Average Cost of Capital) berechnet. Anschließend wird das verzinsliche Fremdkapital subtrahiert, um den Marktwert des Eigenkapitals zu bestimmen.⁶¹ Der Restwert (Ewige Rente bzw. Terminal Value) wird dabei durch den Free Cashflow des letzten Prognosejahres um die langfristige Wachstumsrate fortgeschrieben und durch die Differenz aus WACC und der ewigen Wachstumsrate geteilt.

⁵⁷ Vgl. Heesen [2021], S. 3

⁵⁸ Vgl. Institut der Wirtschaftsprüfer in Deutschland [2017], Abschnitt 6.5

⁵⁹ Vgl. Schmeisser et al. [2009], S. 138

⁶⁰ Vgl. Wöltje [2021], S. 107 f.

⁶¹ Vgl. Wöltje [2021], S. 107

Mathematisch ausgedrückt sieht der Entity Ansatz des DCF-Verfahrens wie folgt aus:

$$UW_{DCF} = \sum_{t=1}^n \frac{(FCF_t)}{(1 + WACC)^t} + \frac{(TV_n)}{(1 + WACC)^n} - FK$$

UW_{DCF}	Unternehmenswert Discounted Cashflow Verfahren
FCF_t	Free Cashflow in Periode t
TV_n	Terminal Value (Ertrag der ewigen Rente ab der letzten Planungsperiode)
$WACC$	Weighted Average Cost of Capital
FK	Fremdkapital
t	Periode
n	Endperiode

Formel 4: Unternehmenswert Discounted Cashflow Verfahren (Entity Ansatz)

Wobei der Terminal Value dargestellt wird durch folgende Formel:

$$TV_{Entity} = \frac{FCFF_T(1 + g)}{WACC - g}$$

TV_{Entity}	Unternehmenswert Discounted Cashflow Verfahren
$FCFF_T$	Free Cashflow to Firm in letzter Periode T
$WACC$	Weighted Average Cost of Capital
g	Ewige Wachstumsrate

Formel 5: Terminal Value (Entity Ansatz)

Im Gegensatz zum risikoadjustierten Zinssatz, der nachfolgend im Equity Ansatz genutzt wird, bildet der WACC und somit der relevante Diskontierungszinssatz die Basis im Entity Ansatz des Discount Cashflow Modells. Bei der Berechnung des WACC wird zudem der Steuervorteil, der sogenannte Tax Shield, welcher sich zwangsläufig aus einer möglichen Fremdkapitalfinanzierung ergibt, ebenfalls berücksichtigt. Nachfolgend in Formel 6 ist der WACC mit Tax Shield dargestellt:

$$WACC_{mit\ Tax\ Shield} = \frac{EK_{Markt}}{GK_{Markt}} * i_{EK,v} + \frac{FK_{Markt}}{GK_{Markt}} * i_{FK} * (1 - s)$$

EK_{Markt}	Marktwert des Eigenkapitals
FK_{Markt}	Marktwert des Fremdkapitals
GK_{Markt}	Marktwert des Gesamtkapitals
$i_{EK,v}$	Eigenkapitalkostensatz des verschuldeten Unternehmens
i_{FK}	Fremdkapitalkostensatz
$(1 - s)$	Steuervorteil, mit s = durchschnittlicher Ertragssteuersatz des Unternehmens

Formel 6: WACC mit Tax Shield

Da der WACC von der Kapitalstruktur (Verhältnis von Eigen- und Fremdkapital) abhängt und der Unternehmenswert wiederum für die Ableitung der Kapitalstruktur benötigt wird, besteht eine zirkuläre Abhängigkeit. Dieser Rückkopplungseffekt muss durch iterative

Verfahren aufgelöst werden, was die praktische Anwendung erschwert, wie auch die nachfolgende Abbildung zeigt.⁶²

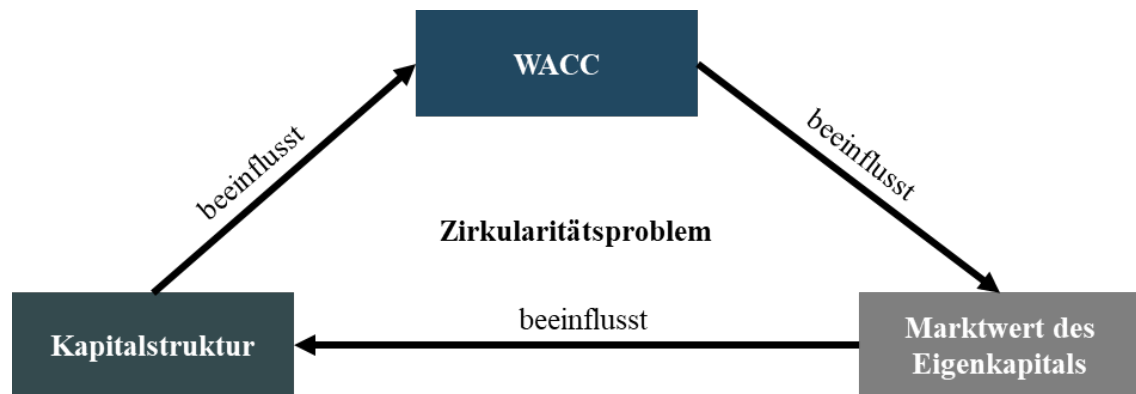


Abbildung 1: WACC Zirkularitätsproblem
(Quelle: Eigene Darstellung in Anlehnung an Wöltje [2021], S. 123)

Die Entity-Methode setzt implizit eine konstante Kapitalstruktur über den gesamten Prognosezeitraum voraus. Diese Annahme ist besonders bei wachsenden, kapitalintensiven oder restrukturierenden Unternehmen realitätsfern. Eine sich verändernde Verschuldung wirkt sich aber direkt auf den WACC aus – und damit auf den Unternehmenswert.⁶³

Das Problem der zirkulären Abhängigkeit zwischen WACC und dem Marktwert des Eigenkapitals kann in der Praxis durch ein iteratives Verfahren behoben werden. Dabei erfolgt die erste Bewertung des Unternehmens auf Basis einer angenommenen Kapitalstruktur bzw. eines geschätzten Eigenkapitalmarktwerts. Anschließend wird in mehreren Rechenschritten der berechnete Unternehmenswert an die jeweils resultierende Kapitalstruktur angepasst, bis sich ein Gleichgewicht zwischen Kapitalstruktur und WACC einstellt.⁶⁴

Beim Equity Ansatz erfolgt eine einstufige Bewertung durch Abzinsung der eigenkapitalbezogenen Zahlungsüberschüsse FCF_e mit der risikoäquivalenten Renditeforderung der Eigenkapitalgeber (CAPM). Das Ergebnis der Equity Methode entspricht daher für börsennotierte Unternehmen dem Marktwert des Eigenkapitals. Damit stellt dieser Ansatz das direkte Pendant zur Ertragswertmethode dar. Anstelle der Erträge bzw. Jahresüberschüsse werden entsprechend die Free Cashflows

⁶² Vgl. Wöltje [2021], S. 123.

⁶³ Vgl. Schmeisser et al. [2009], S. 147

⁶⁴ Vgl. Wöltje [2021], S. 123

berücksichtigt.⁶⁵ Die sogenannte ewige Rente wird analog zum Entity Ansatz kalkuliert, anstelle des WACCs aber aus der Differenz des Diskontierungszinssatzes r und der ewigen Wachstumsrate g kalkuliert.

Der Equity Ansatz lässt sich mathematisch wie folgt ausdrücken:

$$UW_{DCF} = \sum_{t=1}^n \frac{FCF_t}{(1+r)^t} + \frac{TV_n}{(1+r)^n}$$

UW_{DCF}	Unternehmenswert Discounted Cashflow Verfahren
FCF_t	Free Cashflow in Periode t
TV_n	Terminal Value (Ertrag der ewigen Rente ab der letzten Planungsperiode)
r	Diskontierungszinssatz
t	Periode
n	Endperiode

Formel 7: Unternehmenswert Discounted Cashflow Verfahren (Equity Ansatz)

Der Terminal Value im Equity Ansatz wird wie folgt dargestellt:

$$TV_{Equity} = \frac{FCFE_T(1+g)}{r-g}$$

TV_{Equity}	Unternehmenswert Discounted Cashflow Verfahren
$FCFE_T$	Free Cashflow to Equity in letzter Periode T
r	Risikoadjustierter Zinssatz der Eigenkapitalkosten
g	Ewige Wachstumsrate

Formel 8: Terminal Value (Equity Ansatz)

Insgesamt, trotz seiner breiten Akzeptanz in Theorie und Praxis, ist das Discounted-Cashflow-Verfahren mit einer Reihe struktureller Herausforderungen verbunden. Besonders kritisch ist die hohe Prognoseabhängigkeit: Da der Unternehmenswert wesentlich durch die Planungsannahmen über zukünftige Free Cashflows und die langfristige Wachstumsrate abgeleitet wird, kann bereits eine geringe Veränderung in den Inputgrößen zu signifikant abweichenden Ergebnissen führen. Diese Sensitivität macht das Modell anfällig für Verzerrungen oder strategische Annahmen – insbesondere in Bewertungs- oder Transaktionskontexten mit asymmetrischer Informationsverteilung.⁶⁶

Ein weiterer kritischer Punkt ist die häufig vereinfachende Behandlung der Kapitalkosten, wie zuvor dargestellt über den risikoadjustierten Zinssatz im Equity Modell oder den WACC im Entity Model. Die zugrunde liegende Annahme einer konstanten Kapitalstruktur über die gesamte Planungs- und Ewige-Rente-Phase widerspricht in vielen Fällen der Realität dynamischer Unternehmensentwicklung. Zudem stellt der WACC keine rein marktorientierte Größe dar, sondern basiert auf Modellannahmen (z. B.

⁶⁵ Vgl. Schmeisser et al. [2009], S. 138 f.

⁶⁶ Vgl. Crasselt et al. [2018], S. 50-54.

CAPM, Steuersätze), die unternehmensspezifisch variieren und schwer valide zu schätzen sind. Insbesondere bei Unternehmen mit hohem Verschuldungsgrad oder volatiler Finanzierung führt dies zu potenziellen Bewertungsverzerrungen.⁶⁷

Schließlich ist zu bedenken, dass das DCF-Verfahren trotz seiner mathematischen Präzision nur unter der Annahme des ‚Going Concern‘ (Unternehmensfortführung) belastbare Aussagen liefert. In Krisen-, Sanierungs- oder Umbruchphasen verliert das Verfahren seine Aussagekraft, da zukünftige Zahlungsströme und risikoadjustierte Diskontierung dann nicht mehr valide prognostiziert werden können. Auch weiche Faktoren wie Innovationsfähigkeit, Marktzugang oder regulatorische Unsicherheiten bleiben im klassischen DCF-Modell weitgehend unberücksichtigt. Aus diesen Gründen ist die DCF-Bewertung stets im Zusammenspiel mit qualitativen Analysen und Alternativverfahren (z. B. ‚Multiples‘ oder Szenarioanalysen) zu interpretieren.⁶⁸

Trotz der genannten Einschränkungen bleibt das Discounted-Cashflow-Verfahren ein zentraler Bewertungsansatz in der finanzwirtschaftlichen Praxis und Theorie. Seine fundamentalanalytische Herleitung auf Basis zukünftiger Zahlungsüberschüsse macht es besonders geeignet für Unternehmensbewertungen, bei denen detaillierte Planungen vorliegen und der Bewertungszweck eine objektspezifische Analyse erfordert. Voraussetzung ist jedoch eine transparente und nachvollziehbare Modellierung, bei der Unsicherheiten in den Annahmen kritisch reflektiert und – soweit möglich – durch Sensitivitätsanalysen oder alternative Szenarien adressiert werden. In dieser Funktion stellt das DCF-Verfahren eine wertvolle Bewertungslogik dar, die jedoch nicht als rein rechentechnische Wahrheit, sondern als strukturierter Argumentationsrahmen für unternehmerische Entscheidungsfindung verstanden werden sollte.

3.5.5 ‚Multiple-basiertes‘ DCF-Modell

In Ergänzung zum zuvor theoretisch dargestellten vollständigen DCF-Modell wird im Rahmen dieser Arbeit ein alternativer, aber dennoch sehr verbreiteter DCF-Ansatz angewandt, um eine vergleichbare, massenhaft durchgeführte Unternehmensbewertung auf Basis homogener Annahmen durchzuführen. Hierbei handelt es sich um die sogenannte ‚Multiple-basierte‘-Methode. Bei dieser Methode wird zur Berechnung des Terminal Values das Vielfache einer Kennzahl (wie z.B. das EBITDA) am Ende der Prognoseperiode angewandt. Diese Methode basiert ursprünglich auf der Annahme, dass

⁶⁷ Vgl. Wöltje [2021], S. 123 f.

⁶⁸ Schmeisser et al. [2009], S. 148 f.

das Unternehmen nach dem Prognosezeitraum verkauft werden kann, wobei der Verkaufspreis durch die Multiplikation einer finanziellen Kennzahl mit einem entsprechenden ‚Exit Multiple‘ bestimmt wird. Das Verhältnis des ‚Enterprise Values‘ (Unternehmenswert) zum EBITDA (Ergebnis vor Steuern, Zinsen & Abschreibung), das sogenannte EV/EBITDA-Multiple, ist dabei eine der am häufigsten verwendeten Kennzahlen, da es eine operative Bewertung des Unternehmens ermöglicht.

Das klassische Discounted-Cashflow-Verfahren erfordert eine detaillierte, individualisierte Modellierung der zukünftigen Zahlungsflüsse sowie der Kapitalkosten. Eine derart präzise Bewertung ist jedoch im Rahmen eines großflächigen, modellvergleichenden Untersuchungsdesigns – wie es dieser Arbeit zugrunde liegt – nicht praktikabel.

Ziel dieses daher angewandten Modells ist es nicht, die präziseste Unternehmensbewertung im Einzelfall zu ermöglichen, sondern eine neutrale Referenzbasis für den Vergleich mit datengetriebenen, ML-basierten Bewertungsansätzen bereitzustellen, da es basierend auf den vorhandenen Einzelunternehmensdaten überhaupt erst die Möglichkeit bietet, einen Massentest in dieser Form durchzuführen.

Die Methodik folgt dabei gängigen wissenschaftlichen und praktischen Anwendungen, bei denen zur Sicherstellung von Reproduzierbarkeit und Objektivität pauschale Annahmen verwendet werden.⁶⁹ Diese Methode wird in der Praxis bevorzugt, da sie auf beobachtbaren Marktdaten basiert und somit eine objektivere Bewertungsgrundlage bietet als rein wachstumsbasierte Modelle wie die ewige Rente bzw. der Terminal Value.

Die zugrundeliegenden Annahmen sind über alle Unternehmen konstant: Es wird ein pauschaler Diskontierungssatz verwendet, der die typischen Eigenkapitalkosten kapitalmarktorientierter Unternehmen abbildet.⁷⁰

Die grundlegende Formel gleicht der genannten Formel des Equity Ansatzes mit Abwandlung des genutzten Terminal Values:

⁶⁹ Vgl. Damodaran [2012], S. 22 f.

⁷⁰ Vgl. Kruschwitz [2019], S. 171

$$UW_{DCF} = \sum_{t=1}^n \frac{FCF_t}{(1+r)^t} + \frac{TV_m}{(1+r)^n}$$

UW_{DCF}	Unternehmenswert Discounted Cashflow Verfahren
FCF_t	Free Cashflow in Periode t
TV_m	Terminal Value aus multiple-basiertem Ansatz
r	Diskontierungszinssatz
t	Periode
n	Endperiode

Formel 9: Unternehmenswert Discounted Cashflow Verfahren (multiple-basiert)

Hierbei wird aber nun der Terminal Value mit der Methode eines ‚EV‘ zu EBITDA Multiples kalkuliert.

$$TV_M = EBITDA_T \times M - (D_T - C_T)$$

TV_M	Terminal Value Multiple-Basiert
FCF_t	EBITDA im letzten Prognosejahr T
D_T	Fremdkapital (Debt) im Jahr T
C_T	Liquidität (Cash) im Jahr T
M	Kalkuliertes EV/EBITDA Multiple

Formel 10: Terminal Value im multiple-basiertem Ansatz

TV_M ergibt sich aus dem im letzten Prognosejahr T erwirtschafteten EBITDA, multipliziert mit dem aus vergleichbaren Unternehmen oder Transaktionen abgeleiteten EV/EBITDA-Multiple M . Um den reinen Eigenkapitalwert zu erhalten, werden anschließend die zum Zeitpunkt T bestehenden Nettofinanzverbindlichkeiten Fremdkapital D_T abzüglich vorhandener Liquidität C_T abgezogen.

3.5.6 Effizientmarkthypothese

Um die Modelle in späteren Kapiteln zielgerichtet auf den Unternehmenswert trainieren zu können, benötigt es die historischen Unternehmenswerte der einzelnen Unternehmen zu den jeweiligen Zeitpunkten der Trainingsdaten. Als methodischen Ansatzpunkt ist die Nutzung der Marktkapitalisierung als Proxy für den tatsächlichen Unternehmenswert gewählt worden. Hierbei wird sich uns auf die sogenannte Effizientmarkthypothese (EMH) von Eugene F. Fama gestützt. Gemäß der Effizienzmarkthypothese von Eugene F. Fama spiegelt die Marktkapitalisierung als aggregiertes Ergebnis rationaler Marktteilnehmer den wahren Unternehmenswert wider, da alle öffentlichen Informationen unverzüglich in den Aktienkurs einfließen.⁷¹

Die EMH differenziert drei spezifische Grade der Informationseffizienz an Kapitalmärkten: Schwache Effizienz, wonach sämtliche historischen Preisdaten bereits

⁷¹ Vgl. Fama [1970], S. 383

in aktuellen Marktpreisen enthalten sind, die Halbstarke Effizienz (Semi-Strong Efficiency), welche postuliert, dass alle öffentlich zugänglichen Informationen in den Preisen abgebildet werden sowie die Starke Effizienz, wonach sämtliche Informationen, einschließlich privater oder interner Informationen, vollständig in die Marktpreise einfließen.⁷²

Im Kontext dieser Arbeit ist insbesondere die Halbstarke Effizienzform entscheidend, denn sie impliziert, dass die aktuelle Marktkapitalisierung eines börsennotierten Unternehmens, berechnet aus dem Produkt des aktuellen Aktienkurses und der Anzahl der ausgegebenen Aktien, eine zuverlässige Annäherung an dessen fairen Wert darstellt.⁷³

Diese Annahme erlaubt es, die Marktkapitalisierung als geeigneten Proxy für den Unternehmenswert zu betrachten, da bei hinreichend effizienten Märkten systematische Bewertungsfehler selten und nur vorübergehend auftreten sollten. In einer effizienten Marktsituation werden Fehleinschätzungen rasch durch Arbitrageprozesse korrigiert, wodurch Marktpreise sich dem gerechtfertigten Wert schnell annähern.⁷⁴

Insgesamt ist es wichtig zu betonen, dass sich die verschiedenen Bewertungsverfahren in ihrer Anwendung je nach Unternehmenssituation unterscheiden. So sind beispielsweise bestimmte Verfahren, die sich für die Bewertung in Liquidations- oder Krisensituationen eignen, eher darauf ausgerichtet, den Zerschlagungswert oder den Substanzwert eines Unternehmens zu ermitteln. Im Gegensatz dazu geht das Discounted-Cashflow-Verfahren stets von der Fortführung des Unternehmens (Going Concern) aus und ist daher für Unternehmen in Krisen nicht immer geeignet. Durch die Betrachtung dieser unterschiedlichen Verfahren wird deutlich, wie eng die Wahl der Bewertungsmethode mit der jeweiligen Unternehmenslage verknüpft ist.

4 Künstliche Intelligenz und Maschinelles Lernen

Zur besseren Einordnung des maschinellen Lernens und zur Abgrenzung gegenüber anderen KI-Modellen wird in diesem Kapitel zunächst auf den Begriff Künstliche Intelligenz eingegangen und folgend maschinelles Lernen näher erläutert. Abschließend erfolgt die Vorstellung existierender Algorithmen.

⁷² Vgl. Fama [1970], S. 383 ff.

⁷³ Vgl. Damodaran [2012], S. 100 ff.

⁷⁴ Vgl. Brealey et al. [2023], S. 328 ff.

4.1 Der Begriff Künstliche Intelligenz

Spätestens seit der Veröffentlichung von OpenAIs Large Language Model „ChatGPT“ ist KI in aller Munde. In der Literatur finden sich für Künstliche Intelligenz (KI) – im englischen Sprachraum als Artificial Intelligence (AI) bekannt – sehr verschiedene Definitionsansätze.

Ursprünglich wurde der Begriff 1955 von John McCarthy geprägt, der KI als die Wissenschaft und Technik verstand, Maschinen dazu zu bringen, Aufgaben zu lösen, die normalerweise menschliche Intelligenz erfordern.⁷⁵ Seither wurde der Begriff mehrfach weiterentwickelt und angepasst.

Für diese Masterthesis wird die Definition zugrunde gelegt, die 2019 von der von der Europäischen Kommission eingesetzten hochrangigen Expertengruppe für KI veröffentlicht wurde: „Künstliche Intelligenz (KI oder Artificial Intelligence, AI) bezeichnet Systeme, die intelligentes Verhalten zeigen, indem sie ihre Umgebung analysieren und – mit einem gewissen Grad an Autonomie – Maßnahmen ergreifen, um bestimmte Ziele zu erreichen. [...]“⁷⁶

Allgemein umfasst Künstliche Intelligenz die Fähigkeit von Maschinen, Aufgaben zu übernehmen, die kognitive Fähigkeiten wie Lernen, Problemlösen, Planen, Verstehen von Sprache und Wahrnehmung erfordern.⁷⁷ Dabei wird zwischen starker KI und schwacher KI unterschieden: Starke KI strebt danach, echte menschenähnliche Intelligenz zu erreichen, während schwache KI auf die Ausführung sehr spezifischer Aufgaben beschränkt ist.⁷⁸

Eine wichtige Eigenschaft von KI-Systemen ist ihre Fähigkeit zur Generalisierung, also der Anwendung erlernter Muster auf unbekannte Situationen. Entscheidende Bestandteile heutiger KI-Anwendungen sind Methoden des maschinellen Lernens, sowie regelbasierte Systeme.

Im Zuge des technologischen Fortschritts haben sich die Anwendungsoptionen von Künstlicher Intelligenz grundlegend weiterentwickelt. Inzwischen stehen diverse methodische Konzepte zur Verfügung, um komplexe Probleme zu lösen. Die Einteilung dieser Ansätze erfolgt häufig anhand der Art und Verfügbarkeit der zugrunde liegenden

⁷⁵ Vgl. Ertel [2008], S. 2

⁷⁶ High-Level Expert Group on Artificial Intelligence [2019], S.1

⁷⁷ Vgl. Buxmann & Schmidt [2021], S.3 f.

⁷⁸ Paaß & Hecker [2020], S. 418

Daten, auf deren Basis die jeweiligen Algorithmen konstruiert werden. Grundsätzlich lässt sich dabei zwischen regelbasierten Verfahren und maschinellem Lernen unterscheiden. Während einige Autoren das regelverarbeitende Lernen als Teilbereich des maschinellen Lernens einordnen, wird in dieser Arbeit eine funktionale Trennung vorgenommen, die sich an der konkreten Datenlage orientiert. Eine grafische Darstellung dieser Systematik bietet folgende Abbildung:

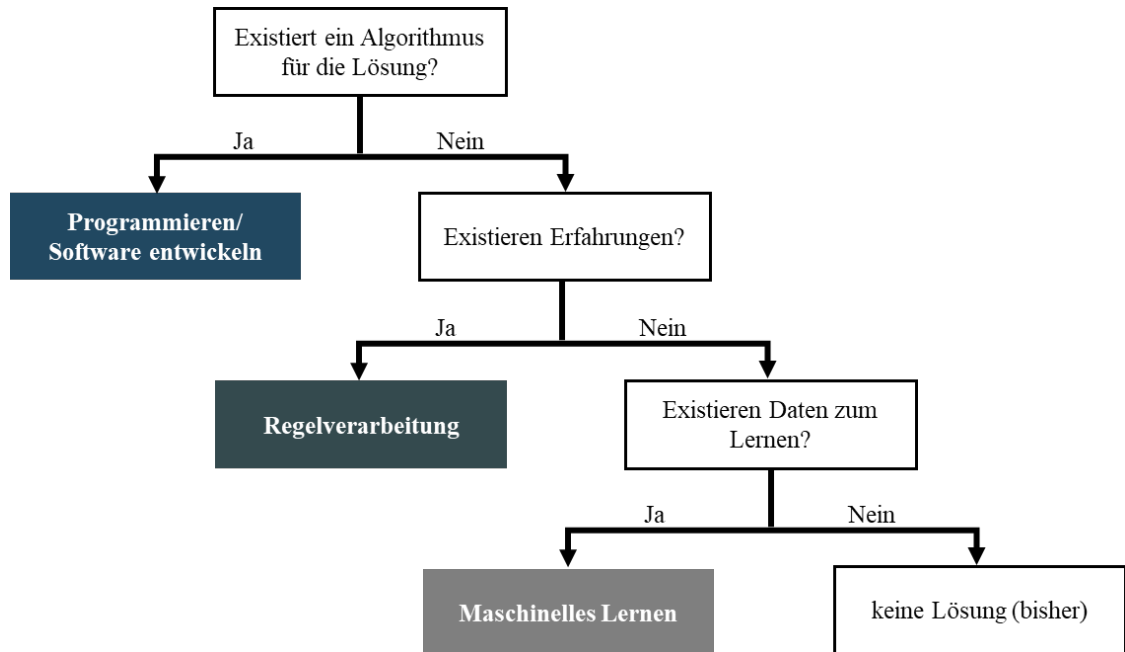


Abbildung 2: Unterscheidung von Regelbasiertem und Maschinellern in Abhängigkeit der vorliegenden Daten
(Quelle: Eigene Darstellung in Anlehnung an Lämmel/Cleve [2020] S. 13)

Abseits der bekannten Large Language Models wie OpenAIs ChatGPT, Anthropic Claude oder Googles Gemini gab es im maschinellen Lernen schon früher große Erfolge – beispielsweise in der Bild- und Spracherkennung – und auch in Verfahren wie Random Forest und Gradient Boosting, die man als Ensemble-Lernverfahren bezeichnet, weil sie mehrere Basis-Modelle zu einem stärkeren Gesamtsystem zusammenführen.⁷⁹ Regelbasierte Systeme (auch Expertensysteme genannt) reichen noch weiter zurück: Erste markante Beispiele wie DENDRAL zur chemischen Analyse oder MYCIN zur medizinischen Diagnose entstanden bereits in den 1970er-Jahren, als man begann, Wissen in Form von Wenn-Dann-Regeln symbolisch abzubilden.⁸⁰

Dennoch gibt es zu den jüngsten Erfolgen der generativen AI weiterhin Herausforderungen. Dazu zählen Fragen der Erklärbarkeit von Modellen (Explainable

⁷⁹ Vgl. Hastie et al. [2009], S. 587 - 595

⁸⁰ Vgl. Russell & Norvig [2003], S.26 ff.

AI), der ethischen Verantwortung sowie der gesellschaftlichen Auswirkungen durch Automatisierung und algorithmische Entscheidungsfindung.⁸¹ Auf diese Aspekte soll im späteren Hauptteil noch einmal gesondert eingegangen werden.

4.2 Regelbasiertes Lernen

Zu der besseren Abgrenzung des Teilbereichs des maschinellen Lernens von anderen Bereichen im Forschungsfeld der künstlichen Intelligenz wird in diesem Abschnitt zunächst der Begriff des Regelbasierten Lernens näher erläutert.

Regelbasiertes Lernen ist einen Teilbereich der Künstlichen Intelligenz und bezeichnet Verfahren, die auf einer expliziten Programmierung und festgelegten Regeln beruhen. Dabei sind sowohl die Bedingungen als auch die Konsequenzen klar definiert, sodass Entscheidungen und Aktionen automatisch abgeleitet werden können, wenn bestimmte Bedingungen erfüllt sind.⁸²

Im Gegensatz zum maschinellen Lernen, das auf datengetriebenen Methoden beruht, basiert regelbasiertes Lernen auf der logischen Schlussfolgerung (Deduktion). Ein bekanntes Beispiel für regelbasiertes Lernen sind Expertensysteme, die in der Lage sind komplexe Fragestellungen zu bearbeiten, indem sie auf eine umfassende Datenbank mit definierten Regeln zugreifen. Solche Systeme finden Verwendung in Bereichen, in denen klare, regelmäßige Zusammenhänge bestehen, wie zum Beispiel in der Steuerberatung, im juristischen Bereich oder in einfachen technischen Diagnosesystemen. Darstellbar ist dies etwa durch mehrere „Wenn-Dann“ Bedingungen, wie zum Beispiel beim Training zur Erkennung von Verkehrsschildern, wie die folgende Abbildung zeigt.

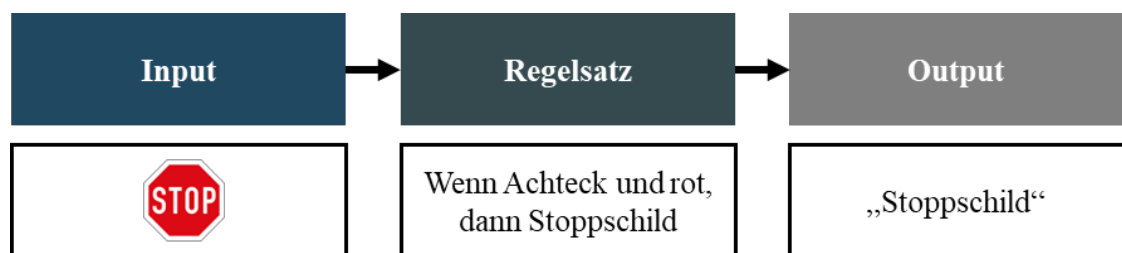


Abbildung 3: Beispiel für Regelbasiertes Lernen
(Quelle: eigene Darstellung)

Regelbasierte Systeme bieten den Vorteil einer hohen Transparenz und Nachvollziehbarkeit, da jede Entscheidung direkt auf eine explizit definierte Regel

⁸¹ Vgl. Buxmann & Schmidt [2021], S. 20 ff.

⁸² Vgl. Ertel [2008], S. 35 ff.

zurückgeführt werden kann. Allerdings ist der entscheidende Nachteil regelbasierter Systeme ihre geringe Flexibilität und Anpassungsfähigkeit, insbesondere bei komplexen, nichtlinearen Problemen, wie sie auch in der Praxis der Unternehmensbewertung häufig auftreten. Änderungen und Anpassungen müssen manuell durch Fachleute erfolgen, was einen erheblichen Aufwand darstellt und die Skalierbarkeit begrenzt.⁸³

4.3 Maschinelles Lernen

Maschinelles Lernen (ML) ist ein Teilgebiet der Künstlichen Intelligenz und bezeichnet Algorithmen, die Computern die Fähigkeit verleihen, aus Daten zu lernen und Muster zu erkennen, ohne explizit darauf programmiert zu sein.⁸⁴ Im Gegensatz zum regelbasierten Lernen, das, wie bereits erläutert, auf fest definierten Regeln beruht, erkennt maschinelles Lernen eigenständig Muster und Gesetzmäßigkeiten durch die Analyse großer Datenmengen. Dies ermöglicht eine deutlich höhere Flexibilität und Anpassungsfähigkeit bei komplexen Problemstellungen, die regelmäßig in der Praxis der Unternehmensbewertung aufkommen.

Das maschinelle Lernen wird in drei Arten unterschieden (s. Abbildung 4): Überwachtes, unüberwachtes und verstärkendes Lernen. Jede dieser Lernarten basiert auf spezifischen Prinzipien und Methoden zur Verarbeitung, Strukturierung und Interpretation von Daten. Sie unterscheiden sich sowohl in der Art und Weise, welche Daten zur Verfügung gestellt werden, wie sie mit vorhandenen Informationen umgehen, als auch in ihren Zielen und Einsatzmöglichkeiten.

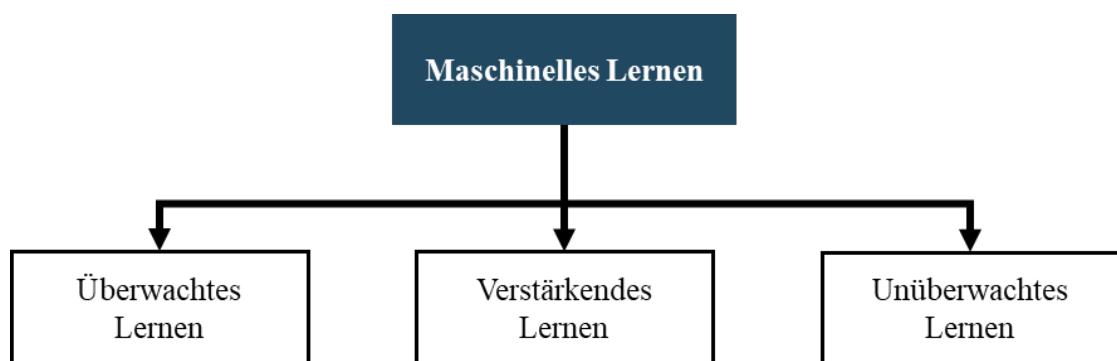


Abbildung 4: Differenzierung des Maschinellen Lernens in Unterarten
(Quelle: eigene Darstellung)

Die Leistungsfähigkeit maschineller Lernverfahren zeigt sich insbesondere bei der Lösung komplexer Mustererkennungsprobleme, die selbst für menschliche Analysten

⁸³ Vgl. Wittpahl [2019], S.45 f.

⁸⁴ Vgl. Murphy [2014], S. 1 f.

herausfordernd sind. Ein anschauliches Beispiel ist die visuelle Unterscheidung zwischen einem Hund und einem Wischmopp (s. Abbildung 5) – eine Aufgabe, die scheinbar triviale Objekterkennung erfordert, aber bei ähnlichen Formen und Texturen hohe Abstraktionsfähigkeit voraussetzt. Maschinelle Lernalgorithmen bewältigen solche Probleme durch Extraktion hierarchischer Merkmale aus Rohdaten, ohne explizite Regeln zu benötigen. Zentral für alle Kategorien ist daher das Training mit Daten.



Abbildung 5: Hund vs. Wischmopp
(Quelle: <https://www.beano.com/random/fun/sheepdog-or-mop>)

Beim überwachten Lernen werden Algorithmen auf Basis von Datenpaaren (Eingabedaten und zugehörige Ergebnisse) trainiert, um den Zusammenhang zwischen unabhängigen Variablen und einer abhängigen Zielgröße zu lernen.⁸⁵ Es werden also beschriftete bzw. kategorisierte Daten genutzt, um Algorithmen zu trainieren.

⁸⁵ Vgl. Aggarwal [2018], S. 42 ff.



Abbildung 6: Prinzip des Überwachten Lernens
(Quelle: eigene Darstellung)

Analog zum oben genannten Beispiel „Hund vs. Wischmopp“ kann man sich Trainingsbilder vorstellen, die vorher anhand der Kategorie „Wischmopp“ oder „Hund“ beschriftet wurden. Der Algorithmus des maschinellen Lernens findet dann selbstständig diejenigen Eigenschaften, die darstellen, warum es sich bei einzelnen Bildern um einen Hund oder Wischmopp handelt. Entsprechend ist ein sofortiges Feedback gegeben um welchen Output es sich handelt. Für die Unternehmensbewertung eignet sich dieses Verfahren indem es historische Finanzkennzahlen nutzt, um auf Basis gelabelter Transaktionsdaten den Kaufpreis einer Zielgesellschaft vorherzusagen.

Das unüberwachte Lernen bezeichnet Verfahren, bei denen Algorithmen ohne vordefinierte Zielgrößen auskommen. Im Gegensatz zum überwachten Lernen, das auf bekannten Ausgangsdaten basiert, besteht das Ziel des unüberwachten Lernens darin, verborgene Strukturen oder Muster in Daten eigenständig zu entdecken, wie auch Abbildung 7 darstellt.

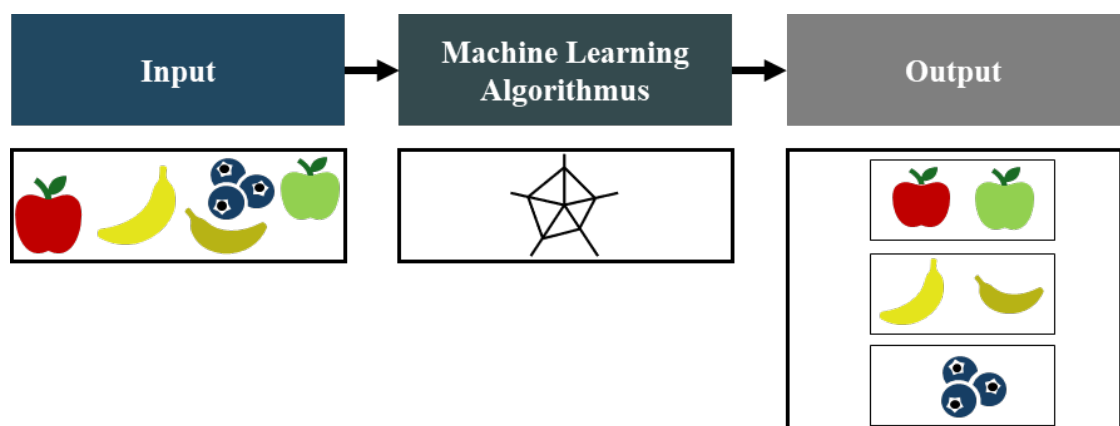


Abbildung 7: Prinzip des Unüberwachten Lernens
(Quelle: eigene Darstellung)

In der Unternehmensbewertung könnte unüberwachtes Lernen beispielsweise zur Segmentierung von Unternehmen nach Risikoprofilen oder Marktverhalten eingesetzt werden.⁸⁶

Das verstärkende Lernen ist ein Algorithmus, der in einer dynamischen Umgebung durch wiederholte Interaktionen optimalen Output generiert. Der Algorithmus erhält Feedback in Form von Belohnungen oder Strafen, basierend auf den erzielten Ergebnissen seiner Aktionen. Ziel ist es, eine langfristig optimale Strategie zu entwickeln, welche die kumulierten Belohnungen maximiert. In der Unternehmensbewertung könnte verstärkendes Lernen beispielsweise zur Optimierung von Verhandlungsstrategien oder zur Entscheidungsfindung unter Unsicherheit genutzt werden.⁸⁷

4.4 Existierende Algorithmen

In diesem Kapitel sollen die unterschiedlichen Machine Learning Algorithmen, die zum einen in dieser Arbeit praktische Anwendung finden, als auch generell relevante Machine Learning Algorithmen näher vorgestellt werden

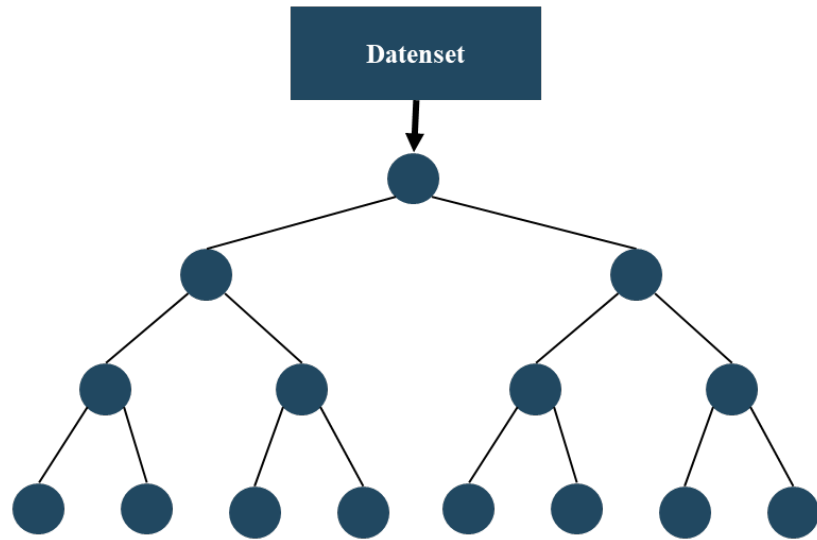
4.4.1 Random Forests

Random Forests (zu Deutsch „Zufallswälder“) sind ein prominentes Ensemble-Verfahren innerhalb des maschinellen Lernens, das erstmals von Leo Breiman im Jahr 2001 vorgestellt wurde. Es lässt sich den Algorithmen des Überwachten Lernens aus dem Vorkapitel zuordnen. Diese Methode kombiniert eine Vielzahl einzelner Entscheidungsbäume, um eine robuste Vorhersagegenauigkeit und erhöhte Generalisierungsfähigkeit gegenüber einzelnen Entscheidungsbäumen zu erzielen.

⁸⁶ Vgl. Aggarwal [2018], S. 52 ff.

⁸⁷ Vgl. Ertel [2008], S. 281 ff.

Entscheidungsbaum



Random Forest

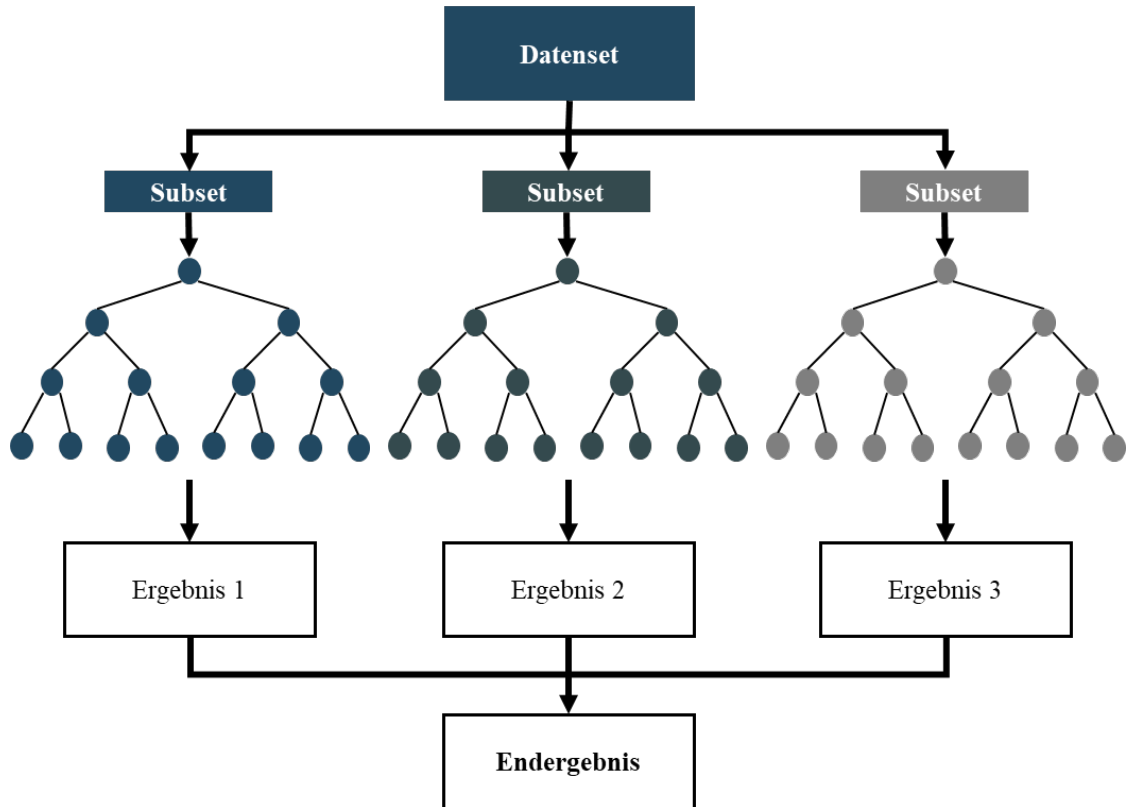


Abbildung 8: Visualisierung des Random Forest Prinzips
(Quelle: eigene Darstellung in Anlehnung an Hartshorn [2017], S. 27)

Die Grundidee eines Random Forest besteht darin, dass jeder Entscheidungsbaum aus einem zufälligen, unabhängig gezogenen Teil der Trainingsdaten erstellt wird – ein Verfahren, das als Bagging (Bootstrap Aggregating) bezeichnet wird. Zusätzlich wird bei jeder Aufteilung (engl. split) in den Entscheidungsbäumen eine zufällige Teilmenge der

zur Verfügung stehenden Merkmale berücksichtigt. Dadurch entsteht Diversität unter den Bäumen, welche wiederum die Varianz der Prognosen reduziert und die Stabilität der Ergebnisse erhöht.⁸⁸

Jeder einzelne Baum im Random Forest liefert eine Vorhersage für eine bestimmte Beobachtung. Die finale Vorhersage ergibt sich schließlich aus der Aggregation der Ergebnisse aller Bäume. Für Klassifikationsprobleme geschieht dies in der Regel durch Mehrheitsabstimmung, während bei Regressionsproblemen die Ergebnisse der einzelnen Bäume gemittelt werden.⁸⁹

Ein wesentlicher Vorteil von Random Forests ist deren Fähigkeit, sogenannte ‚Feature Importance‘, wie sie in Kapitel 4.5.1 erläutert werden, zu berechnen. Diese erlauben eine Interpretation, wie stark einzelne Merkmale zur Verbesserung der Prognosequalität beitragen. Häufig genutzte Metriken sind die Mean Decrease Impurity (MDI) sowie die Permutation Importance.⁹⁰ Besonders durch diese Eigenschaft sind Random Forests nicht nur für Prognosezwecke geeignet, sondern bieten zusätzlich Möglichkeiten zur Interpretation komplexer Zusammenhänge innerhalb großer Datensätze.

Allerdings bringt auch der Random-Forest-Ansatz gewisse Limitationen mit sich: Trotz ihrer robusten Natur neigen sie bei stark verrauschten Daten gelegentlich dazu, komplexe Zusammenhänge zu überinterpretieren. Daher sind eine sorgfältige Validierung sowie eine bewusste Begrenzung der Komplexität, etwa durch Parameter wie maximale Tiefe oder Mindestgröße eines Blattes, wichtig.⁹¹

Insgesamt gelten Random Forests aufgrund ihrer Flexibilität, Robustheit gegenüber überflüssigen Merkmalen und Fähigkeit zur Modellinterpretation als ein leistungsfähiges und weitverbreitetes Werkzeug im maschinellen Lernen. Sie werden insbesondere dann bevorzugt, wenn hohe Prognosegenauigkeit bei gleichzeitiger Interpretierbarkeit der Merkmalsbeiträge gefragt ist.⁹²

Ein Nachteil der Random Forests liegt in ihrer begrenzten Interpretierbarkeit im Vergleich zu einfachen Entscheidungsbäumen oder linearen Modellen. Außerdem kann

⁸⁸ Vgl. Hastie et al. [2009], S. 587 ff.

⁸⁹ Vgl. Breiman [2001], S.10 ff.

⁹⁰ Vgl. Hastie et al. [2009], S. 593 ff.

⁹¹ Vgl. James et al. [2021], S. 338 ff.

⁹² Vgl. Bishop [2006], S. 663 f.

das Training bei sehr großen Datensätzen und einer hohen Anzahl von Bäumen ressourcenintensiv sein.

4.4.2 Gradient Boosting

Gradient Boosting ist – ähnlich wie Random Forests – ein sogenanntes Ensemble-Verfahren, bei dem eine Vielzahl schwacher Modelle meist einfacher Entscheidungsbäume kombiniert werden, um ein leistungsstarkes Vorhersagemodell zu erzeugen. Entwickelt wurde Gradient Boosting maßgeblich durch Jerome Friedman, dessen Arbeiten fundamentale theoretische und praktische Grundlagen für das Verfahren geschaffen haben.⁹³

Gradient Boosting

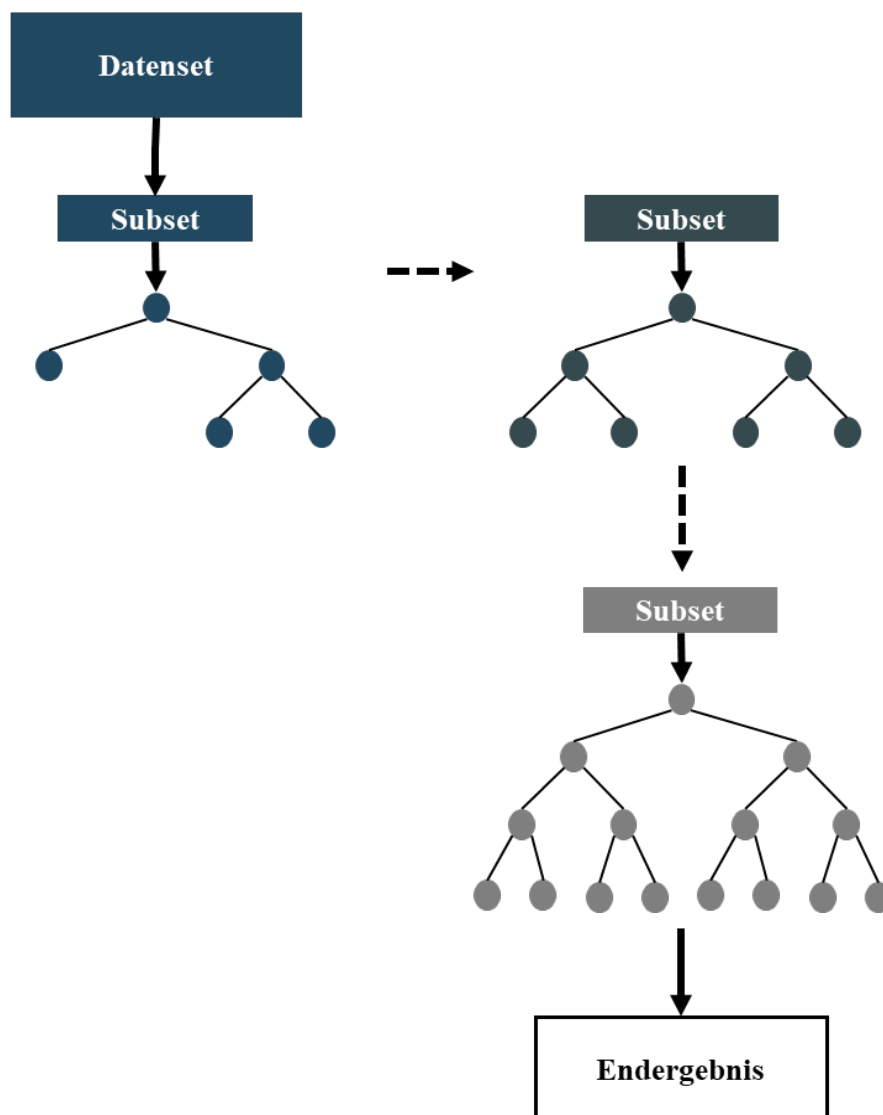


Abbildung 9: Visualisierung der Gradient Boosting Methode
(Quelle: eigene Darstellung in Anlehnung an Hartshorn [2017], S. 19 ff.)

⁹³ Vgl. Friedman [2001], S. 1189 ff.

Im Gegensatz zum bereits erläuterten Bagging, welches bei Random Forests Anwendung findet, basiert Gradient Boosting auf dem sogenannten Boosting-Verfahren. Während beim Bagging die einzelnen Modelle unabhängig voneinander parallel trainiert werden, erfolgt beim Boosting das Training sequentiell: Jeder neue Baum wird gezielt daraufhin trainiert, die Vorhersagefehler des vorherigen Ensembles zu korrigieren.⁹⁴

Konkret wird beim Gradient Boosting zunächst ein einfaches Ausgangsmodell erstellt, etwa ein Entscheidungsbaum mit geringer Tiefe. Danach werden iterativ neue Bäume hinzugefügt, wobei jeder neue Baum so angepasst wird, dass er die Residuen, also die Fehler des bisherigen Modells, möglichst effektiv reduziert. Der Begriff „Gradient“ im Namen verweist darauf, dass dieser iterative Prozess einer Minimierung eines Verlustkriteriums, der sogenannten Loss-Funktion folgt, indem dessen Gradienten absteigend verfolgt werden.⁹⁵

Gradient Boosting zeichnet sich durch eine hohe Prognosegenauigkeit aus und ist besonders erfolgreich bei strukturierten Datensätzen, bei denen komplexe Zusammenhänge vorliegen. Das Verfahren wurde durch seine besonders gute Performance in vielen Wettbewerben populär und stellt oft den aktuellen Goldstandard für tabellarische Daten dar.⁹⁶

Allerdings neigt Gradient Boosting stärker zu dem Phänomen ‚Overfitting‘, also der Überanpassung, welches auch in Kapitel 4.7 gesondert erläutert wird, als Random Forests, da es sehr aggressiv daran arbeitet Fehler vollständig zu beseitigen. Um Overfitting zu reduzieren, werden häufig Regularisierungstechniken eingesetzt, etwa das Einschränken der Anzahl der Bäume, die Begrenzung der Baumtiefe oder das sogenannte Shrinkage, auch Lernrate genannt, wodurch jeder einzelne Baum nur einen kleinen Teil der Fehlerkorrektur übernimmt.⁹⁷

Wie Random Forests liefert auch Gradient Boosting eine Metrik zur Bestimmung der ‚Feature Importance‘, welche dabei hilft, die relevanten Merkmale im Modell zu interpretieren. Auch hier kann entweder Mean Decrease Impurity oder Permutation Importance verwendet werden.⁹⁸

⁹⁴ Hastie et al. [2009], S. 353 ff.

⁹⁵ Vgl. Friedman [2001], S. 1199 ff.

⁹⁶ Vgl. James et al. [2021], S. 348 ff.

⁹⁷ Vgl. James et al. [2021], S. 352 ff.

⁹⁸ Vgl. Hastie et al. [2009], S. 367 f.

Insgesamt bietet Gradient Boosting ein leistungsfähiges Werkzeug zur datenbasierten Unternehmensbewertung, insbesondere dann, wenn umfangreiche historische Daten und vielfältige Einflussfaktoren berücksichtigt werden sollen. Seine hohe Prognosekraft und Flexibilität machen es zu einem der beliebtesten Verfahren im Bereich Machine Learning für anspruchsvolle Regressions- und Klassifikationsaufgaben.

4.4.3 Künstliche Neuronale Netze

Künstliche Neuronale Netze (KNN) orientieren sich, wie der Name bereits andeutet, am Vorbild biologischer Nervenzellen und bestehen aus vielen miteinander verknüpften Einheiten, den künstlichen Neuronen, die über gewichtete Verbindungen Signale austauschen und so gemeinsam komplexe Aufgaben lösen können.

Den Grundstein legten die Neurophysiologen und Logiker Warren McCulloch und Walter Pitts bereits 1943, als sie in „A Logical Calculus of the Ideas Immanent in Nervous Activity“ zeigten, dass sich durch einfache binäre Bausteine und logische Verknüpfungen beliebige logische und arithmetische Funktionen nachbilden lassen.⁹⁹

Ein vollständiges Netzwerk gliedert sich in eine Eingabeschicht, eine oder mehrere verborgene Schichten („Hidden Layers“) und eine Ausgabeschicht. In der Eingabeschicht werden die Rohdaten – beispielsweise historische Umsätze oder Bilanzkennzahlen – eingespeist. Die verborgenen Schichten übernimmt dabei die eigentliche Transformation: Sie erlauben durch hintereinandergeschaltete Aktivierungen die Modellierung hochkomplexer, nichtlinearer Zusammenhänge. Die Ausgabeschicht liefert das finale Ergebnis, sei es eine Prognose für künftige Cashflows, eine Klassifikation oder eine Regression.

Wie in der folgenden Abbildung dargestellt empfängt jedes künstliche Neuron mehrere Eingangssignale (blau), die jeweils mit einem Gewicht multipliziert werden, bevor sie gesammelt weiterverarbeitet werden. Anschließend entscheidet eine Aktivierungsfunktion basierend auf den Informationen der Eingabeschicht, ob die Neuronen der Eingabeschicht aktiviert oder deaktiviert werden und leitet entsprechend der Informationen der aktivierten Neuronen an eine verborgene Schicht (grün), den sogenannten „Hidden Layer“, weiter. Über eine eigene Aktivierungsfunktion legt diese verborgene Schicht fest, ob die (grünen) Neuronen aktiv sind oder nicht, und leitet die Informationen anschließend an die graue Ausgabeschicht weiter.

⁹⁹ Vgl. McCulloch & Pitts [1943], S. 115 - 133

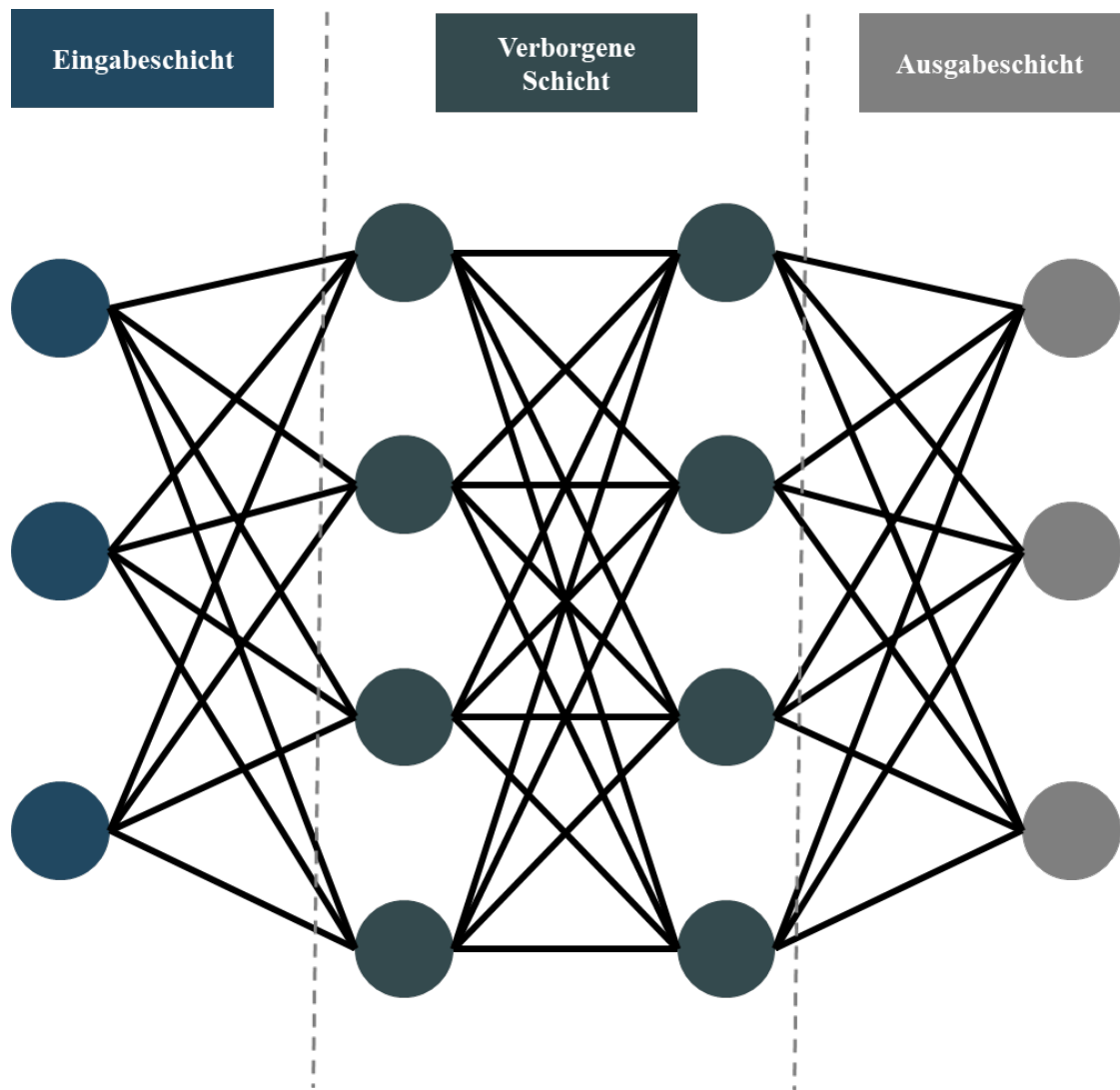


Abbildung 10: Prinzip künstlicher neuronaler Netze
 (Quelle: Eigene Darstellung in Anlehnung an Buxmann/Schmidt [2021] S.15)

Das Lernen eines Netzes erfolgt iterativ: Nach jedem Durchlauf wird das Ergebnis mit dem erwarteten Wert verglichen und die Diskrepanz als Fehler zurück durch alle Verbindungen geleitet. Dabei werden die Gewichte so angepasst, dass der Fehler schrittweise abnimmt und das Netz seine Vorhersagen stetig verbessert

In der Unternehmensbewertung spielen künstliche neuronale Netze vor allem dort ihre Stärken aus, wo es gilt, nichtlineare Muster in umfangreichen und heterogenen Datensätzen zu erkennen. Sie eignen sich etwa zur Vorhersage von Umsätzen, Gewinnen oder Cashflows, wenn klassische lineare Modelle an ihre Grenzen stoßen.¹⁰⁰ Allerdings erfordern sie für verlässliche Ergebnisse eine große Menge qualitativ hochwertiger Trainingsdaten sowie ausreichende Rechenkapazitäten, weshalb sie in dieser Arbeit aufgrund begrenzter Datengrundlagen nicht zum Einsatz kommen.

¹⁰⁰ Vgl. Goodfellow et al. [2016], S. 75

Künstliche neuronale Netze können beliebig viele versteckte Schichten beinhalten, dies wird im folgenden Kapitel ‚Deep Learning‘ näher erläutert.

4.4.4 Deep Learning

‚Deep Learning‘ ist ein spezieller Teilbereich des maschinellen Lernens, der auf tiefen künstlichen neuronalen Netzen basiert. Charakteristisch für Deep Learning-Modelle ist der Einsatz vieler versteckter Schichten, wodurch sie in der Lage sind, hochkomplexe und nichtlineare Zusammenhänge in großen Datenmengen zu erkennen und zu modellieren.¹⁰¹

Deep Learning-Modelle haben insbesondere im Bereich der Bild-, Sprach- und Textverarbeitung revolutionäre Fortschritte ermöglicht. Ihre Fähigkeit, Merkmale automatisch aus Rohdaten zu extrahieren und zu hierarchisieren, unterscheidet sie von klassischen Machine-Learning-Ansätzen, die oft eine aufwändigere manuelle Merkmalsauswahl benötigen.

In der Unternehmensbewertung könnte Deep Learning insbesondere bei der Auswertung unstrukturierter Daten, wie Textinformationen aus Geschäftsberichten, Analystenmeinungen oder Marktberichten neue Anwendungsmöglichkeiten eröffnen. Beispielsweise könnten zeitliche Entwicklungen und Trends in Finanzdaten präziser vorherzusagen.¹⁰²

Ein großer Vorteil von Deep Learning ist seine außerordentliche Prognosegenauigkeit bei komplexen Aufgaben. Allerdings benötigen Deep-Learning-Modelle große Mengen an Trainingsdaten sowie erhebliche Rechenressourcen, was ihren Einsatz in der Praxis der Unternehmensbewertung bislang einschränkt. Zudem bleibt die Nachvollziehbarkeit der Modellentscheidungen, die sogenannte Black-Box-Problematik, eine große Herausforderung, insbesondere in regulierten Bereichen wie der Finanzberichterstattung.

4.5 Explainable AI

Ein zentrales Hindernis für den breiten Einsatz maschineller Lernverfahren in betriebswirtschaftlichen Entscheidungsprozessen ist ihre begrenzte Nachvollziehbarkeit. Während klassische Bewertungsverfahren wie das Discounted-Cashflow-Modell auf expliziten Annahmen und transparenten Rechenwegen beruhen, gelten viele Machine-Learning-Modelle – insbesondere tiefere neuronale Netze oder komplexe Ensemble-Methoden – als „Black Boxes“, deren innere Funktionsweise für Anwender kaum

¹⁰¹ Vgl. Aggarwal [2018], S. 1 ff.

¹⁰² Vgl. Skansi [2018], S. 70 ff.

einschbar ist. Diese mangelnde Transparenz stellt insbesondere im regulierten Umfeld der Finanzberichterstattung ein erhebliches Akzeptanzproblem dar.¹⁰³

Unter dem Begriff Explainable Artificial Intelligence (XAI) werden Ansätze und Methoden zusammengefasst, die darauf abzielen, die Entscheidungen komplexer Machine Learning-Modelle nachvollziehbar zu machen. Ziel ist es, die zugrunde liegenden Entscheidungslogiken zu visualisieren, zu quantifizieren und für nicht-technische Nutzer interpretierbar aufzubereiten. Dies ist insbesondere dann relevant, wenn ML-basierte Bewertungsmodelle als Grundlage für externe Kommunikation (z. B. in Gutachten oder Jahresabschlüssen) oder interne Investitionsentscheidungen dienen sollen.¹⁰⁴

In der Praxis haben sich insbesondere zwei methodische Ansätze zur Erhöhung der Erklärbarkeit durchgesetzt: das Modellagnostische Verfahren wie LIME (Local Interpretable Model-Agnostic Explanations) und globale Strukturvisualisierungen wie Entscheidungsbaumanalysen, SHAP oder ‚Feature-Importances‘ bei Random Forests oder Boosting-Modellen.¹⁰⁵ Diese werden im Folgekapitel näher erläutert, da diese für die in dieser Arbeit angewandter Algorithmen von hoher Bedeutung sind.

4.5.1 Feature Importance

Ein Element für den Erfolg maschineller Lernverfahren ist die Qualität der verwendeten Eingangsdaten. In der Unternehmensbewertung bedeutet das nicht nur die Verfügbarkeit historischer Finanzkennzahlen, sondern vor allem deren systematische Aufbereitung zu modellrelevanten Inputvariablen. Dieser Prozess wird unter dem Begriff ‚Feature Engineering‘ zusammengefasst. Feature Engineering bezeichnet alle Maßnahmen zur Auswahl, Transformation, Konstruktion und gegebenenfalls Reduktion von Merkmalen, mit dem Ziel, die Prognosegüte und Robustheit eines Machine-Learning-Modells zu verbessern.¹⁰⁶

Insbesondere bei strukturierten Finanzdaten – wie sie im Rahmen der Unternehmensbewertung vorliegen – ist Feature Engineering essenziell, um lineare wie nichtlineare Zusammenhänge, saisonale Effekte oder zeitverzögerte Wirkmechanismen explizit zu modellieren.

¹⁰³ Vgl. Molnar [2022], S. 3 f.

¹⁰⁴ Vgl. Buxmann [2021], S. 197 f.

¹⁰⁵ Vgl. Molnar [2022], S. 101-130

¹⁰⁶ Vgl. Kuhn & Johnson [2013], S. 367 f.

Ein entscheidender Vorteil datengetriebener Modelle wie Random Forest oder Gradient Boosting liegt in ihrer Fähigkeit, auch mit einer großen Anzahl potenziell irrelevanter oder korrelierter Merkmale umzugehen. Dennoch kann ein zu umfangreicher Feature-Satz zu sogenanntem Overfitting, längeren Trainingszeiten oder schlechterer Generalisierbarkeit führen. Daher ist eine zielgerichtete Merkmalsselektion erforderlich.¹⁰⁷

Besondere Bedeutung kommt in der Zeitreihenanalyse – etwa bei der Prognose von Unternehmenswerten auf Basis historischer Daten – den sogenannten Lag-Features zu. Diese erlauben es, vergangene Werte einer Variable, z. B. Bruttomarge im Vorjahr als eigenständige Prädiktoren einzubeziehen und so zeitliche Dynamiken zu erfassen. Eine häufige Erweiterung stellt die Bildung von gleitenden Durchschnitten (Rolling Windows) oder Volatilitätsmaßen dar, die kurzfristige Trends und Schwankungen abbilden, wie sie auch in dieser Arbeit Anwendung finden.¹⁰⁸

Im Rahmen dieser Arbeit wurde das Feature Engineering als eigenständiger Modellierungsschritt zwischen Datenextraktion und Modellentwicklung verstanden. Es stellt somit einen integralen Bestandteil des Bewertungsprozesses dar, da die Qualität der Feature-Konstruktion maßgeblich über die Erklärungs- und Prognosekraft der eingesetzten Machine-Learning-Modelle entscheidet. Die Relevanz dieses Schritts lässt sich prägnant mit einem bekannten Zitat des Datenwissenschaftlers Pedro Domingos zusammenfassen: „A dumb algorithm with lots and lots of data beats a clever one with modest amounts“¹⁰⁹.

4.5.2 SHAP SHapley Additive exPlanations

SHAP (SHapley Additive exPlanations) ist ein Ansatz zur Interpretation von Machine-Learning-Modellen, der auf den Shapley-Werten aus der kooperativen Spieltheorie basiert. SHAP weist jedem Feature für eine bestimmte Vorhersage einen Relevanzwert zu und quantifiziert so, wie stark jedes einzelne Merkmal zum Modelloutput beiträgt. Die Methode zeichnet sich dadurch aus, dass sie eine neue Klasse additiver Feature-Attributsverfahren definiert und mathematisch beweist, dass es innerhalb dieser Klasse nur eine Lösung gibt, die alle gewünschten Eigenschaften wie Konsistenz und Additivität erfüllt. SHAP kann dadurch sowohl globale als auch lokale Erklärungen liefern und ist

¹⁰⁷ Vgl. James et al. [2021], S. 216 - 220

¹⁰⁸ Vgl. Hyndman & Athanasopoulos [2021], S. 142 ff.

¹⁰⁹ Domingos [2012], S. 84

für eine Vielzahl von Modellen, darunter auch komplexe Ensemble-Modelle wie Gradient Boosting und Random Forest, anwendbar.¹¹⁰

4.5.3 LIME (Local Interpretable Model-agnostic Explanations)

LIME (Local Interpretable Model-agnostic Explanations) verfolgt einen anderen Ansatz, um die Entscheidungen von Machine-Learning-Modellen nachvollziehbar zu machen. LIME erklärt einzelne Modellvorhersagen, indem es das Verhalten des komplexen Modells lokal um eine konkrete Instanz herum durch eine lineare Regression, approximiert. Dazu werden künstlich erzeugte, leicht veränderte Datenpunkte in der Umgebung der zu erklärenden Instanz generiert und deren Vorhersagen durch das Originalmodell bewertet. Dieses lokale Modell wird dann auf diese lokal gewichteten Datenpunkte trainiert, sodass es die Entscheidungslogik des komplexen Modells in diesem lokalen Bereich möglichst gut abbildet. LIME ist dabei modellagnostisch und kann für beliebige Klassifikatoren eingesetzt werden.¹¹¹

4.6 Hyperparameter

Beim Einsatz moderner Machine-Learning-Methoden wie Random Forest und Gradient Boosting spielt die richtige Einstellung sogenannter Hyperparameter eine zentrale Rolle für die Modellgüte. Hyperparameter sind dabei jene Einstellungen, die vor dem eigentlichen Training festgelegt werden – beispielsweise die Anzahl der Entscheidungsbäume, die maximale Tiefe der Bäume oder die Lernrate, mit der das Modell aus Fehlern lernt. Sie steuern maßgeblich, wie flexibel und leistungsfähig das Modell am Ende ist.¹¹²

Beim Random Forest sind insbesondere die Anzahl der Bäume sowie die Auswahl der Merkmale bei jeder Entscheidung im Baum wichtige Stellschrauben. Oft liefern bereits die Standardwerte solide Ergebnisse, doch gezieltes „Tuning“ dieser Parameter kann die Vorhersagegenauigkeit weiter verbessern.¹¹³ In der Praxis wird das Tuning meist automatisiert durchgeführt, indem verschiedene Einstellungen ausprobiert und anhand von Messwerten wie Genauigkeit oder Fehlerquote bewertet werden.¹¹⁴

¹¹⁰ Vgl. Luxburg et al. [2018], S. 4765 - 4774

¹¹¹ Vgl. Krishnapuram [2016], S. 1135 - 1144

¹¹² Vgl. Hastie et al. [2009], S. 587 ff.

¹¹³ Vgl. Probst et al. [2019], S.6 f.

¹¹⁴ Vgl. Géron [2019], S. 109 ff.

Gradient Boosting-Algorithmen reagieren noch empfindlicher auf die Wahl der Hyperparameter. Hier sind vor allem die Lernrate, die Baumtiefe und die Anzahl der Bäume entscheidend.¹¹⁵ Eine zu hohe Lernrate oder zu tiefe Bäume führen schnell zu Überanpassung, während zu niedrigen Werten die Modellleistung einschränken. Daher ist das Hyperparameter-Tuning bei Gradient Boosting besonders wichtig und erfolgt häufig mit modernen Optimierungsverfahren wie ‚Random Search‘ oder Bayes'sche Optimierung, die gezielt kombinatorisch erfolgversprechende Einstellungen testen.¹¹⁶ Auf Random Search oder Bayes'sche Optimierung soll an dieser Stelle allerdings nicht eingegangen werden, vielmehr wird die Bedeutung der Hyperparameter anhand klassischer Methoden wie der systematischen Anwendung verschiedene Kombinationen, um die beste Modellleistung zu erzielen, im Hauptteil dieser Arbeit ersichtlich.

Für beide Methoden stehen heute zahlreiche Softwarepakete zur Verfügung, die das Hyperparameter-Training weitgehend automatisieren. Die Auswahl der besten Einstellungen erfolgt meist mit Hilfe von Kreuzvalidierung oder speziellen Testmethoden, um eine Überanpassung an die Trainingsdaten zu vermeiden und die Übertragbarkeit auf neue Daten sicherzustellen.¹¹⁷

Insgesamt zeigt sich, dass die sorgfältige Abstimmung der Hyperparameter ein zentraler Erfolgsfaktor für den praktischen Einsatz von Random Forest und Gradient Boosting ist.¹¹⁸ Moderne Optimierungsmethoden ermöglichen es, auch bei komplexen Datensätzen effiziente und leistungsfähige Modelle zu entwickeln.¹¹⁹

4.7 Overfitting

Overfitting stellt eine der Herausforderungen bei der Entwicklung und Anwendung maschineller Lernmodelle dar. Es beschreibt das Phänomen, dass ein Modell nicht nur die zugrundeliegenden Zusammenhänge in den Trainingsdaten erlernt, sondern auch zufällige Schwankungen und Rauschen als vermeintlich relevante Muster interpretiert.¹²⁰ Dadurch erreicht das Modell zwar eine sehr hohe Genauigkeit auf den Trainingsdaten, verliert jedoch die Fähigkeit, auf neuen, unbekanntem Daten zuverlässige Vorhersagen zu treffen – es generalisiert also schlecht. Entsprechend passt sich das Modell zu stark an die Trainingsdaten an und reagiert empfindlich auf kleinste Veränderungen oder

¹¹⁵ Vgl. Hastie et al. [2009], S. 387 ff.

¹¹⁶ Vgl. Géron [2019], S. 112 ff.

¹¹⁷ Vgl. Géron [2019], S. 114 ff.

¹¹⁸ Vgl. Probst et al. [2019], S. 1

¹¹⁹ Vgl. Hastie et al. [2009], S. 389

¹²⁰ Vgl. Hastie et al. [2009], S. 37 f.

Ausreißer. Besonders komplexe Modelle, wie tief verschachtelte Entscheidungsbäume oder neuronale Netze mit vielen Parametern, sind anfällig für Overfitting, wenn keine geeigneten Gegenmaßnahmen angewendet werden. Zu den gängigen Strategien zur Vermeidung von Overfitting zählen die Anwendung von Regularisierungstechniken, die Nutzung von Cross-Validation zur Modellbewertung sowie die gezielte Auswahl und Reduktion von Merkmalen, letztere findet auch in dieser Arbeit Anwendung.¹²¹

Im Kontext der Unternehmensbewertung ist Overfitting besonders kritisch, da Modelle, die zu stark auf die Besonderheiten der historischen Daten angepasst sind, in der Praxis zu fehlerhaften und nicht robusten Bewertungen führen können. Daher ist es essenziell, bereits im Feature Engineering und bei der Modellauswahl auf eine ausgewogene Balance zwischen Modellkomplexität und Generalisierungsfähigkeit zu achten.

4.8 Erfolgsmessung maschineller Lernmodelle

Um die Qualität und den Erfolg eines maschinellen Lernmodells objektiv bewerten zu können, werden verschiedene Kennzahlen und Metriken verwendet. Neben der bereits erläuterten Merkmalswichtigkeit mittels Mean Decrease Impurity (MDI) und Permutation Importance kommt dem Bestimmtheitsmaß R^2 eine wichtige Rolle zu.

4.8.1 Mean Decrease Impurity (MDI)

Die Mean Decrease Impurity (zu Deutsch „mittlere Abnahme der Unreinheit“) bewertet die Wichtigkeit eines Merkmals anhand seiner Fähigkeit, die Unreinheit in den Entscheidungsbäumen eines Random Forests oder Gradient Boosting-Modells zu reduzieren.¹²²

Formell betrachtet berechnet sich die *MDI* als gewichteter Durchschnitt der Reduktionen der Unreinheit über alle Knoten, in denen das Merkmal verwendet wurde. Merkmale, die häufig zu großen Verringerungen der Unreinheit führen, erhalten hohe Werte der Merkmalswichtigkeit.¹²³

Ein wesentlicher Vorteil der MDI-Methode liegt in ihrer schnellen und einfachen Berechnung während des Modelltrainings, wodurch sie für große Datensätze besonders geeignet ist. Eine Limitation besteht jedoch darin, dass die *MDI* zu einer Verzerrung neigen kann, insbesondere wenn Merkmale stark korreliert sind. In solchen Fällen ist die

¹²¹ Vgl. Kuhn & Johnson [2013], S. 28

¹²² Vgl. Hastie et al. [2009], S. 593 ff.

¹²³ Vgl. Hastie et al. [2009], S. 593

Interpretation schwierig, da stark korrelierte Merkmale oft ähnliche Importance-Werte erhalten.¹²⁴

4.8.2 Das Bestimmtheitsmaß R^2

Das Bestimmtheitsmaß (engl. coefficient of determination), häufig mit R^2 bezeichnet, stellt eine standardisierte Maßzahl dar, die angibt, welcher Anteil der Varianz der abhängigen Variablen durch das verwendete Regressionsmodell erklärt wird.¹²⁵

$$R^2 = 1 - \frac{SSR}{SST}$$

SSR	Summe der quadrierten Residuen
SST	Totale Varianz der Zielvariable

Formel 11: Bestimmtheitsmaß R^2

Dabei misst die Summe der quadrierten Residuen (SSR), wie stark die Vorhersagen des Modells von den tatsächlichen Werten abweichen, und die totale Varianz (SST) gibt an, wie stark die Zielvariable insgesamt von ihrem Mittelwert abweicht.¹²⁶

Der reguläre Wertebereich von R^2 liegt zwischen 0 und 1. Werte nahe 1 bedeuten, dass das Modell die Zielvariable sehr gut erklärt, während Werte nahe 0 darauf hinweisen, dass das Modell kaum besser als eine einfache Durchschnittsschätzung ist. Es sind jedoch auch negative Werte möglich, was auf besonders schlechte Prognosen hinweist.¹²⁷ R^2 lässt sich dabei wie folgt interpretieren:

$R^2 = 1$: Perfekte Prognose

$R^2 = 0$: Das Modell hat keine Vorhersagekraft gegenüber einer konstanten Schätzung

$R^2 < 0$: Das Modell ist schlechter als eine einfache Mittelwertschätzung

Ein Vorteil des R^2 -Maßes ist dessen einfache Interpretierbarkeit und weite Verbreitung, wodurch es als standardisierte Vergleichsgröße verschiedener Modelle dient. Eine Limitation von R^2 liegt jedoch darin, dass es automatisch bei Hinzufügen neuer Variablen steigt. Um diesem Nachteil entgegenzuwirken, wird häufig das adjustierte R^2 eingesetzt,

¹²⁴ Vgl. James et al. [2021], S. 341

¹²⁵ Vgl. James et al. [2021], S. 72 f.

¹²⁶ Vgl. James et al. [2021], S. 72

¹²⁷ Vgl. Bishop [2006], S. 41 f.

welches eine Anpassung für zusätzliche Variablen enthält und so Modellvergleiche bei unterschiedlicher Merkmalszahl ermöglicht.¹²⁸

Nachfolgend findet sich in Tabelle 2 ein Vergleich der Messzahlen.

Merkmals	Mean Decrease Impurity (MDI)	Bestimmtheitsmaß (R^2)
Messprinzip	Reduktion der Unreinheit in Entscheidungsbäumen	Anteil der erklärten Varianz
Interpretierbarkeit	Gut, jedoch verzerrt bei korrelierten Merkmalen	Sehr intuitiv, direkte Interpretation
Anwendungsfeld	Merkmalsinterpretation und -selektion	Modellbewertung und -vergleich
Robustheit bei Korrelation	Gering (Verzerrung möglich)	Nicht betroffen (betrifft nur Gesamtmodell)
Berechnungsaufwand	Niedrig (direkt aus Modelltraining)	Niedrig (aus Residuen direkt berechenbar)
Typischer Wertebereich	≥ 0 (relative Skala)	≤ 1 (typisch 0–1, negative Werte bei schlechter Prognose)
Stärke	Schnell, einfach berechenbar	Standardisierte, weit verbreitete und intuitive Vergleichbarkeit
Schwäche	Verzerrung bei korrelierten Merkmalen	Steigt automatisch mit zusätzlichen Merkmalen (

Tabelle 2: Vergleich MDI vs. R^2

4.8.3 Mean Absolute Error (MAE)

Der *MAE* ist der durchschnittliche absolute Fehler zwischen den vorhergesagten Werten \hat{y}_i und den tatsächlichen Werten y_i . Er gibt anschaulich an, um wie viele Einheiten das Modell im Mittel abweicht und ist weniger anfällig für Ausreißer als das quadratische Mittel.¹²⁹

Mathematisch kann der *MAE* wie folgt ausgedrückt werden:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

y_i tatsächlicher Wert
 \hat{y}_i vorhergesagter Wert

Formel 12: Mean Absolute Error (MAE)

¹²⁸ James et al. [2021], S. 79 f.

¹²⁹ Vgl. James et al. [2021], S. 54

4.8.4 Mean Absolute Percentage Error (MAPE)

Der Mean Absolute Percentage Error (*MAPE*) gibt den *MAE* prozentual auf den tatsächlichen Wert wieder. So lässt sich leicht beurteilen, um wieviel Prozentpunkte die Prognose im Schnitt von der Realität abweicht. Nachteilig ist jedoch, dass *MAPE* bei Werten nahe Null stark verzerrt sein kann, da die Division mit Werten nahe Null gegen unendlich konvergieren.¹³⁰

$$MAPE = \frac{100\%}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

y_i tatsächlicher Wert
 \hat{y}_i vorhergesagter Wert

Formel 13: Mean Absolute Percentage Error (MAPE)

4.8.5 Weighted Mean Absolute Percentage Error (WMAPE)

Der Weighted Mean Absolute Percentage Error (*WMAPE*) gibt, wie der Name vermuten lässt, den *WMAPE* gewichtet auf den prozentual auf den tatsächlichen Wert wieder. bei der die absoluten Fehler mit Gewichten w_i versehen werden, um Beobachtungen unterschiedlicher Bedeutung angemessen zu berücksichtigen. Üblicherweise wählt man als Gewicht das tatsächliche Volumen y_i , sodass große Werte stärker ins Gesamtmaß eingehen und kleine Werte – bei denen *MAPE* gegen Unendlich laufen kann – weniger verzerren. Dadurch erhält man eine robuste Kennzahl, die angibt, wie viel Prozent der gesamten tatsächlichen Summe im Schnitt als absoluter Fehler verbleiben.¹³¹

$$WMAPE = \frac{100\%}{\sum_{i=1}^n w_i} \sum_{i=1}^n w_i |y_i - \hat{y}_i|$$

y_i tatsächlicher Wert
 \hat{y}_i vorhergesagter Wert
 w_i Gewicht (meist und in dieser Thesis y_i)

Formel 14: Weighted Mean Absolute Percentage Error (WMAPE)

5 Methodik

Im Folgenden werden zur Analyse der Nutzenpotenziale des maschinellen Lernens für moderne Ansätze der Unternehmensbewertung Algorithmen des maschinellen Lernens auf Unternehmensdaten trainiert und eine Bewertung basierend auf Basis dieser

¹³⁰ Vgl. James et al. [2021], S. 56

¹³¹ Vgl. James et al. [2021], S. 57 f.

trainierten Algorithmen vorgenommen. Anhand der Bewertung traditioneller Methoden werden die Ergebnisse verglichen und die Nutzenpotenziale herausgearbeitet, um die Forschungshypothesen abschließend zu beantworten. Für beispielhafte Anwendung des maschinellen Lernens für die Unternehmensbewertung wurden insgesamt drei Algorithmen mit denselben Sätzen an Trainingsdaten trainiert und ausgewertet. In diesem Kapitel wird zunächst auf die Beschaffung und Aufstellung der Trainingsdaten näher eingegangen und anschließend die Machine Learning Algorithmen samt ihren Ergebnissen vorgestellt.

Im Rahmen dieser Arbeit wurden börsennotierte US-amerikanische Unternehmen mit einer Marktkapitalisierung im Bereich von 2 bis 200 Milliarden US-Dollar analysiert. Die betrachtete Stichprobe erstreckt sich somit über das sogenannte Mid Cap- sowie das Large Cap-Segment. Mid Cap-Unternehmen gelten dabei allgemein als wachstumsstarke, etablierte Unternehmen mit einer gewissen Marktdurchdringung, während Large Caps meist als global tätige Konzerne mit stabilen Geschäftsmodellen und hoher Kapitalmarktrelevanz klassifiziert werden.

Die Auswahl dieser Kapitalisierungssegmente erfolgte bewusst, um eine methodisch saubere und analytisch belastbare Datengrundlage zu schaffen. Unternehmen unterhalb der 2-Milliarden-US-Dollar-Schwelle – sogenannte Small oder Micro Caps – weisen oftmals eine erhöhte Kursvolatilität, geringere Liquidität sowie eine eingeschränkte Datenverfügbarkeit auf. Dies erschwert nicht nur die modellbasierte Analyse, sondern kann auch zu statistischen Verzerrungen führen, etwa durch unvollständige Berichterstattung oder illiquide Märkte.¹³² Gleichzeitig wurden Unternehmen oberhalb der 200-Milliarden-US-Dollar-Marke – die sogenannten Mega Caps – gezielt ausgeschlossen, da diese in jüngerer Zeit nicht selten Gegenstand spekulativer Marktbewegungen und mediengetriebener Überbewertungen waren. Prominente Beispiele wie Apple, Tesla oder Nvidia sind in dieser Hinsicht bekannt für stark schwankende Bewertungsniveaus, die weniger aus den Fundamentaldaten als vielmehr aus den sogenannten „exogenen Narrativen“, wie der Wirtschaftsnobelpreisträger Robert J. Shiller in seinem Werk „Narrative Economics: How Stories Go Viral & Drive Major Economic Events eindrucksvoll zeigt, gespeist werden.¹³³

¹³² Vgl. Perridon et al. [2022], S. 296

¹³³ Vgl. Shiller [2019], S. 62 f.

Durch die Fokussierung auf Mid und Large Caps wurde somit ein bewusstes Gleichgewicht zwischen Repräsentativität, Datenqualität und Bewertungsstabilität angestrebt. Diese Unternehmensgruppen zeichnen sich typischerweise durch konsistente Berichterstattung, eine breite Investorenbasis sowie eine hinreichende Historie an Finanz- und Marktdaten aus – zentrale Voraussetzungen für den Einsatz quantitativer Bewertungsverfahren wie dem Discounted-Cashflow-Modell oder datengetriebenen Regressionsmodellen. Die Auswahl reflektiert zudem die gängige Forschungspraxis im Bereich empirischer Kapitalmarktforschung, die sich häufig auf liquide, börsennotierte Gesellschaften mit hoher Informationsdichte konzentriert, um Vergleichbarkeit und statistische Aussagekraft zu maximieren.¹³⁴

Die Auswahl börsennotierter Unternehmen mit Hauptsitz in den Vereinigten Staaten erfolgte zur Sicherstellung einer konsistenten und vergleichbaren Datengrundlage. Diese Entscheidung basiert primär auf dem Ziel, eine einheitliche Bilanzierungs- und Ausweislogik in den zugrunde liegenden Jahresabschlüssen zu gewährleisten. Die betrachteten Unternehmen bilanzierten sämtlich nach den Vorschriften der United States ‚Generally Accepted Accounting Principles‘ (US-GAAP), was insbesondere im Hinblick auf die Gliederung der Gewinn- und Verlustrechnung sowie der Definition zentraler Kennzahlen von Bedeutung ist.¹³⁵

In diesem Zusammenhang ist die einheitliche Anwendung des Umsatzkostenverfahrens ebenso bedeutend. Im Gegensatz zu internationalen Vergleichsstudien, bei denen Unternehmen aus unterschiedlichen Rechtsräumen sowohl nach dem Gesamtkostenverfahren als auch nach dem Umsatzkostenverfahren bilanzieren, ermöglicht die Fokussierung auf US-GAAP eine größtenteils homogene Darstellung der operativen Leistungskraft. US-GAAP schreiben für kapitalmarktorientierte Unternehmen grundsätzlich die Verwendung des Umsatzkostenverfahrens vor, bei dem die betrieblichen Aufwendungen nach Funktionsbereichen (z. B. Herstellung, Vertrieb, Verwaltung) ausgewiesen werden.¹³⁶ Diese systematische Gliederung erlaubt eine methodisch saubere Abgrenzung des Bruttogewinns („Gross Profit“) als Differenz zwischen Umsatzerlösen und Herstellungskosten, die wiederum als belastbare Grundlage

¹³⁴ Vgl. Damodaran [2012], S. 89

¹³⁵ Vgl. Küting [2015], S. 72 ff.

¹³⁶ Vgl. Pronobis [2022], S. 97 ff.

für nachgelagerte Analysen und das spätere „Feature Engineering“ im Rahmen datenbasierter Bewertungsmodelle dient.¹³⁷

Neben dieser Konsistenz bietet die Fokussierung auf US-Unternehmen weitere Vorteile im Hinblick auf die Datenverfügbarkeit und -qualität. Aufgrund regulatorischer Anforderungen an börsennotierte Gesellschaften – wie dem Securities Exchange Act und der Überwachung durch die U.S. ‚Securities and Exchange Commission‘ (SEC) – liegt für US-Firmen in der Regel eine vollständige, vergleichbare und strukturiert zugängliche Historie an Finanzkennzahlen vor.¹³⁸ Darüber hinaus ist die Verfügbarkeit Analystenschätzungen und Marktindikatoren für den US-Kapitalmarkt signifikant höher als für viele andere Regionen, was die externe Validierung der Modelle zusätzlich erleichtert.¹³⁹

Durch die Fokussierung auf einheitlich bilanzierende US-Unternehmen wird somit sichergestellt, dass sämtliche für die Modellierung relevanten Inputdaten – insbesondere Umsatz, Bruttogewinn, operative Marge, Investitionen und Working Capital – auf einer vergleichbaren Rechnungslegungssystematik beruhen. Dies minimiert das Risiko struktureller Verzerrungen durch bilanzpolitische Unterschiede und stärkt die Aussagekraft der datenbasierten Bewertungsergebnisse im Vergleich zu klassischen Bewertungsverfahren.

Aus diesem vordefinierten Rahmen wurden konkret insgesamt 490 Unternehmen mit dem Betrachtungszeitraum vom 01.01.2012 bis zum 31.12.2023 ausgewählt, um einen möglichst großen Datensatz zu gewährleisten.¹⁴⁰

Die Trainingsdaten bilden hierbei eine zufällig ausgewählte Menge von rund zwei Dritteln der Gesamtzahl der Unternehmen (331 Unternehmen). Die Testdaten entsprechen daher der Menge von 159 Unternehmen. Der tatsächliche Trainingszeitraum wird auf 01.01.2012 bis zum 31.12.2020 gewählt. Für den abschließenden Test und den Hauptteil bietet sich hierdurch eine Menge disjunkter Unternehmen einschließlich eines disjunkten Zeitraums, wodurch sichergestellt ist, dass der Algorithmus keine Informationen hat, die ein Jahresabschlussprüfer oder Analyst mit traditionellen Bewertungsmethoden nicht hätte haben können.

¹³⁷ Vgl. Drukarczyk & Schüler [2021], S. 161

¹³⁸ Vgl. Exchange Act Reporting and Registration [o.J.]

¹³⁹ Vgl. Valentine [2011], S. 8 ff.

¹⁴⁰ Vgl. Datensatz ‚Total Data.csv‘ im Anhang

Inhaltlich wurden pro Unternehmen eine Vielzahl unternehmensrelevanter Kennzahlen und makroökonomisch relevanter Kennzahlen berücksichtigt um abschließend im Feature Engineering der Algorithmen entsprechend Kürzungen vorzunehmen. Es wurde darauf abgezielt initial möglichst viele Daten bereitzuhalten und erst nach eingehender Analyse im „Feature Engineering“ Prozess entsprechend zu kürzen.

Unter anderem berücksichtigt sind Daten zur Bilanz, Gewinn- und Verlustrechnung, der Kapitalflussrechnung, historischen Börsen und Marktkapitalisierungswerten, sowie makroökonomische Indikatoren, wie Leitzinssatz und Inflation. Eine vollständige Liste aller initial verwandten Daten findet sich im Anhang.¹⁴¹

Angereichert wurden diese Daten durch die jährliche Veränderung in Absoluten und in Prozentzahlen, sowie durch sogenannte „Lag Werte“, also der Anreicherung durch zeitliche Dynamik der einzelnen Kennzahlen. Erst hierdurch können bspw. Erst Konjunkturzyklen vom Modell erkannt werden.¹⁴²

Dieses Vorgehen folgt dem etablierten ‚Kitchen Sink Approach‘ im prädiktiven Modellbau. Initial werden alle verfügbaren finanziellen und makroökonomischen Variablen erfasst, um Informationsverluste zu vermeiden. Erst im Feature-Engineering erfolgt eine systematische Reduktion überflüssiger Prädiktoren, um Overfitting zu verhindern. Die Anreicherung durch absolute/prozentuale Veränderungen und Lag-Werte erfasst zeitliche Dynamiken, die für Finanzprognosen essenziell sind.¹⁴³

Hierdurch ergeben sich pro Unternehmen 732 Datenpunkte pro Jahr, entsprechend für die Trainingsdaten (8 Jahre) und die Gesamtzahl der Unternehmen 331 im Trainingsset (1,938,336 Gesamtdatenpunkte). Für die notwendige Güte des Random Forest Models und Gradient Boosting Modells wird dies als ausreichend erachtet.¹⁴⁴

Die zur Analyse verwendeten Daten wurden vollständig über frei zugängliche Open-Source-Python-Bibliotheken bezogen und weiterverarbeitet. Im Zentrum der Datenbeschaffung standen die Python-Libraries „Financial Toolkit“ und „Finance Database“, beide entwickelt und gepflegt durch den Finanzanalysten und Mathematiker Jeroen Bouma.

¹⁴¹ Vgl. Datenset ‘Total Data.csv’ im Anhang

¹⁴² Vgl. Kuhn & Johnson [2013], S. 148 f.

¹⁴³ Vgl. Kuhn & Johnson [2013], S. 29 f.

¹⁴⁴ Vgl. Géron [2019], S. 24 f.

Die Bibliothek Financial Toolkit bezieht Unternehmenskennzahlen, Bewertungskennzahlen sowie historische Zeitreihen primär über die API des Anbieters ‚Financial Modeling Prep‘ (FMP). Ergänzende Kursdaten und Marktpreise werden automatisiert über Yahoo Finance integriert. Darüber hinaus ermöglicht das Toolkit den Zugriff auf makroökonomische Indikatoren – etwa Leitzinsen, Inflationsraten oder BIP-Werte – über die öffentlich zugängliche Datenbank der Organisation für wirtschaftliche Zusammenarbeit und Entwicklung (OECD). Der Zugriff erfolgt programmgesteuert über standardisierte REST-Schnittstellen innerhalb der Toolkit-Architektur.¹⁴⁵

Die Finance Database stellt eine umfassend gepflegte Sammlung strukturierter Unternehmens- und Marktklassifikationsdaten dar. Sie enthält Angaben zu Börsenplätzen, Sektoren, Regionen und Kapitalisierungsklassen (z. B. Small, Mid, Large Caps). Diese Daten wurden im Rahmen dieser Arbeit insbesondere zur Abgrenzung der analysierten Unternehmen nach Marktkapitalisierungssegmenten verwendet. Die Datenbank selbst wird auf Basis öffentlich zugänglicher Finanzinformationen kuratiert und regelmäßig durch die Community aktualisiert.¹⁴⁶ Beide Bibliotheken sowie der zugehörige Quellcode sind vollständig einsehbar und dokumentiert und auf der Entwicklerplattform GitHub veröffentlicht.¹⁴⁷

Die Transformation und Bereinigung der Daten erfolgte im Rahmen dieser Arbeit mit Hilfe gängiger numerischer Python-Bibliotheken, insbesondere ‚pandas‘ für tabellarische Datenverarbeitung¹⁴⁸ und ‚numpy‘ für numerische Rechenoperationen.¹⁴⁹ Auf Basis dieser Tools wurden unter anderem Lag-Features, gleitende Durchschnitte, prozentuale Veränderungen und andere Inputvariablen generiert, die als Feature Engineering Grundlage für die datenbasierten Bewertungsmodelle dienen.

Die Entwicklung, das Training und die Evaluation der in dieser Arbeit eingesetzten Machine-Learning-Modelle erfolgten unter Verwendung offener, in der wissenschaftlichen Forschung etablierten Python-Bibliotheken. Im Zentrum standen hierbei die bereits in Kapitel 4.4.1 und 4.4.2 erwähnten algorithmischen Ansätze: das

¹⁴⁵ Financial Toolkit: <https://github.com/JerBouma/financial-toolkit> & Finance Database: <https://github.com/JerBouma/FinanceDatabase>

¹⁴⁶ Financial Toolkit: <https://github.com/JerBouma/financial-toolkit> & Finance Database: <https://github.com/JerBouma/FinanceDatabase>

¹⁴⁷ Financial Toolkit: <https://github.com/JerBouma/financial-toolkit> & Finance Database: <https://github.com/JerBouma/FinanceDatabase>

¹⁴⁸ Vgl. pandas GitHub Repository <https://github.com/pandas-dev/pandas>

¹⁴⁹ Vgl. Numpy GitHub Repository <https://github.com/numpy/numpy>

Random Forest Regressionsmodell sowie ein Gradient Boosting Regressor, jeweils in ihrer Standardimplementierung.

Für das Random Forest Modell wurde die Implementierung aus der Bibliothek `scikit-learn`¹⁵⁰ verwendet. Random Forests gelten aufgrund ihrer geringen Anfälligkeit für Overfitting und ihrer hohen Interpretierbarkeit als Standardansatz in der Regressionsanalyse strukturierter Daten.¹⁵¹

Für das Gradient Boosting Modell wurde auf die leistungsfähige Open-Source-Bibliothek `scikit-learn`¹⁵² und der Funktion `GradientBoostingRegressor` zurückgegriffen. `GradientBoostingRegressor` verwendet ein iteratives Boosting-Verfahren, bei dem schwache Entscheidungsbäume sequenziell trainiert und Fehler aus vorherigen Iterationen korrigiert werden. Aufgrund ihrer Effizienz und Prognosegüte wird `GradientBoostingRegressor` heute vielfach in praxisnahen Machine-Learning-Projekten und Data-Science-Wettbewerben eingesetzt.¹⁵³

Die Trainings- und Testdaten wurden zunächst mittels der Bibliotheken `pandas` und `numpy` für die Modellerstellung vorbereitet. Dabei erfolgte eine systematische Auftrennung in Trainings- und Testsegmente, gefolgt von der Auswahl relevanter Features, der Zielvariablen sowie der Transformation logarithmierter Zielgrößen. Nach Abschluss des Trainings wurden die Modelle zur Vorhersage von Unternehmenswerten auf einem separaten Testdatensatz eingesetzt und anhand etablierter Regressionsmetriken – insbesondere dem Mean Absolute Percentage Error (*MAPE*) und dem Bestimmtheitsmaß R^2 – evaluiert.

Sowohl Random Forest als auch Gradient Boosting bieten, wie bereits erläutert, eine Möglichkeit zur Ermittlung der Feature-Wichtigkeit, die sogenannte Feature Importance, welche im Rahmen dieser Arbeit zur Auswahl und Bewertung relevanter Inputvariablen herangezogen wurde und in den jeweiligen Kapiteln „Feature Engineering“ näher erläutert werden.

¹⁵⁰ Pedregosa, Fabian et al. [2011], S. 2825-2830

¹⁵¹ Vgl. Breiman [2001], S. 5-32

¹⁵² Pedregosa, Fabian et al. [2011], S. 2825-2830

¹⁵³ Vgl. Géron [2019], S. 312

6 Design und Implementation selbst entwickelter

Bewertungsalgorithmen

6.1 Entwicklung des Algorithmus für Random Forest Methode

Basierend auf dem in Kapitel 4.4.1 vorgestellten Random Forest Algorithmus wurde mittels der Bibliothek scikit-learn ein Python Skript entworfen, das die Trainingsdaten verarbeitet und eine Random Forest Regression ausführt.

Zur Modellierung wurde ein Random Forest Regressor mit 400 Entscheidungsbäumen (*n_estimators=400*) verwendet. Die maximale Tiefe der Bäume wurde auf *None* (*max_depth=None*) gesetzt, wodurch das Wachstum der einzelnen Bäume nicht begrenzt und sie so bis zur maximalen Reinheit der Blätter weiter aufgeteilt werden. Um die Parallelisierung der Rechenprozesse zu maximieren, wurde der Parameter *n_jobs=-1* gesetzt, wodurch sämtliche logischen Prozessoren des Systems genutzt wurden. Die Wahl einer höheren Baumanzahl erfolgte bewusst, um robustere Schätzungen der Feature-Wichtigkeit zu erzielen. Sie basiert auf einer kombinatorischen Optimierung der Modellgenauigkeit, wie in Kapitel 4.7 dargestellt. Zur Sicherstellung der Reproduzierbarkeit wurde *random_state=42* festgelegt.

Zur Verbesserung der Modellinterpretierbarkeit und zur Reduktion potenziell irrelevanter oder redundanter Prädiktoren wurde eine Merkmalsselektion auf Basis der „Feature Importance“ des trainierten Random-Forest-Modells durchgeführt.

Dieses Vorgehen basiert auf etablierten Ansätzen in der Literatur, wie es auch in Kapitel 4.5.1 dargestellt wurde. Random Forests liefern von Natur aus eine robuste Metrik zur Variablenbedeutung, welche sich empirisch bewährt hat. Auch Hastie et al. empfehlen die Eliminierung wenig beitragender Merkmale, um Overfitting zu vermeiden und die Modellkomplexität zu reduzieren.¹⁵⁴ Darüber hinaus entspricht die Merkmalsreduktion den Grundsätzen der Datenqualitätsstandards nach ISO/IEC 25024 hinsichtlich Verständlichkeit und Effizienz.¹⁵⁵

Für den tatsächlichen Unternehmenswert wurde gemäß der Effizienzmarkthypothese von Eugene F. Fama, wie in Kapitel 3.5.6 beschrieben, die Marktkapitalisierung als Proxy verwendet und zudem durch eine logarithmierte Version ersetzt, um eine

¹⁵⁴ Vgl. Hastie et al. [2009], S 243 ff.

¹⁵⁵ Vgl. ISO/IEC [2015], o.S.

Standardisierung herzustellen. Dies ist auch in der Wissenschaft ein Standardvorgehen, denn die logarithmische Transformation von stark schief verteilten Größen wie der Marktkapitalisierung ermöglicht eine Annäherung an die Normalverteilung, verbessert die Vergleichbarkeit zwischen Unternehmen unterschiedlicher Größenordnungen und erhöht die Robustheit statistischer Analysen.¹⁵⁶

Als Feature Werte wurden zunächst alle erhobenen Unternehmensfinanzzahlen aus der Bilanz, Gewinn- und Verlustrechnung, sowie berechnete Kennzahlen genutzt und durch makroökonomische Werte ergänzt.¹⁵⁷ Zusätzlich wurden Werte, die mathematisch oder rechnerisch im direkten Zusammenhang mit der Marktkapitalisierung stehen, exkludiert. Dies umfasst unter anderem die Aktienkurse, Enterprise Value to EBITDA sowie die Marktkapitalisierung selbst. Eine vollständige Liste findet sich im Anhang.¹⁵⁸ Im Rahmen des nun folgenden „Feature Engineerings“ wurden Features, die offenbar keine bzw. eine zu geringe Relevanz für das ML-Modell liefern, für das finale Modell exkludiert.

Als Schwellenwert für die Relevanz wurde ein Mindestbeitrag von 1 % zur Gesamtwichtigkeit definiert. Merkmale, deren relative Bedeutung unterhalb dieser Schwelle lagen, wurden aus dem finalen Trainingsdatensatz entfernt.¹⁵⁹ Zunächst wurde das Modell ohne Schwellenwert berechnet, wodurch sich die Wichtigkeit aller Features ausrechnen ließ.

Die folgende Abbildung zeigt exemplarisch die vom Modell ausgegebene Liste der Top Features mit berechneter Relevanz (*Importance*) > 0,1%.

¹⁵⁶ Vgl. Damodaran [2012], S. 112 ff.

¹⁵⁷ Siehe Trainingsdaten Set im Anhang

¹⁵⁸ Siehe exkludiertes Feature Set im Anhang

¹⁵⁹ Vgl Skript „RF Engineering.ipynb“ im Anhang

	Feature	Importance
51	EBITDA	0.223668
101	Operating Income	0.102544
21	Cash Flow from Operations	0.075617
96	Operating Cash Flow	0.070327
245	Income Before Tax_lag_1	0.040614
91	Net Income	0.038631
56	Earnings Yield	0.037256
140	Stock Based Compensation	0.025759
65	Free Cash Flow	0.024914
73	Income Before Tax	0.023886
263	Net Income_lag_1	0.023442
312	Stock Based Compensation_lag_1	0.018181
228	Earnings Yield_lag_1	0.015767
243	Gross Profit_lag_1	0.014522
318	Total Current Liabilities_lag_1	0.012990
66	Free Cash Flow Yield	0.012651
31	Cumulative Return	0.009365
193	Cash Flow from Operations_lag_1	0.009157
71	Gross Profit	0.009050
24	Cash and Short Term Investments	0.007906
268	Operating Cash Flow_lag_1	0.007658
158	Weighted Average Shares Diluted	0.007447
223	EBITDA_lag_1	0.006860
237	Free Cash Flow_lag_1	0.006549
273	Operating Income_lag_1	0.005853
155	Total Shareholder Equity	0.005495
148	Total Equity	0.005023
330	Weighted Average Shares Diluted_lag_1	0.004780
151	Total Liabilities and Equity	0.003602
157	Weighted Average Shares	0.003356
238	Free Cash Flow Yield_lag_1	0.003103
145	Total Current Assets	0.002916
152	Total Liabilities and Shareholder Equity	0.002869
323	Total Liabilities and Equity_lag_1	0.002831
317	Total Current Assets_lag_1	0.002549
329	Weighted Average Shares_lag_1	0.002473
324	Total Liabilities and Shareholder Equity_lag_1	0.002443
144	Total Assets	0.002005
63	Fixed Assets	0.002003
375	Cumulative Return_delta	0.002002
44	Depreciation and Amortization	0.001979

Abbildung 11: Top Features Random Forests und berechnete Importance zur Bewertung der Merkmalsrelevanz mit relativer Bedeutung > 0,1%
(Quelle: Random Forest Modell auf Basis ‚Training Data.csv‘)

Zusätzlich zu der reinen Wichtigkeit besteht ebenfalls eine Korrelation zwischen den einzelnen Features. Wie auch aus betriebswirtschaftlicher Perspektive sehr direkt ersichtlich, hängen bspw. die Features „EBITDA“ und „Operating Income“ eng zusammen, da ersteres eine Abwandlung des Operating Incomes ohne Abschreibung auf immaterielle und materielle Werte darstellt.

Aus diesem Grund wurde zudem eine Korrelationsanalyse erstellt. Werte, die eine Korrelation von über 90 % aufweisen, wurden eliminiert, wobei der erste Wert des Korrelationspaares erhalten blieb. Ein Grenzwert von zwischen 75 % und 90 % hat sich

in der Praxis etabliert.¹⁶⁰ In dieser Arbeit wurde ein Wert von 90 % gewählt, um möglichst viele Features noch zu erhalten.

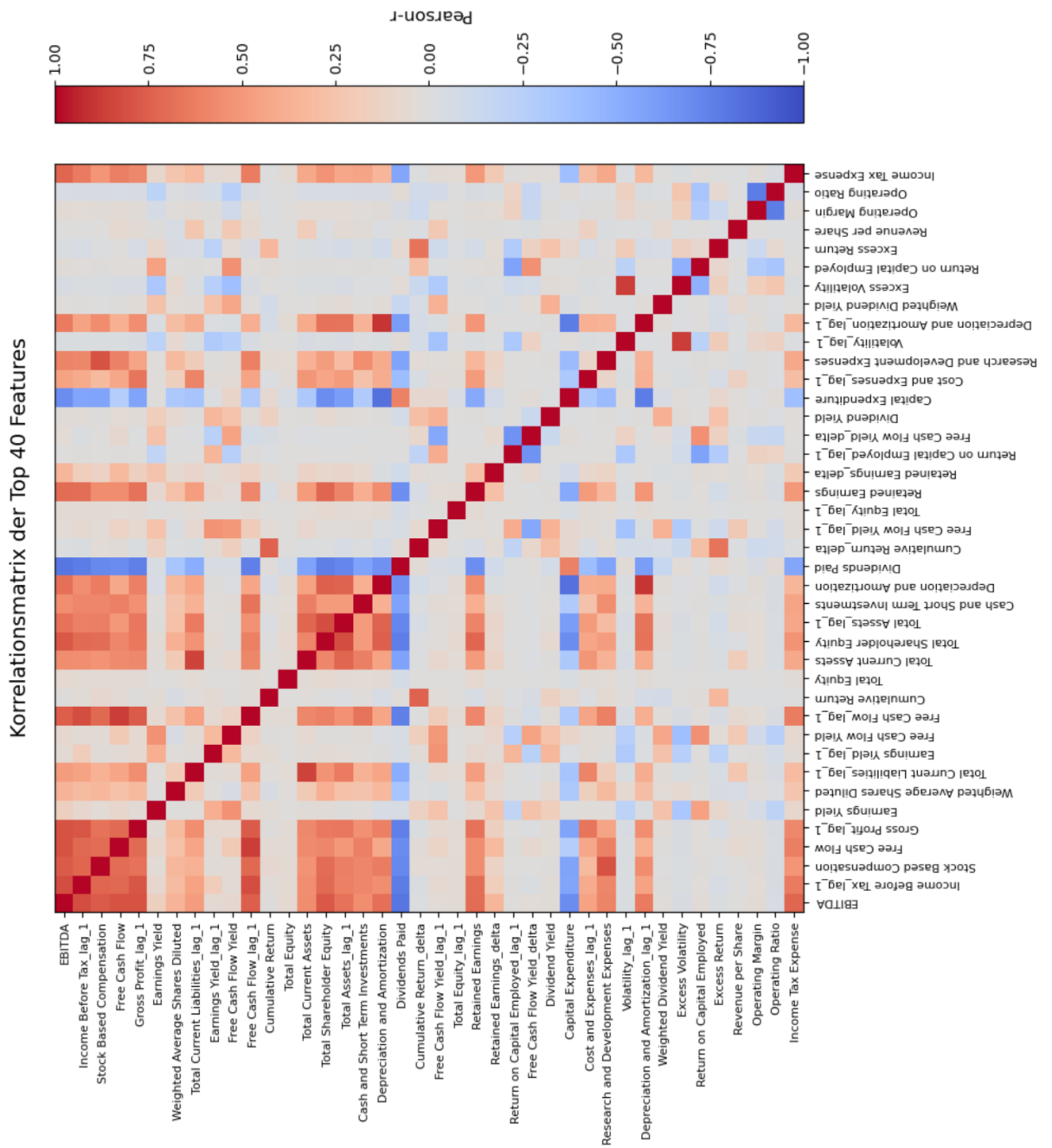


Abbildung 12: Korrelationsmatrix Features Random Forest
(Quelle: Eigene Darstellung)

Anschließend wurde das Random Forest Modell erneut mit den Features exklusive der Korrelationswerte trainiert und nun Features unterhalb der avisierten Schwelle von 1 %

¹⁶⁰ Vgl. Kuhn & Johnson [2021], S.133 ff.

eliminiert. Daraus ergibt sich die in der folgenden Abbildung festgestellte Feature Liste samt neu berechneter Feature Wichtigkeit:

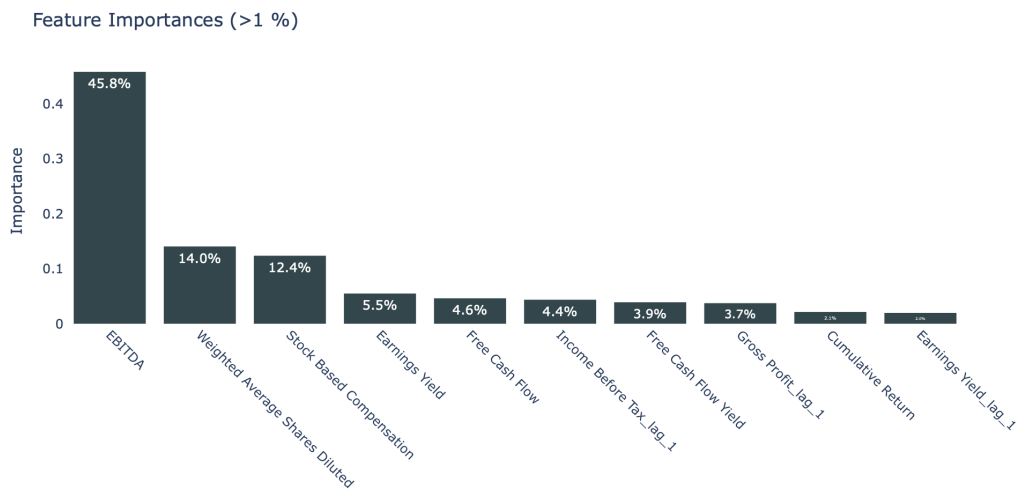


Abbildung 13 Finale Feature Importance Random Forest
(Quelle Eigene Darstellung)

6.2 Entwicklung des Algorithmus für Gradient Boosting Methode

Analog zur Vorgehensweise beim Random-Forest-Modell (vgl. Kapitel 6.1) wurde auch für das Gradient Boosting Verfahren ein entsprechendes Python-Skript mit der scikit-learn-Bibliothek entwickelt. Zur Anwendung kam ein ‚GradientBoostingRegressor‘, wobei ebenfalls der Parameter *random_state=42* gesetzt wurde, um die Reproduzierbarkeit sicherzustellen.

Im Unterschied zum Random Forest, bei dem mehrere unabhängige Entscheidungsbäume parallel trainiert werden, basiert Gradient Boosting auf einem sequentiellen Lernprozess, bei dem jeweils neue Bäume auf den Fehlern der vorherigen Bäume aufbauen. Die verwendete Konfiguration orientierte sich dabei an Best Practices in der einschlägigen Literatur mit 300 Basisbäumen (*n_estimators=300*) und einer moderaten Lernrate (*learning_rate=0.1*), um eine gute Balance zwischen Modellkomplexität und Generalisierbarkeit zu erreichen.¹⁶¹

Auch für das Gradient Boosting Modell wurde die Feature-Auswahl in mehreren Schritten durchgeführt. Zunächst wurde das Modell mit sämtlichen zur Verfügung stehenden Features trainiert – dabei kamen dieselben Eingangsvariablen wie im Random Forest Modell zum Einsatz: Finanzkennzahlen aus Bilanz und GuV, berechnete KPIs sowie ausgewählte makroökonomische Variablen. Werte mit direktem mathematischem

¹⁶¹ Vgl. Friedman [2001], S. 1189-1232

Zusammenhang zur Zielgröße Marktkapitalisierung (z. B. Aktienkurs oder EV/EBITDA) wurden analog ausgeschlossen.¹⁶²

Die Bewertung der Merkmalsrelevanz erfolgte hier über die im Modell integrierte `feature_importances_`-Methode, welche auf der Reduktion des Loss-Funktionalwerts basiert. Die initial berechneten Feature-Wichtigkeiten dienten wiederum als Grundlage für die erste Schwellenwertprüfung (1 %-Grenze), wie bereits im vorhergehenden Abschnitt beschrieben.

	Feature	Importance
51	EBITDA	0.195725
101	Operating Income	0.120021
21	Cash Flow from Operations	0.059707
245	Income Before Tax_lag_1	0.059323
91	Net Income	0.059208
56	Earnings Yield	0.050728
140	Stock Based Compensation	0.046075
268	Operating Cash Flow_lag_1	0.030199
155	Total Shareholder Equity	0.027742
65	Free Cash Flow	0.027711
31	Cumulative Return	0.024533
312	Stock Based Compensation_lag_1	0.024069
228	Earnings Yield_lag_1	0.023917
158	Weighted Average Shares Diluted	0.023141
263	Net Income_lag_1	0.022561
318	Total Current Liabilities_lag_1	0.020623
66	Free Cash Flow Yield	0.017416
73	Income Before Tax	0.017297
273	Operating Income_lag_1	0.016841
148	Total Equity	0.013959
152	Total Liabilities and Shareholder Equity	0.009398
243	Gross Profit_lag_1	0.007829
24	Cash and Short Term Investments	0.007464
71	Gross Profit	0.004991
159	Weighted Dividend Yield	0.004481
237	Free Cash Flow_lag_1	0.003742
317	Total Current Assets_lag_1	0.003449
49	Dividends Paid	0.003344
330	Weighted Average Shares Diluted_lag_1	0.003299
157	Weighted Average Shares	0.003275

Abbildung 14: Feature und berechnete Importance zur Bewertung der Merkmalsrelevanz mit relativer Bedeutung > 0,1%

(Quelle: Gradient Boosting Modell auf Basis ‚Training Data.csv‘)

Analog zum Random-Forest-Modell wurde zunächst eine Korrelationsanalyse durchgeführt, um stark redundante Merkmale ($r > 0.9$) zu identifizieren und zu eliminieren. In jedem hochkorrelierten Merkmal-Paar wurde das jeweils zweitplatzierte Merkmal entfernt.

¹⁶² Siehe exkludiertes Feature Set im Anhang

Nach der Bereinigung wurde das Gradient Boosting Modell erneut trainiert und eine finale Feature-Auswahl mit der Schwellenwertprüfung von 1% durchgeführt.

Die berechneten Feature-Wichtigkeiten unterscheiden sich leicht vom Random Forest Modell, bestätigen jedoch größtenteils die dort getroffenen Annahmen zur Relevanz einzelner Finanzkennzahlen, was die Robustheit der Merkmalsauswahl unterstreicht. Eine exemplarische Übersicht der wichtigsten Merkmale ist im Anhang dokumentiert.

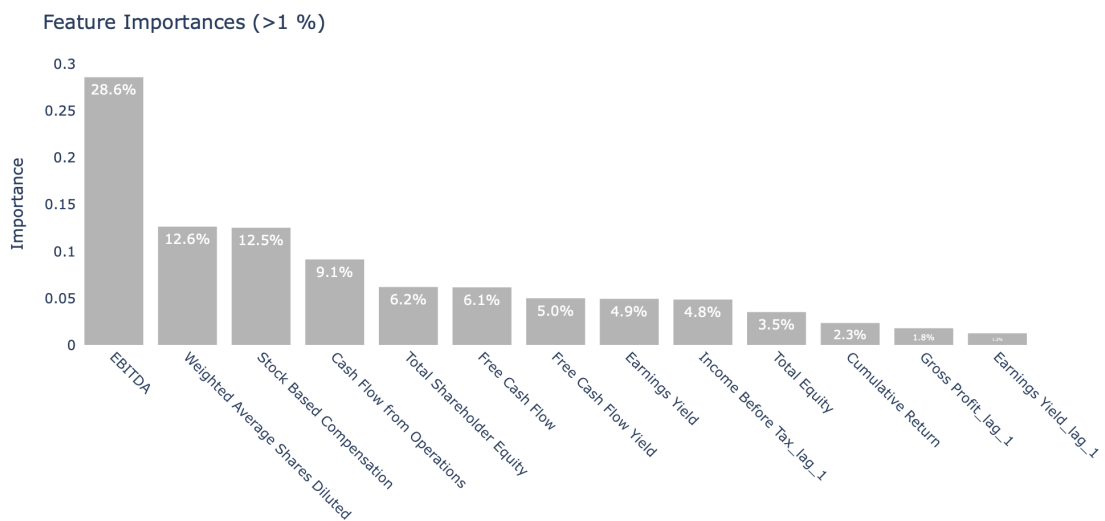


Abbildung 15: Feature Importance Gradient Boosting
(Quelle: Eigene Darstellung)

6.3 Anwendung traditioneller Bewertung

Analog zu den datengetriebenen Verfahren Random Forest und Gradient Boosting wurde ein Discounted-Cashflow-Modell implementiert, um den Eigenkapitalwert von Unternehmen auf Basis finanzwirtschaftlicher Fundamentaldaten zu schätzen. Im vorliegenden Fall wurde ein Multiple-basierter DCF-Ansatz gewählt, der, wie auch in Kapitel 3.5.5 erläutert wurde, auf dem Equity Ansatz basiert, aber anstelle der Ewigen Rente ein Multiple aus dem historischen Unternehmenswert zu EBITDA Verhältnis kalkuliert wurde.

Es wird ein pauschaler Diskontierungssatz von 11% verwendet, den typischen Eigenkapitalkosten kapitalmarktorientierter Unternehmen in den USA. Die Fachliteratur zeigt, dass Eigenkapitalkosten-Spannbreiten von 11% bis 20% für Unternehmensbewertungen als angemessen betrachtet werden. Für etablierte Mid- und Large-Cap-Unternehmen liegt dieser Wert häufig am unteren Ende dieser Spanne.¹⁶³

¹⁶³ Vgl. Calhoun & Harkins [2021], o.S.

Die Standardisierung der Parameter erhöht nicht nur die methodische Vergleichbarkeit mit den ML-basierten Bewertungsmodellen, sondern reduziert auch den Einfluss subjektiver Schätzentscheidungen. Auf diese Weise dient das genutzte DCF-Modell in dieser Arbeit als objektiven Vergleich für die Bewertungsgüte datengetriebener Verfahren und ermöglicht eine faire methodische Gegenüberstellung. An dieser Stelle sei nochmals erwähnt, dass das klassische DCF-Modell in der Theorie genauere Angaben erwartet, dies ist aber in der Praxis und vor allem in einem Mastentest für ein objektiven Vergleich nicht darstellbar. Die Multiple-basierte Methode ermöglicht es Analysten, die Bewertung an den aktuellen Marktgegebenheiten und vergleichbaren Transaktionen auszurichten, wodurch eine marktnahe Bewertung des Terminal Value erreicht wird und somit gewissermaßen den Kreis zur genutzten Marktkapitalisierung als Proxy zur Unternehmensbewertung schließt.

Die Prognose erfolgte über einen fünfjährigen expliziten Forecast-Zeitraum (2021–2025), wobei für die Jahre ab 2021 geschätzte Free Cash-Flows to Equity verwendet wurden. Die Free Cash-Flows to Equity für die Jahre 2021–2025 werden aus den historischen Werten (bis 2020) mittels eines gleitenden Durchschnitts (*window* = 5) extrapoliert. Um übermäßige Ausreißer zu vermeiden, wird jede Jahresprognose auf $\pm 25\%$ des Vorjahreswerts begrenzt (*use_clip* = *True* & *clip_pct* = 0.25).

Die diskontierten Cashflows bis Ende 2023 sowie ein am Ende des Prognosehorizonts berechneter Terminal Value (basierend auf dem historischen EV/EBITDA-Multiple und einer geschätzten Multiplikator-Wachstumsrate) werden zum Stichtag 31.12.2020 abgezinst. Der resultierende Barwert bildet die Grundlage für ein iteratives Roll-Forward: Jährliches Aufzinsen um den gewählten Discount-Satz (*discount_rate* = 0.10) und Abzug des im Vorjahr prognostizierten *FCFE* simulieren die Veränderung der Marktkapitalisierung für 2021, 2022 und 2023.

Die eigentliche Bewertung erfolgte durch die Rückdiskontierung der Cashflows auf das die Jahre 2021, 2022 und 2023.

Abweichungen zwischen Prognose und Marktwert wurden in Form absoluter und relativer Fehlerkennzahlen dokumentiert, sowie einer Berechnung von R^2 , *MAE* sowie *MAPE* analog zum Random Forest oder Gradient Boosting Model, um die Prognosequalität des DCF-Modells im Vergleich zu datengetriebenen Verfahren zu bewerten.

7 Ergebnisse

Im Folgenden werden die Resultate der Unternehmensbewertung mithilfe des DCF-Modells sowie der beiden Machine-Learning-Modelle (Random Forest und Gradient Boosting) tabellarisch und grafisch dargestellt. Die Modelle wurden, wie erläutert, auf Basis historischer Finanzdaten trainiert und auf die außerhalb der Trainingsdaten befindliche Periode 2021–2023 angewendet (Trainingsdaten umfassen 2012–2020). Die Prognosegüte der Verfahren wurde anhand gängiger Regressionsmetriken evaluiert, insbesondere dem Bestimmtheitsmaß R^2 sowie Fehlermessgrößen wie Mean Absolute Error (MAE), Mean Absolute Percentage Error ($MAPE$), sowie Weighted Mean Absolute Percentage Error ($WMAPE$). Darüber hinaus wurden zur Beurteilung der Abweichungen sowohl durchschnittliche als auch mittlere (Median-) Fehler sowie die Verteilung der Fehler analysiert. Die folgende Tabelle gibt einen Überblick über die erzielten Kennzahlen, gemittelt über die Prognosejahre.

Modell	R^2 (erkl. Varianz)	MAPE (Abw. in %)	MAE [Mrd. USD]	WMAPE (Abw. in %)
Random Forest	0,858	29,0 %	4,25	27,9 %
Gradient Boosting	0,912	22,4 %	3,45	22,7 %
DCF-Modell	0,708	40,5 %	5,56	36,5 %

Tabelle 3: Vergleich der drei Modelle DCF, Random Forest und Gradient Boosting hinsichtlich der Prognosegüte für Unternehmensbewertungen
(Quelle: eigene Darstellung)

Zuallererst zeigt sich, dass alle Modelle durchschnittlich eine Überbewertung vornehmen. Ebenso ist ein deutlicher Leistungsunterschied zwischen den datengetriebenen ML-Verfahren und dem DCF-Ansatz zu erkennen. Beide ML-Modelle weisen über alle Testfälle hinweg eine erheblich höhere Vorhersagegüte auf als das DCF-Modell. So erreichte der Random-Forest-Ansatz durchschnittlich ein R^2 von rund $0,86$, während das Gradient-Boosting-Modell mit ca. $0,91$ sogar noch höher lag. Demgegenüber fiel das Bestimmtheitsmaß des DCF-Verfahrens mit etwa $0,71$ deutlich geringer aus. Mit anderen Worten konnten die Machine-Learning-Modelle einen viel größeren Anteil der Varianz der tatsächlichen Unternehmenswerte erklären als das vergleichsweise schwächere DCF-Modell. Dieses Muster zeigte sich konsistent in allen Testdurchläufen: die R^2 -Werte der ML-Modelle lagen in jedem Testjahr zwischen $0,84$ und $0,92$, wohingegen das DCF maximal etwa $0,76$ erreichte, wie auch folgende Abbildung zeigt.

Bestimmtheitsmaß (R^2) pro Jahr



Abbildung 16: Bestimmtheitsmaß R^2 pro Jahr pro Modell
(Quelle: eigene Darstellung)

Auch bei den Fehlerkennzahlen erzielten Random Forest und Gradient Boosting deutlich günstigere Resultate als der DCF-Ansatz. Die durchschnittliche prozentuale Abweichung des geschätzten vom tatsächlichen Unternehmenswert – gemessen als *MAPE* – betrug beim Random Forest rund 25–33 % (über die verschiedenen Testjahre), im Mittel ca. 29 %, und beim Gradient Boosting lediglich etwa 20–24 %, im Mittel ca. 22 %. Damit lagen die ML-Modelle in der Größenordnung nur etwa halb so hoch wie der entsprechende Fehler des DCF-Modells, das durchschnittlich eine *MAPE* von 40 % aufwies.

MAPE pro Jahr

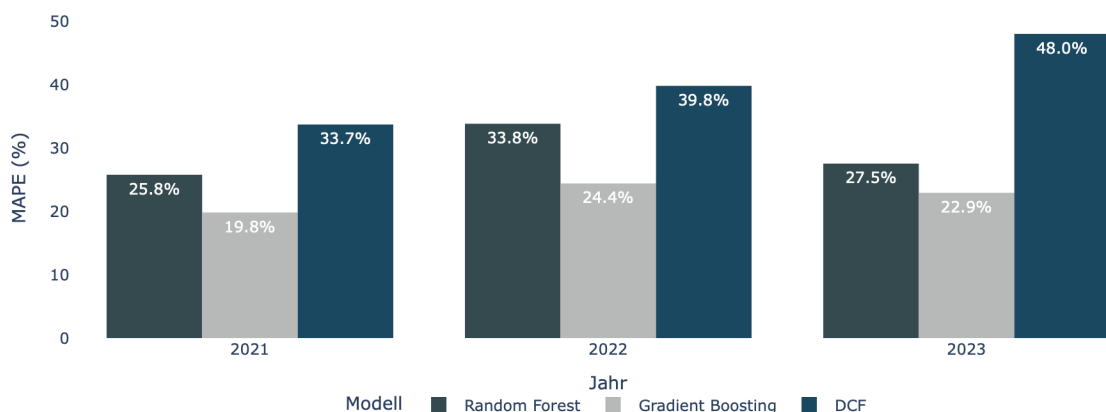


Abbildung 17: *MAPE* pro Jahr pro Modell
(Quelle: eigene Darstellung)

Ähnliche Differenzen zeigen sich für den gewichteten mittleren Fehler (*WMAPE*), der bei Random Forest und Gradient Boosting um 23–28 % schwankte, während er für DCF knapp 36 % betrug. Die geringeren *MAPE/WMAPE*-Werte der ML-Modelle deuten

darauf hin, dass deren Schätzungen im Schnitt wesentlich näher am echten Marktwert lagen als die auf dem DCF basierenden Bewertungen. Darüber hinaus ist der absolute Vorhersagefehler in monetären Einheiten bei den ML-Methoden deutlich geringer.

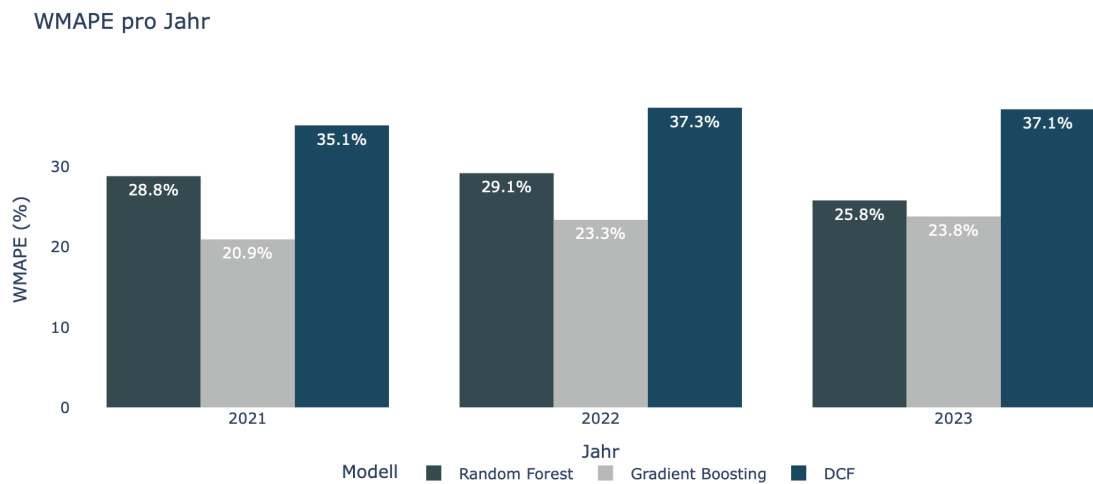


Abbildung 18: *WMAPE* pro Jahr pro Modell
(Quelle: eigene Darstellung)

Der Mean Absolute Error (*MAE*) des Random-Forest-Modells betrug etwa 4,1–4,6 Milliarden USD pro Unternehmen (durchschnittlich ca. 4,25 Mrd. USD), beim Gradient Boosting 3,3–3,7 Mrd. USD (Mittelwert ca. 3,45 Mrd. USD). Demgegenüber lag der MAE des DCF-Ansatzes bei 5,6 Mrd. USD. Mit anderen Worten betrug die durchschnittliche Abweichung der durch das DCF-Verfahren berechneten Unternehmenswerte vom tatsächlichen Marktwert rund 5,6 Milliarden US-Dollar, während die Fehler der ML-Modelle typischerweise um mehrere Milliarden niedriger waren.

Es wird deutlich, dass Gradient Boosting die insgesamt beste Prognosequalität erzielte (höchstes R^2 , niedrigste Fehlermaße), gefolgt vom Random Forest. Das DCF-Modell schnitt in allen Metriken am schwächsten ab. Diese Ergebnisse sind rein deskriptiv zu verstehen; mögliche Ursachen und Implikationen werden im anschließenden Kapitel 8 diskutiert.

Neben den Mittelwerten empfiehlt sich ein Blick auf die Streuung der Fehler, wie in der folgenden Abbildung dargestellt.

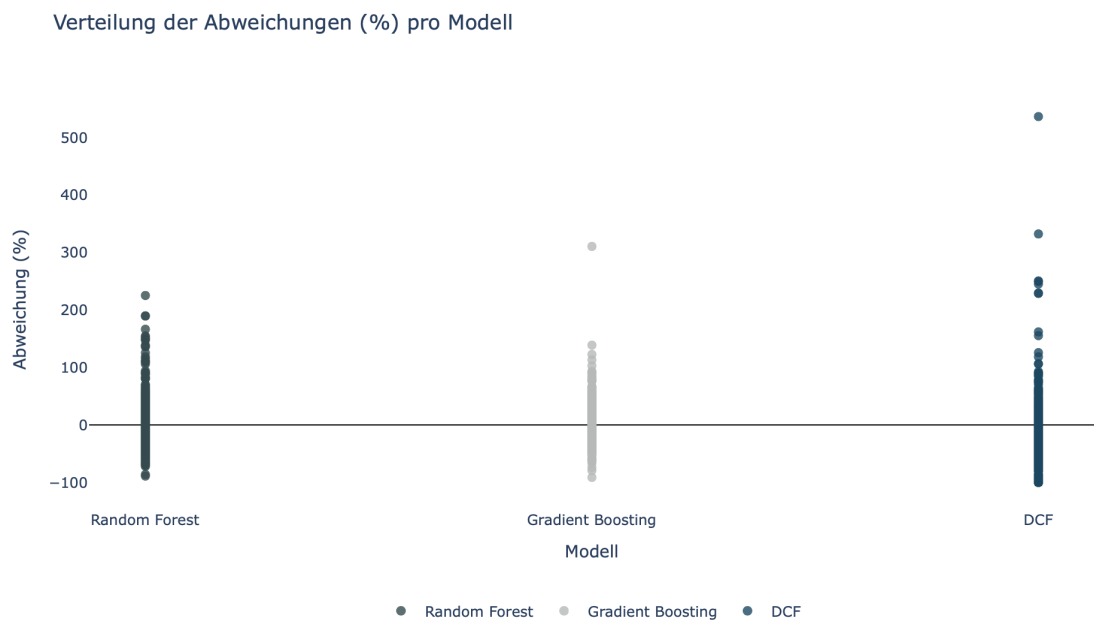


Abbildung 19: Verteilung der %-Abweichungen pro Modell
(Quelle: eigene Darstellung)

Beide ML-Verfahren zeigten nicht nur geringere Durchschnittsabweichungen, sondern auch weniger extreme Ausreißer als das DCF. Die maximal beobachtete prozentuale Abweichung lag beim DCF bei über 536 % (in einem Extremfall verfünffachte das DCF-Modell den wahren Wert), während die größten Fehler der ML-Modelle bei etwa 225 % (Random Forest) bzw. 310 % (Gradient Boosting) lagen.

8 Diskussion

In diesem Abschnitt werden die theoretischen und praktischen Nutzenpotenziale von Machine-Learning-Verfahren (insbesondere Random Forests und Gradient Boosting) im Kontext der Forschungsfrage analysiert, inwiefern Verfahren des maschinellen Lernens Random Forest und Gradient Boosting einen betriebswirtschaftlich relevanten Nutzen gegenüber der klassischen Methode der Unternehmensbewertung dem Discounted-Cashflow-Modell bieten. Dabei wird auch auf die Ergebnisse des vorangegangenen Kapitels der eigenen Modellentwicklung und dem Discounted Cashflow Modell Bezug genommen.

Zunächst werden die potenziellen Mehrwerte moderner ML-Methoden gegenüber klassischen Bewertungsansätzen herausgearbeitet. Im Anschluss erfolgt eine kritische Diskussion struktureller Schwächen beider Verfahren – etwa hinsichtlich Annahmesensitivität und begrenzter Skalierbarkeit traditioneller Verfahren und die Black-Box-Problematik mancher ML-Modelle. Zudem werden aktuelle Erkenntnisse und Veröffentlichungen einbezogen, um die Praxisrelevanz der Thematik zu untermauern und

sowohl Nutzenpotenziale als auch Limitationen von ML in der Unternehmensbewertung aufzuzeigen.

Aus theoretischer Sicht bieten ML-Algorithmen die Möglichkeit, komplexe nicht-lineare Zusammenhänge in Unternehmensdaten aufzudecken, die mit traditionellen Bewertungsmodellen schwer abzubilden sind. Während klassische Methoden wie DCF auf wenigen zentralen Inputgrößen und einfachen linearen Verknüpfungen beruhen, können Machine-Learning-Modelle eine Fülle von Finanzkennzahlen, Marktdaten und sogar “weichen” Faktoren (z. B. ESG-Ratings oder Medienberichterstattung) simultan analysieren. Dadurch ist es möglich, Muster und Werttreiber zu identifizieren, die in klassischen Modellen unentdeckt blieben. Random Forests etwa ziehen zahlreiche Entscheidungsbäume zu Rate und können damit hochdimensionalen Daten gerecht werden, ohne dass explizite Annahmen über die Zusammenhänge der Variablen getroffen werden müssen. Gradient Boosting-Modelle lernen sequentiell aus den Fehlern vorheriger Bäume und können so schrittweise auch subtile Einflüsse erfassen. Theoretisch führt diese datengetriebene Natur dazu, dass ML-Modelle dynamischere und adaptivere Bewertungsansätze ermöglichen: Sie passen sich an beobachtete empirische Zusammenhänge an, statt wie DCF auf festen und teils unsicheren Prämissen aufzubauen. Zudem erlaubt ML den Einbezug historischer Muster – etwa indem aus vergangenen Marktzyklen oder Krisen gelernt wird, was für zukünftige Bewertungen relevant sein könnte. Dies ist ein bedeutender Vorteil gegenüber traditionellen Verfahren, die zukünftige Entwicklungen primär durch subjektive Schätzungen abbilden. Insgesamt ergibt sich das Potenzial, dass ML-basierte Bewertungsmodelle realistischere und aktuellere Unternehmenswerte liefern, sofern ausreichende und qualitativ hochwertige Daten vorliegen. Diese theoretischen Überlegungen legen nahe, dass maschinelles Lernen dazu beitragen kann, Bewertungsmodelle präziser, umfassender und evidenzbasierter zu machen.

Die praktische Anwendung der Modelle in dieser Arbeit zeigt klar, dass datengetriebene Modelle (Random Forest, Gradient Boosting) im Vergleich zum vereinfachten DCF-Modell eine deutlich höhere Prognosegüte erreichen. Beide ML-Modelle erzielen über die Testjahre (2021–2023) hinweg signifikant höhere Bestimmtheitsmaße (R^2) und geringere Fehlerkennzahlen (MAE , $MAPE$) als das DCF. So liegen in unseren Ergebnissen die R^2 -Werte der Random-Forest-Modelle meist um $0,85$ – $0,88$ und der Gradient-Boosting-Modelle um $0,90$ – $0,92$, während das DCF im gleichen Datensatz nur auf etwa $0,64$ – $0,76$ kommt. Dies entspricht ein deutlich geringerer mittlerer prozentualer

Fehler (*MAPE*) von etwa 20–34% bei RF/GB gegenüber 34–48% beim DCF. Dies bestätigt empirisch den theoretischen Nutzen: ML-Modelle arbeiten mit einer Vielzahl nicht-linearer Zusammenhänge und können aus historischen Daten lernen, statt wie DCF auf wenige lineare Annahmen zu bauen. Dementsprechend liefern die ML-Modelle in der Untersuchung kleinere Vorhersagefehler bzw. eine höhere Genauigkeit als das DCF. Auch auffallend ist, dass alle Modelle durchschnittlich den Unternehmenswert zu hoch bewerten. Unter Berücksichtigung des Testzeitraumes von 2021 bis 2023 und der Covid19 Pandemie, lässt sich auch oberflächlich eine Erklärung dafür finden, da während dieser Jahre die Marktkapitalisierung teils drastisch eingefallen ist.¹⁶⁴ Konkret zeigt sich dies auch in den absoluten Fehlerschätzungen für einzelne Unternehmen. Einige Beispiele illustrieren die Stärken und Schwächen der Ansätze: So wiesen die ML-Ansätze unerwartete Ausreißer auf. So überschätzte etwa das Gradient-Boosting-Modell IAC/InterActiveCorp für 2023 den Wert um über 310 % (Prognose 15,24 Mrd. \$ vs. tatsächliche 3,71 Mrd. \$), während das DCF eine moderatere Schätzung erzielte. Ähnlich stark überhöhte RF/GB Aktienwerte wie EchoStar Corporation (+225 % RF-Fehler 2022) oder Lumen Technologies (LUMN, +189 % RF-Fehler 2022). Diese extremen Abweichungen deuten auf spezifische Limitationen der ML-Modelle hin (z. B. unzureichende Features oder Überanpassung) und zeigen, dass auch ML-Modelle nicht frei von großen Prognosefehlern sind. Gleichzeitig traten auch beim DCF-Modell bei Großkonzernen mit außerordentlichen Ereignissen große Fehler auf. Beispielsweise wurde der Unternehmenswert der Boeing Company für 2021 per DCF als Null prognostiziert (Abweichung 100 % vom tatsächlichen Marktwert), da die angenommenen Cashflows negativ waren. Im selben Jahr unterschätzten auch RF und GB den Wert stark (etwa –89% bis –91 %), jedoch verblieben hier zumindest positive Werte. Analog überschätzte das DCF bei The Bank of New York Mellon in einem Jahr den Wert ebenfalls mit einer 100%-Fehlschätzung, weil keine nachhaltigen Cashflows unterstellt wurden. Im Gegensatz dazu haben RF/GB für diese Unternehmen realistischere, wenn auch noch fehlerbehaftete Schätzungen geliefert. Diese Beispiele verdeutlichen, dass das DCF-Modell in Extremsituationen (z. B. negative Cashflows, außerordentliche Ereignisse) unplausible Ergebnisse liefern kann. ML-Modelle zeigen hier zwar ebenfalls Ausreißer, aber seltener solch vollständiges „Versagen“. Andererseits können ML-Modelle in vielen Fällen deutlich präzisere Schätzungen liefern. So erzielte Gradient Boosting für Sabra Health Care REIT 2023 mit 0,66 % Fehler praktisch eine nahezu

¹⁶⁴ Vgl. Shu et al. [2021], S. 2 ff.

punktgenaue Prognose des Marktwerts, während das DCF in diesem Fall noch eine Abweichung von 12 % aufwies. Ähnlich exakte Ergebnisse fanden sich bei Unternehmen wie H&R Block (2023: GB-Fehler 0,06 %) oder Apple Hospitality REIT (2033: RF-Fehler 0,05 %). Diese Beispiele zeigen, dass ML-Modelle unter günstigen Bedingungen außergewöhnlich genau arbeiten können.

Aus den Daten geht hervor, dass Gradient Boosting über die meisten Branchen hinweg die geringsten mittleren prozentualen Fehler aufweist und damit überwiegend das verlässlichste Modell ist. Random Forest nimmt in vielen Industrien eine Mittelfeldrolle ein: In Bereichen wie „Banks“ oder „Paper & Forest Products“ erzielt es sogar die besten oder zweitbesten Ergebnisse, während DCF hier oft deutlich schlechter abschneidet. Besonders auffällig sind die großen Leistungsunterschiede in Finanz- und Telekommunikationssektoren: In „Consumer Finance“, „Thriffs & Mortgage Finance“ sowie „Communications Equipment“ liegen die Fehlerraten von DCF teilweise bei über 100 %, während Gradient Boosting und Random Forest deutlich niedriger verbleiben. Branchen wie „Healthcare Equipment & Services“ oder „Retail“ bilden hingegen eine Ausnahme, da hier alle drei Modelle relativ eng beieinanderliegen und geringe Modellunterschiede aufweisen. Zusammengefasst ist Gradient Boosting das konsistent leistungsstärkste Modell, Random Forest liefert solide Mittelwerte, und DCF zeigt in einigen Sektoren extreme Fehler – was besonders in volatileren Bereichen eine gezielte Modellauswahl erfordert. Exemplarisch ist in der folgenden Abbildung die gesamthafte Streuung je Industrie dargestellt.

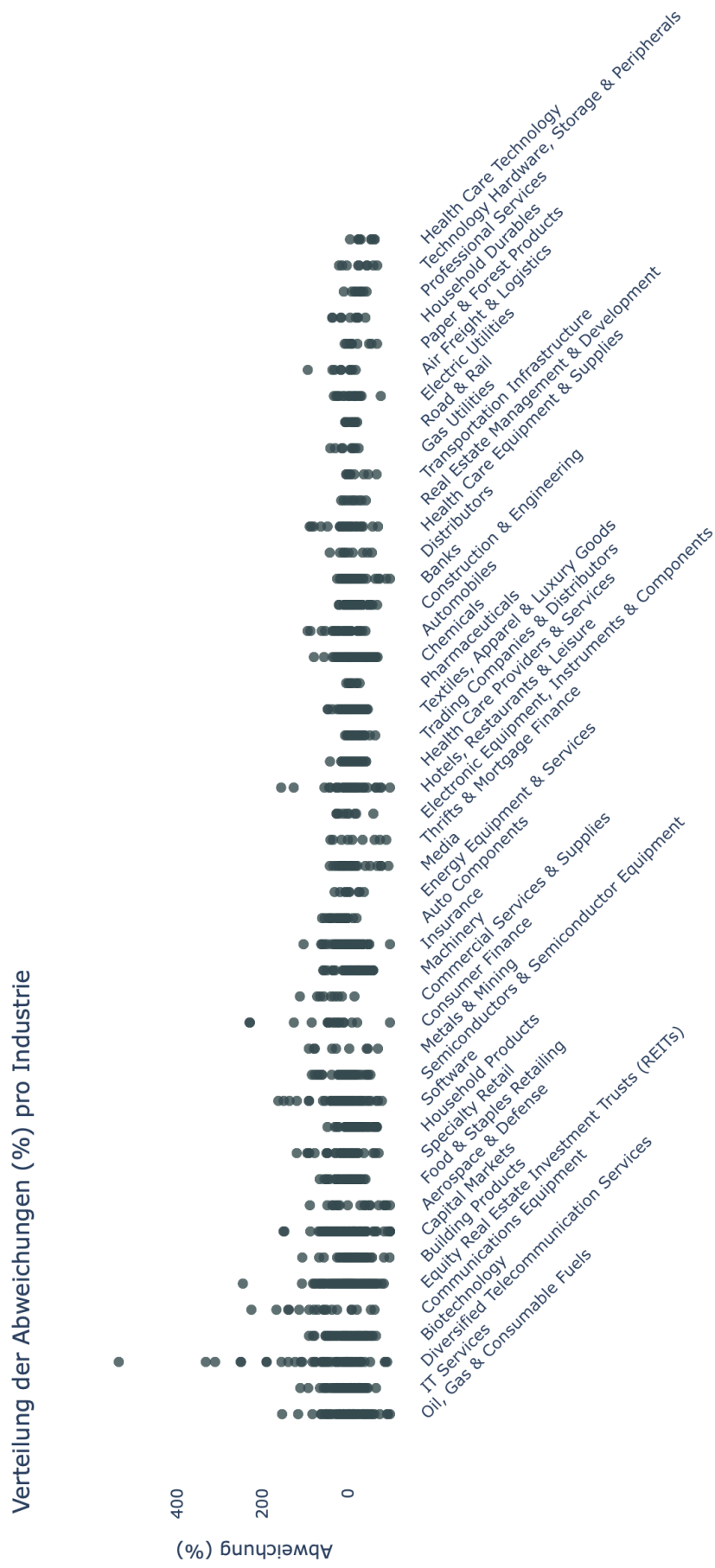


Abbildung 20: Verteilung der %-Abweichungen je Industrie
(Quelle: eigene Darstellung)

Insgesamt bestätigt sich die Arbeitshypothese, dass ML-basierte Verfahren im Durchschnitt geringere Abweichungen aufweisen als das traditionelle DCF. Neben der reinen Prognosequalität bietet maschinelles Lernen erhebliche betriebswirtschaftliche Effizienzvorteile. Zeit- und Ressourceneinsatz spielen in der Praxis eine entscheidende Rolle. Einmal trainiert, können ML-Modelle automatisiert hunderte oder tausende Unternehmen in Sekundenschnelle bewerten, da die Vorhersage nur auf dem Durchlauf fertiger Algorithmen basiert. Im Gegensatz dazu erfordert ein DCF-Ansatz für jedes Unternehmen eine aufwändige manuelle Analyse: Annahmen müssen individuell festgelegt, Cashflow-Prognosen berechnet und Validierungen durchgeführt werden. Dieses Verfahren ist pro Bewertung zeitintensiv und personalaufwendig. Gerade bei großvolumigen Analysen ist der Unterschied enorm. Maschinelles Lernen spart wiederkehrende Routineaufgaben und erlaubt eine zentrale Verarbeitung großer Datenmengen in kurzer Zeit. Zwar beansprucht das initiale Training der ML-Modelle Rechenkapazität, doch dieser Aufwand fällt einmalig an und skaliert nicht mit der Anzahl zu bewertender Firmen. Dadurch bestätigt sich die zweite Arbeitshypothese: ML-Verfahren ermöglichen eine deutlich schnellere, automatisierte Bewertung bei vergleichbarer oder höherer Genauigkeit. Eng verbunden mit der Effizienz ist die Skalierbarkeit der Modelle. Maschinelles Lernen skaliert praktisch nahezu linear mit der Datenmenge: Mit wachsendem Trainingsdatensatz verbessern sich – bis zu einem Sättigungspunkt – die Vorhersagegenauigkeit und Robustheit der Modelle, während die Vorhersagezeit nur moderat zunimmt. Klassische DCF-Bewertungen hingegen skalieren kaum. Jeder neue Unternehmens-Fall erfordert hohen manuellen Aufwand, da jede Bewertung individuelle Analysen und Anpassungen durch Fachexperten erfordert, was zeit- und ressourcenintensiv ist, die sich nicht ohne Weiteres automatisieren lassen, außer es werden Abstriche am Modell vorgenommen, wie sie auch in dem angewandten DCF-Modell dieser Arbeit vorgenommen wurden.

In volatilen Märkten, in denen kurzfristige Bewertungen für viele Unternehmen notwendig werden, wie etwa bei Krisen oder M&A, stößt das traditionelle Verfahren schnell an Grenzen. ML-Ansätze können hier ihre Stärken ausspielen und schnell aktualisierte Werte liefern. Schließlich eliminieren ML-Modelle menschliche Tippfehler und Inkonsistenzen, die in iterativen DCF-Prozessen entstehen können. Die erhöhte Skalierbarkeit wurde auch durch die empirischen Resultate gestützt, womit die zweite Hypothese dieser Arbeit belegt wurde. Für die praktische Anwendbarkeit ist zudem wichtig, welche Informationen die Modelle nutzen und wie interpretierbar die Vorhersagen sind. ML-Modelle können sehr viele Variablen simultan berücksichtigen –

von klassischen Finanzkennzahlen über Marktindikatoren bis hin zu „weichen“ Einflussgrößen, wie z.B. Branchendaten oder ESG-Faktoren. Große Bewertungsstichproben, wie sie etwa bei der Analyse ganzer Branchen oder Portfolios anstehen, lassen sich mit Maschinellem Lernen somit skalierbar bewältigen. In Zeiten schnelllebigere Märkte ist es oft nötig, Bewertungen für viele Unternehmen in kurzer Zeit durchzuführen – hier stoßen traditionelle Prozesse an ihre Grenzen. Zudem basieren klassische Modelle häufig auf vereinfachenden Annahmen, wie etwa stetiges Wachstum in der ewigen Rente beim DCF, die der Realität nicht immer gerecht werden. Strukturelle Brüche, disruptive Geschäftsmodelländerungen oder neuartige Werttreiber können in den starren Strukturen traditioneller Modelle nur unzureichend berücksichtigt werden. Im Rahmen dieser Arbeit wurde aus diesem Grund ein DCF-Modell mit vereinfachten Annahmen gewählt, wie es auch in der Praxis regelmäßig vorkommt. Trotz der nun vorhandenen Skalierbarkeit waren die ML-Modelle auch hier qualitativ überlegen.

Aus der Analyse geht hervor, dass die ML-Modelle in der Regel ähnliche zentrale Werttreiber wie traditionelle Analysen identifizieren. Dies zeigt sich beispielsweise darin, dass in beiden Verfahren die EBITDA-Kennzahl der wichtigste Einflussfaktor auf den geschätzten Unternehmenswert. Die Feature-Importance-Analyse ergab, dass EBITDA im Random Forest 46,4 % und im Gradient Boosting 25,7 % der Modellentscheidungen trägt. Damit bleibt ML konsistent mit betriebswirtschaftlicher Intuition, dem EBITDA als Proxy für operative Ertragskraft und zeigt, dass datengetriebene Modelle ökonomisch sinnvolle Muster lernen können. Dies erleichtert auch die Akzeptanz bei Finanzfachleuten: Wenn die ML-Modelle bekannte Treiber hervorheben, können Analysten das Ergebnis plausibilisieren. Dennoch sollte berücksichtigt werden, dass Unternehmensbewertungen somit nicht ausschließlich das Ergebnis objektiver Modellrechnungen sind. Vielmehr werden sie durch das Verhalten und die Einschätzungen von Finanzanalysten am Kapitalmarkt mitgeprägt, insbesondere dann, wenn traditionelle Bewertungsverfahren zur Anwendung kommen. Insofern sind Bewertungsprozesse auch performativer Natur – sie erzeugen durch ihre Anwendung die Realität, die sie zu messen vorgeben.

Random Forests und Gradient Boosting liefern empirisch einen betriebswirtschaftlich relevanten Mehrwert gegenüber dem klassischen DCF. Sie prognostizieren Unternehmenswerte präziser (höhere R^2 , niedrigere Fehlerkennzahlen), erlauben automatisierte und somit deutlich ressourcenschonendere Auswertungen, und skalieren besser bei großen Datensätzen. Fallbeispiele aus den Testdaten unterstreichen, dass ML-

Modelle in vielen Fällen genauere Schätzungen bringen – besonders bei datenreichen, „typischen“ Unternehmen – während sie in wenigen Ausreißerfällen (z. B. stark kritische Geschäftsverläufe) noch verbesserungswürdig sind. Durch die Identifikation bekannter Fundamentalkennzahlen als Treiber zeigen die ML-Ansätze, dass sie ökonomisch sinnvolle Muster lernen. Damit können Random Forest und Gradient Boosting die traditionelle Unternehmensbewertung sinnvoll ergänzen: Sie erhöhen die Effizienz und Objektivität des Bewertungsprozesses, ohne das DCF prinzipiell zu ersetzen. In Summe bestätigen die empirischen Resultate das Nutzenpotenzial moderner ML-Methoden in der Finanzanalyse – insbesondere in Szenarien, in denen schnelle, datenbasierte Bewertungen gefragt sind. Im Theorieteil dieser Arbeit wurde erörtert, dass speziell DCF-Modelle in Krisenzeiten nicht mehr valide prognostizieren können, da Planannahmen nicht valide getroffen werden können. Unter Anbetracht der Tatsache, dass die Corona Pandemie in 2021 und 2022 signifikanten Einfluss auf Unternehmenswerte hatte, lässt sich ebenfalls noch einmal hervorheben, dass die gewählten ML-Algorithmen nicht so sensitiv auf diese Krise reagiert haben.

Neben den erörterten und praktisch angewandten ML-Algorithmen Random Forest und Gradient Boosting gibt es auch die bereits erwähnten, aber nicht exemplarisch dargestellten Neuronalen Netze, die signifikante Nutzenpotenziale für die Unternehmensbewertung darstellen können.

Neuronale Netze bieten im Kontext der Unternehmensbewertung und Finanzanalyse eine Reihe von bedeutenden Anwendungsvorteilen. Ihr größter Vorteil liegt in der Fähigkeit, hochkomplexe, nicht-lineare Zusammenhänge in großen und heterogenen Datensätzen zu erkennen und auszuwerten. Während die gezeigten klassischen Verfahren wie Random Forest oder Gradient Boosting häufig auf vordefinierte Merkmale oder lineare Beziehungen angewiesen sind, können neuronale Netze eigenständig Muster und Interaktionen zwischen zahlreichen Einflussgrößen erfassen. Dies ermöglicht deutlich präzisere Prognosen, etwa bei der Unternehmensbewertung, Preisoptimierung oder der Vorhersage von Marktbewegungen.¹⁶⁵ In der Praxis zeigt sich, dass neuronale Netze durch die Analyse umfangreicher strukturierter und unstrukturierter Daten – wie Bilanzkennzahlen, Marktindizes, Kundenverhalten und makroökonomische Faktoren – eine höhere Prognosegenauigkeit erreichen als traditionelle Modelle. Beispielsweise berichten Unternehmen, die neuronale Netze für Preisprognosen einsetzen, von einer Steigerung der Vorhersagegenauigkeit auf über 90 % im Vergleich zu etwa 60 % bei

¹⁶⁵ Vgl. Moro Visconti [2024], S.91-100

klassischen Segmentierungsmodellen.¹⁶⁶ Zudem sind neuronale Netze in der Lage in Echtzeit auf neue Daten und Marktveränderungen zu reagieren und so dynamische Anpassungen vorzunehmen, was insbesondere in volatilen Märkten ein entscheidender Wettbewerbsvorteil ist. Ein weiterer Pluspunkt ist die automatische Merkmalsextraktion. Neuronale Netze benötigen weniger manuelles Feature Engineering, da sie relevante Muster selbstständig aus den Rohdaten herausfiltern können.¹⁶⁷

Trotzdem haben auch Maschinelle Lern Algorithmen Einschränkungen in der Praxis. Die Modellstrukturen sind weniger transparent als klassische formale Rechnungen. Random Forests und Boosting-Modelle gelten gewissermaßen als „Black Box“, sodass ihre Vorhersagen nicht immer intuitiv erklärbar sind. Anders als im DCF, wo jeder einzelne Parameter explizit festgelegt wird, können ML-Modelle komplexe Wechselwirkungen verbergen. Hier helfen zwar Methoden wie Feature-Importance oder Shapley-Werte, doch ein vollständiges Verständnis ist schwieriger. Hinzu kommt, dass ML-Modelle stark auf die Verfügbarkeit hochwertiger historischer Daten angewiesen sind. Fehlende oder verzerrte Inputdaten können zu falschen Schätzungen führen. Neben der Black-Box-Problematik sind Datenanforderungen ein kritischer Punkt. ML-Modelle benötigen umfangreiche historische Datensätze zu Trainingszwecken. In der Unternehmensbewertung sind solche Daten nicht immer in ausreichender Menge oder Qualität verfügbar. So wurde in dieser Arbeit explizit der US-Markt als Trainingsdatenset gewählt, da die europäische Datenverfügbarkeit deutlich eingeschränkter war. Auch der Trainingsprozess ist sehr rechenintensiv und erfordert oftmals spezialisierte Hardware, was die Implementierung für kleinere Unternehmen erschwert.¹⁶⁸ Schließlich besteht das Risiko des Overfittings, insbesondere bei zu komplexen Architekturen oder zu kleinen Datensätzen, wodurch die Generalisierungsfähigkeit auf neue Fälle leidet.¹⁶⁹ Klassische Bewertungsmethoden können im Gegensatz dazu auch auf knapper Informationsbasis mit Expertenwissen arbeiten. Ein datengetriebenes Modell ist zudem immer nur so gut wie die Daten, mit denen es trainiert wurde – Stichwort “garbage in, garbage out”. Beispielsweise würde ein ML-Modell, das auf überwiegend konjunkturellen Aufschwungphasen trainiert wurde, in einer Rezession möglicherweise systematische Fehleinschätzungen produzieren, da es einen solchen Zustand kaum “gelernt” hat. Hier zeigt sich, dass ML-Ansätze anfällig für Strukturbrüche und wechselhafte Entwicklungen

¹⁶⁶ Vgl. Price Optimization & Neural Networks [2025], o.S.

¹⁶⁷ Vgl. Rashid [2024], S. 63 ff.

¹⁶⁸ Vgl. Mehlig [2022], S. 75

¹⁶⁹ Vgl. Grunert [2021], S. 215 f.

sein können, wenn die Datenmenge bzw. der Zeitraum zu kurz gewählt wurde, während menschliche Analysten solche Umbrüche zumindest qualitativ antizipieren können.

In der vorliegenden Arbeit wurde der Zeitraum von 2012 bis 2020 als Trainingsperiode für die Machine-Learning-Algorithmen gewählt, während die Testperiode die Jahre 2021 bis 2023 umfasst. Diese Auswahl könnte auf den ersten Blick als problematisch erscheinen, da die Testjahre von der Corona-Pandemie geprägt waren, die viele Unternehmen und ihre Bewertungen stark beeinflusst hat. Jedoch ist diese Entscheidung keineswegs zufällig oder nachteilig, sondern getroffen, um die Leistungsfähigkeit der Algorithmen unter außergewöhnlichen Bedingungen zu testen. Gerade die Pandemie-Jahre fungieren als eine Art Stresstest, der verdeutlicht, wie robust und anpassungsfähig die Modelle in Krisensituationen agieren. Die Ergebnisse zeigen, dass die Algorithmen trotz der Herausforderungen der Corona-Zeit eine Abweichung von ‚nur‘ rund 20% aufweisen, was unterstreicht, dass die Modelle bereits jetzt eine teils hohe Genauigkeit besitzen und in der Lage sind, auch in schwierigen Marktphasen valide Ergebnisse zu liefern. Auch wird hiermit gezeigt, dass das Training mit mehr Daten, etwa inklusive der Finanzkrisen 2008 bzw. 2009 oder der Zusammenbruch des europäischen Wirtschaftssystems 1992 ggf. mehr Genauigkeit liefern könnte. Dieses Bewusstsein und die damit verbundene Argumentation können somit als zusätzlicher Beleg für die Stärken der datengetriebenen Verfahren herangezogen werden, zeigt aber auch eine gewisse Grundbedingung: Die Verfügbarkeit an ausreichend und qualitativ hochwertigen Daten.

Im Gegensatz dazu ist das DCF-Modell weniger ‚datenhungrig‘ und kann in datenarmen Situationen zumindest eine theoretisch kohärente Prognose abgeben. Außerdem sind regulatorische und organisatorische Hürden zu beachten. Die DCF-Methodik ist in der Finanzwelt weit etabliert und wird von Aufsichtsbehörden, Prüfern und Anlegern erwartet. Ein kompletter Ersatz durch Algorithmen des Maschinellen Lernens ist derzeit unwahrscheinlich; sinnvoller ist eine Kombination: ML-Modelle können als Screening- oder Validierungswerkzeuge dienen, die Analysten auf mögliche Fehleinschätzungen hinweisen oder Trends aufdecken, die in der eigenen DCF-Analyse überprüft werden sollten. Ein zentrales Problem klassischer Ansätze wie dem DCF ist die hohe Annahmesensitivität. Kleine Veränderungen bei Kernparametern (etwa Wachstumsraten, Kapitalkosten oder Margen) können zu großen Schwankungen im ermittelten Unternehmenswert führen. Dadurch besteht die Gefahr, dass Ergebnisse missinterpretiert oder manipuliert werden. Deloitte weist in diesem Zusammenhang darauf hin, dass DCF-Analysen ohne ausreichende Sorgfalt und plausible Annahmen „hochgradig irreführende

Schlussfolgerungen“ liefern können.¹⁷⁰ Dies liegt an der Vielzahl subjektiver Inputfaktoren im DCF, welche das Modell anfällig für Bias und Manipulation machen.

Die sogenannten „Big Four“ Wirtschaftsprüfungsgesellschaften betonen daher die Bedeutung von erklärbaren ML-Modellen: PwC etwa beschreibt, dass die meisten ML-Algorithmen „keine direkten Erklärungen dafür liefern, warum oder wie die Resultate erzielt werden“, was die Notwendigkeit von Interpretationshilfen hervorhebt.¹⁷¹

Die „Big-Four“ treiben aus genau diesem Grund entsprechende Tools voran: KPMG beschreibt in aktuellen Publikationen, wie GenAI für die Analyse und Segmentierung von Finanzdaten, die automatische Erstellung von Berichten und die Durchführung von Benchmark-Analysen eingesetzt wird. Die Plattformen sind darauf ausgelegt, Bewertungsprozesse zu beschleunigen und zu automatisieren, insbesondere durch die Integration von multimodalen KI-Modellen, die verschiedene Datenquellen zusammenführen und auswerten können.¹⁷²

Dadurch können in der Praxis Transaktionsbewertungen oder Werthaltigkeits-Tests schneller und objektiver durchgeführt werden. Ein weiterer praktischer Nutzen ist die Reduzierung von subjektiven Verzerrungen: ML-Modelle kennen weder Euphorie noch Pessimismus, sie extrapolieren ausschließlich aus Daten. So werden Bewertungen reproduzierbarer und weniger von individuellen Einschätzungen einzelner Analysten abhängig. Schließlich eröffnet der Einsatz von ML auch die Möglichkeit, frühzeitig Anomalien oder Trends zu erkennen. Ein ML-Modell könnte etwa systematisch Über- oder Unterbewertungen im Vergleich zu Marktpreisen aufdecken und so als Frühwarnsystem dienen. Insgesamt zeigen sowohl die Ergebnisse dieser Arbeit als auch aktuelle Forschung durch Wirtschaftsprüfungsgesellschaften, dass ML-basierte Verfahren in der Unternehmensbewertung greifbaren Mehrwert bieten können – sei es durch höhere Prognosegenauigkeit, schnellere Analysen oder neue Einblicke in Werttreiber.

Insbesondere komplexere Modelle wie die angewandten Ensemble Modelle Random Forest und Gradient Boosting oder tiefe neuronale Netze liefern zwar ein Ergebnis, aber die Nachvollziehbarkeit, warum ein bestimmter Wert prognostiziert wurde, ist eingeschränkt. Das Verständnis der Modelllogik ist jedoch in der

¹⁷⁰ Knabe & Rondorf [2023], o.S.

¹⁷¹ Chris Oxborough et al. [2019]

¹⁷² Herr & Arias [2025], o.S.

Unternehmensbewertung essenziell – Stakeholder (Gutachter, Prüfer, Investoren) verlangen Erklärungen dafür, wie ein Wert zustande gekommen ist.¹⁷³ Bei intransparenten ML-Modellen besteht die Gefahr, dass Vertrauen und Akzeptanz leiden.

Der theoretische Einsatz von Machine Learning Modellen bietet, wie gezeigt, signifikante Nutzenpotenziale. Für die tatsächliche Umsetzung dieser Modelle ist es im Zuge der betriebswirtschaftlichen Gegebenheiten unerlässlich auch die im theoretischen Teil dieser Arbeit gezeigten regulatorischen Vorgaben für Unternehmensbewertungen einzuhalten.

Die relevanten IDW-Standards (insbesondere IDW S1, PS 880 und EPS 861) schreiben vor, dass ein Bewertungsverfahren durch Nachvollziehbarkeit, Transparenz, Prüfbarkeit durch Dritte, Reproduzierbarkeit und lückenlose Dokumentation gekennzeichnet sein muss, wie auch bereits in Kapitel 3.2 erläutert. Random Forest- und Gradient Boosting-Modelle können diese Anforderungen zwar nicht in derselben Schlichtheit erfüllen wie klassische DCF-Berechnungen, kommen ihnen aber bei konsequenter technischer Umsetzung sehr nahe. Zum einen ist die Nachvollziehbarkeit eingeschränkt erfüllbar. Durch das Berechnen von Feature Importances oder den Einsatz von SHAP-Analysen lassen sich die wichtigsten Einflussgrößen (z. B. EBITDA, Umsatzwachstum) extrahieren und für Dritte verständlich machen. Die exakte interne Logik – hunderte Entscheidungsbäume mit vielen Tiefenstufen – bleibt jedoch weitgehend eine ‚Black Box‘ und kann nicht in jedem Detail mit einer einfachen Formel abgebildet werden.

Des Weiteren gewährleistet die Offenlegung des Modellierungsprozesses Transparenz. Werden Datenherkunft, Feature Engineering, Hyperparameter und Validierungsverfahren klar dokumentiert, entsteht ein nachvollziehbarer Rahmen für das Modell. Die Komplexität der Baumstrukturen selbst erschwert jedoch die intuitive Nachvollziehbarkeit, so dass Transparenz eher auf den Prozess als auf die Entscheidungswege abzielt.

Umso mehr ist die Prüfbarkeit durch Dritte technisch sehr gut gegeben. Ein gespeichertes RF- oder GB-Modell lässt sich exakt laden, mit Testdatensätzen erneut ausführen und so rechnerisch verifizieren. Die fachliche Plausibilität einzelner Vorhersagen bleibt allerdings nur bedingt prüfbar, da ein Prüfer nicht jede einzelne Baumentscheidung leicht nachvollziehen kann.

¹⁷³ Vgl. Moro Visconti [2024], S. 100

Abschließend ist die Reproduzierbarkeit eine der Stärken von ML-Verfahren. Liegen Datenbasis, Code und Hyperparameter vollständig vor und wurden Zufallsvariablen fixiert, liefert jeder erneute Durchlauf identische Ergebnisse. Damit ist sichergestellt, dass bei gleichen Voraussetzungen stets derselbe Unternehmenswert herauskommt.

Um IDW-Konformität zu erreichen, muss der gesamte Lebenszyklus des Modells akribisch protokolliert werden – von der Datenbereinigung über das Training bis hin zu Validierungsergebnissen und Modellversionen. Wird dieser hohe Dokumentationsaufwand betrieben, können RF- und GB-Modelle die formalen Anforderungen in vollem Umfang erfüllen, auch wenn ein Rest an Intransparenz aufgrund der Modellkomplexität verbleibt.

Um maschinelle Lernverfahren entsprechend regulatorisch und praktisch in die Unternehmensbewertungspraxis zu integrieren, ist eine konsequente Orientierung an den Vorgaben des IDW S1 sowie der internationalen Standards nach IFRS 13 erforderlich. Zentrale Voraussetzung ist die daher vollständige Nachvollziehbarkeit und Dokumentation des Bewertungsprozesses. Dies bedeutet, dass nicht nur die Auswahl und Parametrisierung der ML-Modelle (z. B. Random Forest, Gradient Boosting), sondern auch sämtliche Datenquellen, Transformationsschritte und Modellannahmen lückenlos protokolliert werden müssen. Insbesondere Methoden der Explainable AI (XAI) leisten hierbei einen entscheidenden Beitrag: Sie ermöglichen es, die Einflussfaktoren einzelner Prognosen transparent offenzulegen und die Entscheidungslogik auch für Dritte – etwa Prüfer oder Gutachter – verständlich zu machen. Damit kann also die zentrale Forderung des IDW S1 nach einer prüfbar und nachvollziehbaren Herleitung des Unternehmenswerts erfüllt werden. Entscheidend ist dabei, dass die Ergebnisse der ML-Modelle stets im Kontext klassischer Bewertungsverfahren interpretiert und bspw. in einer Entscheidungsmatrix gegenübergestellt werden, um die regulatorische Konformität nach IFRS 13 sicherzustellen. Einen weiteren Schritt in die Richtung liefert der neue IDW-Prüfungsstandard PS 861 „Prüfung von KI-Systemen“, da dieser erstmals verbindliche Mindestanforderungen an vertrauenswürdige KI-Lösungen stellt und sie somit prüfbar macht.¹⁷⁴ Dennoch ist dies bisher nur der erste Schritt um KI-Risiken systematisch zu adressieren und regulatorische Vertrauensnachweise zu erbringen, nicht um eine Vollumfassende Implementierung prüfbar zu machen. Es bleibt an dieser Stelle

¹⁷⁴ IDW PS: Prüfung von KI-Systemen (IDW PS 861 (03.2023) Stand: 10.03.2023 [2023], S.1 ff.

abzuwarten, ob XAI perspektivisch tatsächlich für regulatorische Anforderungen ausreichend ist, oder seinerseits eine weitere Komplexitätsebene einbaut.

In der Praxis zeichnet sich zudem ein Umdenken ab: Anstatt ML und klassische Ansätze gegeneinanderzustellen, werden Hybridlösungen diskutiert.¹⁷⁵ So könnten ML-Modelle genutzt werden, um Inputparameter für DCF-Modelle abzuleiten (z. B. prognostizierte Wachstumsraten), oder umgekehrt DCF-Ergebnisse durch ML-basierte Plausibilitätschecks zu validieren. Auch die Big-Four sehen ML nicht als Allheilmittel, sondern als ergänzendes Werkzeug. So betont Deloitte, dass der Einsatz von Analytics und ML den Bewertungsprozess effizienter machen kann, die menschliche Beurteilung aber weiterhin zentral bleibt – Analysten sollen weniger Zeit auf manuelle Berechnungen verwenden und dafür mehr auf die Interpretation der Ergebnisse und Prüfung der Annahmen, welche in die Modelle einfließen.¹⁷⁶

In der praktischen Anwendung empfiehlt es sich, ML-Modelle zunächst als komplementäre Bewertungsinstrumente einzusetzen, etwa zur Sensitivitätsanalyse bei Unternehmen mit volatilen Cashflows oder schwer prognostizierbaren Werttreibern, wie sie im klassischen DCF-Verfahren häufig zu Unsicherheiten führen. Für datenreiche, standardisiert bilanzierende Unternehmen können ML-Modelle – bei Einhaltung der Dokumentations- und Transparenzpflichten – perspektivisch auch als primäres Bewertungsinstrument herangezogen werden. So entsteht ein auditierbarer Prüfpfad, der sowohl die technische Leistungsfähigkeit datengetriebener Ansätze als auch deren regulatorische und praktische Einbettung in die Bewertungspraxis gewährleisten.

Die Anwendung maschineller Lernverfahren zeigt zudem auch signifikante Praxisrelevanz insbesondere im Bereich M&A und der Start-up-Bewertung, wo traditionelle Methoden an komplexe Datenstrukturen und dynamische Marktbedingungen angepasst werden müssen. So konnten die Autoren Wei Chang und Tingting Zhang in Ihrer empirischen Studie zu „Corporate M&A Valuation“ Genauigkeiten von 91,25–93,81% für Finanzkennzahlen nachweisen.¹⁷⁷ Ergänzend erreichen spezialisierte Bewertungsmodelle für sog. Post-Money-Valuation (Nach-Geld-Bewertung), die unter anderem auf Gradient Boosting-Algorithmen basieren, Vorhersagegenauigkeiten von über 95%, wie in einer Forschungsarbeit von Ang et al. gezeigt werden konnte.¹⁷⁸ Diese

¹⁷⁵ Vgl. Dixon et al. [2020], S. 16 ff.

¹⁷⁶ Vgl. Guszca, J. et al. [2020], o.S.

¹⁷⁷ Vgl. Chang & Zhang [2025], S. 1

¹⁷⁸ Vgl. Ang et al. [2020], S. 1

Anwendungsfälle demonstrieren, wie komplexere ML-Verfahren nicht nur theoretische Präzision bieten, sondern konkrete Entscheidungsprozesse in Finanztransaktionen durch datengetriebene Informationen optimieren.

Zusammenfassend eröffnet Maschinelles Lernen in der Unternehmensbewertung erhebliche Potenziale, etwa durch eine höhere Genauigkeit bei der Erfassung komplexer Zusammenhänge, die Automatisierung von Bewertungsprozessen und die Möglichkeit, große Datenmengen dynamisch auszuwerten. Insbesondere moderne Deep-Learning-Modelle, die mehrere Zwischenschichten enthalten, können Beziehungen erkennen, die mit klassischen und auch den angewandten ML-Bewertungsverfahren nicht abbildbar sind. Damit ermöglichen sie eine detailliertere und oft realistischere Einschätzung des Unternehmenswerts.

Gleichzeitig wurde aber deutlich, dass die beiden Ansätze nicht als Gegensätze, sondern eher als komplementäre Werkzeuge zu betrachten sind. Ein wichtiger Diskussionspunkt war, dass ML-Modelle trotz besserer numerischer Ergebnisse nicht per se „besser“ im holistischen Sinn sein müssen – Aspekte wie Nachvollziehbarkeit, Datenverfügbarkeit und Modellrisiken spielen eine Rolle. Der DCF-Ansatz behält daher aus Sicht des Autors seine Bedeutung als intuitiv nachvollziehbarer Referenzpunkt, während ML als zusätzliche Perspektive wertvolle Dienste leisten kann.

Auf Basis der gewonnenen Erkenntnisse lassen sich mehrere nächste Schritte und Empfehlungen ableiten. Für die praktische Anwendung wäre ein sinnvoller nächster Schritt, Pilotprojekte durchzuführen, in denen DCF- und ML-Bewertungen parallel erstellt und verglichen werden, etwa im Rahmen realer Transaktionen oder Jahresabschlussbewertungen. So könnten Unternehmen und Prüfer lernen, mit Abweichungen umzugehen und Vertrauen in die neuen Methoden aufzubauen. Zudem könnten Schulungen entwickelt werden, um Bewertern die notwendigen Datenanalysefähigkeiten zu vermitteln. Aus technischer Sicht bietet es sich an, die in dieser Arbeit verwendeten ML-Modelle weiter zu verfeinern – z. B. durch Hyperparameter-Tuning, den Einsatz neuronaler Netze oder Ensemble-Techniken, um die Prognosegüte noch weiter zu steigern. Auch der Einbezug zusätzlicher externer Daten (weitere Makroindikatoren, Marktstimmungen aus News/Social Media, etc.) wäre ein nächster Schritt, um das Modell noch breiter aufzustellen. Für die Wissenschaft schließlich ergibt sich die Notwendigkeit, die Robustheit und Validität der ML-Methoden in der Bewertung weiter zu erforschen. Langfristige Studien könnten klären, ob ML-basierte Bewertungen tatsächlich zu besseren finanziellen Entscheidungen führen (etwa gemessen an erzielten

Renditen bei Investitionsentscheidungen oder geringeren Wertberichtigungen im Zeitverlauf). Darüber hinaus sollte die ethische und regulatorische Dimension nicht vernachlässigt werden: zukünftige Forschung und Pioniere müssen sich damit auseinandersetzen, wie die Verantwortung bei automatisierten Bewertungsmodellen verteilt ist und wie sichergestellt werden kann, dass solche Modelle fair und unvoreingenommen funktionieren. Beispielsweise könnten Audit-Standards angepasst werden, um Prüfroutinen für KI-Modelle zu definieren.

9 Fazit

Diese Masterarbeit hatte das Ziel, die Nutzenpotenziale von maschinellem Lernen für moderne Ansätze der Unternehmensbewertung herauszuarbeiten – ein Thema von hoher Aktualität und praktischer Bedeutung. Vor dem Hintergrund der Digitalisierung im Finanzwesen und der Verfügbarkeit immer größerer Datenmengen stellt sich zunehmend die Frage, wie traditionelle Bewertungsmethoden, allen voran das Discounted-Cashflow-Modell, durch datengestützte Ansätze ergänzt, ersetzt oder verbessert werden können. Die Relevanz dieser Fragestellung zeigt sich sowohl in der Wissenschaft – wo interdisziplinäre Forschung zwischen Finance und Data Science an Bedeutung gewinnt – als auch in der Praxis, in der große Wirtschaftsprüfungsgesellschaften und Investmenthäuser verstärkt in KI-gestützte Bewertungsinstrumente investieren. Unternehmensbewertungen spielen eine zentrale Rolle bei Investitionsentscheidungen, Fusionen und Übernahmen, sowie der finanziellen Berichterstattung; bereits kleine Verbesserungen in der Bewertungsgenauigkeit oder -effizienz können daher große wirtschaftliche Auswirkungen haben. Vor diesem Hintergrund leistet die vorliegende Arbeit einen relevanten Beitrag, indem sie die klassischen Bewertungsansätze den Machine-Learning-Methoden gegenüberstellt und deren Stärken und Schwächen systematisch beleuchtet.

Ein Kernbestandteil der Arbeit war der Vergleich des traditionellen DCF-Ansatzes mit einem ML-basierten Ansatz. Das DCF-Verfahren gilt seit Jahrzehnten als Standard zur Ermittlung des fundamentalen Unternehmenswerts, da es auf einem soliden theoretischen Fundament basiert und in Führungskreisen vertraut ist. Im Theorieteil wurde herausgestellt, dass DCF-Schätzungen stark von subjektiven Annahmen abhängen und daher eine gewisse Resultatunschärfe und Anfälligkeit für Fehleinschätzungen aufweisen. Demgegenüber bieten ML-Modelle einen datenorientierten Zugang. Denn sie versuchen, den Unternehmenswert indirekt aus historischen Mustern und Beziehungen in

den Daten zu „lernen“, anstatt ihn direkt durch prognostizierte Cashflows zu berechnen. Die Arbeit hat beide Ansätze konzeptionell und empirisch gegenübergestellt. Konzeptionell zeigt sich, dass DCF und ML unterschiedliche Stärken mitbringen – das DCF-Modell punktet mit Transparenz und theoretischer Fundierung, während ML-Modelle durch Flexibilität und Anpassungsfähigkeit überzeugen. Ein zentrales Ergebnis der vorliegenden Arbeit ist, dass ein einfach implementiertes ML-Modell in der empirischen Analyse bessere Prognoseergebnisse lieferte als ein vereinfachtes Discounted-Cashflow-Verfahren auf derselben Datengrundlage. Empirisch konnte in dieser Thesis demonstriert werden, dass der ML-Ansatz dem DCF-Ansatz in einem direkten Vergleich auf derselben Datenbasis überlegen sein kann. Dieses Ergebnis ist insofern bemerkenswert, als ein relativ einfaches Random-Forest-Modell bzw. Gradient Boosting Modell genutzt wurde und dennoch die Prognosegüte eines, wenn auch vereinfachten, DCF-Modells übertraf. Damit untermauern die Befunde die zentrale These der Arbeit: Maschinelles Lernen besitzt erhebliche Nutzenpotenziale in der Unternehmensbewertung und kann ggü. traditionelle Methoden in bestimmten Aspekten eine bessere Leistung erzielen

Die im empirischen Teil entwickelte Modellanwendung lieferte konkrete Hinweise darauf, wie ML und DCF in der Praxis unterschiedlich performen. Der Random Forest und das Gradient Boosting zeigte auf dem gegebenen Datensatz geringere Abweichungen zu den tatsächlichen Unternehmenswerten als das Vergleichs-DCF-Modell. Dieses Resultat stützt die Annahme, dass ML-Verfahren komplexe Werttreiber-Kombinationen besser erfassen können. Allerdings war das ML-Modell in dieser Arbeit bewusst einfach gehalten, um die Vergleichbarkeit zu gewährleisten – es handelt sich also um eine konservative Untersuchung, die Verbesserungspotenziale für den ML-Ansatz nach oben offenlässt. Die Ergebnisse sind positiv zu bewerten, sollten jedoch ebenso kritisch eingeordnet werden. Zum einen basierte der Vergleich auf einem DCF-Modell mit vereinfachten Annahmen. Es ist denkbar, dass ein vollumfänglich ausgearbeitetes DCF mit Experteneinschätzungen näher an den realen Werten gelegen hätte. Ebenso wurde mit der Unterstellung der Effizienzmarkthypothese gearbeitet, in der praktischen Umsetzung der ML-Modellen ist eine solche Hypothese nur unter akkurater Berücksichtigung der Marktgegebenheit und des entsprechenden Umfelds anzuwenden. Zum anderen erfolgte die Bewertung ex-post mit historischen Daten, um die Genauigkeit zu bestimmen. Es sei an dieser Stelle nochmals betont, dass die ML-Modelle dennoch nur Trainingsdaten außerhalb des Prognosezeitraums erhalten haben. Für eine abschließende Beurteilung der ML-Überlegenheit wäre ein prospektiver Forecast-Vergleich wünschenswert.

Nichtsdestotrotz demonstriert die eigene Modellanwendung praktisch, dass ML-Ansätze sehr wohl in der Lage sind, valide und zuverlässige Bewertungen zu liefern. Zusammenfassend lässt sich feststellen, dass die Machine-Learning-Algorithmen selbst unter den herausfordernden Bedingungen der Corona-Pandemie beachtlich präzise Ergebnisse lieferten, was ihre Robustheit und ihren Mehrwert in der modernen Unternehmensbewertung zusätzlich unterstreicht. Interessant ist zudem, dass die Feature-Importance-Analyse des Random Forests (siehe Kapitel 6.1) wirtschaftlich sinnvolle Treiber identifiziert hat – ein Indiz dafür, dass das ML-Modell inhaltlich nachvollziehbare Zusammenhänge gelernt hat (z. B. zeigten sich EBITDA, Umsatzwachstum und Kapitalstruktur als wichtige Einflussfaktoren, was konform mit finanztheoretischen Erwartungen ist). Insgesamt untermauern die empirischen Resultate der Arbeit, dass die Integration von ML in Bewertungsaufgaben lohnend sein kann. Die Ergebnisse liefern damit sowohl Praktikern als auch Wissenschaftlern wertvolle Anhaltspunkte. Praktiker erhalten einen Beleg dafür, dass Datenanalysen und ML-Tools traditionelle Bewertungen verbessern können, und Wissenschaftler sehen bestätigt, dass weitere Forschung an der Schnittstelle von ML und Corporate Finance sinnvoll ist.

Gleichzeitig wurde aber deutlich, dass die beiden Ansätze nicht als Gegensätze, sondern eher als komplementäre Werkzeuge zu betrachten sind. Ein wichtiger Diskussionspunkt war, dass ML-Modelle trotz besserer numerischer Ergebnisse nicht per se „besser“ im holistischen Sinn sein müssen – Aspekte wie Nachvollziehbarkeit, Datenverfügbarkeit und Modellrisiken spielen eine Rolle. Der DCF-Ansatz behält daher aus Sicht des Autors seine Bedeutung als intuitiv nachvollziehbarer Referenzpunkt, während ML als zusätzliche Perspektive wertvolle Dienste leisten kann.

Für die unternehmerische Praxis und die Bewertungsexperten (z. B. in M&A-Abteilungen, Corporate Finance Beratungen oder Wirtschaftsprüfungsgesellschaften) haben die Erkenntnisse dieser Arbeit mehrere Implikationen. Erstens könnten ML-Modelle künftig als effizientes Ergänzungsinstrument im Bewertungsprozess dienen. Anstatt ausschließlich auf manuelle DCF-Modelle zu setzen, könnten Unternehmen ML-basierte Bewertungen als zweite Meinung oder Benchmark heranziehen. Dies kann helfen, Bewertungsfehlereinflüsse frühzeitig zu erkennen – etwa, wenn ein DCF-Modell aufgrund zu optimistischer Annahmen einen deutlich höheren Wert liefert als das ML-Modell, wäre das ein Signal zur Überprüfung der Annahmen. Zweitens erlaubt der Einsatz von ML eine höhere Verarbeitungsgeschwindigkeit und Skalierbarkeit. In der Praxis könnten so z. B. umfangreiche Unternehmensportfolios automatisiert bewertet und

regelmäßig aktualisiert werden, was mit traditionellen Methoden kaum, oder nur mit vereinfachten Annahmen, realisierbar ist. Dies erhöht die Reaktionsfähigkeit, etwa bei Marktschwankungen oder für turnusmäßige Werthaltigkeitsprüfungen. Drittens fördert die Kombination aus ML und klassischen Methoden die Objektivität. Subjektive Einschätzungen des Bewerter werden durch datenbasierte Modelle ergänzt, was die Abhängigkeit von Einzelmeinungen reduziert. Allerdings zeigt die Arbeit auch, dass für die praktische Nutzung gewisse Voraussetzungen geschaffen sein müssen – insbesondere braucht es Transparenzmechanismen für ML-Modelle, wie etwa erklärende Module (xAI), die die wichtigsten Werttreiber ausgeben und klar definierte Prozesse, wann und wie ML-Ergebnisse in die Bewertung einfließen. Die großen Wirtschaftsprüfungsgesellschaften und Beratungen haben hier bereits ersten Fortschrittverzeichnen können: So berichtet KPMG von erfolgreichen Pilotprojekten, in denen Advanced Analytics die Unternehmensbewertung unterstützt und beschleunigt haben, ohne die Qualität zu beeinträchtigen.

Ein weiterer Forschungsbereich betrifft daher die Erklärbarkeit von ML-Modellen in finanziellen Anwendungen – hier besteht Bedarf an Methoden, die komplexe Modelle so transparent machen, sodass sie auch von Entscheidern akzeptiert werden. Ferner könnten zukünftige Studien die Generalisierbarkeit der Ergebnisse überprüfen. Gilt die Überlegenheit von ML-Modellen auch in anderen Märkten, für andere Bewertungsziele (z. B. Startup-/Venture-Bewertungen) oder über längere Prognosehorizonte? Auch die Frage nach der Integration in Regulierungsrahmen (etwa in der Rechnungslegung bei Werthaltigkeitsprüfungen nach IFRS/US GAAP) ist forschungswürdig. Möglicherweise müssen Bewertungsstandards angepasst werden, wenn KI-Modelle zum festen Repertoire gehören sollen.

Insgesamt bedeutet dies für die Praxis, dass die traditionelle Bewertungsprofession im Wandel ist. Experten müssen in Zukunft verstärkt Datenkompetenzen aufbauen und bereit sein, mit KI-Systemen zusammenzuarbeiten. Wer diese Entwicklung antizipiert und proaktiv nutzt, kann daher Wettbewerbsvorteile erzielen – etwa durch schnellere Deals, bessere Investmententscheidungen und fundiertere Gutachten. Für die Praxis ist die zentrale Botschaft der Arbeit somit: Maschinelles Lernen wird die Unternehmensbewertung nicht ersetzen, aber erheblich bereichern. Unternehmen und Bewertungsexperten sollten die sich bietenden Potenziale nutzen, dabei aber die Grenzen und Risiken im Blick behalten.

Diese Arbeit kann als Ausgangspunkt für derartige Untersuchungen dienen, da sie eine systematische Bestandsaufnahme von Nutzen und Grenzen liefert. Sie zeigt, dass ML kein reiner Hype, sondern fundiert einsetzbar ist, gleichzeitig aber klassischen finanzwirtschaftlichen Prinzipien nicht entgegenstehen darf. Die Finanzforschung ist eingeladen, auf diesen Erkenntnissen aufzubauen, etwa durch die Entwicklung hybrider Modelle oder durch empirische Langzeitstudien, die die Performance von ML-Bewertungen im Zeitverlauf und in verschiedenen Kontexten beobachten. Zusammenfassend trägt die Arbeit dazu bei, die Brücke zwischen traditioneller Unternehmensbewertungslehre und modernen Datenwissenschaften zu schlagen – ein Brückenschlag, der für die zukünftige Forschung viele Ansatzpunkte bietet.

Die vorliegende Arbeit hat gezeigt, dass der Einsatz von maschinellem Lernen in der Unternehmensbewertung kein abstraktes Zukunftsszenario mehr ist, sondern bereits heute greifbare Vorteile bietet. Gleichzeitig wurde klar, dass klassische Methoden wie das DCF weiterhin einen unverzichtbaren Orientierungsrahmen liefern. Die Kombination beider Welten – traditionelles finanzwirtschaftliches Denken und moderne KI-Tools – verspricht die höchste Güte in der Bewertungsarbeit. Letztlich lässt sich festhalten: Maschinelles Lernen kann die Unternehmensbewertung zwar nicht ersetzen, aber sinnvoll ergänzen und verbessern. Wenn die identifizierten Nutzenpotenziale ausgeschöpft und die Herausforderungen gemeistert werden, kann dies zu präziseren, effizienteren und fundierteren Unternehmensbewertungen führen. Dies wäre sowohl im Interesse der wissenschaftlichen Gemeinschaft als auch – und vor allem – im Interesse aller Akteure auf den Finanzmärkten, die auf verlässliche Unternehmenswerte angewiesen sind. Die Ergebnisse dieser Arbeit bilden einen Schritt in diese Richtung und liefern eine Grundlage für weitere Vertiefung und Anwendung in Forschung und Praxis. Insgesamt unterstreicht die Arbeit die Wichtigkeit der Fragestellung und zeigt auf, dass die Zukunft der Unternehmensbewertung wahrscheinlich in einem integrativen Ansatz liegt, der das Beste aus beiden Welten vereint. Moneyball hat im Baseball gezeigt, was möglich ist, wenn man genauer hinsieht. In der Unternehmensbewertung beginnt ein ähnlicher Wandel – getragen von Daten, Mustern und dem Mut, alte Annahmen zu hinterfragen.

V. Literaturverzeichnis

Aggarwal, C. C. [2018]. Neural Networks and Deep Learning: A Textbook. <http://link.springer.com/10.1007/978-3-319-94463-0>. Cham, Springer International Publishing.

Ang, Y. Q., Chia, A., & Saghafian, S. [2020]. Using Machine Learning to Demystify Startups Funding, Post-Money Valuation, and Success. https://research.hks.harvard.edu/publications/getFile.aspx?Id=4964&_gl=1*27w13k*_gcl_au*ODcwNDY5MzUwLjE3NTEzMTMzOTU.*_ga*NzYzNTg0MjAyLjE3NTEzMTMzOTU.*_ga_72NC9RC7VN*cze3NTEzMTMzOTUkbzEkZzEkdDE3NTEzMTM0MjQkajMxJGwwJGgyMDU4NDU2OTM.

Ballwieser, W. & Hachmeister, D. Hrsg. [2019]. Digitalisierung und Unternehmensbewertung: neue Objekte, Prozesse, Parametergewinnung. Stuttgart, Schäffer-Poeschel-Verlag.

Becker, H. P. & Peppmeier, A. [2022]. Investition und Finanzierung: Grundlagen der betrieblichen Finanzwirtschaft. Wiesbaden, Springer Gabler.

Bishop, C. M. [2006]. Pattern recognition and machine learning. New York, Springer.

Brealey, R. A., Myers, S. C., Allen, F., & Edmans, A. [2023]. Principles of corporate finance. New York, NY, McGraw Hill.

Breiman, L. [2001]. Random Forests. <http://link.springer.com/10.1023/A:1010933404324>.

Buxmann, P. [2021]. Künstliche Intelligenz: Mit Algorithmen Zum Wirtschaftlichen Erfolg. Berlin, Heidelberg, Springer Berlin / Heidelberg.

Buxmann, P. & Schmidt, H. Hrsg. [2021]. Künstliche Intelligenz: Mit Algorithmen zum wirtschaftlichen Erfolg. <https://link.springer.com/10.1007/978-3-662-61794-6>. Berlin, Heidelberg, Springer Berlin Heidelberg.

Calhoun, K. & Harkins, D. W. R. [2021]. Understand the Discount Rate Used in a Business Valuation. <https://mercercapital.com/article/understand-the-discount-rate-used-in-a-business-valuation/>.

Chang, W. & Zhang, T. [2025]. Machine learning in corporate M&A valuation: an empirical study based on big data.

<https://combinatorialpress.com/article/jcmcc/Volume%20127/127ap2/Machine%20learning%20in%20corporate%20M&A%20valuation%20an%20empirical%20study%20based%20on%20big%20data.pdf>.

Chris Oxborough, Townsend, A., Westermann, C., Cameron, E., & Rao, D., Anand [2019]. Explainable AI. <https://www.pwc.com/m1/en/services/assurance/risk-assurance/documents/explainable-ai.pdf>.

Crasselt, N., Lukas, E., Mölls, S. H., & Timmreck, C. Hrsg. [2018]. Handbuch Kapitalmarktorientierte Unternehmensbewertung: Grundlagen, Methoden, Regulierung und Branchentrends. Freiburg, Schäffer-Poeschel Verlag für Wirtschaft Steuern Recht GmbH.

Damodaran, A. [2012]. Investment valuation: tools and techniques for determining the value of any asset. Hoboken, N.J, Wiley.

Dixon, M. F., Halperin, I., & Bilokon, P. [2020]. Machine learning in finance: from theory to practice. Cham, Switzerland, Springer.

Domingos, P. [2012]. A few useful things to know about machine learning. <https://dl.acm.org/doi/10.1145/2347736.2347755>.

Drukarczyk, J. & Schüler, A. [2021]. Unternehmensbewertung. München, Verlag Franz Vahlen.

Eayrs, W. E. [2011]. Corporate Finance Training: Planung, Bewertung und Finanzierung von Unternehmen. Freiburg, Schäffer-Poeschel Verlag für Wirtschaft Steuern Recht GmbH.

Ertel, W. [2008]. Grundkurs künstliche Intelligenz: eine praxisorientierte Einführung. Wiesbaden, Vieweg.

Fama, E. F. [1970]. Efficient Capital Markets: A Review of Theory and Empirical Work. <https://www.jstor.org/stable/2325486?origin=crossref>.

Friedman, J. H. [2001]. Greedy function approximation: A gradient boosting machine. <https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full>.

- Friedman, J. H.** [2001]. Greedy function approximation: A gradient boosting machine. <https://projecteuclid.org/journals/annals-of-statistics/volume-29/issue-5/Greedy-function-approximation-A-gradient-boosting-machine/10.1214/aos/1013203451.full>.
- Géron, A.** [2019]. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: concepts, tools, and techniques to build intelligent systems. Beijing [China] ; Sebastopol, CA, O'Reilly Media, Inc.
- Geyer, H.** [2013]. Praxiswissen BWL - mit Arbeitshilfen online: Crashkurs für Führungskräfte und Quereinsteiger. Stuttgart, Haufe.
- Goodfellow, I., Courville, A., & Bengio, Y.** [2016]. Deep learning. Cambridge, Massachusetts, The MIT Press.
- Grobys, K., Kolari, J. W., & Niang, J.** [2022]. Man versus machine: on artificial intelligence and hedge funds performance. <https://www.tandfonline.com/doi/full/10.1080/00036846.2022.2032585>.
- Grunert, P.** [2021]. Machine Learning und Neuronale Netze: Der verständliche Einstieg mit Python. Landshut, BMU Verlag.
- Guszcza, J. et al.** [2020]. Realizing the full potential of AI in the workplace. <https://www.deloitte.com/us/en/insights/focus/technology-and-the-future-of-work/ai-in-the-workplace.html>.
- Hastie, T. J., Tibshirani, R., & Friedman, J. H.** [2009]. The elements of statistical learning: data mining, inference, and prediction. New York, Springer.
- Heesen, B.** [2021]. Basiswissen Unternehmensbewertung: Schneller Einstieg in Die Wertermittlung. Wiesbaden, Springer Fachmedien Wiesbaden GmbH.
- Herr, T. & Arias, M.** [2025]. The transformative potential of GenAI in transfer pricing and valuation. <https://kpmg.com/kpmg-us/content/dam/kpmg/pdf/2025/transformative-potential-genai-transfer-pricing-valuation.pdf>.
- High-Level Expert Group on Artificial Intelligence** [2019]. Ethics Guidelines for Trustworthy AI. <https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai>.
- Hyndman, R. J. & Athanasopoulos, G.** [2021]. Forecasting: principles and practice. Melbourne, Australia, Otexts, Online Open-Access Textbooks.

- Institut der Wirtschaftsprüfer in Deutschland** [2017]. IDW-Standard: Grundsätze zur Durchführung von Unternehmensbewertungen: IDW S 1. Düsseldorf, IDW-Verlag.
- ISO/IEC** [2015]. ISO/IEC 25024:2015. <https://www.iso.org/standard/35749.html>.
- James, G., Witten, D., Hastie, T., & Tibshirani, R.** [2021]. An introduction to statistical learning: with applications in R. New York, Springer.
- Knabe, Dr. M. & Rondorf, S.** [2023]. Factoring in Discounted Cash Flow-Bewertungen. <https://www.deloitte.com/de/de/issues/growth-competition/factoring-in-discounted-cash-flow-bewertungen.html>.
- Krishnapuram, B.** [2016]. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, ACM.
- Kruschwitz, L.** [2019]. Investitionsrechnung. Berlin/München/Boston, Walter de Gruyter GmbH.
- Kuhn, M. & Johnson, K.** [2013]. Applied predictive modeling. New York, Springer.
- Kuhn, M. & Johnson, K.** [2021]. Feature engineering and selection: a practical approach for predictive models. Boca Raton London New York, CRC Press, Taylor & Francis Group.
- Küting, P.** [2015]. Die Bilanzanalyse: Beurteilung von Abschlüssen nach HGB und IFRS. Freiburg, Schäffer-Poeschel Verlag für Wirtschaft Steuern Recht GmbH.
- Lewis, M.** [2004]. Moneyball: the art of winning an unfair game ; [with a new afterword]. New York, NY, Norton.
- Luxburg, U. von, Guyon, I., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S. V. N., Garnett, R., & Neural Information Processing Systems Foundation Hrsg.** [2018]. Advances in neural information processing systems 30: 31st Annual Conference on Neural Information Processing Systems (NIPS 2017): Long Beach, California, USA, 4-9 December 2017. Red Hook, NY, Curran Associates, Inc.
- McCulloch, W. S. & Pitts, W.** [1943]. A logical calculus of the ideas immanent in nervous activity. <http://link.springer.com/10.1007/BF02478259>.

Mehlig, B. [2022]. Machine learning with neural networks: an introduction for scientists and engineers. Cambridge, United Kingdom New York, NY Port Melbourne, Australia New Delhi, India Singapore, Cambridge University Press.

Molnar, C. [2022]. Interpretable machine learning: a guide for making black box models explainable. Munich, Germany, Christoph Molnar.

Moro Visconti, R. [2024]. Artificial intelligence valuation: the impact on automation, BioTech, ChatBots, FinTech, B2B2C, and other industries. Cham, Palgrave Macmillan.

Moxter, A. & Engel-Ciric, D. [2019]. Grundsätze ordnungsgemäßer Bilanzierung: §§ 246-256a HGB. Düsseldorf, IDW Verlag GmbH.

Murphy, K. P. [2014]. Machine Learning - A Probabilistic Perspective. Cambridge, MIT Press.

O.V. [2022]. IFRS STANDARDS. LONDON, IFRS FOUNDATION.

O.V. [2023]. IDW PS: Prüfung von KI-Systemen (IDW PS 861 (03.2023) Stand: 10.03.2023. Düsseldorf, IDW Verlag GmbH.

O.V. [2024]. INTERNATIONAL VALUATION STANDARDS. LONDON, INTL VALUATION STANDARDS.

O.V. [2025]. Price Optimization & Neural Networks. <https://pros.com/price-optimization-neural-networks/>.

O.V. [o.J.]. HGB.

O.V. [o.J.]. BewG.

O.V. [o.J.]. UmwG.

O.V. [o.J.]. AktG.

O.V. [o.J.]. Exchange Act Reporting and Registration. <https://www.sec.gov/resources-small-businesses/going-public/exchange-act-reporting-registration>.

Paaß, G. & Hecker, D. [2020]. Künstliche Intelligenz: Was steckt hinter der Technologie der Zukunft? <http://link.springer.com/10.1007/978-3-658-30211-5>. Wiesbaden, Springer Fachmedien Wiesbaden.

Pedregosa, Fabian, Varoux, Gael, Gramfort, Alexander, Michel, Michel, & Bertrand, Thirion [2011]. Scikit-learn: Machine Learning in Python. <https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf>.

Peemöller, V. H. Hrsg. [2023]. Praxishandbuch der Unternehmensbewertung: Grundlagen und Methoden ; Bewertungsverfahren ; Besonderheiten bei der Bewertung. Herne, NWB Verlag.

Perridon, L., Steiner, M., & Rathgeber, A. W. [2022]. Finanzwirtschaft der Unternehmung. München, Verlag Franz Vahlen.

Probst, P., Wright, M. N., & Boulesteix, A. [2019]. Hyperparameters and tuning strategies for random forest. <https://wires.onlinelibrary.wiley.com/doi/10.1002/widm.1301>.

Pronobis, P. [2022]. Das Umsatzkostenverfahren im internationalen Vergleich: Beschreibung des Aufbaus sowie der einzelnen Posten nach HGB, IFRS und US-GAAP. Berlin, Duncker & Humblot.

Rashid, T. [2024]. Neuronale Netze selbst programmieren: ein verständlicher Einstieg mit Python. Heidelberg, O'Reilly®.

Rothe, R., Merantix, A., & Merantix, M. [2023]. Wird AI die Banker ersetzen? <https://live.handelsblatt.com/wird-ai-die-banker-ersetzen/>.

Russell, S. J. & Norvig, P. [2003]. Artificial intelligence: a modern approach ; [the intelligent agent book]. Upper Saddle River, NJ, Prentice Hall.

Schmeisser, W., Görlitz, B., Spree, J., Clausen, L., & Schindler, F. Hrsg. [2009]. Einführung in die Unternehmensbewertung. <https://doi.org/10.1688/9783866182899>. DE, Rainer Hampp Verlag.

Seppelfricke, P. [2020]. Unternehmensbewertungen: Methoden, Übersichten und Fakten für Praktiker. Stuttgart, Schäffer-Poeschel Verlag.

Shiller, R. J. [2019]. Narrative economics: how stories go viral & drive major economic events. Princeton, Princeton University press.

Shu, M., Song, R., & Zhu, W. [2021]. The „COVID“ Crash of the 2020 U.S. Stock Market. <https://arxiv.org/abs/2101.03625>. arXiv.

Skansi, S. [2018]. Introduction to Deep Learning: From Logical Calculus to Artificial Intelligence. <http://link.springer.com/10.1007/978-3-319-73004-2>. Cham, Springer International Publishing.

Valentine, J. J. [2011]. Best practices for equity research analysts: essentials for buy-side and sell-side analysts. New York, McGraw-Hill.

Wittpahl, V. Hrsg. [2019]. Künstliche Intelligenz: Technologie | Anwendung | Gesellschaft. <http://link.springer.com/10.1007/978-3-662-58042-4>. Berlin, Heidelberg, Springer Berlin Heidelberg.

Wöltje, J. [2021]. Schnelleinstieg Unternehmensbewertung und Finanzkennzahlen. <https://www.beck-elibrary.de/index.php?doi=10.34157/9783648147221>. Haufe Lexware.

VI. Anhang

```
import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor

# Daten laden
df = pd.read_csv("output/Training Data.csv")
df = df[df["Market Cap"] > 0]
df["Market_Cap_log"] = np.log(df["Market Cap"])
df = df[df["Year"] <= 2020]

# Zielnahe und ID-Spalten ausschließen (Market Cap, Enterprise Value, etc)
excluded_cols = [
    "Ticker", "Market_Cap_lag_1", "Market_Cap_delta", "Log_MarketCap",
    "Log_MarketCap_lag_1", "EV-to-EBITDA", "Log_MarketCap_rate"
    "EV-to-Sales", "Price-to-Cash-Flow", "EV-to-Operating-Cash-Flow_lag_1",
    "EV-to-Sales_delta",
    "EV-to-EBITDA_lag_1", "Price-to-Book", "EV-to-EBIT_lag_1", "Price-to-
    Book_lag_1", "Price-to-Free-Cash-Flow",
    "Price-to-Book_delta", "EV-to-Sales", "Price-to-Book_delta",
    "Log_MarketCap_rate",
    "Price-to-Earnings_lag_1", "Price-to-Free-Cash-Flow_lag_1", "Price-to-
    Earnings", "Price-to-Free-Cash-Flow_lag_1",
    "Price-to-Earnings", "EV-to-EBIT", "Price-to-Cash-Flow_lag_1", "EV-to-
    Sales_lag_1", "EV-to-Operating-Cash-Flow",
    "MarketCap_lag_1", "Market_Cap_lag_1", "Market_Cap_rate", "Market Cap",
    "Market_Cap_log", "is_test"
]

features = [col for col in df.columns if col not in excluded_cols]

# Trainingsdaten bereinigen
train_df = df.copy()
X = train_df[features]
y = np.log(train_df["Market Cap"])
X_clean = X.dropna()
y_clean = y.loc[X_clean.index]

# Random Forest trainieren
model = RandomForestRegressor(
    n_estimators=300,
    random_state=42,
    n_jobs=-1 # nutzt alle verfügbaren CPU-Kerne
)

model.fit(X_clean, y_clean)

# Feature Importance extrahieren
importances = model.feature_importances_
importance_df = pd.DataFrame({
    "Feature": X_clean.columns,
    "Importance": importances
}).sort_values(by="Importance", ascending=False)
```

```

#
selected_temp_features_df = importance_df[importance_df["Importance"] > 0]
print(selected_temp_features_df)
# Korrelationen zusätzlich prüfen
selected_features = selected_temp_features_df["Feature"].tolist()
# Filter auf ausgewählte Features
X_selected = X_clean[selected_features]
# 1. Heatmap der Korrelationsmatrix
corr_matrix = X_selected.corr()
# 2. Mini-Check: Feature-Paare mit hoher Korrelation (> 0.9)
high_corr_pairs = (
    corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
    .stack()
    .reset_index()
)
high_corr_pairs.columns = ["Feature_1", "Feature_2", "Korrelation"]
high_corr_pairs = high_corr_pairs[high_corr_pairs["Korrelation"].abs() > 0.9]
print("🔍 Feature-Paare mit |Korrelation| > 0.9:")
print(high_corr_pairs)
redundant_features = high_corr_pairs["Feature_2"].unique().tolist()
#Neuen Random Forest Run durchführen dieses Mal gelistete Redundant Features
ausgeschlossen
features = [col for col in df.columns if col not in excluded_cols and col not
in redundant_features]
# Trainingsdaten bereinigen
train_df = df.copy()
X = train_df[features]
y = np.log(train_df["Market Cap"])
X_clean = X.dropna()
y_clean = y.loc[X_clean.index]
# Random Forest trainieren
model = RandomForestRegressor(n_estimators=300, random_state=42, n_jobs=-1)
model.fit(X_clean, y_clean)
# Feature Importance extrahieren
importances = model.feature_importances_
importance_df = pd.DataFrame({
    "Feature": X_clean.columns,
    "Importance": importances
}).sort_values(by="Importance", ascending=False)
# Nur Features mit >1% Wichtigkeit behalten
selected_features_df = importance_df[importance_df["Importance"] > 0.01]
print(selected_features_df)
# Finale Liste der zu verwendenden Features (als Array für Copy/Paste)
final_features = selected_features_df["Feature"].tolist()
print("\n Final selected features (Copy-Paste ready):\n")
print(final_features)
# In Prozent umrechnen und sortieren

```

```

importance_df["Importance_pct"] = 100 * importance_df["Importance"]
importance_df = importance_df.sort_values(by="Importance_pct",
ascending=False)
# Auf 2 Dezimalstellen runden
importance_df["Importance_pct"] = importance_df["Importance_pct"].round(2)
# Anzeigen
print(importance_df)
import pandas as pd
import numpy as np
import financedatabase as fd
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_absolute_error
from IPython.display import display
from sklearn.model_selection import RandomizedSearchCV

# -----
# 1) Globale Anzeigeoptionen:
# - Zahlen mit 2 Dezimalstellen, Punkt als Tausendertrennzeichen und Komma
als Dezimaltrennzeichen
pd.options.display.float_format = lambda x: f"{x:,.2f}".replace('.',
'X').replace(',', '.').replace('X', ',')

# - Funktion, um Prozentwerte (z.B. 12.3456) als "12,35 %" auszugeben
pct_fmt = lambda x: f"{x:,.2f}".replace('.', 'X').replace(',',
'.').replace('X', ',') + " %"

# -----
# 2) FinanceDatabase einbinden und Metadaten laden
# Stelle sicher, dass du 'pip install financedatabase' ausgeführt hast
equities_db = fd.Equities()
# hole alle Equity-Metadaten (inkl. symbol, sector, industry, ...)
df_fd = equities_db.select()

# symbol ist im Index, daher zuerst Reset des Indexes, dann Spalten
auswählen
df_meta = df_fd.reset_index()[['symbol', 'name', 'sector',
'industry']].rename(columns={'symbol': 'Ticker'})

# -----
# 3) Haupt-Datensatz einlesen
df = pd.read_csv('output/Total Data_Set.csv')
bad_tickers = ['ADS', 'ANTM', 'CNA', 'JOE', 'KBR', 'ORI', 'ROLL'] # Es wurden
nachträglich 6 Ticker mit unvollständigen Daten gefunden, diese wurden
rausgeworfen
df = df[~df['Ticker'].isin(bad_tickers)] #
# 4) Features definieren
feature_list = [

```

```

    'EBITDA',
    'Income Before Tax_lag_1',
    'Stock Based Compensation',
    'Free Cash Flow',
    'Gross Profit_lag_1',
    'Earnings Yield',
    'Weighted Average Shares Diluted',
    'Total Current Liabilities_lag_1',
    'Earnings Yield_lag_1',
    'Free Cash Flow Yield',
    'Free Cash Flow_lag_1',
    'Cumulative Return'
]

# -----
# 5) Vorverarbeitung: Nur positive Market Cap und Log-Target
df = df[df['Market Cap'] > 0].copy()
df['Market_Cap_log'] = np.log(df['Market Cap'])

# 6) Trainingsdaten (is_test == False und Year <= 2020) vorbereiten
df_train = df[(df["is_test"] == False) & (df["Year"] <= 2020)].copy()
df_train = df_train.dropna(subset=feature_list + ['Market_Cap_log'])

X_train = df_train[feature_list]
y_train = df_train['Market_Cap_log']

# -----
# 7) Random-Forest-Modell trainieren
rf = RandomForestRegressor(
    n_estimators=400,
    max_depth=None,
    random_state=42,
    n_jobs=-1
)
rf.fit(X_train, y_train)

# -----
# 8) Evaluation für die Jahre 2021, 2022, 2023
results_per_year = {}
metrics_per_year = {}
best_worst_per_year = {}

for year in [2021, 2022, 2023]:
    # Testdaten für das jeweilige Jahr filtern (nur is_test == True)
    df_test = df[(df["Year"] == year) & (df["is_test"] == True)].copy()
    df_test = df_test.dropna(subset=feature_list + ['Market_Cap_log'])

```

```

X_test = df_test[feature_list]
y_true_log = df_test['Market_Cap_log']

# Vorhersage auf Log-Skala
y_pred_log = rf.predict(X_test)

# Rücktransformation in Original-Skala
df_test['Predicted_MarketCap'] = np.exp(y_pred_log)
df_test['True_MarketCap']      = df_test['Market_Cap']
df_test['Abw_pct']             = 100 * (
    df_test['Predicted_MarketCap'] - df_test['True_MarketCap']
) / df_test['True_MarketCap']
# Absolute Abweichung in Prozent
df_test['Abs_Abw_pct'] = df_test['Abw_pct'].abs()

# Kennzahlen berechnen
r2    = r2_score(y_true_log, y_pred_log)
mape  = np.mean(
    np.abs(df_test['Predicted_MarketCap'] - df_test['True_MarketCap']) /
    df_test['True_MarketCap']
) * 100
mae   = mean_absolute_error(
    df_test['True_MarketCap'],
    df_test['Predicted_MarketCap']
)

# Weighted MAPE (WMAPE): Summe der absoluten Fehler geteilt durch
Summe der echten Werte
df_test['Abs_Error'] = np.abs(df_test['Predicted_MarketCap'] -
df_test['True_MarketCap'])
wmape = df_test['Abs_Error'].sum() / df_test['True_MarketCap'].sum() * 100
# Pro-Ticker-Übersicht komplett sortiert nach absoluter Abweichung
ticker_summary = df_test[[
    'Ticker', 'True_MarketCap', 'Predicted_MarketCap', 'Abw_pct',
'Abs_Abw_pct'
]].copy()
ticker_summary = ticker_summary.sort_values(
    by='Abw_pct',
    ascending=False
).reset_index(drop=True)

# Top 20 schlechteste (höchste |Abw_pct|)
worst20 = ticker_summary.head(20).copy()
# Top 20 beste (niedrigste |Abw_pct|)
best20  = ticker_summary.nsmallest(20, 'Abs_Abw_pct').copy()

# -----
# 9) Merge der Top-20-Tabellen mit FinanceDatabase-Metadaten

```

```

    worst20 = worst20.merge(df_meta, on='Ticker', how='left')
    best20 = best20.merge(df_meta, on='Ticker', how='left')

    # Spaltenreihenfolge anpassen: Ticker, Sector, Industry, True, Predicted,
    Abw_pct
    worst20 = worst20[[
        'Ticker', 'name', 'sector', 'industry', 'True_MarketCap',
        'Predicted_MarketCap', 'Abw_pct'
    ]]
    best20 = best20[[
        'Ticker', 'name', 'sector', 'industry', 'True_MarketCap',
        'Predicted_MarketCap', 'Abw_pct'
    ]]

    # Speichern (für spätere Anzeige)
    best_worst_per_year[year] = {
        'best20': best20,
        'worst20': worst20
    }

    results_per_year[year] = ticker_summary
    metrics_per_year[year] = {
        'R2': r2,
        'MAPE (%)': mape,
        'MAE': mae,
        'WMAPE (%)': wmape
    }

# -----
# 10) Metriken in DataFrame zusammenführen
metrics_df = pd.DataFrame.from_dict(metrics_per_year, orient='index')
metrics_df.index.name = 'Year'

# 11) Alle Abweichungen pro Jahr in einer Tabelle kombinieren (inkl. Meta)
all_errors_list = []
for year, df_err in results_per_year.items():
    tmp = df_err.merge(df_meta, on='Ticker', how='left')
    tmp['Year'] = year
    all_errors_list.append(tmp)
all_errors_df = pd.concat(all_errors_list, ignore_index=True)

# -----
# 12) Formatierte DataFrames für die Anzeige vorbereiten

# 12a) Metriken formatieren: R2 und MAE bleiben numerisch, MAPE_pct und
WMAPE_pct als Prozentstring
metrics_display = metrics_df.copy()
metrics_display['MAPE (%)'] = metrics_display['MAPE (%)'].apply(pct_fmt)

```

```

metrics_display['WMAPE (%)'] = metrics_display['WMAPE (%)'].apply(pct_fmt)

# 12b) Funktion, um Ticker-Tabellen (best/worst/alle) vor der Anzeige zu
formatieren
def format_ticker_display(df_in):
    df_out = df_in.copy()
    # True_MarketCap und Predicted_MarketCap: Tausenderpunkt + Komma als
    Dezimal
    df_out['True_MarketCap'] = df_out['True_MarketCap'].map(
        lambda x: f"{x:,.2f}".replace('.', 'X').replace(',', '.').replace('X',
        ',')
    )
    df_out['Predicted_MarketCap'] = df_out['Predicted_MarketCap'].map(
        lambda x: f"{x:,.2f}".replace('.', 'X').replace(',', '.').replace('X',
        ',')
    )
    # Abw_pct als Prozentstring
    df_out['Abw_pct'] = df_out['Abw_pct'].apply(pct_fmt)
    # Year als Integer (falls vorhanden)
    if 'Year' in df_out.columns:
        df_out['Year'] = df_out['Year'].astype(int)
    return df_out

# -----
# 13) Ergebnisse anzeigen (hübsche Tabellen mit display)

# 13a) Kennzahlen-Tabelle
print("=== Kennzahlen (R2, MAPE (%), MAE) pro Jahr ===")
display(metrics_display)
print()

# 13b) Top 20 beste und schlechteste Ticker für jedes Jahr (inkl.
    Sektor/Industrie)
for year in [2021, 2022, 2023]:
    print(f"=== {year}: Top 20 Schlechteste Ticker (größte absolute
    Abweichung) ===")
    display(format_ticker_display(best_worst_per_year[year]['worst20']))
    print()

# 13c) Top 20 beste ker für jedes Jahr (inkl. Sektor/Industrie)
for year in [2021, 2022, 2023]:
    print(f"=== {year}: Top 20 Beste Ticker (kleinste absolute Abweichung)
    ===")
    display(format_ticker_display(best_worst_per_year[year]['best20']))
    print()

# 13c) Gesamtübersicht: Alle Jahre, sortiert nach Jahr und absteigender
    Abweichung

```

```

print("=== Alle Jahre: Abweichungen je Ticker inklusive Sektor/Industrie ===")
display(
    format_ticker_display(all_errors_df).sort_values(
        ['Year', 'Abs_Abw_pct'],
        import pandas as pd
import numpy as np
from sklearn.ensemble import GradientBoostingRegressor
# Daten laden
df = pd.read_csv("output/Training Data.csv")
df = df[df["Market Cap"] > 0]
df["Market_Cap_log"] = np.log(df["Market Cap"])
df = df[df["Year"] <= 2020]
# Zielnahe und ID-Spalten ausschließen
excluded_cols = [
    "Ticker", "Market_Cap_lag_1", "Market_Cap_delta", "Log_MarketCap",
    "Log_MarketCap_lag_1", "EV-to-EBITDA", "Log_MarketCap_rate"
    "EV-to-Sales", "Price-to-Cash-Flow", "EV-to-Operating-Cash-Flow_lag_1",
    "EV-to-Sales_delta",
    "EV-to-EBITDA_lag_1", "Price-to-Book", "EV-to-EBIT_lag_1", "Price-to-
Book_lag_1", "Price-to-Free-Cash-Flow",
    "Price-to-Book_delta", "EV-to-Sales", "Price-to-Book_delta",
    "Log_MarketCap_rate",
    "Price-to-Earnings_lag_1", "Price-to-Free-Cash-Flow_lag_1", "Price-to-
Earnings", "Price-to-Free-Cash-Flow_lag_1",
    "Price-to-Earnings", "EV-to-EBIT", "Price-to-Cash-Flow_lag_1", "EV-to-
Sales_lag_1", "EV-to-Operating-Cash-Flow",
    "MarketCap_lag_1", "Market_Cap_lag_1", "Market_Cap_rate", "Market Cap",
    "Market_Cap_log", "is_test"
]
features = [col for col in df.columns if col not in excluded_cols]
# Trainingsdaten bereinigen
train_df = df.copy()
X = train_df[features]
y = np.log(train_df["Market Cap"])
X_clean = X.dropna()
y_clean = y.loc[X_clean.index]
# Gradient Boosting trainieren
model = GradientBoostingRegressor(
    n_estimators=300,
    random_state=42,
    max_depth=4, # vergleichbar mit default RF
    learning_rate=0.1
)
model.fit(X_clean, y_clean)
# Feature Importance extrahieren
importances = model.feature_importances_
importance_df = pd.DataFrame({

```

```

        "Feature": X_clean.columns,
        "Importance": importances
    }).sort_values(by="Importance", ascending=False)
# Nur Features mit >0.1% Wichtigkeit behalten
selected_temp_features_df = importance_df[importance_df["Importance"] > 0.001]
print(selected_temp_features_df)
# Korrelationen zusätzlich prüfen
selected_features = selected_temp_features_df["Feature"].tolist()
# Filter auf ausgewählte Features
X_selected = X_clean[selected_features]
# 1. Heatmap der Korrelationsmatrix (optional)
corr_matrix = X_selected.corr()
# 2. Feature-Paare mit hoher Korrelation
high_corr_pairs = (
    corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
    .stack()
    .reset_index()
)
high_corr_pairs.columns = ["Feature_1", "Feature_2", "Korrelation"]
high_corr_pairs = high_corr_pairs[high_corr_pairs["Korrelation"].abs() > 0.9]
print("🔍 Feature-Paare mit |Korrelation| > 0.9:")
print(high_corr_pairs)
redundant_features = high_corr_pairs["Feature_2"].unique().tolist()
# Neue Auswahl ohne redundante Features
features = [col for col in df.columns if col not in excluded_cols and col not
in redundant_features]
# Trainingsdaten neu bereinigen
X = train_df[features]
y = np.log(train_df["Market Cap"])
X_clean = X.dropna()
y_clean = y.loc[X_clean.index]
# Gradient Boosting erneut trainieren
model = GradientBoostingRegressor(
    n_estimators=100,
    random_state=42,
    max_depth=4,
    learning_rate=0.1
)
model.fit(X_clean, y_clean)
# Feature Importance extrahieren
importances = model.feature_importances_
importance_df = pd.DataFrame({
    "Feature": X_clean.columns,
    "Importance": importances
}).sort_values(by="Importance", ascending=False)

# Nur Features mit >1% Wichtigkeit behalten

```

```

selected_features_df = importance_df[importance_df["Importance"] > 0.01]
print(selected_features_df)
# Finale Feature-Liste
final_features = selected_features_df["Feature"].tolist()
print("\n Final selected features (Copy-Paste ready):\n")
print(final_features)

import pandas as pd
import numpy as np
import financedatabase as fd
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import r2_score, mean_absolute_error
from IPython.display import display

# -----
# 1) Globale Anzeigeoptionen:
# - Zahlen mit 2 Dezimalstellen, Punkt als Tausendertrennzeichen und Komma
  als Dezimaltrennzeichen
pd.options.display.float_format = lambda x: f"{x:,.2f}".replace('.',
'X').replace(',', ' ').replace('X', ',')

# - Funktion, um Prozentwerte (z.B. 12.3456) als "12,35 %" auszugeben
pct_fmt = lambda x: f"{x:,.2f}".replace('.', 'X').replace(',',
' ').replace('X', ',') + " %"

# -----
# 2) FinanceDatabase einbinden und Metadaten laden
# Stelle sicher, dass du 'pip install financedatabase' ausgeführt hast
equities_db = fd.Equities()
df_fd = equities_db.select()

# symbol ist im Index, daher Index zurücksetzen und nur
symbol/name/sector/industry auswählen
df_meta = df_fd.reset_index()[['symbol', 'name', 'sector',
'industry']].rename(columns={'symbol': 'Ticker'})

# -----
# 3) Haupt-Datensatz einlesen
df = pd.read_csv('output/Total Data_Set.csv')
bad_tickers = ['ADS', 'ANTM', 'CNA', 'JOE', 'KBR', 'ORI', 'ROLL'] # Es wurden
nachträglich 7 Ticker mit unvollständigen Daten gefunden, diese wurden
rausgeworfen
df = df[~df['Ticker'].isin(bad_tickers)] #
# 4) Features definieren
feature_list = ['EBITDA',
                'Income Before Tax_lag_1',
                'Cash Flow from Operations',
                'Stock Based Compensation',
                'Weighted Average Shares Diluted',
                'Free Cash Flow',

```

```

        'Total Equity',
        'Cumulative Return',
        'Free Cash Flow Yield',
        'Operating Cash Flow_lag_1',
        'Earnings Yield',
        'Total Current Liabilities_lag_1',
        'Earnings Yield_lag_1',
        'Total Shareholder Equity',
        'Gross Profit_lag_1']

# -----
# 5) Vorverarbeitung: Nur positive Market Cap und Log-Target
df = df[df['Market Cap'] > 0].copy()
df['Market_Cap_log'] = np.log(df['Market Cap'])

# 6) Trainingsdaten (is_test == False und Year <= 2020) vorbereiten
df_train = df[(df["is_test"] == False) & (df["Year"] <= 2020)].copy()
df_train = df_train.dropna(subset=feature_list + ['Market_Cap_log'])

X_train = df_train[feature_list]
y_train = df_train['Market_Cap_log']
# -----
# 7) Gradient-Boosting-Modell trainieren
gb = GradientBoostingRegressor(
    n_estimators=400,
    max_depth=4,
    learning_rate=0.1,
    random_state=42
)
gb.fit(X_train, y_train)
# -----
# 8) Evaluation für die Jahre 2021, 2022, 2023
results_per_year = {}
metrics_per_year = {}
best_worst_per_year = {}
for year in [2021, 2022, 2023]:
    # Testdaten für das jeweilige Jahr filtern (nur is_test == True)
    df_test = df[(df["Year"] == year) & (df["is_test"] == True)].copy()
    df_test = df_test.dropna(subset=feature_list + ['Market_Cap_log'])
    X_test = df_test[feature_list]
    y_true_log = df_test['Market_Cap_log']
    # Vorhersage auf Log-Skala
    y_pred_log = gb.predict(X_test)
    # Rücktransformation in Original-Skala
    df_test['Predicted_MarketCap'] = np.exp(y_pred_log)
    df_test['True_MarketCap'] = df_test['Market Cap']
    df_test['Abw_pct'] = 100 * (
        df_test['Predicted_MarketCap'] - df_test['True_MarketCap']

```

```

) / df_test['True_MarketCap']
# Absolute Abweichung in Prozent
df_test['Abs_Abw_pct'] = df_test['Abw_pct'].abs()
# Kennzahlen berechnen
r2 = r2_score(y_true_log, y_pred_log)
mape = np.mean(
    np.abs(df_test['Predicted_MarketCap'] - df_test['True_MarketCap']) /
    df_test['True_MarketCap']
) * 100
mae = mean_absolute_error(
    df_test['True_MarketCap'],
    df_test['Predicted_MarketCap']
)

# Weighted MAPE (WMAPE): Summe der absoluten Fehler geteilt durch
Summe der echten Werte
df_test['Abs_Error'] = np.abs(df_test['Predicted_MarketCap'] -
df_test['True_MarketCap'])
wmape = df_test['Abs_Error'].sum() / df_test['True_MarketCap'].sum() * 100
# Pro-Ticker-Übersicht komplett sortiert nach absoluter Abweichung
ticker_summary = df_test[[
    'Ticker', 'True_MarketCap', 'Predicted_MarketCap', 'Abw_pct',
'Abs_Abw_pct'
]].copy()
ticker_summary = ticker_summary.sort_values(
    by='Abs_Abw_pct',
    ascending=False
).reset_index(drop=True)
# Top 20 schlechteste (höchste |Abw_pct|)
worst20 = ticker_summary.head(20).copy()
# Top 20 beste (niedrigste |Abw_pct|)
best20 = ticker_summary.nsmallest(20, 'Abs_Abw_pct').copy()
# -----
# 9) Merge der Top-20-Tabellen mit FinanceDatabase-Metadaten
# Um Namens-Konflikte zu vermeiden, wählen wir hier nur benötigte
Spalten aus
worst20 = worst20.merge(
    df_meta[['Ticker', 'name', 'sector', 'industry']],
    on='Ticker',
    how='left'
)
best20 = best20.merge(
    df_meta[['Ticker', 'name', 'sector', 'industry']],
    on='Ticker',
    how='left'
)
# Spaltenreihenfolge anpassen: Ticker, Name, Sector, Industry, True,
Predicted, Abw_pct

```

```

    worst20 = worst20[[
        'Ticker', 'name', 'sector', 'industry', 'True_MarketCap',
'Predicted_MarketCap', 'Abw_pct'
    ]]
    best20 = best20[[
        'Ticker', 'name', 'sector', 'industry', 'True_MarketCap',
'Predicted_MarketCap', 'Abw_pct'
    ]]
    # Speichern (für spätere Anzeige)
    best_worst_per_year[year] = {
        'best20': best20,
        'worst20': worst20
    }
    results_per_year[year] = ticker_summary
    metrics_per_year[year] = {
        'R2': r2,
        'MAPE (%)': mape,
        'MAE': mae,
        'WMAPE (%)': wmape
    }
}

# -----
# 10) Metriken in DataFrame zusammenführen
metrics_df = pd.DataFrame.from_dict(metrics_per_year, orient='index')
metrics_df.index.name = 'Year'
# 11) Alle Abweichungen pro Jahr in einer Tabelle kombinieren (inkl. Meta)
all_errors_list = []
for year, df_err in results_per_year.items():
    tmp = df_err.merge(
        df_meta[['Ticker', 'name', 'sector', 'industry']],
        on='Ticker',
        how='left'
    )
    tmp['Year'] = year
    all_errors_list.append(tmp)
all_errors_df = pd.concat(all_errors_list, ignore_index=True)
# -----
# 12) Formatierte DataFrames für die Anzeige vorbereiten
# 12a) Metriken formatieren: R2 und MAE bleiben numerisch, MAPE_pct und
WMAPE_pct als Prozentstring
metrics_display = metrics_df.copy()
metrics_display['MAPE (%)'] = metrics_display['MAPE (%)'].apply(pct_fmt)
metrics_display['WMAPE (%)'] = metrics_display['WMAPE (%)'].apply(pct_fmt)
# 12b) Funktion, um Ticker-Tabellen (best/worst/alle) vor der Anzeige zu
formatieren
def format_ticker_display(df_in):
    df_out = df_in.copy()
    # True_MarketCap und Predicted_MarketCap: Tausenderpunkt + Komma als

```

```

Dezimal
    df_out['True_MarketCap']      = df_out['True_MarketCap'].map(
        lambda x: f"{x:,.2f}".replace('.', 'X').replace(',', '.').replace('X',
        ',')
    )
    df_out['Predicted_MarketCap'] = df_out['Predicted_MarketCap'].map(
        lambda x: f"{x:,.2f}".replace('.', 'X').replace(',', '.').replace('X',
        ',')
    )
    # Abw_pct als Prozentstring
    df_out['Abw_pct']            = df_out['Abw_pct'].apply(pct_fmt)
    # Year als Integer (falls vorhanden)
    if 'Year' in df_out.columns:
        df_out['Year'] = df_out['Year'].astype(int)
    return df_out

# -----
# 13) Ergebnisse anzeigen (hübsche Tabellen mit display)
# 13a) Kennzahlen-Tabelle
print("=== Kennzahlen (R², MAPE (%), MAE) pro Jahr ===")
display(metrics_display)
print()
# 13b) Top 20 Schlechteste Ticker für jedes Jahr (inkl. Name,
Sektor/Industrie)
for year in [2021, 2022, 2023]:
    print(f"=== {year}: Top 20 Schlechteste Ticker (größte absolute
Abweichung) ===")
    display(format_ticker_display(best_worst_per_year[year]['worst20']))
    print()
# 13c) Top 20 Beste Ticker für jedes Jahr (inkl. Name, Sektor/Industrie)
for year in [2021, 2022, 2023]:
    print(f"=== {year}: Top 20 Beste Ticker (kleinste absolute Abweichung)
===")
    display(format_ticker_display(best_worst_per_year[year]['best20']))
    print()

# 13d) Gesamtübersicht: Alle Jahre, sortiert nach Jahr und absteigender
Abweichung
print("=== Alle Jahre: Abweichungen je Ticker inklusive Name, Sektor/Industrie
===")
display(
    format_ticker_display(all_errors_df).sort_values(
        ['Year', 'Abs_Abw_pct'],
        ascending=False
    ).reset_index(drop=True)
)
# ----- Export Gesamt-Results für Modellvergleich -----
---
```

```

# all_errors_df enthält bereits alle Jahre, Abweichungen und Meta-Spalten
(name, sector, industry)
all_errors_df['Model'] = 'Gradient Boosting'
metrics_display['Model'] = 'Gradient Boosting'
# Datei schreiben (Pfad nach Wunsch anpassen)
all_errors_df.to_csv('output/GB_results.csv', index=False)
metrics_display.to_csv('output/GB_Metrics.csv', index=False)

import pandas as pd
import numpy as np
from sklearn.ensemble import GradientBoostingRegressor
# Daten laden
df = pd.read_csv("output/Training Data.csv")
df = df[df["Market Cap"] > 0]
df["Market_Cap_log"] = np.log(df["Market Cap"])
df = df[df["Year"] <= 2020]
# Zielnahe und ID-Spalten ausschließen
excluded_cols = [
    "Ticker", "Market_Cap_lag_1", "Market_Cap_delta", "Log_MarketCap",
    "Log_MarketCap_lag_1", "EV-to-EBITDA", "Log_MarketCap_rate"
    "EV-to-Sales", "Price-to-Cash-Flow", "EV-to-Operating-Cash-Flow_lag_1",
    "EV-to-Sales_delta",
    "EV-to-EBITDA_lag_1", "Price-to-Book", "EV-to-EBIT_lag_1", "Price-to-
    Book_lag_1", "Price-to-Free-Cash-Flow",
    "Price-to-Book_delta", "EV-to-Sales", "Price-to-Book_delta",
    "Log_MarketCap_rate",
    "Price-to-Earnings_lag_1", "Price-to-Free-Cash-Flow_lag_1", "Price-to-
    Earnings", "Price-to-Free-Cash-Flow_lag_1",
    "Price-to-Earnings", "EV-to-EBIT", "Price-to-Cash-Flow_lag_1", "EV-to-
    Sales_lag_1", "EV-to-Operating-Cash-Flow",
    "MarketCap_lag_1", "Market_Cap_lag_1", "Market_Cap_rate", "Market Cap",
    "Market_Cap_log", "is_test"
]
features = [col for col in df.columns if col not in excluded_cols]
# Trainingsdaten bereinigen
train_df = df.copy()
X = train_df[features]
y = np.log(train_df["Market Cap"])
X_clean = X.dropna()
y_clean = y.loc[X_clean.index]
# Gradient Boosting trainieren
model = GradientBoostingRegressor(
    n_estimators=300,
    random_state=42,
    max_depth=4, # vergleichbar mit default RF
    learning_rate=0.1
)
model.fit(X_clean, y_clean)

```

```

# Feature Importance extrahieren
importances = model.feature_importances_
importance_df = pd.DataFrame({
    "Feature": X_clean.columns,
    "Importance": importances
}).sort_values(by="Importance", ascending=False)
# Nur Features mit >0.1% Wichtigkeit behalten
selected_temp_features_df = importance_df[importance_df["Importance"] > 0.001]
print(selected_temp_features_df)
# Korrelationen zusätzlich prüfen
selected_features = selected_temp_features_df["Feature"].tolist()
# Filter auf ausgewählte Features
X_selected = X_clean[selected_features]
# 1. Heatmap der Korrelationsmatrix (optional)
corr_matrix = X_selected.corr()
# 2. Feature-Paare mit hoher Korrelation
high_corr_pairs = (
    corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
    .stack()
    .reset_index()
)
high_corr_pairs.columns = ["Feature_1", "Feature_2", "Korrelation"]
high_corr_pairs = high_corr_pairs[high_corr_pairs["Korrelation"].abs() > 0.9]
print("\n Feature-Paare mit |Korrelation| > 0.9:")
print(high_corr_pairs)
redundant_features = high_corr_pairs["Feature_2"].unique().tolist()
# Neue Auswahl ohne redundante Features
features = [col for col in df.columns if col not in excluded_cols and col not
in redundant_features]
# Trainingsdaten neu bereinigen
X = train_df[features]
y = np.log(train_df["Market Cap"])
X_clean = X.dropna()
y_clean = y.loc[X_clean.index]
# Gradient Boosting erneut trainieren
model = GradientBoostingRegressor(
    n_estimators=100,
    random_state=42,
    max_depth=4,
    learning_rate=0.1
)
model.fit(X_clean, y_clean)
# Feature Importance extrahieren
importances = model.feature_importances_
importance_df = pd.DataFrame({
    "Feature": X_clean.columns,
    "Importance": importances

```

```

}).sort_values(by="Importance", ascending=False)

# Nur Features mit >1% Wichtigkeit behalten
selected_features_df = importance_df[importance_df["Importance"] > 0.01]
print(selected_features_df)
# Finale Feature-Liste
final_features = selected_features_df["Feature"].tolist()
print("\n✅ Final selected features (Copy-Paste ready):\n")
print(final_features)
import pandas as pd
import numpy as np
import finencedatabase as fd
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import r2_score, mean_absolute_error
from IPython.display import display

# -----
# 1) Globale Anzeigeeoptionen:
#   - Zahlen mit 2 Dezimalstellen, Punkt als Tausendertrennzeichen und Komma
#     als Dezimaltrennzeichen
pd.options.display.float_format = lambda x: f"{x:,.2f}".replace('.',
'X').replace(',', '.').replace('X', ',')

#   - Funktion, um Prozentwerte (z.B. 12.3456) als "12,35 %" auszugeben
pct_fmt = lambda x: f"{x:,.2f}".replace('.', 'X').replace(',',
'.').replace('X', ',') + " %"

# -----
# 2) FinanceDatabase einbinden und Metadaten laden
#   Stelle sicher, dass du 'pip install finencedatabase' ausgeführt hast
equities_db = fd.Equities()
df_fd = equities_db.select()

#   symbol ist im Index, daher Index zurücksetzen und nur
#   symbol/name/sector/industry auswählen
df_meta = df_fd.reset_index()[['symbol', 'name', 'sector',
'industry']].rename(columns={'symbol': 'Ticker'})

# -----
# 3) Haupt-Datensatz einlesen
df = pd.read_csv('output/Total Data_Set.csv')
bad_tickers = ['ADS', 'ANTM', 'CNA', 'JOE', 'KBR', 'ORI', 'ROLL'] # Es wurden
nachträglich 7 Ticker mit unvollständigen Daten gefunden, diese wurden
rausgeworfen
df = df[~df['Ticker'].isin(bad_tickers)] #
# 4) Features definieren
feature_list = ['EBITDA',
                'Income Before Tax_lag_1',
                'Cash Flow from Operations',

```

```

        'Stock Based Compensation',
        'Weighted Average Shares Diluted',
        'Free Cash Flow',
        'Total Equity',
        'Cumulative Return',
        'Free Cash Flow Yield',
        'Operating Cash Flow_lag_1',
        'Earnings Yield',
        'Total Current Liabilities_lag_1',
        'Earnings Yield_lag_1',
        'Total Shareholder Equity',
        'Gross Profit_lag_1']

# -----
# 5) Vorverarbeitung: Nur positive Market Cap und Log-Target
df = df[df['Market Cap'] > 0].copy()
df['Market_Cap_log'] = np.log(df['Market Cap'])

# 6) Trainingsdaten (is_test == False und Year <= 2020) vorbereiten
df_train = df[(df["is_test"] == False) & (df["Year"] <= 2020)].copy()
df_train = df_train.dropna(subset=feature_list + ['Market_Cap_log'])

X_train = df_train[feature_list]
y_train = df_train['Market_Cap_log']

# -----
# 7) Gradient-Boosting-Modell trainieren
gb = GradientBoostingRegressor(
    n_estimators=400,
    max_depth=4,
    learning_rate=0.1,
    random_state=42
)
gb.fit(X_train, y_train)

# -----
# 8) Evaluation für die Jahre 2021, 2022, 2023
results_per_year = {}
metrics_per_year = {}
best_worst_per_year = {}
for year in [2021, 2022, 2023]:
    # Testdaten für das jeweilige Jahr filtern (nur is_test == True)
    df_test = df[(df["Year"] == year) & (df["is_test"] == True)].copy()
    df_test = df_test.dropna(subset=feature_list + ['Market_Cap_log'])
    X_test = df_test[feature_list]
    y_true_log = df_test['Market_Cap_log']
    # Vorhersage auf Log-Skala
    y_pred_log = gb.predict(X_test)
    # Rücktransformation in Original-Skala
    df_test['Predicted_MarketCap'] = np.exp(y_pred_log)

```

```

df_test['True_MarketCap']      = df_test['Market Cap']
df_test['Abw_pct']            = 100 * (
    df_test['Predicted_MarketCap'] - df_test['True_MarketCap']
) / df_test['True_MarketCap']
# Absolute Abweichung in Prozent
df_test['Abs_Abw_pct'] = df_test['Abw_pct'].abs()
# Kennzahlen berechnen
r2    = r2_score(y_true_log, y_pred_log)
mape  = np.mean(
    np.abs(df_test['Predicted_MarketCap'] - df_test['True_MarketCap']) /
    df_test['True_MarketCap']
) * 100
mae   = mean_absolute_error(
    df_test['True_MarketCap'],
    df_test['Predicted_MarketCap']
)

# Weighted MAPE (WMAPE): Summe der absoluten Fehler geteilt durch
Summe der echten Werte
df_test['Abs_Error'] = np.abs(df_test['Predicted_MarketCap'] -
df_test['True_MarketCap'])
wmape = df_test['Abs_Error'].sum() / df_test['True_MarketCap'].sum() * 100
# Pro-Ticker-Übersicht komplett sortiert nach absoluter Abweichung
ticker_summary = df_test[[
    'Ticker', 'True_MarketCap', 'Predicted_MarketCap', 'Abw_pct',
'Abs_Abw_pct'
]].copy()
ticker_summary = ticker_summary.sort_values(
    by='Abs_Abw_pct',
    ascending=False
).reset_index(drop=True)
# Top 20 schlechteste (höchste |Abw_pct|)
worst20 = ticker_summary.head(20).copy()
# Top 20 beste (niedrigste |Abw_pct|)
best20  = ticker_summary.nsmallest(20, 'Abs_Abw_pct').copy()
# -----
# 9) Merge der Top-20-Tabellen mit FinanceDatabase-Metadaten
#    Um Namens-Konflikte zu vermeiden, wählen wir hier nur benötigte
Spalten aus
worst20 = worst20.merge(
    df_meta[['Ticker', 'name', 'sector', 'industry']],
    on='Ticker',
    how='left'
)
best20 = best20.merge(
    df_meta[['Ticker', 'name', 'sector', 'industry']],
    on='Ticker',
    how='left'

```

```

)
# Spaltenreihenfolge anpassen: Ticker, Name, Sector, Industry, True,
Predicted, Abw_pct
worst20 = worst20[[
    'Ticker', 'name', 'sector', 'industry', 'True_MarketCap',
'Predicted_MarketCap', 'Abw_pct'
]]
best20 = best20[[
    'Ticker', 'name', 'sector', 'industry', 'True_MarketCap',
'Predicted_MarketCap', 'Abw_pct'
]]
# Speichern (für spätere Anzeige)
best_worst_per_year[year] = {
    'best20': best20,
    'worst20': worst20
}
results_per_year[year] = ticker_summary
metrics_per_year[year] = {
    'R2': r2,
    'MAPE (%)': mape,
    'MAE': mae,
    'WMAPE (%)': wmape
}
}
# -----
# 10) Metriken in DataFrame zusammenführen
metrics_df = pd.DataFrame.from_dict(metrics_per_year, orient='index')
metrics_df.index.name = 'Year'
# 11) Alle Abweichungen pro Jahr in einer Tabelle kombinieren (inkl. Meta)
all_errors_list = []
for year, df_err in results_per_year.items():
    tmp = df_err.merge(
        df_meta[['Ticker', 'name', 'sector', 'industry']],
        on='Ticker',
        how='left'
    )
    tmp['Year'] = year
    all_errors_list.append(tmp)
all_errors_df = pd.concat(all_errors_list, ignore_index=True)
# -----
# 12) Formatierte DataFrames für die Anzeige vorbereiten
# 12a) Metriken formatieren: R2 und MAE bleiben numerisch, MAPE_pct und
WMAPE_pct als Prozentstring
metrics_display = metrics_df.copy()
metrics_display['MAPE (%)'] = metrics_display['MAPE (%)'].apply(pct_fmt)
metrics_display['WMAPE (%)'] = metrics_display['WMAPE (%)'].apply(pct_fmt)
# 12b) Funktion, um Ticker-Tabellen (best/worst/alle) vor der Anzeige zu
formatieren

```

```

def format_ticker_display(df_in):
    df_out = df_in.copy()
    # True_MarketCap und Predicted_MarketCap: Tausenderpunkt + Komma als
    # Dezimal
    df_out['True_MarketCap'] = df_out['True_MarketCap'].map(
        lambda x: f"{x:,.2f}".replace('.', 'X').replace(',', '.').replace('X',
        ',')
    )
    df_out['Predicted_MarketCap'] = df_out['Predicted_MarketCap'].map(
        lambda x: f"{x:,.2f}".replace('.', 'X').replace(',', '.').replace('X',
        ',')
    )
    # Abw_pct als Prozentstring
    df_out['Abw_pct'] = df_out['Abw_pct'].apply(pct_fmt)
    # Year als Integer (falls vorhanden)
    if 'Year' in df_out.columns:
        df_out['Year'] = df_out['Year'].astype(int)
    return df_out

# -----
# 13) Ergebnisse anzeigen (hübsche Tabellen mit display)
# 13a) Kennzahlen-Tabelle
print("=== Kennzahlen (R2, MAPE (%), MAE) pro Jahr ===")
display(metrics_display)
print()
# 13b) Top 20 Schlechteste Ticker für jedes Jahr (inkl. Name,
# Sektor/Industrie)
for year in [2021, 2022, 2023]:
    print(f"=== {year}: Top 20 Schlechteste Ticker (größte absolute
    Abweichung) ===")
    display(format_ticker_display(best_worst_per_year[year]['worst20']))
    print()
# 13c) Top 20 Beste Ticker für jedes Jahr (inkl. Name, Sektor/Industrie)
for year in [2021, 2022, 2023]:
    print(f"=== {year}: Top 20 Beste Ticker (kleinste absolute Abweichung)
    ===")
    display(format_ticker_display(best_worst_per_year[year]['best20']))
    print()

# 13d) Gesamtübersicht: Alle Jahre, sortiert nach Jahr und absteigender
# Abweichung
print("=== Alle Jahre: Abweichungen je Ticker inklusive Name, Sektor/Industrie
===")
display(
    format_ticker_display(all_errors_df).sort_values(
        ['Year', 'Abs_Abw_pct'],
        ascending=False
    ).reset_index(drop=True)
)

```

```

)
# ----- Export Gesamt-Results für Modellvergleich -----
---
# all_errors_df enthält bereits alle Jahre, Abweichungen und Meta-Spalten
(name, sector, industry)
all_errors_df['Model'] = 'Gradient Boosting'
metrics_display['Model'] = 'Gradient Boosting'
# Datei schreiben (Pfad nach Wunsch anpassen)
all_errors_df.to_csv('output/GB_results.csv', index=False)
metrics_display.to_csv('output/GB_Metrics.csv', index=False)

```

Gradient Boosting Model

```

import pandas as pd
import numpy as np
from sklearn.ensemble import GradientBoostingRegressor
# Daten laden
df = pd.read_csv("output/Training Data.csv")
df = df[df["Market Cap"] > 0]
df["Market_Cap_log"] = np.log(df["Market Cap"])
df = df[df["Year"] <= 2020]
# Zielnahe und ID-Spalten ausschließen
excluded_cols = [
    "Ticker", "Market_Cap_lag_1", "Market_Cap_delta", "Log_MarketCap",
    "Log_MarketCap_lag_1", "EV-to-EBITDA", "Log_MarketCap_rate"
    "EV-to-Sales", "Price-to-Cash-Flow", "EV-to-Operating-Cash-Flow_lag_1",
    "EV-to-Sales_delta",
    "EV-to-EBITDA_lag_1", "Price-to-Book", "EV-to-EBIT_lag_1", "Price-to-
    Book_lag_1", "Price-to-Free-Cash-Flow",
    "Price-to-Book_delta", "EV-to-Sales", "Price-to-Book_delta",
    "Log_MarketCap_rate",
    "Price-to-Earnings_lag_1", "Price-to-Free-Cash-Flow_lag_1", "Price-to-
    Earnings", "Price-to-Free-Cash-Flow_lag_1",
    "Price-to-Earnings", "EV-to-EBIT", "Price-to-Cash-Flow_lag_1", "EV-to-
    Sales_lag_1", "EV-to-Operating-Cash-Flow",
    "MarketCap_lag_1", "Market_Cap_lag_1", "Market_Cap_rate", "Market Cap",
    "Market_Cap_log", "is_test"
]
features = [col for col in df.columns if col not in excluded_cols]
# Trainingsdaten bereinigen
train_df = df.copy()
X = train_df[features]
y = np.log(train_df["Market Cap"])
X_clean = X.dropna()
y_clean = y.loc[X_clean.index]
# Gradient Boosting trainieren
model = GradientBoostingRegressor(
    n_estimators=300,
    random_state=42,

```

```

    max_depth=4, # vergleichbar mit default RF
    learning_rate=0.1
)
model.fit(X_clean, y_clean)
# Feature Importance extrahieren
importances = model.feature_importances_
importance_df = pd.DataFrame({
    "Feature": X_clean.columns,
    "Importance": importances
}).sort_values(by="Importance", ascending=False)
# Nur Features mit >0.1% Wichtigkeit behalten
selected_temp_features_df = importance_df[importance_df["Importance"] > 0.001]
print(selected_temp_features_df)
# Korrelationen zusätzlich prüfen
selected_features = selected_temp_features_df["Feature"].tolist()
# Filter auf ausgewählte Features
X_selected = X_clean[selected_features]
# 1. Heatmap der Korrelationsmatrix (optional)
corr_matrix = X_selected.corr()
# 2. Feature-Paare mit hoher Korrelation
high_corr_pairs = (
    corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(bool))
    .stack()
    .reset_index()
)
high_corr_pairs.columns = ["Feature_1", "Feature_2", "Korrelation"]
high_corr_pairs = high_corr_pairs[high_corr_pairs["Korrelation"].abs() > 0.9]
print("🔍 Feature-Paare mit |Korrelation| > 0.9:")
print(high_corr_pairs)
redundant_features = high_corr_pairs["Feature_2"].unique().tolist()
# Neue Auswahl ohne redundante Features
features = [col for col in df.columns if col not in excluded_cols and col not
in redundant_features]
# Trainingsdaten neu bereinigen
X = train_df[features]
y = np.log(train_df["Market Cap"])
X_clean = X.dropna()
y_clean = y.loc[X_clean.index]
# Gradient Boosting erneut trainieren
model = GradientBoostingRegressor(
    n_estimators=100,
    random_state=42,
    max_depth=4,
    learning_rate=0.1
)
model.fit(X_clean, y_clean)
# Feature Importance extrahieren

```

```

importances = model.feature_importances_
importance_df = pd.DataFrame({
    "Feature": X_clean.columns,
    "Importance": importances
}).sort_values(by="Importance", ascending=False)

# Nur Features mit >1% Wichtigkeit behalten
selected_features_df = importance_df[importance_df["Importance"] > 0.01]
print(selected_features_df)
# Finale Feature-Liste
final_features = selected_features_df["Feature"].tolist()
print("\n Final selected features (Copy-Paste ready):\n")
print(final_features)

import pandas as pd
import numpy as np
import finencedatabase as fd
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import r2_score, mean_absolute_error
from IPython.display import display

# -----
# 1) Globale Anzeigeeoptionen:
# - Zahlen mit 2 Dezimalstellen, Punkt als Tausendertrennzeichen und Komma
als Dezimaltrennzeichen
pd.options.display.float_format = lambda x: f"{x:,.2f}".replace('.',
'X').replace(',', '.').replace('X', ',')

# - Funktion, um Prozentwerte (z.B. 12.3456) als "12,35 %" auszugeben
pct_fmt = lambda x: f"{x:,.2f}".replace('.', 'X').replace(',',
'.').replace('X', ',') + " %"

# -----
# 2) FinanceDatabase einbinden und Metadaten laden
# Stelle sicher, dass du 'pip install finencedatabase' ausgeführt hast
equities_db = fd.Equities()
df_fd = equities_db.select()

# symbol ist im Index, daher Index zurücksetzen und nur
symbol/name/sector/industry auswählen
df_meta = df_fd.reset_index()[['symbol', 'name', 'sector',
'industry']].rename(columns={'symbol': 'Ticker'})

# -----
# 3) Haupt-Datensatz einlesen
df = pd.read_csv('output/Total Data_Set.csv')
bad_tickers = ['ADS', 'ANTM', 'CNA', 'JOE', 'KBR', 'ORI', 'ROLL'] # Es wurden
nachträglich 7 Ticker mit unvollständigen Daten gefunden, diese wurden
rausgeworfen
df = df[~df['Ticker'].isin(bad_tickers)] #

```

```

# 4) Features definieren
feature_list = ['EBITDA',
                'Income Before Tax_lag_1',
                'Cash Flow from Operations',
                'Stock Based Compensation',
                'Weighted Average Shares Diluted',
                'Free Cash Flow',
                'Total Equity',
                'Cumulative Return',
                'Free Cash Flow Yield',
                'Operating Cash Flow_lag_1',
                'Earnings Yield',
                'Total Current Liabilities_lag_1',
                'Earnings Yield_lag_1',
                'Total Shareholder Equity',
                'Gross Profit_lag_1']

# -----
# 5) Vorverarbeitung: Nur positive Market Cap und Log-Target
df = df[df['Market Cap'] > 0].copy()
df['Market_Cap_log'] = np.log(df['Market Cap'])

# 6) Trainingsdaten (is_test == False und Year <= 2020) vorbereiten
df_train = df[(df["is_test"] == False) & (df["Year"] <= 2020)].copy()
df_train = df_train.dropna(subset=feature_list + ['Market_Cap_log'])

X_train = df_train[feature_list]
y_train = df_train['Market_Cap_log']

# -----
# 7) Gradient-Boosting-Modell trainieren
gb = GradientBoostingRegressor(
    n_estimators=400,
    max_depth=4,
    learning_rate=0.1,
    random_state=42
)
gb.fit(X_train, y_train)

# -----
# 8) Evaluation für die Jahre 2021, 2022, 2023
results_per_year = {}
metrics_per_year = {}
best_worst_per_year = {}
for year in [2021, 2022, 2023]:
    # Testdaten für das jeweilige Jahr filtern (nur is_test == True)
    df_test = df[(df["Year"] == year) & (df["is_test"] == True)].copy()
    df_test = df_test.dropna(subset=feature_list + ['Market_Cap_log'])
    X_test = df_test[feature_list]
    y_true_log = df_test['Market_Cap_log']

```

```

# Vorhersage auf Log-Skala
y_pred_log = gb.predict(X_test)
# Rücktransformation in Original-Skala
df_test['Predicted_MarketCap'] = np.exp(y_pred_log)
df_test['True_MarketCap']      = df_test['Market Cap']
df_test['Abw_pct']            = 100 * (
    df_test['Predicted_MarketCap'] - df_test['True_MarketCap']
) / df_test['True_MarketCap']
# Absolute Abweichung in Prozent
df_test['Abs_Abw_pct'] = df_test['Abw_pct'].abs()
# Kennzahlen berechnen
r2 = r2_score(y_true_log, y_pred_log)
mape = np.mean(
    np.abs(df_test['Predicted_MarketCap'] - df_test['True_MarketCap']) /
    df_test['True_MarketCap']
) * 100
mae = mean_absolute_error(
    df_test['True_MarketCap'],
    df_test['Predicted_MarketCap']
)
# Weighted MAPE (WMAPE): Summe der absoluten Fehler geteilt durch
Summe der echten Werte
df_test['Abs_Error'] = np.abs(df_test['Predicted_MarketCap'] -
df_test['True_MarketCap'])
wmape = df_test['Abs_Error'].sum() / df_test['True_MarketCap'].sum() * 100
# Pro-Ticker-Übersicht komplett sortiert nach absoluter Abweichung
ticker_summary = df_test[[
    'Ticker', 'True_MarketCap', 'Predicted_MarketCap', 'Abw_pct',
'Abs_Abw_pct'
]].copy()
ticker_summary = ticker_summary.sort_values(
    by='Abs_Abw_pct',
    ascending=False
).reset_index(drop=True)
# Top 20 schlechteste (höchste |Abw_pct|)
worst20 = ticker_summary.head(20).copy()
# Top 20 beste (niedrigste |Abw_pct|)
best20 = ticker_summary.nsmallest(20, 'Abs_Abw_pct').copy()
# -----
# 9) Merge der Top-20-Tabellen mit FinanceDatabase-Metadaten
# Um Namens-Konflikte zu vermeiden, wählen wir hier nur benötigte
Spalten aus
worst20 = worst20.merge(
    df_meta[['Ticker', 'name', 'sector', 'industry']],
    on='Ticker',
    how='left'
)

```

```

best20 = best20.merge(
    df_meta[['Ticker', 'name', 'sector', 'industry']],
    on='Ticker',
    how='left'
)
# Spaltenreihenfolge anpassen: Ticker, Name, Sector, Industry, True,
Predicted, Abw_pct
worst20 = worst20[[
    'Ticker', 'name', 'sector', 'industry', 'True_MarketCap',
'Predicted_MarketCap', 'Abw_pct'
]]
best20 = best20[[
    'Ticker', 'name', 'sector', 'industry', 'True_MarketCap',
'Predicted_MarketCap', 'Abw_pct'
]]
# Speichern (für spätere Anzeige)
best_worst_per_year[year] = {
    'best20': best20,
    'worst20': worst20
}
results_per_year[year] = ticker_summary
metrics_per_year[year] = {
    'R2': r2,
    'MAPE (%)': mape,
    'MAE': mae,
    'WMAPE (%)': wmape
}
# -----
# 10) Metriken in DataFrame zusammenführen
metrics_df = pd.DataFrame.from_dict(metrics_per_year, orient='index')
metrics_df.index.name = 'Year'
# 11) Alle Abweichungen pro Jahr in einer Tabelle kombinieren (inkl. Meta)
all_errors_list = []
for year, df_err in results_per_year.items():
    tmp = df_err.merge(
        df_meta[['Ticker', 'name', 'sector', 'industry']],
        on='Ticker',
        how='left'
    )
    tmp['Year'] = year
    all_errors_list.append(tmp)
all_errors_df = pd.concat(all_errors_list, ignore_index=True)
# -----
# 12) Formatierte DataFrames für die Anzeige vorbereiten
# 12a) Metriken formatieren: R2 und MAE bleiben numerisch, MAPE_pct und
WMAPE_pct als Prozentstring
metrics_display = metrics_df.copy()

```

```

metrics_display['MAPE (%)'] = metrics_display['MAPE (%)'].apply(pct_fmt)
metrics_display['WMAPE (%)'] = metrics_display['WMAPE (%)'].apply(pct_fmt)
# 12b) Funktion, um Ticker-Tabellen (best/worst/alle) vor der Anzeige zu
formatieren
def format_ticker_display(df_in):
    df_out = df_in.copy()
    # True_MarketCap und Predicted_MarketCap: Tausenderpunkt + Komma als
    Dezimal
    df_out['True_MarketCap'] = df_out['True_MarketCap'].map(
        lambda x: f"{x:,.2f}".replace('.', 'X').replace(',', '.').replace('X',
        ',')
    )
    df_out['Predicted_MarketCap'] = df_out['Predicted_MarketCap'].map(
        lambda x: f"{x:,.2f}".replace('.', 'X').replace(',', '.').replace('X',
        ',')
    )
    # Abw_pct als Prozentstring
    df_out['Abw_pct'] = df_out['Abw_pct'].apply(pct_fmt)
    # Year als Integer (falls vorhanden)
    if 'Year' in df_out.columns:
        df_out['Year'] = df_out['Year'].astype(int)
    return df_out

# -----
# 13) Ergebnisse anzeigen (hübsche Tabellen mit display)
# 13a) Kennzahlen-Tabelle
print("=== Kennzahlen (R2, MAPE (%), MAE) pro Jahr ===")
display(metrics_display)
print()
# 13b) Top 20 Schlechteste Ticker für jedes Jahr (inkl. Name,
Sektor/Industrie)
for year in [2021, 2022, 2023]:
    print(f"=== {year}: Top 20 Schlechteste Ticker (größte absolute
    Abweichung) ===")
    display(format_ticker_display(best_worst_per_year[year]['worst20']))
    print()
# 13c) Top 20 Beste Ticker für jedes Jahr (inkl. Name, Sektor/Industrie)
for year in [2021, 2022, 2023]:
    print(f"=== {year}: Top 20 Beste Ticker (kleinste absolute Abweichung)
    ===")
    display(format_ticker_display(best_worst_per_year[year]['best20']))
    print()

# 13d) Gesamtübersicht: Alle Jahre, sortiert nach Jahr und absteigender
Abweichung
print("=== Alle Jahre: Abweichungen je Ticker inklusive Name, Sektor/Industrie
===")
display(

```

```

format_ticker_display(all_errors_df).sort_values(
    ['Year', 'Abs_Abw_pct'],
    ascending=False
).reset_index(drop=True)
)
# ----- Export Gesamt-Results für Modellvergleich -----
---
# all_errors_df enthält bereits alle Jahre, Abweichungen und Meta-Spalten
(name, sector, industry)
all_errors_df['Model'] = 'Gradient Boosting'
metrics_display['Model'] = 'Gradient Boosting'
# Datei schreiben (Pfad nach Wunsch anpassen)
all_errors_df.to_csv('output/GB_results.csv', index=False)
metrics_display.to_csv('output/GB_Metrics.csv', index=False)

```

DCF Model

```

import pandas as pd
import numpy as np
import financedatabase as fd # <<<< FinanceDatabase import
from sklearn.metrics import r2_score, mean_absolute_error
from IPython.display import display
# -----
# 1) Globale Anzeigeoptionen (Zahlenformatierung)
pd.options.display.float_format = lambda x: f"{x:,.2f}".replace('.',
'X').replace(',', '.').replace('X', ',')

pct_fmt = lambda x: f"{x:.1f}%".replace('.', ',') # Prozentformat mit Komma
# -----
# 2) FinanceDatabase einbinden und Metadaten laden
equities_db = fd.Equities()
df_fd = equities_db.select()
df_meta = (
    df_fd.reset_index()[['symbol', 'name', 'sector', 'industry']]
    .rename(columns={'symbol': 'Ticker'})
)

# ----- 3) Parameter festlegen -----
---
discount_rate = 0.10 # Discount Faktor (Eigenkapitalkostenrate)
window = 5 # Moving Average Fenster
forecast_horizon = 5 # Prognosehorizont
use_clip = True
clip_pct = 0.25
tv_method = "EV"

# ----- 4) Daten laden und aufbereiten -----
---
df = pd.read_csv("output/Test Data.csv")

```

```

df["EV"]          = df["Market Cap"] + df["Net Debt"]
df["EV_EBITDA"] = df["EV"] / df["EBITDA"].replace(0, np.nan)
df = df[[
    "Ticker", "Year", "Free Cash Flow Equity", "Market Cap",
    "Net Debt", "EBITDA", "EV", "EV_EBITDA", "is_test"
]]

historical      = df[(df["Year"] <= 2020) & (~df.isna().any(axis=1))].copy()
future_actual  = df[df["Year"].isin([2021, 2022, 2023])].copy()

# ----- 5) Hilfsfunktionen -----
--
def _clip_value(pred, last, pct):
    return min(max(pred, last*(1-pct)), last*(1+pct))
def _calc_cagr(series):
    if len(series)<2 or series.iloc[0]<=0 or series.iloc[-1]<=0:
        return 0.05
    yrs = len(series)-1
    return (series.iloc[-1]/series.iloc[0])** (1/yrs)-1
def forecast_fcfe(ticker_group):
    ticker_group = ticker_group.sort_values("Year")
    last_year    = int(ticker_group["Year"].max())
    last_fcfe    = ticker_group.loc[ticker_group["Year"]==last_year, "Free Cash
Flow Equity"].iloc[0]
    fcfe = {last_year: last_fcfe}
    recent = ticker_group[ticker_group["Year"]>=last_year-(window-1)]
    ma_fcfe = recent["Free Cash Flow Equity"].mean()
    prev = last_fcfe
    for y in range(last_year+1, last_year+1+forecast_horizon):
        pred = ma_fcfe
        if use_clip:
            pred = _clip_value(pred, prev, clip_pct)
        fcfe[y]=pred
        prev=pred
    return fcfe

# ----- 6) DCF-Berechnung pro Ticker -----
--
all_results = []
for ticker, grp in historical.groupby("Ticker"):
    try:
        fcfe = forecast_fcfe(grp)
        # PV der FCFE-Ströme 2021-2023
        pv = sum(
            fcfe[y]/((1+discount_rate)**(y-2020))
            for y in range(2021, 2024)
            if y in fcfe
        )

```

```

# Terminal Value via EV/EBITDA
recent_mult = grp.loc[grp["Year"]>=grp["Year"].max()-(window-
1),"EV_EBITDA"].dropna()
g_mult = np.clip(_calc_cagr(recent_mult), -0.2, 0.2)
last_mult= recent_mult.iloc[-1] if len(recent_mult)>0 else 0
mult_T = last_mult*(1+g_mult)
last_ebt = grp.loc[grp["Year"]==grp["Year"].max(),"EBITDA"].iloc[0]
ev_T = mult_T*(last_ebt*(1+g_mult))
netd = grp.loc[grp["Year"]==grp["Year"].max(),"Net Debt"].iloc[0]
tv = max(ev_T-netd, 0)
pv_total = pv + tv/((1+discount_rate)**forecast_horizon)
# Roll-Forward für 2021-2023
val = pv_total
for year in [2021,2022,2023]:
    if (year-1) in fcfe:
        val = val*(1+discount_rate) - fcfe[year-1]
        val = max(val, 0)
        actual = future_actual.query("Ticker==@ticker and
Year==@year")["Market Cap"]
        if not actual.empty:
            true_mc = actual.iloc[0]
            pred_mc = val
            abw_pct = 100*(pred_mc-true_mc)/true_mc
            all_results.append({
                "Ticker": ticker,
                "Year": int(year),
                "True_MarketCap": true_mc,
                "Predicted_MarketCap": pred_mc,
                "Abw_pct": abw_pct
            })
    except:
        continue
results_df = pd.DataFrame(all_results)
# ----- 7) Kennzahlen & Top/Flop je Jahr -----
--
metrics_per_year = {}
results_per_year = {}
best_worst_per_year = {}
for year in [2021,2022,2023]:
    df_y = results_df[results_df["Year"]==year].copy()
    r2 = r2_score(df_y["True_MarketCap"], df_y["Predicted_MarketCap"])
    mae = mean_absolute_error(df_y["True_MarketCap"],
df_y["Predicted_MarketCap"])
    mape = np.mean(np.abs(df_y["Predicted_MarketCap"]-
df_y["True_MarketCap"])/df_y["True_MarketCap"])*100
    # Weighted MAPE (WMAPE)
    df_y["Abs_Error"] = np.abs(df_y["Predicted_MarketCap"] -

```

```

df_y["True_MarketCap"])
    wmape = df_y["Abs_Error"].sum() / df_y["True_MarketCap"].sum() * 100
    metrics_per_year[year] = {"R2":r2, "MAPE (%)":mape, "MAE":mae, "WMAPE
(%)":wmape}
    df_y["Abs_Abw_pct"]=df_y["Abw_pct"].abs()
    sorted_y =
df_y.sort_values("Abs_Abw_pct",ascending=False).reset_index(drop=True)
    # Top/Flop
    worst20 = sorted_y.nlargest(20,"Abs_Abw_pct")
    best20  = sorted_y.nsmallest(20,"Abs_Abw_pct")
    # ---- Merge mit Metadaten ----
    worst20 = worst20.merge(df_meta, on="Ticker", how="left")
    best20  = best20.merge(df_meta, on="Ticker", how="left")
    cols    =
["Ticker","name","sector","industry","True_MarketCap","Predicted_MarketCap","A
bw_pct"]
    worst20 = worst20[cols]
    best20  = best20[cols]
    best_worst_per_year[year] = {"worst20":worst20,"best20":best20}
    results_per_year[year]    = sorted_y
# ----- 8) Gesamtübersicht (inkl. Meta) -----
-
all_errors_list=[]
for year, df_err in results_per_year.items():
    tmp = df_err.merge(df_meta, on="Ticker", how="left")
    tmp["Year"]=year
    all_errors_list.append(tmp)
all_errors_df = pd.concat(all_errors_list,ignore_index=True)
# ----- 9) Formatierung & Ausgabe -----
metrics_df      = pd.DataFrame.from_dict(metrics_per_year, orient="index")
metrics_df.index.name="Year"
metrics_display=metrics_df.copy()
metrics_display["MAPE (%)"]=metrics_display["MAPE (%)"].apply(pct_fmt)
metrics_display["WMAPE (%)"]=metrics_display["WMAPE (%)"].apply(pct_fmt)
def format_ticker_display(df_in):
    df = df_in.copy()
    df["True_MarketCap"]      = df["True_MarketCap"].map(
        lambda x: f"{x:,.2f}".replace(".", "X").replace(",", ".").replace("X",
",")
    )
    df["Predicted_MarketCap"] = df["Predicted_MarketCap"].map(
        lambda x: f"{x:,.2f}".replace(".", "X").replace(",", ".").replace("X",
",")
    )
    df["Abw_pct"] = df["Abw_pct"].apply(pct_fmt)
    if "Year" in df.columns:
        df["Year"]=df["Year"].astype(int)

```

```

    return df
print("=== Kennzahlen (R², MAPE (%), MAE) pro Jahr ===")
display(metrics_display); print()

for year in [2021,2022,2023]:
    print(f"=== {year}: Top 20 Schlechteste ===")
    display(format_ticker_display(best_worst_per_year[year]["worst20"]));
print()
for year in [2021,2022,2023]:
    print(f"=== {year}: Top 20 Beste ===")
    display(format_ticker_display(best_worst_per_year[year]["best20"]));
print()
print("=== Alle Jahre: Abweichungen inkl. Sektor/Industrie ===")
display(
    format_ticker_display(
        all_errors_df.sort_values(["Year","Abs_Abw_pct"],
ascending=[True,False])
        ).reset_index(drop=True)
    )
# ----- Export Gesamt-Results für Modellvergleich -----
---
# all_errors_df enthält bereits alle Jahre, Abweichungen und Meta-Spalten
(name, sector, industry)
all_errors_df['Model'] = 'DCF'
metrics_display['Model'] = 'DCF'
# Datei schreiben (Pfad nach Wunsch anpassen)
all_errors_df.to_csv('output/DCF_results.csv', index=False)
metrics_display.to_csv('output/DCF_Metrics.csv', index=False)

```

Datensäuberung

```

import pandas as pd
import numpy as np

# Säuberungskript und Erstellung von Logarithmischen Market Caps
# 1. Trainingsdaten einlesen
df = pd.read_csv("output/Company_Data.csv")

# 2. Alle Ticker identifizieren, bei denen mindestens ein Market Cap
NaN ist
tickers_with_nan = df.loc[df["Market Cap"].isna(), "Ticker"].unique()

# Ausgabe der Ticker, die entfernt werden
print("Folgende Ticker werden entfernt (mindestens ein fehlender
Market Cap-Wert):")
print(tickers_with_nan)

```

```

# 3. Entfernen aller Zeilen, bei denen der Ticker in tickers_with_nan
enthalten ist
df_clean = df[~df["Ticker"].isin(tickers_with_nan)].copy()

# 4. Optional: Überprüfen, ob in der bereinigten Datei kein NaN mehr
in Market Cap vorkommt
remaining_nan = df_clean["Market Cap"].isna().sum()
print(f"Anzahl verbleibender NaN in 'Market Cap': {remaining_nan}")

# 5. Doppelte Spalten entfernen
df = df.loc[:, ~df.columns.duplicated()]

# 6. Zielvariable berechnen
df["Log_MarketCap"] = np.log(df["Market Cap"].replace({0: np.nan}))
df["Log_MarketCap"] = df["Log_MarketCap"].replace([np.inf, -np.inf],
np.nan)

# 7. Spalten, die nicht als Input-Features verwendet werden sollen
exclude_columns = [
    "Unnamed: 0", "Close", "Open", "High", "Low", "Volume", "Adj
Close"
]

# 8. Alle numerischen Spalten (inkl. Log_MarketCap)
numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()

# 9. Input-Features definieren (Ticker, Year und Log_MarketCap
ausschließen)
selected_features = [col for col in numeric_cols if col not in
exclude_columns + ["Year", "Ticker", "Log_MarketCap", "Enterprise
Value"]]

# 10. Sauberen DataFrame zusammenstellen
df_clean = df[selected_features + ["Log_MarketCap", "Ticker", "Year"]]

# 11. Ungültige Werte rauswerfen
df_clean = df_clean.replace([np.inf, -np.inf], np.nan)
df_clean = df_clean.dropna()

# 12. Export (optional)

```

```
df_clean.to_csv("output/Training_Data_cleaned_for_model.csv",
index=False)
```

Anhang 1 Daten Anreicherung

```
import pandas as pd
import numpy as np

# Daten laden
df = pd.read_csv("output/Training_Data_cleaned_for_model.csv")
df = df.drop(columns=["Unnamed: 0"], errors="ignore")
df = df.sort_values(by=["Ticker", "Year"])

# Numerische Spalten
numerical_cols = df.drop(columns=["Ticker",
"Year"]).select_dtypes(include="number").columns

# Feature-Generierung
lags = df.groupby("Ticker")[numerical_cols].shift(1).add_suffix("_lag_1")
deltas = df.groupby("Ticker")[numerical_cols].diff().add_suffix("_delta")
rates =
df.groupby("Ticker")[numerical_cols].pct_change(fill_method=None).add_suffix("_rate")
#ma3 = df.groupby("Ticker")[numerical_cols].transform(lambda x: x.rolling(3,
min_periods=1).mean()).add_suffix("_ma3")

# Zusammenführen
df_final = pd.concat([df, lags, deltas, rates], axis=1).copy()
df_final.replace([np.inf, -np.inf], np.nan, inplace=True)

# NaNs optional mit 0 auffüllen (nur bei den technischen Features)
fill_cols = [col for col in df_final.columns if col.endswith(('_lag_1',
'_delta', '_rate'))]
df_final[fill_cols] = df_final[fill_cols].fillna(0)

# Jeden dritten Ticker als Testdaten kennzeichnen
unique_tickers = df_final["Ticker"].dropna().unique()
np.random.seed(42)
test_tickers = np.random.permutation(unique_tickers)[:3]

is_test_col = df_final["Ticker"].isin(test_tickers)
df_final = pd.concat([df_final, is_test_col.rename("is_test")], axis=1).copy()
df_test = df_final[df_final["is_test"]]
df_training = df_final[~df_final["is_test"]]

# Speichern
```

```
cols = df_final.columns.tolist()
cols = ['Ticker', 'Year'] + [col for col in cols if col not in ['Ticker',
'Year']]
bad_tickers = ['ADS', 'ANTM', 'CNA', 'JOE', 'KBR', 'ORI', 'ROLL']
df_final = df_final[cols]
#[~df_test['Ticker'].isin(bad_tickers)]
df_test = df_final[df_final["is_test"]]
df_test = df_test[~df_test['Ticker'].isin(bad_tickers)]
df_training = df_final[~df_final["is_test"]]
df_final.to_csv("output/Total Data Set.csv", index=False)
df_training.to_csv("output/Training Data.csv", index=False)
df_test.to_csv("output/Test Data.csv", index=False)
```

Alle Outputs (Datenset, Testdaten, Trainingsdaten etc.) und weitere Skripte finden sich separat Anhang.

VII. Glossar

Unternehmensbewertung – Grundbegriffe

Unternehmenswert: Monetärer Wert eines Unternehmens, ermittelt durch verschiedene Bewertungsverfahren (z.B. DCF, Ertragswertverfahren).

Bewertungsanlass: Grund für eine Unternehmensbewertung, z.B. Kauf/Verkauf, steuerliche Zwecke, gerichtliche Auseinandersetzungen.

Going-Concern-Prinzip: Annahme, dass das Unternehmen fortgeführt wird und nicht liquidiert werden soll.

Marktkapitalisierung: Börsenwert eines Unternehmens: Aktienkurs \times Anzahl der ausgegebenen Aktien. Dient oft als Proxy für den „wahren“ Unternehmenswert.

Traditionelle Bewertungsverfahren

Substanzwertverfahren: Summe der Vermögenswerte abzüglich Schulden (Teilreproduktionswert = nur betriebsnotwendige Vermögenswerte; Vollreproduktionswert = inkl. immaterieller Werte).

Liquidationswertverfahren: Wert bei sofortiger Zerschlagung des Unternehmens, basierend auf Veräußerungserlösen abzüglich Kosten.

Mittelwertverfahren: Durchschnitt aus Substanzwert und Ertragswert.

Übergewinnverfahren: Substanzwert plus Barwert der künftigen Übergewinne (Gewinne über einer Mindestverzinsung).

Stuttgarter Verfahren: Steuerlich motiviertes Verfahren, das bilanzielles Vermögen und Ertragskraft kombiniert.

Ertragswertverfahren: Barwert der erwarteten, den Eigentümern zufließenden Erträge, abgezinst mit risikoadjustiertem Zinssatz.

Discounted-Cashflow-Verfahren (DCF): Barwert aller zukünftigen, dem Unternehmen zufließenden Free Cashflows, abgezinst mit einem risikoadjustierten Zinssatz (z.B. WACC).

WACC (Weighted Average Cost of Capital): Durchschnittlicher Kapitalkostensatz eines Unternehmens, gewichtet nach Eigen- und Fremdkapitalanteil.

CAPM (Capital Asset Pricing Model): Modell zur Bestimmung der Eigenkapitalkosten auf Basis von Marktrisiko (Beta-Faktor, Marktrisikoprämie).

Terminal Value: Barwert aller zukünftigen Cashflows ab dem letzten explizit geplanten Jahr (ewige Rente).

Regulatorik und Standards

IDW S1: Deutscher Standard für Unternehmensbewertungen, herausgegeben vom Institut der Wirtschaftsprüfer.

IFRS 13: Internationaler Rechnungslegungsstandard zur Fair-Value-Bewertung.

IVS (International Valuation Standards): Weltweite Standards für Bewertungsverfahren.

Effizientmarkthypothese (EMH): Theorie, dass Marktpreise alle verfügbaren Informationen widerspiegeln (schwache, halbstarke, starke Form).

Künstliche Intelligenz & Maschinelles Lernen

Künstliche Intelligenz (KI, engl. AI): Systeme, die Aufgaben mit einem gewissen Grad an Autonomie und „intelligentem Verhalten“ ausführen.

Maschinelles Lernen (ML): Teilgebiet der KI; Algorithmen lernen Muster aus Daten, ohne explizite Programmierung.

Regelbasiertes Lernen: KI-Verfahren, die auf fest programmierten Wenn-Dann-Regeln basieren (z.B. Expertensysteme).

Überwachtes Lernen (Supervised Learning): ML-Ansatz, bei dem Modelle mit gelabelten Daten (Input/Output) trainiert werden.

Unüberwachtes Lernen (Unsupervised Learning): ML-Ansatz, bei dem Modelle Strukturen/Muster in unbeschrifteten Daten erkennen.

Verstärkendes Lernen (Reinforcement Learning)

ML-Ansatz, bei dem Algorithmen durch Belohnung/Strafe lernen, optimale Entscheidungen zu treffen.

Random Forest: Ensemble-ML-Verfahren, das viele Entscheidungsbäume kombiniert, um Vorhersagen zu stabilisieren.

Gradient Boosting: Ensemble-ML-Verfahren, bei dem schwache Modelle sequentiell trainiert werden, um Fehler zu minimieren.

Künstliche Neuronale Netze (KNN): ML-Modelle, die an biologische Nervenzellen angelehnt sind, mit mehreren Schichten („Hidden Layers“).

Deep Learning: ML-Ansatz mit tiefen neuronalen Netzen, besonders leistungsfähig bei großen, komplexen Datensätzen.

ML-Methodik & Bewertung

Feature Engineering: Auswahl, Transformation und Konstruktion von Inputvariablen (Features), um ML-Modelle zu verbessern.

Lag-Features: Historische Werte einer Variable, die als zusätzliche Features genutzt werden (z.B. Umsatz des Vorjahres).

Feature Importance: Kennzahl, wie stark ein bestimmtes Feature zur Prognosegüte eines ML-Modells beiträgt.

Overfitting: Modell passt sich zu stark an Trainingsdaten an und verliert Generalisierungsfähigkeit auf neue Daten.

Explainable AI (XAI): Methoden, die ML-Modelle und deren Entscheidungen für Menschen nachvollziehbar machen.

ML-Erfolgsmessung

Mean Decrease Impurity (MDI): Maß für die Bedeutung eines Features, basierend auf der Reduktion der Unreinheit in Entscheidungsbäumen.

Mean Absolute Error (MAE): Durchschnittlicher absoluter Fehler zwischen Vorhersage und tatsächlichem Wert.

Mean Absolute Percentage Error (MAPE): Durchschnittlicher prozentualer Fehler zwischen Vorhersage und tatsächlichem Wert.

Bestimmtheitsmaß (R^2 , „coefficient of determination“): Anteil der erklärten Varianz durch das Modell (0 = keine Erklärung, 1 = perfekte Erklärung).

Daten & Tools

US-GAAP: Rechnungslegungsstandard für US-Unternehmen, sorgt für Vergleichbarkeit der Finanzdaten.

Python, scikit-learn, pandas, numpy: Gängige Open-Source-Programmbibliotheken für Datenanalyse und maschinelles Lernen.

VIII. Eidesstaatliche Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe.

Die Stellen der Arbeit, einschließlich der Tabellen und Abbildungen, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen sind, habe ich in jedem einzelnen Fall kenntlich gemacht und die Herkunft nachgewiesen.

Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen und wurde auch noch nicht veröffentlicht.

Rosenheim, 11.07.2025

A handwritten signature in black ink, consisting of several overlapping, fluid strokes that form a stylized, somewhat abstract shape.

Tim Stäcker